ALGORITHMIC HEURISTICS IN DEEP LEARNING: REGULARIZATION AND ROBUSTNESS

by
Poorya Mianjy

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
August, 2022

Abstract

While deep learning continues to advance our technological world, its theoretical underpinnings are far from understood. In this thesis, we focus on *regularization* and *robustness* due to *algorithmic heuristics* that are often leveraged in state-of-the-art deep learning systems. In particular, we take steps towards a formal understanding of regularization due dropout, which is a popular local-search heuristics in deep learning. We also present a theoretical study of adversarial training, an effective local-search heuristic to train models that are more robust against adversarial perturbations. The thesis is organized as follows.

In Chapters 2 and 3, we focus on the explicit regularization due to dropout in shallow and deep linear networks. We show that dropout, as a learning rule, amounts to regularizing the objective with a data-dependent term, which includes products of the weights along certain cycles in the network graph. We then show that under certain conditions, this regularizer boils down to a trace-norm penalty, which provides a rich inductive bias in matrix learning problems.

In Chapter 4, we study the learning theoretic implications of the explicit regularizer. In particular, focusing on the matrix completion problem, we provide precise ϵ-suboptimality results for the dropout rule. We also provide extensive empirical evidence establishing that even in this simple application, algorithmic heuristics such as dropout can dramatically boost the generalization performance of gradient-based optimization methods. We further provide generalization error guarantees for the dropout rule in the two-layer neural networks with ReLU activation. We provide

extensive numerical evaluations verifying that the proposed theoretical bound is predictive of the observed generalization gap.

In Chapter 5, we focus on the computational aspects of dropout. We provide precise iteration complexity rates for training two-layer ReLU neural networks with dropout, under certain distributional assumptions and over-parameterization requirements. We also show that dropout implicitly compresses the network. In particular, we show that there exists a sub-network, i.e., one of the iterates of dropout training, that can generalize as well as any complete network.

Finally, in Chapter 6, we switch gears towards adversarial training in two-layer neural networks with Leaky ReLU activation. We provide precise iteration complexity results for end-to-end adversarial training when the underlying distribution is separable. Our results include a convergence guarantee for the PGD attack, which is a popular local-search heuristic for finding adversarial perturbations, and guarantees suboptimality in terms of the robust generalization error, both of which are the first of their kind. More importantly, our results hold for any width and initialization.

Thesis Readers

Dr. Raman Arora (Primary Advisor)
Department of Computer Science
Johns Hopkins University

Dr. Amitabh Basu
Department of Applied Mathematics and Statistics
Johns Hopkins University

Dr. René Vidal
Department of Biomedical Engineering
Johns Hopkins University

To my parents and my wife

Acknowledgements

First and foremost, I would like to thank my advisor, Raman Arora, for his continued support throughout the course of this study. He has been a constant source of inspiration in all projects that I've worked on. Raman has helped me to develop my critical thinking skill, which has had profound effects in my life, beyond the scope of this thesis. His patience has given me the courage to take on some challenging problems, which at first appeared intimidating to me. He gave me the freedom to experience a wide range of topics in machine learning, which has broadened my horizons and will be immensely valuable for my future professional advancements.

I'm grateful for the opportunity of working with my senior collaborators and coauthors, Peter Bartlett, Amitabh Basu, Nati Srebro, and René Vidal. They have guided me on a number of projects that I worked on during my Ph.D. studies. I'm particularly owing gratitude to Amitabh and René, from whom I've learned a lot of useful tools in optimization theory.

I would like to thank my colleagues and collaborators at JHU. In particular, my labmates and coauthors Teodor Marinov, Enayat Ullah, and Yunjuan Wang, and my fellow machine learning researcher and coauthor Anirbit Mukherjee.

Finally, I'm grateful to my family and my wife, for supporting me through thick and thin. Without them, this work was not possible.

Contents

Abstract . . . . . **ii**

Dedication . . . . . **iv**

Acknowledgements . . . . . **v**

Contents . . . . . **vi**

List of Tables . . . . . **x**

List of Figures . . . . . **xi**

Chapter 1 Introduction **1**

1.1 Preliminaries 6

1.1.1 Statistical Perspective of Generalization 8

1.1.2 Computational Perspective of Generalization 15

1.1.3 Robust Learning 18

1.2 Related Work 22

1.2.1 Regularization due to Dropout 23

1.2.2 Adversarial Training 23

1.3 Contributions 24

1.3.1 Explicit Regularization Due to Dropout 25

1.3.2 Statistical Guarantees for Dropout 26

1.3.3 Computational Guarantees for Dropout 27

1.3.4 Robustness Guarantees for Adversarial Training 27

Chapter 2 Dropout Regularizer: Shallow Linear Networks 29

2.1 Linear autoencoders with tied weights 32

2.2 General Two-Layer Networks 36

2.3 The Optimization Landscape 39

2.3.1 Implicit bias in local optima 39

2.3.2 Landscape properties 40

2.4 Matrix Factorization with Dropout 41

2.4.1 Comparison with Previous Work 44

2.5 Proofs 44

2.5.1 Proofs of Theorems in Section 2.1 45

2.5.2 Proofs of Theorems in Section 2.2 50

2.5.3 Proofs of Theorems in Sections 2.3 53

2.6 Empirical Results 61

2.7 Discussion 63

Chapter 3 Dropout Regularizer: Deep Linear Networks 65

3.1 The explicit regularizer 69

3.2 The induced regularizer 74

3.3 Global optimality 84

3.4 Experimental Results 90

3.4.1 Spectral shrinkage and rank control 91

3.4.2 Convergence to equalized networks 92

3.5 Discussion 94

Chapter 4 Statistical Guarantees for Dropout 95

4.1 Related Work 97

4.2 Matrix Sensing 98

4.2.1 Comparison with Previous Work . . . 101
4.3 Non-linear Networks . . . 102
4.3.1 Comparison with Previous Work . . . 108
4.4 Role of Parametrization . . . 109
4.5 Proofs . . . 110
4.5.1 Matrix Sensing . . . 110
4.5.2 Non-linear Neural Networks . . . 113
4.6 Experimental Results . . . 120
4.7 Discussion . . . 123

Chapter 5 Computational Guarantees for Dropout . . . 125
5.1 Related Work . . . 127
5.2 Poblem Setup . . . 129
5.2.1 Notation . . . 132
5.3 Main Results . . . 133
5.4 Proofs . . . 135
5.5 Experimental Results . . . 150
5.6 Discussion . . . 152

Chapter 6 Robustness Guarantees for Adversarial Training . . . 154
6.1 Related Work . . . 156
6.2 Problem Setup . . . 157
6.3 Main Results . . . 159
6.3.1 Comparison with Previous Work . . . 163
6.4 Proofs . . . 163
6.5 Empirical Results . . . 173
6.5.1 Grid Search Optimization . . . 173
6.5.2 Binary Classification . . . 174

6.5.3 Extension to multi-label setting 176
6.6 Discussion 178

Chapter 7 Conclusion 181
7.1 Main Contributions 181
7.2 Other Contributions 182
7.3 Future Work 183

Bibliography 185
.1 Table of Notations 213
.2 Auxiliary Results 214

List of Tables

4-I Test RMSE of plain SGD as well as dropout on the MovieLens dataset 120

6-I Robust test error of adversarially trained models with and without reflecting the loss (Binary Classification) 174

6-II Robust test accuracy of adversarially trained models with and without reflecting the loss (Multiclass Classification) 177

List of Figures

Figure 1-1 Adversarial Examples: imperceptible perturbations that can fool the model 20

Figure 2-1 Optimization landscape of the dropout objective for a single hidden-layer linear autoencoder 34

Figure 2-2 Convergence of dropout from two different initialization to a global optimum 41

Figure 2-3 Dropout converges to global optima for different dropout rates and different widths of the hidden layer 62

Figure 3-1 Illustration of the explicit regularizer due to dropout in deep linear networks 70

Figure 3-2 Distribution of the singular values of a deep linear network trained using dropout 92

Figure 3-3 Dropout converges to the set of equalized networks 93

Figure 4-1 Comparing peformance of plain SGD and dropout on the MovieLens dataset 120

Figure 4-2 Evaluating co-adaptation, generalization gap, and the theoretical gap on the MNIST dataset 121

Figure 5-1 Linear upperbound on the logistic loss 146

Figure 5-2 Test accuracy of the full network as well as the sub-networks drawn by dropout iterates 150

Figure 5-3 Test accuracy of the full network as well as 100 random i.i.d. sub-networks 152

Figure 6-1 The 0-1 loss, the cross-entropy loss, and the reflected cross-entropy loss 158

Figure 6-2 Number of the top-k attack vectors that are optimal for the cross entropy loss and the reflected version 174

Chapter 1

Introduction

Deep learning is revolutionizing the technological world with recent advances in artificial intelligence. However, a formal understanding of when or why deep learning algorithms succeed has remained elusive. Developing a theory around deep learning will help the field grow faster by reducing the amount of trial-and-error involved in training deep neural networks. With this motivation, in this dissertation, we take some steps towards developing a theoretical foundation for deep learning, focusing on *regularization* and *robustness* imparted by *algorithmic heuristics*.

In deep learning, the hypothesis class is represented by deep neural networks. Given an input, a deep neural network computes a composition of *layers*, each of which performs a parameterized transformation of their input. From an approximation theoretic perspective, deep neural networks provide a rich hypothesis class for machine learning applications: even with a single hidden layer of finite size, a neural network can represent any continuous function arbitrary well [Cyb89, Hor91]. Moreover, success of deep learning is in part attributed to the *depth* of the networks, i.e. number of *layers*, which is shown to extract hierarchical features from the data [Ben09]. On the contrary, traditional machine learning models such as support vector machines and kernel machines can be viewed as *shallow* networks, where only a linear transformation is learned on a fixed layer of nonlinear feature extraction.

The goal of machine learning is to find a hypothesis within the target hypothesis class, which has a small expected error (often referred to as *generalization error*). The underlying data distribution is unknown, and the learner can only access it through a sample, known as the training data. A natural, common practice is to minimize the empirical error, or a surrogate, associated with the model on the sample. For traditional, shallow models, this *empirical risk minimization* often reduces to a convex optimization problem. Such convex learning problems can be efficiently solved using principled first-order optimization methods such as Gradient Descent (GD) and its variants. However, the objective landscape associated with a deep neural network is often highly non-convex, with spurious local-optima and saddle-points [Kaw16, SS18]. Therefore, in principle, first-order methods are doomed to fail in learning deep neural networks. Yet, in practice, local-search heuristics are quite successful in minimizing the empirical objective, and finding models that can also generalize well.

From a theoretical perspective, this empirical success is quite surprising, due to both *computational* and *statistical* challenges involved in training neural networks. From a *computational* viewpoint, it is well-known that even training a 3-node neural network is NP-complete [BR92]. In fact, under some cryptographic assumptions, even improper learning of small neural networks is hard [KS09, DLSS14]. From a *statistical* perspective, classical learning theory attributes generalization error to some form of model-class complexity, measured e.g. in terms of combinatorial properties such as the VC-dimension [Vap13], or scale-sensitive measures such as Rademacher complexity [BM02]. The VC-dimension of deep neural networks (with hard-threshold activation) is equal, up to logarithmic factors, to the number of network parameters [ABB+99, SSBD14]. Thus, VC-theory falls short in resolving the generalization puzzle in the over-parameterized settings, where number of parameters far exceeds the sample size. Moreover, despite all efforts in tightening the upperbounds on the scale-sensitive measures [Bar98, NTS15, GRS18, NBS17], these bounds often yield

trivial, vacuous generalization guarantees [NK19].

A recent strand of research attributes the generalization ability of neural networks to the *implicit bias* of optimization algorithms (through the geometry of local search methods) [NTS14, ZBH+16, NTSS17]. Implicit bias refers to the tendency of the optimization algorithm towards solutions with certain structural properties, e.g., having a small norm. While it has been shown in simpler linear models that GD is implicitly biased towards low-norm solutions [SHN+18, JT19a], there is growing evidence that implicit bias may be unable to explain generalization even in a simpler setting of stochastic convex optimization [DFKL20]. Furthermore, most real-world state-of-the-art deep learning systems do employ various *explicit* architectural and algorithmic heuristics – from normalization layers [IS15, BKH16, SK16] to residual connections [HZRS16a] and exotic regularization techniques such as dropout [HSK+12, SHK+14]. Therefore, implicit bias might not give the whole picture when we aim to understand the empirical success of deep learning.

Although neural networks trained by first-order methods generalize well, they are often highly susceptible to small, imperceptible, adversarial perturbations of data at test time [SZS+14]. Such vulnerability to *adversarial examples* imposes severe limitations on the deployment of neural networks-based systems, especially in critical high-stakes applications such as autonomous driving, where safe and reliable operation is paramount. An abundance of studies demonstrating adversarial examples across different tasks and application domains [GSS15, MDFF16, CW17] has led to a renewed focus on robust learning as an active area of research within machine learning. The goal of *robust learning* is to find models that yield reliable predictions on test data notwithstanding adversarial perturbations.

The computational and statistical challenges involved in learning deep neural networks are even more severe when robust generalization is the desired objective. In particular, from a computational perspective, even checking the robustness of a given

model at a given test sample is NP-hard [ADV19]. In fact, there exist learning problems where standard learning can be done efficiently, whereas robust learning becomes computationally intractable [BLPR19, Nak19]. Also, from a statistical perspective, the robust variants of complexity measures such as robust Rademacher complexity, often yield even more pessimistic bounds [YKB19, AFM20] compared to the standard counterparts.

There has been a flurry of recent publications on designing defense strategies against adversarial examples, including Distillation [PMW+16], randomization at inference time time [XWZ+18], thermometer encoding [BRRG18], adversarial training [MMS+18], and convex outer approximation [WK18]. In particular, adversarial training is a principled approach for learning models that are robust to adversarial examples, wherein the robust learning problem is formulated as a min-max optimization problem, and alternating local-search heuristics are employed to solve it. Despite the aforementioned computational and statistical challenges involved in robust learning, in practice, such first-order heuristics used in adversarial training has shown to improve robust generalization across a wide range of experimental settings.

A comprehensive theory of deep learning – one that addresses generalization and robustness – needs to take into account idiosyncrasies involved in various local-search heuristics used widely by practitioners. How do these heuristics provide deep neural networks with such remarkable generalization ability? How do they impart robustness to models that are adversarially trained? Addressing these questions in a principled fashion is paramount in theoretical deep learning, as we do in this thesis. In particular, our focus is on the following two aspects.

Regularization due to Dropout. Dropout is one of the most popular algorithmic heuristics in deep learning. Drawing insights from the success of the sexual reproduction model in the evolution of advanced organisms, dropout aims at breaking co-adaptation between neurons by randomly dropping a subset of neurons at the

time of training. First introduced by Hinton et al. [HSK+12], dropout has been widely used in state-of-the-art models for several tasks including large-scale visual recognition [SLJ+15], large vocabulary continuous speech recognition [DSH13], image question answering [YHG+16], handwriting recognition [PBKL14], sentiment prediction and question classification [KGB14], dependency parsing [CM14], and brain tumor segmentation [HDWF+17]. Following the empirical success of dropout, there have been several studies in recent years aimed at explaining why and how dropout helps with generalization [BS13, McA13, WWL13, WFWL14, HL15, HL17]. However, our understanding of the theoretical foundations of dropout still remains limited. In the first four chapters of this thesis, we present several recent results which contributes to the growing body of literature on grounding the underpinnings of dropout.

Robustness Imparted by Adversarial Training. Local-search heuristics play a crucial role in robust learning – both in finding adversarial examples, and in defense strategies against those adversarial examples [MMS+18, SZS+14, GSS15, KGB17, CW17]. In particular, in adversarial training, the robust generalization problem is formulated as a min-max optimization problem, and local-search heuristics are often employed to solve both the inner-max and the outer-min problems. Adversarial training has been shown effective in learning models that are more robust against adversarial examples across a wide range of experimental settings [MMS+18, CW17, ACW18]. Following this empirical success, several recent works study the convergence properties of local-search heuristics used in adversarial training of deep neural networks [GCL+19, ZPD+20]. While these results shed light on the dynamics of adversarial training in the over-parameterized settings, as we point out later, they are limited in several directions. In the last chapter of this thesis, we take a step towards a better understanding of adversarial training by providing precise robust generalization guarantees for two-layer neural networks.

1.1 Preliminaries

We denote matrices, vectors, scalar variables and sets by Roman capital letters, Roman small letters, small letters and script letters respectively (e.g. X, x, x, and $\mathcal{X}$). We denote the i-th column of a matrix X and the j-th entry of vector y with x_i and y_j, respectively. Let $\langle \cdot, \cdot \rangle$ denote the standard inner product. For $p \geq 1$, we denote the ℓ_p-norm of vector x as $\|\mathrm{x}\|_p$; we often drop the subscript when $p = 2$, and denote the Euclidean norm as $\|\mathrm{x}\|$. Furthermore, we denote the dual conjugate of p with p^*, where $\frac{1}{p} + \frac{1}{p^*} = 1$. We represent the spectral norm and the Frobenius norm of matrix X by $\|\mathrm{X}\|_2$ and $\|\mathrm{X}\|_F$, respectively. Furthermore, $\|\mathrm{X}\|_{p,q} := \left(\sum_j \left(\sum_i |\mathrm{X}_{i,j}|^p \right)^{q/p} \right)^{1/q}$ is the ℓ_q-norm of the vector that collects the ℓ_p-norm of the columns of X.

Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and label spaces, respectively. While the input space is often a subset of $\mathbb{R}^d$, the label space depends on the underlying task. For example, in classification and regression, the label spaces are best represented as $\mathcal{Y} = \{1, \ldots, k\}$ and $\mathcal{Y} \subseteq \mathbb{R}^k$, respectively. We assume that the data is jointly distributed according to an unknown distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$.

In a typical machine learning application, the learner first chooses a parameterized *hypothesis class* $\mathcal{F} := \{f_{\mathrm{w}} : \mathcal{X} \to \mathcal{Y},\ \mathrm{w} \in \mathcal{W}\}$ suitable for the learning task, where $\mathcal{W}$ denotes the parameter space. In this thesis, we are interested in feed-forward neural networks parameterized by a set of *weight matrices* $\mathrm{w} := \{\mathrm{W}_i\}_{i=1}^{k+1}$, $\mathrm{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}$ for all $i \in \{1, \ldots, k+1\}$, computing the function:

$$f_{\mathrm{w}} : \mathrm{x} \mapsto \mathrm{W}_{k+1}\sigma(\mathrm{W}_k\sigma(\cdots\sigma(\mathrm{W}_1\mathrm{x})\cdots))$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is an *activation function* acting elementwise on the input, d_0 and d_{k+1} correspond to the input and output dimensionality, and $d_1, \ldots, d_k$ denote the *widths* of the *hidden layers*. We further let $d := \max\{d_1, \ldots, d_k\}$ denote the overal width of the network.

Given n i.i.d. examples $\mathcal{S}_n = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^n \sim \mathcal{D}^n$ and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$,

the goal of learning is to find parameters $\widehat{\mathrm{w}} \in \mathcal{W}$ such that the hypothesis $f_{\widehat{\mathrm{w}}} \in \mathcal{F}$ enjoys a small *generalization error* $L(\mathrm{w}) := \mathbb{E}_{\mathcal{D}}[\ell(f_{\mathrm{w}}(\mathrm{x}), \mathrm{y})]$. A common approach to this learning problem is *empirical risk minimization* (ERM), which returns the model with smallest empirical loss on the sample:

$$\widehat{\mathrm{w}}_{\mathrm{ERM}} \in \operatorname*{argmin}_{\mathrm{w} \in \mathcal{W}} \widehat{L}(\mathrm{w}) := \frac{1}{n} \sum_{i=1}^{n} \ell(f_{\mathrm{w}}(\mathrm{x}_i), \mathrm{y}_i)$$

The goodness of any model $f_{\widehat{\mathrm{w}}}$, including the ERM solution $f_{\widehat{\mathrm{w}}_{\mathrm{ERM}}}$, can be measured in terms of its *excess error*, i.e., the error that $f_{\widehat{\mathrm{w}}}$ incurs in excess of the best hypothesis in class. In particular, a generalization error guarantee establishes – in expectation or with high probability over the draw of a random sample – that:

$$L(\widehat{\mathrm{w}}) - \min_{\mathrm{w} \in \mathcal{W}} L(\mathrm{w}) \leq \epsilon,$$

for some *small* ϵ that depends on the sample size and other problem-specific parameters. The *sample complexity* of the corresponding learning rule is the sample size required to achieve the desired ϵ-*suboptimality* in the generalization error.

When the empirical problem is convex, first-order methods provide a principled approach to find an approximate ERM solution. In particular, gradient descent (GD) is an iterative first-order method, which starts at an initialization w_1, and updates the parameters as $\mathrm{w}_{t+1} \leftarrow \mathrm{w}_t - \eta_t \nabla \widehat{L}(\mathrm{w}_t)$, where $\eta_t > 0$ is the so-called *learning rate*. Stochastic gradient descent (SGD) – a staple learning algorithm – is a computationally attractive variant of GD, which at each iterate, updates the parameters based on only a few samples drawn uniformly at random from the empirical distribution[1]. Under additional regularity conditions (boundedness, Lipschitzness, smoothness), one can appeal to standard analysis of (S)GD in convex optimization literature and provide precise *iteration complexity* guarantees for (S)GD, i.e., bound the number of iterations required to achieve certain ϵ-suboptimality in the objective (see [SSBD14] and the references therein).

[1] or based on fresh samples drawn i.i.d. from the underlying distribution.

In deep learning, despite the non-convexity of the loss landscape, gradient-based methods are still used to train deep neural networks. In particular, the back-propagation algorithm [RHW86] – the main powerhouse behind deep learning systems – is simply an instance of gradient descent with an efficient, inductive procedure for computing the gradient of the loss with respect to the parameters of each layer.

The rest of this section is organized as follows. In Section 1.1.1, we introduce the statistical aspects of the learning problem. In particular, we present some useful tools from statistical learning theory to bound the sample complexity of a given learning rule, ignoring the computational cost of the rule. In Section 1.1.2, we shift gears towards the computational aspects of the learning problem, wherein the goal is to understand the iteration complexity of the learning rule. In each of these sections, we review several recent schools of thought for understanding generalization in deep learning, and discuss their limitations. Finally, in Section 1.1.3, we rigorously lay out the robust learning framework by extending the standard learning setup presented above, and present adversarial training, which is a principled approach to solve this problem.

1.1.1 Statistical Perspective of Generalization

If the hypothesis class is sufficiently expressive, we expect an approximate ERM solution to have a small empirical loss. This is indeed the case for deep neural networks of even a moderate size. In particular, [YSJ19] showed that 3-layer ReLU networks with $\Omega(\sqrt{n})$ hidden nodes can perfectly fit most datasets. It is then useful to bound the deviation between the generalization error and the empirical error of the model, i.e., $L(\widehat{\mathrm{w}}) - \widehat{L}(\widehat{\mathrm{w}})$, known as the *generalization gap*. In particular, for ERM, it is easy to see that a small generalization gap implies a small excess error (see, e.g., [MRT18]). A crude upperbound on the generalization gap can be obtained via a

uniform deviation bound:

$$L(\widehat{\mathrm{w}}) - \widehat{L}(\widehat{\mathrm{w}}) \leq \sup_{\mathrm{w} \in \mathcal{W}} |L(\mathrm{w}) - \widehat{L}(\mathrm{w})|.$$

The *uniform convergence* of the empirical risk to the expected risk is well-studied in statistical learning theory. This framework establishes that a model with small empirical error also generalizes well to unseen data, provided that the hypothesis class $\mathcal{F}$ is not too *complex* [Vap13]. This result suggests that the learner should find a trade-off between the empirical fit and the model complexity. A common approach is to penalize complex models by adding a *regularization term* to the empirical error, and solving the following *regularized empirical risk minimization (R-ERM)* problem:

$$\min_{\mathrm{w} \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \ell(f_{\mathrm{w}}(\mathrm{x}_i), \mathrm{y}_i) + \lambda R(\mathrm{w}),$$

where $R(\cdot)$ is the *regularizer*, and λ is the associated *regularization parameter*. The regularizer often captures some norm of the model parameters. For example, in linear regression, ℓ_1-penalty, or ℓ_2-penalty, or a combination of both, are often used to regularize the empirical risk [NH92].

Statistical learning theory provides several useful analytical tools to control the uniform deviation bound associated with a hypothesis class. In particular, *Rademacher complexity* is a sample-dependent *measure of complexity* of a hypothesis class that gives tight upper- and lower-bounds on uniform convergence [BM02, KP00, Kol01]. The empirical Rademacher complexity of a function class $\mathcal{F}$ with respect to a sample $\mathcal{S}$ of size n is defined as:

$$\widehat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}) := \mathbb{E}_{\sigma} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(\mathrm{x}_i) \right],$$

where σ_i are i.i.d. Rademacher random variables. Roughly speaking, Rademacher complexity measures how well, on average, functions from the hypothesis class $\mathcal{F}$ restricted to the sample $\mathcal{S}$ correlate with random noise.

We summarize a few important recent advances in theoretical deep learning that bound the generalization gap using tools from statistical learning theory. In particular, controlling the Rademacher complexity of neural networks with bounded parameter norms – directly, or indirectly through other complexity measures – has been an active area of research [Bar98, NTS15, GRS18, BFT17]. Work of [NTS15] introduced the notion of ℓ_p-*path norm* of a neural network, which is simply the ℓ_p-norm of the vector that collects the product of the weights along each path from input to output. More formally, ℓ_p-*path norm* of a network computes $\psi_p(\mathrm{w}) := \|\pi(\mathrm{w})\|_p$, where $\pi(\mathrm{w})$ has one entry per each path from an input node to the output, whose value is given by the product of the weights along that path. They give Rademacher complexity bounds that scale with this quantity as follows:

$$\text{generalization gap} \leq O\left(\sqrt{\frac{(2d^{1/p^*})^{2k}\psi_p(\mathrm{w})^2 \log(d_0)}{n}}\right), \tag{1.1}$$

where d represents the width of the network, and p^* is the dual conjugate of p, that is, $\frac{1}{p} + \frac{1}{p^*} = 1$. Unfortunately, this bound directly depends on both the width and the depth of the network. In particular, unless $p = 1$, the bound scales with d^k, which is an excessively large number even for moderate size networks. Even when $p = 1$, the bound still suffers from an exponential blow up in the network depth, due to the 4^k factor under the squared root.

Following the work of [NTS15], there has been a flurry of research papers aiming to address the explicit dependence of the generalization bound to the architectural parameters, i.e., the depth and the width of the network. In particular, the work of [BFT17] uses a covering number argument and bounds the Rademacher complexity, and therefore the generalization gap, as follows:

$$\text{generalization gap} \leq O\left(\frac{\left(\prod_{i=1}^{k+1} \|\mathrm{W}_i\|_2\right)\left(\sum_{i=1}^{k+1} \frac{\|\mathrm{W}_i^\top\|_{2,1}^{2/3}}{\|\mathrm{W}_i\|_2^{2/3}}\right)^{3/2}}{\sqrt{n}}\right), \tag{1.2}$$

where $\|\mathrm{W}\|_2$ denotes the spectral norm of W, and $\|\mathrm{W}\|_{p,q} := \left(\sum_j \left(\sum_i |\mathrm{W}_{i,j}|^p\right)^{q/p}\right)^{1/q}$

is the ℓ_q-norm of the vector that collects the ℓ_p-norm of the columns of W. In a related line of research, [NBS17] leverage PAC-Bayes theory and prove the following bound:

$$\text{generalization gap} \leq O\left(\frac{\left(\prod_{i=1}^{k+1}\|\mathrm{W}_i\|_2\right)\left(dk^2\sum_{i=1}^{k+1}\frac{\|\mathrm{W}_i\|_F^2}{\|\mathrm{W}_i\|_2^2}\right)^{1/2}}{\sqrt{n}}\right), \tag{1.3}$$

where $\|\mathrm{W}\|_F$ denotes the Frobenius norm of matrix W. We would like to make a few remarks regarding the latter two bounds in Equation 1.2 and Equation 1.3.

- First, the work of [BFT17] also shows that Rademacher complexity is lower bounded by the product of the spectral norm of weight matrices, i.e., $\prod_{i=1}^{k+1}\|\mathrm{W}_i\|_2$. Therefore, when bounding the generalization gap via a uniform deviation bound, an implicit exponential dependence on the depth of the network is unavoidable. In particular, this exponential blow-up show up explicitly in the bounds provided by [BFT17] and [NBS17] in Equation 1.2 and Equation 1.3.

- Second, we note that the bound in [BFT17] is never worse than the bound in [NBS17]. This can be shown by observing that $\|\mathrm{W}_i^\top\|_{2,1}^2 \leq d\|\mathrm{W}_i\|_F^2$, for any individual weight matrix W_i. The claim follows from the fact that $\|\mathrm{v}\|_{2/3} \leq (k+1)\|\mathrm{v}\|_2$, where $v_i = \|\mathrm{W}_i\|_F$ for all layers $i \in [k+1]$.

- Third, as noted in [GRS18], both of these bounds quickly explode as network size increases. For example in [BFT17] – even if we ignore the product of the spectral norms – the right hand side of Equation 1.2 is at least in the order of $\Omega(\sqrt{\frac{k^3}{n}})$, since $\frac{\|\mathrm{W}_i^\top\|_{2,1}}{\|\mathrm{W}_i\|_2} \geq 1$ for any i. As for the bound in [NBS17], ignoring the product of spectral norms again, the right hand side of Equation 1.3 scales at least as $\Omega(\sqrt{\frac{dk^3}{n}})$, since $\frac{\|\mathrm{W}_i\|_F}{\|\mathrm{W}_i\|_2} \geq 1$ for any i. These bounds, therefore, become trivial for large values of d and/or k.

In a subsequent work, aiming to address the issues in the third remark above, [GRS18] give an alternative analysis of Rademacher complexity, and provide the

following guarantee on the generalization gap:

$$\text{generalization gap} \leq O\left(\frac{\sqrt{k}\prod_{i=1}^{k+1}\|\mathrm{W}_i\|_F}{\sqrt{n}}\right), \tag{1.4}$$

While this bound avoids an explicit dependence on the network width, it still suffers from an implicit dependence on d. In fact, the implicit exponential dependence on the network depth is even more exacerbating compared to [BFT17] and [NTS15], as the Frobenius norm in the right hand side of Equation 1.4 can be $\sqrt{\text{width}}$ larger than the spectral norm. It is, however, not possible to give a direct comparison – one that is consistent across all regimes of width and depth parameters – between [GRS18] and the other two bounds in Equation 1.2 and Equation 1.3.

Finally, we would like to remark that in practice, evaluating these bounds on common network architectures that are trained on real datasets, often yield vacuous generalization bounds. In fact, there are empirical evidences showing that the norms of weight matrices can *increase* with the sample size, suggesting that such norm-based generalization bounds might fail to explain generalization in deep learning [NK19].

We remind the reader that despite the non-convexity of the loss landscape associated with deep neural networks, local-search heuristics often succeed in finding networks that can nearly perfectly fit the sample. From a practical perspective, therefore, the main question here is *how to ensure neural networks that are trained to a reasonably small empirical error can also generalize to unseen data?* The literature suggests a few potential answers to this question, as we detail below.

First, it has been widely observed that over-parameterized neural networks trained with pure first-order optimization methods such as (S)GD – in absence of any explicit regularization – can still generalize reasonably well [ZBH+16]. In fact, increasing the network size beyond the interpolation point – where the models can perfectly fit the dataset – does not seem to hurt the generalization ability of models that are trained using (S)GD, somewhat contradictory to the traditional view of over-fitting in

statistical learning theory [NTS14].

It was then conjectured that the optimization algorithms used to train deep neural networks are implicitly biased towards model with small generalization error [NTS14]. Thenceforth, characterizing the *implicit bias* of optimization algorithms in learning over-parameterized models for specific learning problems – from linear regression [GLSS18a] and matrix factorization [GWB+17, ACHL19], to linear binary classification [SHN+18, JT19a] and neural networks [CB20] – has been a central theme in the machine learning literature. At a high level, a typical result of this kind establishes that, first-order methods tend to find near-optimal empirical fits that are close to initialization [GLSS18a, AH19], or have a small norm [SHN+18, JT19a].

On the other hand, there is growing evidence that implicit bias may be unable to explain generalization even in a simpler setting of stochastic convex optimization [DFKL20]. In particular, [SPR18] exhibit a learning problem where gradient flow – that is, gradient descent with an infinitely small step size – diverges from the closest point to the initialization. In a subsequent work, [DFKL20] provide a systematic approach to assess if the implicit bias of an optimization algorithm can be represented by a *reasonable* regularizer. At a high level, if a regularizer $R(\cdot)$ is meant to capture the implicit bias of an optimization algorithm $\mathcal{A}$, then the set of models that simultaneously beat $\mathcal{A}$ on both the empirical loss $\widehat{L}(\cdot)$ and the regularizer $R(\cdot)$ should be small. By constructing convex learning problems where SGD provably converges to solutions which are simultaneously suboptimal both in terms of the empirical loss and any reasonable regularizer, the work of [DFKL20] rejects the possibility that the implicit bias of SGD can be captured by any reasonable regularizer.

Theoretical aspects aside, from an empirical perspective, most real-world state-of-the-art deep learning systems do employ various forms of *explicit* architectural and algorithmic regularization – the list includes but is not limited to early stopping, weight decay, max-norm regularization [SHK+14], weight normalization [IS15, BKH16, SK16],

residual connections [HZRS16a], Jacobian penalty [RVM+11], and dropout [HSK+12]. Therefore, implicit bias might not give the whole picture when we aim to understand the empirical success of deep learning.

Second, as suggested by the R-ERM framework, one can choose to explicitly regularize the empirical objective. In particular, weight-decay, which essentially penalizes the ℓ_2-norm of the parameters, is often leveraged in practice and has shown effective in helping generalization. Other norm-based regularizer, such as ℓ_2 max-norm regularization (also known as per-unit regularization), has also been reported useful in improving generalization [SHK+14]. Both of these measures can indeed be used to provide generalization error bounds for deep neural networks [NTS15]. In fact, max-norm regularization is closely related to path-norm [NTS15], for which we have already introduced a generalization error bound in Equation 1.1.

Another tempting idea, motivated by the Structural Risk Minimization framework [VC74] , is to directly regularize the empirical objective by the complexity measures given in different upperbounds on the generalization gap, such as those presented in Equations 1.1, 1.2, 1.3, and 1.4. However, when the complexity measure is mathematically complicated, as in the case of spectrally-normalized generalization error bounds in Equation 1.2 and Equation 1.3, it can impose additional computational overhead on the learning algorithm. Furthermore, such a complicated regularizer can potentially introduce new optimization barriers by changing the loss landscape in non-trivial ways. More importantly, as suggested by a recent careful empirical analysis [JNM+19], some of the complexity measures present in the upperbounds, including the product of the spectral norms, can even negatively correlate with the generalization of the trained models. Regardless, we note that there has been a few efforts to efficiently incorporate some of these complexity measures, path-norm in specific, approximately, through the geometry of the optimization method [NSS15].

Third, besides traditional, norm-based regularizers, deep learning practitioners

often employ more exotic, algorithmic heuristics such as early stopping, layer normalization [IS15, BKH16, SK16], residual connections [HZRS16a], and dropout [HSK+12, SHK+14]. This thesis is especially concerned about the theoretical foundations of such algorithmic heuristics. How does an algorithmic heuristic explicitly regularize the empirical objective? How does the explicit regularizer help generalization? How efficiently does the heuristic find a good solution? A comprehensive theory of deep learning needs to rigorously address these questions; yet, at this time, these questions are far from understood.

In this thesis, we focus on dropout, wherein at each step of (S)GD, each node in the network is dropped independently and identically according to a Bernoulli random variable with parameter θ. Formally, let $\mathrm{b} = \{\mathrm{B}_i\}_{i=1}^{k}$, where $\mathrm{B}_i = \mathrm{diag}\left[b_{i,1}, \ldots, b_{i,d_i}\right]$ represents the dropout pattern in the i^{th} layer with Bernoulli random variables on the diagonal. If $\mathrm{B}_i(j,j) = 0$ then the j^{th} hidden node in the i^{th} layer is dropped, i.e., it does not contribute in computing the function, and does not participate in the gradient updates. We refer to the parameter $1-\theta$ as the *dropout rate*; smaller θ means higher rate of dropping the corresponding node. Given this formalism, we view dropout, algorithmically, as an instance of SGD on the following objective over w:

$$\widehat{L}_\theta(\mathrm{w}) := \mathbb{E}_{\mathrm{b}}\left[\frac{1}{n}\sum_{i=1}^{n}\ell(f_{\mathrm{w,b}}(\mathrm{x}_i), \mathrm{y}_i)\right],$$

where $f_{\mathrm{w,b}} : \mathrm{x} \mapsto \mathrm{W}_{k+1}\mathrm{B}_k\sigma(\mathrm{W}_k\sigma(\cdots \mathrm{B}_1\sigma(\mathrm{W}_1\mathrm{x})\cdots))$ represents the neural networks sampled according to the dropout pattern.

1.1.2 Computational Perspective of Generalization

The loss landscape associated with deep neural networks is often non-convex. In practice, the back-propagation algorithm, which is an efficient implementation of gradient descent for hierarchical architectures, is used to train deep neural networks. Over-parameterized neural networks trained using back-propagation often fit the sample almost perfectly and also generalize well. A central theme for theoretical

research in deep learning is then the following question: *how can a simple local-search method often succeed in finding an approximate global minimizer despite the non-convexity of the problem*? In recent years, there has been a flurry of studies trying to rigorously address this question. These efforts mainly fall into two categories, as we detail below.

According to the first camp, the success of first-order methods in deep learning is associated with the geometric properties of the loss landscape [CHM+15]. In particular, it was conjectured that sub-optimal critical points are benign, in the sense that all local optima are global, and all saddle points are *strict*. Formally, let $L : \mathcal{W} \to \mathbb{R}$ be a twice differentiable function and let $\mathrm{w} \in \mathcal{W}$ be a critical point of L. Then, w is a *strict saddle point* of L if the Hessian of L at w has at least one negative eigenvalue, i.e. $\lambda_{\min}(\nabla^2 L(\mathrm{w})) < 0$. Furthermore, L satisfies *strict saddle property* if all saddle points of f are strict saddle. This property allows first-order methods to efficiently escape the saddle points and converge to a local optimum [LSJR16, GHJY15], which will be global if the landscape doesn't have any poor local optimum.

Following this benign landscape conjecture, there has been a flurry of works on studying the landscape of different machine learning problems, including low rank matrix recovery [BNS16], generalized phase retrieval problem [SQW16], matrix completion [GLM16], deep linear networks [Kaw16], matrix sensing and robust PCA [GJZ17] and tensor decomposition [GHJY15], making a case for global optimality of first-order methods in deep learning.

However, there is ample evidence for refuting the benign landscape conjecture in deep learning [SCP16, Kaw16, ZL17, YSJ18, SS18]. In particular, work of [Kaw16] showed that even for deep *linear* networks, the loss landscape has non-strict saddle points where the Hessian has no negative eigenvalues. It was also shown, empirically and theoretically, that even in two-layer ReLU networks, spurious local optima are common [SS18]. Furthermore, the landscape approach ignores the role of initialization,

which has shown to be crucial for the success of gradient-based methods in deep learning [SMDH13].

Given the limitations of the landscape approach, the second camp posits that understanding optimization in deep learning requires a careful analysis of the trajectories traversed at time of training [ZSJ+17, BG17, Tia17, BGMSS18, AGCH19, ACHL19]. In particular, there has been significant recent progress in such trajectory-based analysis of local-search heuristics in the so-called *lazy regime*; wherein under certain initialization, learning rate, and over-parameterization requirements, the iterates of (S)GD tend to stay close to initialization. In such settings, therefore, a first-order Taylor expansion of the t-th iterate around initialization, i.e. $f_{\mathrm{w}_t}(\mathrm{x}) \approx f_{\mathrm{w}_1}(\mathrm{x}) + \langle \nabla f_{\mathrm{w}_1}(\mathrm{x}), \mathrm{w}_t - \mathrm{w}_1 \rangle$, can be used as a proxy to track the evolution of the network predictions [COB18, LXS+19].

Leveraging the approximate linearity of the neural networks in a small neighborhood of the initialization, training in lazy regime reduces to finding a linear predictor in the reproducing kernel Hilbert space (RKHS) associated with $\nabla f_{\mathrm{w}_1}(\cdot)$, the gradient of the network at initialization. In the limit of over-parameterization, the induced kernel $k(\mathrm{x}, \mathrm{x}') = \langle \nabla f_{\mathrm{w}_1}(\mathrm{x}), \nabla f_{\mathrm{w}_1}(\mathrm{x}') \rangle$ is often referred to as the *Neural Tangent Kernel (NTK)* [JGH18]. The dynamics of (S)GD is then completely governed by the NTK, and for that reason, lazy regime is also referred to as the *kernel regime*.

Building on the standard convergence guarantees of (S)GD in learning linear models, there has been a flurry of recent work establishing that in the lazy regime, gradient-based methods can efficiently find a solution with vanishing empirical and expected error [LL18, DZPS19, Dan17, ZCZG18, AZLL19, SY19, ADH+19, CG19, OS20, NS19, JT19b]. Furthermore, akin to implicit bias of (S)GD in learning linear models, it is also established that in the lazy regime, (S)GD is implicitly biased towards minimum norm solutions (with respect to the RKHS norm induced by the NTK) [COB18, ADH+19, CB20]. These results suggest an equivalence between deep learning and learning with classical kernel machines: instead of training deep

neural networks and learning hierarchical features, one can use kernel methods by leveraging fixed features that are solely determined by the gradient of the model at the initialization.

Although the idea of reducing deep learning to learning linear predictors in an appropriate RKHS seems promising, it is far from capturing the reality. In particular, empirical evidence suggests that in a typical real-word deep learning scenario, deep neural networks leave the lazy regime and explore the parameter space beyond a small neighborhood of initialization (see, e.g., [NK19]).

Furthermore, a series of empirical and theoretical works suggest that training over-parameterized neural networks with gradient-based methods induce *rich* implicit biases that cannot be captured by any RKHS norm. For example, the implicit bias of gradient descent in training deep linear convolutional networks corresponds to bridge regularization in the frequency domain [GLSS18b]. In the over-parameterized matrix learning problems, [GWB+17] conjectured and provided empirical evidence that gradient descent is implicitly biased towards solutions with minimum nuclear norm; this conjecture was later formally proved by [LMZ18] under the restricted isometry property. In a subsequent work, for matrix sensing and matrix completion problems, [ACHL19] showed that learning deep linear networks with gradient descent has an implicit tendency towards low rank solutions. Finally, [CB20] showed that gradient flow in two-layer ReLU networks is implicitly biased towards the max-margin solution with respect to a variation norm. We note that none of these implicit biases can be represented as an RKHS norm, and therefore, cannot be explained by the lazy regime.

1.1.3 Robust Learning

Recent advances of deep learning in core machine learning tasks such as computer vision [KSH12, KTS+14, OBLS14], natural language processing [ZZL15, CVMG+14],

and reinforcement learning [MKS+15, SHM+16] has produced some major technological breakthroughs in artificial intelligence. However, deep learning systems can be extremely brittle against small distribution shift in data, which can arise naturally or adversarially, due to a change in environment or a malicious actor [BCM+13, SZS+14, NYC15]. In particular, recent studies has shown that neural networks are highly susceptible to *adversarial examples* – inputs that are contaminated with tiny, imperceptible adversarial perturbations, yet can *fool* the model to make a wrong prediction [GSS15, MDFF16, CW17]. Figure 1-1 illustrates one such adversarial example: a perturbed image of a pig (on the right) – indistinguishable from the correctly classified original image (on the left) – is wrongly classified as a wombat by the same model.

The rise of deep learning in artificial intelligence has made it an indispensable tool in autonomous systems with minimal to no human assistance, including self-driving cars, domestic robots, automated medical delivery systems, and surgical robots. In such critical high-stakes applications, risk of failure due to adversarial examples can be catastrophic, and should be avoided at all costs. As a result, in recent years, *robust learning* has arisen as an active area of research within the machine learning community. The goal of *robust learning* is to find models that yield reliable predictions on test data notwithstanding adversarial perturbations. In the following, we formally introduce robust learning by extending the standard learning setting discussed in the previous sections.

First, we need to choose a *threat model*, a class of adversarial perturbations that we allow at the test time, and we aim to be robust against. At an abstract level, an adversarial perturbation is simply a function that maps an input $\mathrm{x} \in \mathcal{X}$ to another *look-alike* input $\mathrm{x}' \in \mathcal{X}$. A natural threat model in many applications is the class of ℓ_p-bounded additive perturbations $\Delta_{p,\nu}(\mathrm{x}) := \{\mathrm{x} + \delta,\ \|\delta\|_p \leq \nu\}$, where $\nu \geq 0$ is the perturbation budget that limits the power of the adversary. Under this model,

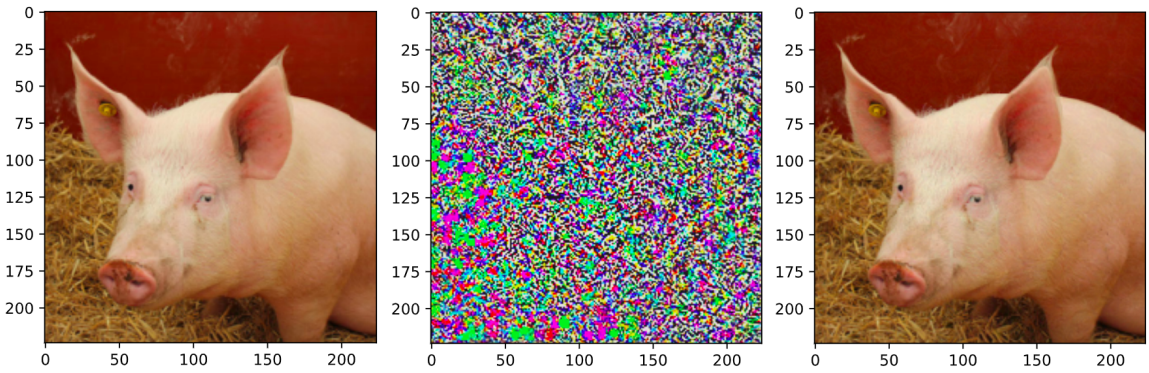

Figure 1-1. (**left**) An image of a pig, predicted correctly by the model; (**middle**) image of an adversarial perturbation scaled by a factor of 50 to be visible; and (**right**) the image in the left after perturbation; predicted as a wombat by the same model. Images are downloaded from Adversarial Robustness - Theory and Practice.

an adversarial example is generated as $\mathrm{x}' = \mathrm{x} + \delta$ for an appropriate choice of a norm-bounded δ. For example, in an image classification problem like the one in Figure 1-1, the class of ℓ_∞-bounded attacks is a natural choice for imperceptible perturbations. In this thesis, we focus on ℓ_2-bounded adversarial attacks, which is denoted as $\Delta := \Delta_{2,\nu}$ for the simplicity of the notation.

Given a hypothesis class and a threat model, robust learning can be viewed as a game played between a learner and an adversary. The single goal of the adversary is to *trick* the learner into making a wrong prediction on a perturbed test sample. That is, at any given datapoint $(\mathrm{x}, y) \sim \mathcal{D}$, the adversary aims to solve the following problem against the learner f_{w}:

$$\max_{\mathrm{x}' \in \Delta_{p,\nu}(\mathrm{x})} \mathbb{I}\{f_{\mathrm{w}}(\mathrm{x}') \neq y\}, \tag{1.5}$$

which is precisely the loss that the learner incurs at (x, y). Here, $\mathbb{I}\{\cdot\}$ denotes the indicator function. In particular, the maximization problem above can only take values in $\{0, 1\}$. If the maximum equals zero, then the model f_{w} is said to be robust against the threat model $\Delta_{p,\nu}$ at (x, y). Otherwise, there exists an $\mathrm{x}' \in \Delta_{p,\nu}(\mathrm{x})$ such that $f_{\mathrm{x}}(\mathrm{x}') \neq y$, and therefore, the model fails to robustly predict the label of x. The

goal of the learner is then to minimize the *robust error* in expectation over samples drawn from $\mathcal{D}$:

$$\min_{\mathrm{w}\in\mathcal{W}} L_{\mathrm{rob}}(\mathrm{w}) := \mathbb{E}_{(\mathrm{x},y)\sim\mathcal{D}}\left[\max_{\mathrm{x}'\in\Delta_{p,\nu}(\mathrm{x})} \mathbb{I}\{f_{\mathrm{w}}(\mathrm{x}') \neq y\}\right]. \tag{1.6}$$

Given n samples $\{(\mathrm{x}_i, y_i)\}_{i=1}^n$ drawn i.i.d. from the source distribution $\mathcal{D}$, the goal of robust learning is then to find $\widehat{\mathrm{w}}$ which enjoys a small *robust risk* $L_{\mathrm{rob}}(\cdot)$ defined above. In particular, a hypothesis $f_{\widehat{\mathrm{w}}}$ with vanishing $L_{\mathrm{rob}}(\widehat{\mathrm{w}})$ is robust against the threat model $\Delta_{p,\nu}$, in expectation over the random draw of a test sample.

Adversarial training is a recent promising approach that addresses the above robust learning problem [MMS+18] in a principled fashion. In adversarial training, the $0-1$ loss inside the expectation is replaced with a convex surrogate such as the cross entropy loss, and the expected value is estimated using a sample average, which leads to the following min-max optimization problem:

$$\min_{\mathrm{w}\in\mathcal{W}} \widehat{L}_{\mathrm{rob}}(\mathrm{w}) := \frac{1}{n}\sum_{i=1}^{n} \max_{\mathrm{x}'_i\in\Delta_{p,\nu}(\mathrm{x}_i)} \ell(y_i, f_{\mathrm{w}}(\mathrm{x}'_i)). \tag{1.7}$$

This formalism gives a unifying view over much of the prior work on both adversarial attacks and defense strategies. In particular, the difference between different strategies mainly stems from the particular choice of the surrogate loss and the optimization method used to solve the saddle-point problem in Equation 1.7 [MMS+18].

In practice, alternating local-search heuristics are often employed to solve the above min-max problem – both for finding an attack in the inner-max problem, and finding a robust model in the outer-max problem. In a typical algorithm of this kind, at each iterate, the learner first attempts to simulate the adversary by finding adversarial examples $\{\widehat{\mathrm{x}}_1, \ldots, \widehat{\mathrm{x}}_n\}$ that approximately maximize the inner-max problems. The learner then updates the weights to minimize the empirical loss evaluated on the perturbed dataset, i.e., $\widehat{L}(\mathrm{w}; \{(\widehat{\mathrm{x}}_i, y_i)\}_{i=1}^n) := \frac{1}{n}\sum_{i=1}^n \ell(f_{\mathrm{w}}(\widehat{\mathrm{x}}_i), y_i)$, and continues with the next iterate.

Finally, we emphasize that most attack strategies can be viewed as some form of the projected gradient descent (PGD) algorithm for solving the constrained inner-max problem. Given sample (x, y), such methods initialize the adversarial example at an arbitrary point $\mathrm{x}'_1 \in \Delta_{p,\nu}(\mathrm{x})$, and iteratively update it by taking a step in the direction of the gradient of the surrogate loss, followed by a projection onto the feasible set of the corresponding threat model:

$$\mathrm{x}'_{t+1} = \Pi_{\Delta_{p,\nu}}[\mathrm{x}'_t + \eta_t \nabla \ell(f_{\mathrm{w}}(\mathrm{x}'_t), y)],$$

where η_t is the step size, and $\Pi_{\mathcal{C}}(\mathrm{v})$ projects v onto $\mathcal{S}$ with respect to the ℓ_2-norm. For example, for an ℓ_∞ threat model ($\Delta_{\infty,\nu}$), the projection simply corresponds to clipping entries of the adversarial example x' if they leave the range $(\mathrm{x} - \nu, \mathrm{x} + \nu)$. In particular, a popular instance of PGD for the ℓ_∞ threat model, the Fast Gradient Signed Method (FGSM) [GSS15] computes an adversarial example as follows:

$$\mathrm{x}' = \mathrm{x} + \nu \operatorname{sgn}[\nabla_{\mathrm{x}} \ell(f_{\mathrm{w}}(\mathrm{x}), y)],$$

which is equivalent to a single step of PGD with infinite step size for an ℓ_∞ bounded adversary.

1.2 Related Work

In this dissertation, we develop a theoretical foundation around deep learning, focusing on regularization and robustness due to local-search heuristics often used in practice. Following the empirical success of such algorithmic heuristics, there has been numerous theoretical studies aiming to explain why and when they work. In this section, we survey the previous art around the main focus of the thesis; we review the literature related to the theoretical foundations of dropout in Section 1.2.1, and the previous work on adversarial training in Section 1.2.2, respectively.

1.2.1 Regularization due to Dropout

Early theoretical studies on dropout focus on understanding the regularization due to dropout in simpler models. After introducing dropout [HSK+12], in a follow-up work, the authors showed that dropout in linear regression amounts to a weight decay penalty [SHK+14]. A similar result was also shown by [BS13], who further establish that for a single sigmoidal unit trained to minimize the cross entropy loss, dropout induces a weight decay penalty which is adaptively scaled by the second moment of the input and the dropout rates. A more general result by [WWL13] showed that for generalized linear models, dropout performs an adaptive regularization which is equivalent to a data-dependent scaling of the weight decay penalty. In contrast, [HL15] argued that in linear classification, the regularizer due to dropout can radically differ from weight decay, in the sense that it can be non-convex and non-monotone in individual weight parameters.

In a related strand of research, several early studies focus on understanding how dropout helps with generalization, using tools from statistical learning theory. In particular, the work of [McA13] leverages the PAC-Bayes framework and provides a generalization bound for dropout that decays with the dropout rate; interestingly, the connection between dropout and weight decay is also evident in their analysis. In a follow-up work by the authors of [WWL13], they show that under a certain topic model assumption on the data, dropout in linear classification can improve the decay of the excess risk of the empirical risk minimizer [WFWL14].

1.2.2 Adversarial Training

Adversarial training, and theoretical studies around it, are fairly new topics in the machine learning literature. Some of the earliest studies focus on adversarial training of linear models, where the optimal attack has a simple closed-form expression, which mitigates the challenge of analyzing the optimization method used for the

inner-max problem. In particular, [CRWP19, LXXZ20] give robust generalization error guarantees for adversarially trained linear models under a margin separability assumption. The hard margin assumption was relaxed by [ZFG21] who give robust generalization guarantees for distributions with agnostic label noise.

There has also been a few efforts to understand adversarial training in non-linear neural networks. The works of [GCL+19] and [ZPD+20] study the convergence of adversarial training in the lazy regime. Under specific initialization and width requirements, these works guarantee small robust training error with respect to the attack strategy that is used in the inner-loop, without explicitly analyzing the convergence of the attack. [GCL+19] assume that the activation function is smooth and require that the width of the network, as well as the overall computational cost, is exponential in the input dimension. The work of [ZPD+20] partially addresses these issues. In particular, their results hold for ReLU neural networks, and they only require the width and the computational cost to be polynomial in the input parameters.

1.3 Contributions

In this dissertation, we build a theoretical framework around the regularization and robustness imparted by local-search heuristics in deep learning. First, focusing on dropout, in the next four chapters of this thesis we rigorously analyze several important theoretical questions that were poorly understood before this work. *How does dropout explicitly regularize the empirical objective?* We formally answer this question in Chapter 2 and Chapter 3, focusing on shallow and deep linear networks as a case study. *How does the explicit regularizer due to dropout provide capacity control in deep learning?* In Chapter 4 of this thesis, we answer this question by providing precise sample complexity bounds for the dropout rule, in the context of matrix sensing problem, and for two-layer ReLU neural networks. *Does dropout, as an iterative*

local-search heuristic, converge to a solution with small generalization error? In Chapter 5, we give precise iteration complexity results for learning a two-layer ReLU neural network with dropout. Finally, in the last chapter of this thesis, we shift gears towards the robustness due to local-search heuristics. In particular, we seek to answer *can adversarial training provably robustly learn neural networks?* In Chapter 6, we answer this question affirmatively by providing precise iteration complexity guarantees for end-to-end adversarial training of two-layer neural networks with Leaky-ReLU activation. In the following, we detail the main contributions of this dissertation.

1.3.1 Explicit Regularization Due to Dropout

A natural first step toward understanding generalization due to dropout, is to instantiate the explicit form of the regularizer due to dropout, and analyze the *dropout objective*, i.e., the resulting regularized risk minimization problem that dropout aims to solve. This is precisely the focus of our study in Chapters 2 and 3, as we detail below.

In Chapter 2, we study dropout in linear regression with shallow linear neural networks. We show that the regularizer due to dropout is equal to the ℓ_2-path norm of the network. We then prove that at the minima of the dropout objective, the regularizer induced by dropout amounts to a nuclear norm penalty. This allows us to completely characterize the global minima of the dropout objective, despite the objective being non-convex (Theorem 6).

We also describe the optimization landscape of the dropout problem in the case of two-layer autoencoders with tied weights. In particular, we show that for a sufficiently small dropout rate, all local minima of the dropout objective are global and all saddle points are non-degenerate (Theorem 7). This allows dropout to efficiently escape saddle points and converge to a global optimum.

In Chapter 3, we extend the results of Chapter 2 to deep linear networks with

arbitrary architecture. First, we show that dropout induces a data-dependent regularizer that includes, among other terms, the ℓ_2-path norm of the network. We then completely characterize the global minima of the dropout objective, under a simple eigengap condition (see Theorem 10). This gap condition depends on the model, the data distribution, the network architecture and the dropout rate, and is always satisfied by two-layer linear networks, as well as deep linear networks with one output neuron. In particular, under this gap condition, at any global minimum of the objective, the regularizer induced by dropout boils down to the nuclear norm of the network.

1.3.2 Statistical Guarantees for Dropout

As we formally show in Chapters 2 and 3, training deep linear networks with dropout induces a rich inductive bias that is captured by the nuclear norm of the network. The next natural step is to investigate how such an inductive bias controls the capacity of the underlying model. In Chapter 4, we study capacity control due to dropout through the lens of Rademacher complexity, and establish precise generalization bounds for the matrix sensing problem (Theorem 11) and two-layer neural networks with ReLU activations (Theorem 12).

We formally argue that dropout training alone does not directly control the norms of the weight vectors (Proposition 2); and therefore, to prove capacity control due to dropout, one cannot simply appeal to norm-based generalization error bounds discussed in Section 1.1.1. Our generalization bounds in Chapter 4 are solely in terms of the value of the dropout regularizer and without additional norm constraints on the predictors. In the case of two-layer neural networks with ReLU activation, this is a significant departure from most of the prior work wherein dropout is analyzed in conjunction with additional norm-based capacity control, e.g., max-norm [WZZ+13, GZ16], or ℓ_p norm on the weights of the model [ZW18].

We also provide extensive numerical evaluations for validating our theory including verifying that the proposed theoretical bound on the Rademacher complexity is predictive of the observed generalization gap.

1.3.3 Computational Guarantees for Dropout

In Chapters 2 - 4 we rigorously argue for the inductive bias due to dropout and provide precise sample complexity guarantees for dropout training; however, our results in those chapters completely ignores the computational complexity of learning with dropout. In Chapter 5, we study dropout through the lens of computational learning theory and focus on the iteration complexity of learning non-linear neural networks with dropout.

We leverage recent advances in the theory of deep learning in the lazy regime and extend convergence guarantees and generalization bounds for GD-based methods with explicit regularization due to dropout. In Theorem 14, we give precise non-asymptotic convergence rates for achieving ϵ-subotimality in the generalization error via dropout training in two-layer ReLU networks. Furthermore, we show that dropout training implicitly compresses the network. In particular, we show in Theorem 15 that there exists a sub-network, i.e., one of the iterates of dropout training, that can generalize as well as any complete network. We also provide empirical evidence (see Figure 5-2) to support the compression results.

1.3.4 Robustness Guarantees for Adversarial Training

Recall that in adversarial training, robust learning is formulated as a min-max optimization problem wherein the classification error is replaced by a *convex upperbound*, and alternating local-search heuristics are used to solve the resulting saddle point problem. In Chapter 6, we propose a simple modification of adversarial training: when solving the inner-max problem searching for an "optimal" perturbation vector

to attack the current model, we reflect the convex upperbound about the origin, which yields a *concave lowerbound* for the misclassification error (see Figure 6-1 and Algorithm 5).

With this simple modification, under a margin separability assumption, we provide convergence guarantees for PGD attacks on two-layer neural networks with leaky ReLU activation (Lemma 18), which is the first of its kind in the literature. Furthermore, in Theorem 16 we give global convergence guarantees and establish learning rates for adversarial training. Notably, our guarantees hold for *any bounded initialization* and *any width* – a property that is not present in the previous works in the NTK regime [GCL+19, ZPD+20].

We also provide extensive empirical evidence evaluating the idea of reflecting the surrogate loss in the inner loop. First, we show that reflecting the loss can indeed lead to finding "better" attacks in the inner-max loop (Figure 6-2), resulting in adversarially trained models that are more robust (Table 6-I). We then propose a simple, greedy heuristic to extend our approach to the multi-class setting. We show empirically, that this simple extension does not have a significant impact on the test time performance of the adversarially trained models (Table 6-II).

Chapter 2

Dropout Regularizer: Shallow Linear Networks

Dropout regularizes the model by dropping a random subset of hidden nodes at each iterate of back-propagation. A natural first step to understand how dropout helps with generalization is to extract the explicit regularization due to dropout, and study the optima of the resulting regularized risk minimization problem. We argue that a prerequisite for understanding regularization due to dropout, is to analyze its behavior in simpler models. Therefore, in this chapter, we focus on dropout in linear-regression with two-layer linear networks. For simplicity of analysis, we also assume that the input marginals are isotropic – a condition that we lift in the next chapter, when we study dropout in deep linear networks.

Formally, we consider the following learning problem. Let $\mathrm{x} \in \mathbb{R}^{d_0}$ represent an input feature vector with some unknown distribution $\mathcal{D}$ such that $\mathbb{E}_{\mathrm{x}\sim\mathcal{D}}[\mathrm{x}\mathrm{x}^\top] = \mathrm{I}$. The output label vector $\mathrm{y} \in \mathbb{R}^{d_2}$ is given as $\mathrm{y} = \mathrm{M}\mathrm{x}$ for some $\mathrm{M} \in \mathbb{R}^{d_2 \times d_0}$. We consider the hypothesis class represented by a single hidden-layer linear network parametrized as $h_{\mathrm{U},\mathrm{V}}(\mathrm{x}) = \mathrm{U}\mathrm{V}^\top\mathrm{x}$, where $\mathrm{V} \in \mathbb{R}^{d_0 \times r}$ and $\mathrm{U} \in \mathbb{R}^{d_2 \times r}$ are the weight matrices in the first and the second layers, respectively. The goal of learning is to find weight matrices U, V that minimize the *expected loss*

$$L(\mathrm{U}, \mathrm{V}) := \mathbb{E}_{\mathrm{x}\sim\mathcal{D}}[\|\mathrm{y} - h_{\mathrm{U},\mathrm{V}}(\mathrm{x})\|^2] = \mathbb{E}_{\mathrm{x}\sim\mathcal{D}}[\|\mathrm{y} - \mathrm{U}\mathrm{V}^\top\mathrm{x}\|^2].$$

A natural learning algorithm to consider is back-propagation with dropout, which can be seen as an instance of stochastic gradient descent on the following objective:

$$L_\theta(\mathrm{U},\mathrm{V}) := \mathbb{E}_{b_i\sim\mathrm{Ber}(\theta),\mathrm{x}\sim\mathcal{D}}\left[\left\|\mathrm{y}-\frac{1}{\theta}\mathrm{U}\,\mathrm{diag}(\mathrm{b})\mathrm{V}^\top\mathrm{x}\right\|^2\right], \tag{2.1}$$

where the expectation is w.r.t. the underlying distribution on data as well as randomization due to dropout (each hidden unit is dropped independently with probability $1-\theta$). This procedure, which we simply refer to as dropout in this chapter, is given in Algorithm 1.

It is easy to check (see Lemma 2 in Section 2.5) that the objective in equation (2.1) can be written as

$$L_\theta(\mathrm{U},\mathrm{V}) = L(\mathrm{U},\mathrm{V}) + \lambda\sum_{i=1}^{r}\|\mathrm{u}_i\|^2\|\mathrm{v}_i\|^2, \tag{2.2}$$

where $\lambda = \frac{1-\theta}{\theta}$ is the regularization parameter, and u_i and v_i represent the i^{th} columns of U and V, respectively. Note that while the goal was to minimize the expected squared loss, using dropout with gradient descent amounts to finding a minimum of the objective in equation (2.2); we argue that the additional term in the objective serves as a regularizer, $R(\mathrm{U},\mathrm{V}) := \lambda\sum_{i=1}^{r}\|\mathrm{u}_i\|^2\|\mathrm{v}_i\|^2$, and is an explicit instantiation of the implicit bias of dropout. Furthermore, we note that this regularizer is closely related to *path regularization* which is given as the square-root of the sum over all paths, from input to output, of the product of the squared weights along the path [NTS15]. Formally, for a single layer network, path regularization is given as

$$\psi_2(\mathrm{U},\mathrm{V}) = \left(\sum_{i=1}^{r}\sum_{j=1}^{d_2}\sum_{k=1}^{d_0}u_{ji}^2v_{ki}^2\right)^{\frac{1}{2}}. \tag{2.3}$$

Interestingly, the dropout regularizer is equal to the square of the path regularizer, i.e. $R(\mathrm{U},\mathrm{V}) = \lambda\psi_2^2(\mathrm{U},\mathrm{V})$. While this observation is rather immediate, it has profound implications owing to the fact that path regularization provides size-independent capacity control in deep learning, thereby supporting empirical evidence that dropout finds good solutions in over-parametrized settings.

In this chapter, we focus on studying the optimization landscape of the objective in equation (2.2) for a single hidden-layer linear network with dropout and the special case of an autoencoder with tied weights. Furthermore, we are interested in characterizing the solutions to which dropout (i.e. Algorithm 1) converges. We make the following progress toward addressing these questions.

1. We formally characterize the inductive bias of dropout. We argue that, when minimizing the expected loss $L_\theta(\mathrm{U}, \mathrm{V})$ with dropout, any global minimum $(\tilde{\mathrm{U}}, \tilde{\mathrm{V}})$ satisfies $\psi_2(\tilde{\mathrm{U}}, \tilde{\mathrm{V}}) = \min\{\psi_2(\mathrm{U}, \mathrm{V}) \text{ s.t. } \mathrm{U}\mathrm{V}^\top = \tilde{\mathrm{U}}\tilde{\mathrm{V}}^\top\}$. More importantly, for auto-encoders with tied weights, we show that all *local* minima inherit this property.

2. Despite the non-convex nature of the problem, we completely characterize the global optima by giving necessary and sufficient conditions for optimality.

3. We describe the optimization landscape of the dropout problem. In particular, we show that for a sufficiently small dropout rate, all local minima of the dropout objective in equation (2.2) are global and all saddle points are non-degenerate. This allows Algorithm 1 to efficiently escape saddle points and converge to a global optimum.

The rest of the chapter is organized as follows. In Section 2.1, we study dropout for single hidden-layer linear auto-encoder networks with weights tied between the first and the second layers. This gives us the tools to study the dropout problem in a more general setting of single hidden-layer linear networks in Section 2.2. In Section 2.3, we characterize the optimization landscape of the objective in (2.2), show that it satisfies the strict saddle property, and that there are no spurious local minima. We specialize our results to matrix factorization in Section 2.4, and in Section 2.6, we discuss preliminary experiments to support our theoretical results.

Algorithm 1: Training a single hidden layer network with dropout

Input: Data $\{(\mathrm{x}_t, \mathrm{y}_t)\}_{t=0}^{T-1}$, dropout rate $1-\theta$, learning rate η

1: Initialize $\mathrm{U}_0, \mathrm{V}_0$

2: **for** $t = 0, 1, \ldots, T-1$ **do**

3: sample b_t element-wise from Bernoulli(θ)

4: update the weights

$$\mathrm{U}_{t+1} \leftarrow \mathrm{U}_t - \eta\left(\frac{1}{\theta}\mathrm{U}_t \operatorname{diag}(\mathrm{b}_t)\mathrm{V}_t^\top \mathrm{x}_t - \mathrm{y}_t\right)\mathrm{x}_t^\top \mathrm{V}_t \operatorname{diag}(\mathrm{b}_t)$$

$$\mathrm{V}_{t+1} \leftarrow \mathrm{V}_t - \eta\mathrm{x}_t\left(\frac{1}{\theta}\mathrm{x}_t^\top \mathrm{V}_t \operatorname{diag}(\mathrm{b}_t)\mathrm{U}_t^\top - \mathrm{y}_t^\top\right)\mathrm{U}_t \operatorname{diag}(\mathrm{b}_t)$$

5: **end for**

Output: $\mathrm{U}_T, \mathrm{V}_T$

2.1 Linear autoencoders with tied weights

We begin with a simpler hypothesis family of single hidden-layer linear auto-encoders with weights tied such that $\mathrm{U} = \mathrm{V}$. Studying the problem in this setting helps our intuition about the implicit bias that dropout induces on weight matrices U. This analysis will be extended to the more general setting of single hidden-layer linear networks in the next section.

Recall that the goal here is to find an autoencoder network represented by a weight matrix $\mathrm{U} \in \mathbb{R}^{d_0 \times r}$ that solves:

$$\min_{\mathrm{U} \in \mathbb{R}^{d_0 \times r}} L_\theta(\mathrm{U}, \mathrm{U}) = L(\mathrm{U}, \mathrm{U}) + \lambda \sum_{i=1}^{r} \|\mathrm{u}_i\|^4, \tag{2.4}$$

where u_i is the i^{th} column of U. Note that the loss function $L(\mathrm{U}, \mathrm{U})$ is invariant under rotations, i.e., for any orthogonal transformation $\mathrm{Q} \in \mathbb{R}^{d \times d}, \mathrm{Q}^\top \mathrm{Q} = \mathrm{Q}\mathrm{Q}^\top = \mathrm{I}_d$, it holds that

$$L(\mathrm{U}, \mathrm{U}) = \mathbb{E}_{\mathrm{x} \sim \mathcal{D}}[\|\mathrm{y} - \mathrm{U}\mathrm{Q}\mathrm{Q}^\top \mathrm{U}^\top \mathrm{x}\|^2] = L(\mathrm{U}\mathrm{Q}, \mathrm{U}\mathrm{Q}),$$

so that applying a rotation matrix to a candidate solution U does not change the value of the loss function. However, the regularizer is not rotation-invariant and clearly depends on the choice of Q. Therefore, in order to solve Problem (2.4), we need to

Algorithm 2: $\texttt{EQZ}(\mathrm{U})$ equalizer of an auto-encoder $h_{\mathrm{U},\mathrm{U}}$

Input: $\mathrm{U} \in \mathbb{R}^{d \times r}$

$\mathrm{G} \leftarrow \mathrm{U}^\top \mathrm{U}$

$\mathrm{Q} \leftarrow \mathrm{I}_r$

for $i = 1$ to r **do**

  $[\mathrm{V}, \Lambda] \leftarrow \texttt{eig}(\mathrm{G})$ {$\mathrm{G} = \mathrm{V}\Lambda\mathrm{V}^\top$ eigendecomposition}

  $\mathrm{w} = \frac{1}{\sqrt{r-i+1}} \sum_{i=1}^{r-i+1} \mathrm{v}_i$

  $\mathrm{Q}_i \leftarrow [\mathrm{w}\ \mathrm{w}_\perp]$ {$\mathrm{w}_\perp \in \mathbb{R}^{(r-i+1)\times(r-i)}$ orthonormal basis for the Null space of w}

  $\mathrm{G} \leftarrow \mathrm{Q}_i^\top \mathrm{G} \mathrm{Q}_i$ {Making first diagonal element zero}

  $\mathrm{G} \leftarrow \mathrm{G}(2:\mathrm{end}, 2:\mathrm{end})$ {First principal submatrix}

  $\mathrm{Q} \leftarrow \mathrm{Q} \begin{bmatrix} \mathrm{I}_{i-1} & 0 \\ 0 & \mathrm{Q}_i \end{bmatrix}$

end for

Output: Q {such that UQ is equalized}

find a rotation matrix that minimizes the value of the regularizer for a given weight matrix.

To that end, let us denote the squared column norms of the weight matrix U by $\mathrm{n_u} = (\|\mathrm{u}_1\|^2, \ldots, \|\mathrm{u}_r\|^2)$ and let $1_r \in \mathbb{R}^r$ be the vector of all ones. Then, for any U,

$$\begin{aligned} R(\mathrm{U}, \mathrm{U}) &= \lambda \sum_{i=1}^{r} \|\mathrm{u}_i\|^4 = \frac{\lambda}{r} \|1_r\|^2 \|\mathrm{n_u}\|^2 \\ &\geq \frac{\lambda}{r} \langle 1_r, \mathrm{n_u} \rangle^2 = \frac{\lambda}{r} \left(\sum_{i=1}^{r} \|\mathrm{u}_i\|^2 \right)^2 = \frac{\lambda}{r} \|\mathrm{U}\|_F^4, \end{aligned}$$

where the inequality follows from Cauchy-Schwartz inequality. Hence, the regularizer is lower bounded by $\frac{\lambda}{r}\|\mathrm{U}\|_F^4$, with equality if and only if $\mathrm{n_u}$ is parallel to 1_r, i.e. when all the columns of U have equal norms. Since the loss function is rotation invariant, one can always decrease the value of the overall objective by rotating U such that UQ has a smaller regularizer. A natural question to ask, therefore, is *if there always exists a rotation matrix Q such that the matrix UQ has equal column norms.* In order to formally address this question, we introduce the following definition.

Definition 1 (Equalized weight matrix, equalized autoencoder, equalizer)**.** *A weight matrix U is said to be equalized if all its columns have equal norms. An autoencoder with tied weights is said to be equalized if the norm of the incoming weight vector*

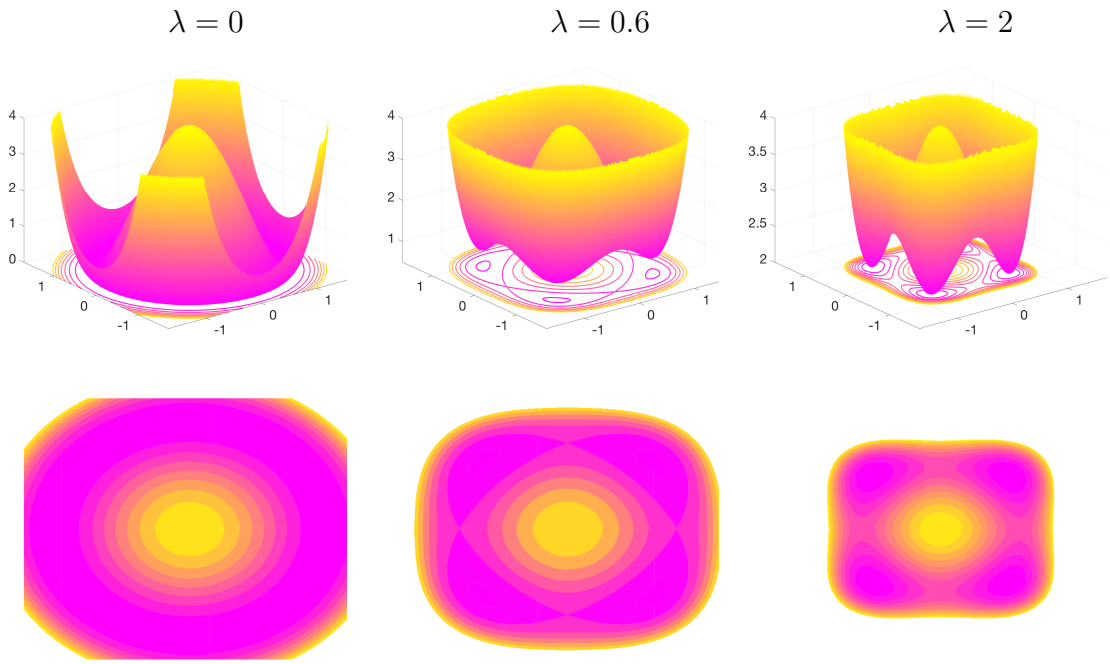

Figure 2-1. Optimization landscape (top) and contour plot (bottom) for a single hidden-layer linear autoencoder network with one dimensional input and output and a hidden layer of width $r = 2$ with dropout, for different values of the regularization parameter λ. Left: for $\lambda = 0$ the problem reduces to squared loss minimization, which is rotation invariant as suggested by the level sets. Middle: for $\lambda > 0$ the global optima shrink toward the origin. All local minima are global, and are equalized, i.e. the weights are parallel to the vector $(\pm 1, \pm 1)$. Right: as λ increases, global optima shrink further.

is equal across all hidden nodes in the network. An orthogonal transformation Q is said to be an equalizer *of U (equivalently, of the corresponding autoencoder) if UQ is equalized.*

Next, we show that any matrix U can be equalized.

Theorem 1. *Any weight matrix $U \in \mathbb{R}^{d \times r}$ (equivalently, the corresponding autoencoder network $h_{U,U}$) can be equalized. Furthermore, there exists a polynomial time algorithm (Algorithm 2) that returns an equalizer for a given matrix.*

The key insight here is that if $\mathrm{G_U} := \mathrm{U}^\top \mathrm{U}$ is the Gram matrix associated with the weight matrix U, then $h_{\mathrm{U,U}}$ is equalized by Q if and only if all diagonal elements of $\mathrm{Q}^\top \mathrm{G_U Q}$ are equal. More importantly, if $\mathrm{G_U} = \mathrm{V} \Lambda \mathrm{V}^\top$ is an eigendecomposition of $\mathrm{G_U}$,

then for $\mathrm{w} = \frac{1}{\sqrt{r}} \sum_{i=1}^{r} \mathrm{v}_i$, it holds that $\mathrm{w}^\top \mathrm{G}_\mathrm{U} \mathrm{w} = \frac{\mathrm{Trace}\, \mathrm{G}_\mathrm{U}}{r}$; Proof of Theorem 1 uses this property to recursively equalize all diagonal elements of G_U.

Finally, we argue that the implicit bias induced by dropout is closely related to the notion of equalized network introduced above. In particular, our main result of the section states that the dropout enforces any globally optimal network to be equalized. Formally, we show the following.

Theorem 2. *If U is a global optimum of Problem 2.4, then U is equalized. Furthermore, it holds that*

$$R(U) = \frac{\lambda}{r} \| U \|_F^4.$$

Theorem 2 characterizes the effect of regularization induced by dropout in learning autoencoders with tied weights. It states that for any globally optimal network, the columns of the corresponding weight matrix have equal norms. In other words, dropout tends to give equal weights to all hidden nodes – it shows that dropout implicitly biases the optimal networks towards having hidden nodes with limited overall influence rather than a few important ones.

While Theorem 2 makes explicit the bias of dropout and gives a necessary condition for global optimality in terms of the weight matrix U_*, it does not characterize the bias induced in terms of the network (i.e. in terms of $\mathrm{U}_* \mathrm{U}_*^\top$). The following theorem completes the characterization by describing globally optimal autoencoder networks. Since the goal is to understand the implicit bias of dropout, we specify the global optimum in terms of the true concept, M.

Theorem 3. *For any $j \in [r]$, let $\kappa_j := \frac{1}{j} \sum_{i=1}^{j} \lambda_i(M)$. Furthermore, define $\rho := \max\{j \in [r] : \lambda_j(M) > \frac{\lambda j \kappa_j}{r + \lambda j}\}$. Then, if U_* is a global optimum of Problem 2.4, it satisfies that $U_* U_*^\top = \mathcal{S}_{\frac{\lambda \rho \kappa_\rho}{r + \lambda \rho}}(M)$.*

We note that the global optimality result presented in Theorem 3 extends the

results in [CHL+18], which holds only for sufficiently large factorization size r. Detailed comparisons are deferred to Section 2.4.

Remark 1. *In light of Theorem 2, the proof of Theorem 3 entails solving the following optimization problem*

$$\min_{U \in \mathbb{R}^{d \times r}} L(U, U) + \frac{\lambda}{r} \| U \|_F^4, \tag{2.5}$$

instead of Problem 2.4. This follows since the loss function $L(U, U)$ is invariant under rotations, hence a weight matrix U cannot be optimal if there exists a rotation matrix Q such that $R(UQ, UQ) < R(U, U)$. Now, while the objective in Problem 2.5 is a lower bound on the objective in Problem 2.4, by Theorem 1, we know that any weight matrix can be equalized. Thus, it follows that the minimum of the two problems coincide. Although Problem 2.5 is still non-convex, it is easier to study owing to a simpler form of the regularizer. Figure 2-1 shows how optimization landscape changes with different dropout rates for a single hidden layer linear autoencoder with one dimensional input and output and with a hidden layer of width two.

2.2 General Two-Layer Networks

Next, we consider the more general setting of a shallow linear network with a single hidden layer. Recall, that the goal is to find weight matrices U, V that solve

$$\min_{\mathrm{U} \in \mathbb{R}^{d_2 \times r}, \mathrm{V} \in \mathbb{R}^{d_0 \times r}} L(\mathrm{U}, \mathrm{V}) + \lambda \sum_{i=1}^{r} \|\mathrm{u}_i\|^2 \|\mathrm{v}_i\|^2. \tag{2.6}$$

As in the previous section, we note that the loss function is rotation invariant, i.e. $L(\mathrm{UQ}, \mathrm{VQ}) = L(\mathrm{U}, \mathrm{V})$ for any rotation matrix Q, however the regularizer is not invariant to rotations. Furthermore, it is easy to verify that both the loss function and the regularizer are invariant under rescaling of the incoming and outgoing weights to hidden neurons.

Remark 2 (Rescaling invariance)**.** *The objective function in Problem (2.2) is invariant under rescaling of weight matrices, i.e. invariant to transformations of the form*

$\bar{U} = UD$, $\bar{V} = VD^{-1}$, *where* D *is a diagonal matrix with positive entries. This follows since* $\bar{U}\bar{V}^\top = UDD^{-\top}V^\top = UV^\top$, *so that* $L(\bar{U}, \bar{V}) = L(U, V)$, *and also* $R(\bar{U}, \bar{V}) = R(U, V)$ *since*

$$\sum_{i=1}^{r} \|\bar{u}_i\|^2 \|\bar{v}_i\|^2 = \sum_{i=1}^{r} \|d_i u_i\|^2 \|\frac{1}{d_i} v_i\|^2 = \sum_{i=1}^{r} \|u_i\|^2 \|v_i\|^2.$$

As a result of rescaling invariance, $L_\theta(\bar{\mathrm{U}}, \bar{\mathrm{V}}) = L_\theta(\mathrm{U}, \mathrm{V})$. Now, following similar arguments as in the previous section, we define $\mathrm{n_{u,v}} = (\|\mathrm{u}_1\|\|\mathrm{v}_1\|, \ldots, \|\mathrm{u}_r\|\|\mathrm{v}_r\|)$, and note that

$$\begin{aligned} R(\mathrm{U}, \mathrm{V}) &= \lambda \sum_{i=1}^{r} \|\mathrm{u}_i\|^2 \|\mathrm{v}_i\|^2 = \frac{\lambda}{r} \|1_r\|^2 \|\mathrm{n_{u,v}}\|^2 \\ &\geq \frac{\lambda}{r} \langle 1_r, \mathrm{n_{u,v}} \rangle^2 = \frac{\lambda}{r} \left(\sum_{i=1}^{r} \|\mathrm{u}_i\| \|\mathrm{v}_i\| \right)^2 \\ &= \frac{\lambda}{r} \left(\sum_{i=1}^{r} \|\mathrm{u}_i \mathrm{v}_i^\top\|_* \right)^2 \\ &\geq \frac{\lambda}{r} \left(\|\sum_{i=1}^{r} \mathrm{u}_i \mathrm{v}_i^\top\|_* \right)^2 = \frac{\lambda}{r} \left(\|\mathrm{U}\mathrm{V}^\top\|_* \right)^2, \end{aligned}$$

where the first inequality is due to Cauchy-Schwartz, the second inequality follows from the triangle inequality, and the lower bound is achieved if and only if $\mathrm{n_{u,v}}$ is a scalar multiple of 1_r and $\|\mathrm{u}_i\|\|\mathrm{v}_i\| = \frac{1}{r}\|\mathrm{U}\mathrm{V}^\top\|_*$ for all $i \in [r]$. This observation motivates the following definition.

Definition 2 (Jointly equalized weight matrices, equalized linear networks)**.** *A pair of weight matrices* $(U, V) \in \mathbb{R}^{d_2 \times r} \times \mathbb{R}^{d_0 \times r}$ *is said to be jointly equalized if* $\|u_i\|\|v_i\| = \frac{1}{r}\|UV^\top\|_*$ *for all* $i \in [r]$. *A single hidden-layer linear network is said to be equalized if the product of the norms of the incoming and outgoing weights are equal for all hidden nodes. Equivalently, a single hidden-layer network parametrized by weight matrices* U, V, *is equalized if* U, V *are jointly equalized. An orthogonal transformation* $Q \in \mathbb{R}^{r \times r}$ *is an equalizer of a single hidden-layer network* $h_{U,V}$ *parametrized by weight matrices* U, V, *if* $h_{UQ,VQ}$ *is equalized. The network* $h_{U,V}$ *(the pair*(U, V)*) then are said to be jointly equalizable by* Q.

Note that Theorem 1 only guarantees the existence of an equalizer for an autoencoder with tied weights. It does not inform us regarding the existence of a rotation matrix that jointly equalizes a general network parameterized by a pair of weight matrices (U, V); in fact, it is not true in general that any pair (U, V) is jointly equalizable. Indeed, the general case requires a more careful treatment. It turns out that while a given pair of matrices (U, V) may not be jointly equalizable there exists a pair $(\tilde{\mathrm{U}}, \tilde{\mathrm{V}})$ that is jointly equalizable and implements the same network function, i.e. $h_{\tilde{\mathrm{U}},\tilde{\mathrm{V}}} = h_{\mathrm{U},\mathrm{V}}$. Formally, we state the following result.

Theorem 4. *For any given pair of weight matrices $(U, V) \in \mathbb{R}^{d_2 \times r} \times \mathbb{R}^{d_0 \times r}$, there exists another pair $(\tilde{U}, \tilde{V}) \in \mathbb{R}^{d_2 \times r} \times \mathbb{R}^{d_0 \times r}$ and a rotation matrix $Q \in \mathbb{R}^{r \times r}$ such that $h_{\tilde{U},\tilde{V}} = h_{U,V}$ and $h_{\tilde{U},\tilde{V}}$ is jointly equalizable by Q. Furthermore, for $\bar{U} := \tilde{U}Q$ and $\bar{V} := \tilde{V}Q$ it holds that $\|\bar{u}_i\|^2 = \|\bar{v}_i\|^2 = \frac{1}{r}\|UV^\top\|_*$ for $i = 1, \dots, r$.*

Theorem 4 implies that for any network $h_{\mathrm{U},\mathrm{V}}$ there exists an equalized network $h_{\bar{\mathrm{U}},\bar{\mathrm{V}}}$ such that $h_{\bar{\mathrm{U}},\bar{\mathrm{V}}} = h_{\mathrm{U},\mathrm{V}}$. Hence, it is always possible to reduce the objective by equalizing the network, and a network $h_{\mathrm{U},\mathrm{V}}$ is globally optimal only if it is equalized.

Theorem 5. *If (U, V) is a global optimum of Problem 2.6, then U, V are jointly equalized. Furthermore, it holds that*

$$R(U, V) = \frac{\lambda}{r}\left(\sum_{i=1}^{r} \|u_i\|\|v_i\|\right)^2 = \frac{\lambda}{r}\|UV^\top\|_*^2$$

Remark 3. *As in the case of autoencoders with tied weights in Section 2.1, a complete characterization of the implicit bias of dropout is given by considering the global optimality in terms of the network, i.e. in terms of the product of the weight matrices $UV^\top$. Not surprisingly, even in the case of single hidden-layer networks, dropout promotes sparsity, i.e. favors low-rank weight matrices.*

Theorem 6. *For any $j \in [r]$, let $\kappa_j := \frac{1}{j}\sum_{i=1}^{j} \lambda_i(M)$. Furthermore, define $\rho := \max\{j \in [r] : \lambda_j(M) > \frac{\lambda j \kappa_j}{r + \lambda j}\}$. Then, if (U_*, V_*) is a global optimum of Problem 2.6, it satisfies that $U_* V_*^\top = \mathcal{S}_{\frac{\lambda \rho \kappa_\rho}{r + \lambda \rho}}(M)$.*

2.3 The Optimization Landscape

While the focus in Section 2.1 and Section 2.2 was on understanding the implicit bias of dropout in terms of the global optima of the resulting regularized learning problem, here we focus on computational aspects of dropout as an optimization procedure. Since dropout is a first-order method (see Algorithm 1) and the landscape of Problem 2.4 is highly non-convex, we can perhaps only hope to find a *local* minimum, that too provided if the problem has no degenerate saddle points [LSJR16, GHJY15]. Therefore, in this section, we pose the following questions: *What is the implicit bias of dropout in terms of local minima? Do local minima share anything with global minima structurally or in terms of the objective? Can dropout find a local optimum?*

For the sake of simplicity of analysis, we focus on the case of autoencoders with tied weight as in Section 2.1. We show in Section 2.3.1 that (a) local minima of Problem 2.4 inherit the same implicit bias as the global optima, i.e. all local minima are equalized. Then, in Section 2.3.2, we show that for sufficiently small regularization parameter, (b) there are no spurious local minima, i.e. all local minima are global, and (c) all saddle points are non-degenerate (see Definition 3).

2.3.1 Implicit bias in local optima

We begin by recalling that the loss $L(\mathrm{U}, \mathrm{U})$ is rotation invariant, i.e. $L(\mathrm{UQ}, \mathrm{UQ}) = L(\mathrm{U}, \mathrm{U})$ for any rotation matrix Q. Now, if the weight matrix U were not equalized, then there exist indices $i, j \in [r]$ such that $\|\mathrm{u}_i\| > \|\mathrm{u}_j\|$. We show that it is easy to design a rotation matrix (equal to identity everywhere expect for columns i and j) that moves mass from u_i to u_j such that the difference in the norms of the corresponding columns of UQ decreases strictly while leaving the norms of other columns invariant. In other words, this rotation strictly reduces the regularizer and hence the objective. Formally, this implies the following result.

Lemma 1. *All local optima of Problem 2.4 are equalized, i.e. if U is a local optimum, then $\|u_i\| = \|u_j\|$ $\forall i, j \in [r]$.*

Lemma 1 unveils a fundamental property of dropout. As soon as we perform dropout in the hidden layer – *no matter how small the dropout rate* – all local minima become equalized.

2.3.2 Landscape properties

Next, we characterize the solutions to which dropout (i.e. Algorithm 1) converges. We do so by understanding the optimization landscape of Problem 2.4. Central to our analysis, is the following notion of *strict saddle property*.

Definition 3 (Strict saddle point/property)**.** *Let $f : \mathcal{U} \to \mathbb{R}$ be a twice differentiable function and let $U \in \mathcal{U}$ be a critical point of f. Then, U is a strict saddle point of f if the Hessian of f at U has at least one negative eigenvalue, i.e. $\lambda_{\min}(\nabla^2 f(U)) < 0$. Furthermore, f satisfies strict saddle property if all saddle points of f are strict saddle.*

Strict saddle property ensures that for any critical point U that is not a local optimum, the Hessian has a significant negative eigenvalue which allows first order methods such as gradient descent (GD) and stochastic gradient descent (SGD) to escape saddle points and converge to a local minimum [LSJR16, GHJY15]. Following this idea, there has been a flurry of works on studying the landscape of different machine learning problems, including low rank matrix recovery [BNS16], generalized phase retrieval problem [SQW16], matrix completion [GLM16], deep linear networks [Kaw16], matrix sensing and robust PCA [GJZ17] and tensor decomposition [GHJY15], making a case for global optimality of first order methods.

For the special case of no regularization (i.e. $\lambda = 0$; equivalently, no dropout), Problem 2.4 reduces to standard squared loss minimization which has been shown to have no spurious local minima and satisfy strict saddle property (see, e.g. [BH89,

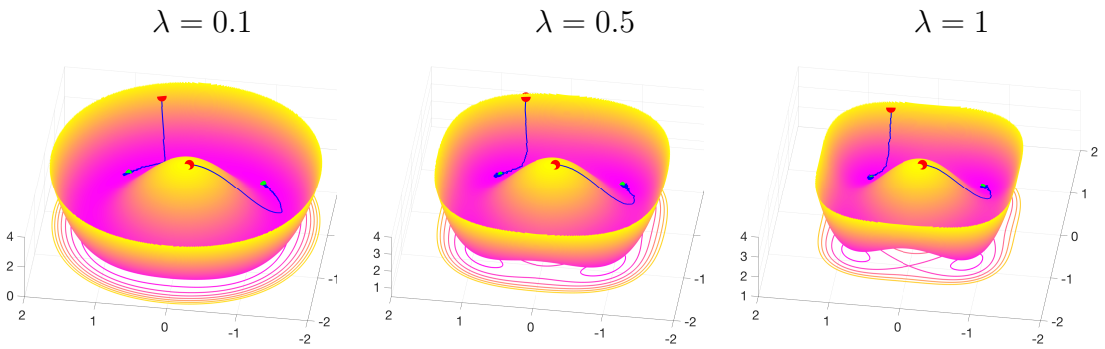

Figure 2-2. Convergence of dropout (Algorithm 1) from two different initialization (marked in red circles) to a global optimum of Problem 2.4 (marked in green circles), for the simple case of scalar M (one dimensional input and output) and $r = 2$. It can be seen that dropout quickly converges to a global optimum, which is equalized (i.e. weights are parallel to $(\pm 1, \pm 1)$) regardless of the value of the regularization parameter, $\lambda = 0.1$ (left), $\lambda = 0.5$ (middle) and $\lambda = 1.0$ (right).

JGN+17]). However, the regularizer induced by dropout can potentially introduce new spurious local minima as well as degenerate saddle points. Our next result establishes that that is not the case, at least when the dropout rate is sufficiently small.

Theorem 7. *For regularization parameter* $\lambda < \frac{r\lambda_r(M)}{\sum_{i=1}^{r} \lambda_i(M) - r\lambda_r(M)}$, *(a) all local minima of Problem 2.4 are global, and (b) all saddle points are strict saddle points.*

A couple of remarks are in order. First, Theorem 7 guarantees that any critical point U that is not a global optimum is a strict saddle point, i.e. $\nabla^2 f(\mathrm{U}, \mathrm{U})$ has a negative eigenvalue. This property allows first order methods, such as dropout given in Algorithm 1, to escape such saddle points. Second, note that the guarantees in Theorem 7 hold when the regularization parameter λ is sufficiently small. Assumptions of this kind are common in the literature (see, for example [GJZ17]). While this is a *sufficient* condition for the result in Theorem 7, it is not clear if it is *necessary*.

2.4 Matrix Factorization with Dropout

The optimization problem associated with learning a shallow network, i.e. Problem 2.6, is closely related to the optimization problem for matrix factorization. Recall

that in matrix factorization, given a matrix $\mathrm{M} \in \mathbb{R}^{d_2 \times d_0}$, one seeks to find factors U, V that minimize $L(\mathrm{U}, \mathrm{V}) = \|\mathrm{M} - \mathrm{U}\mathrm{V}^\top\|_F^2$. Matrix factorization has recently been studied with dropout by [ZZ15, HLL+16] and [CHL+18] where at each iteration of gradient descent on the loss function, the columns of factors U, V are dropped independently and with equal probability. Following [CHL+18], we can write the resulting problem as

$$\min_{\mathrm{U} \in \mathbb{R}^{d_2 \times r}, \mathrm{V} \in \mathbb{R}^{d_0 \times r}} \|\mathrm{M} - \mathrm{U}\mathrm{V}^\top\|_F^2 + \lambda \sum_{i=1}^{r} \|\mathrm{u}_i\|^2 \|\mathrm{v}_i\|^2, \tag{2.7}$$

which is identical to Problem 2.6. However, there are two key distinctions. First, we are interested in stochastic optimization problem whereas the matrix factorization problem is typically posed for a given matrix. Second, for the learning problem that we consider here, it is unreasonable to assume access to the true model (i.e. matrix M). Nonetheless, many of the insights we develop here as well as the technical results and algorithmic contributions apply to matrix factorization. Therefore, the goal in this section is to bring to bear the results in Sections 2.1, 2.2 and 2.3 to matrix factorization.

We note that Theorem 6 and Theorem 4, both of which hold for matrix factorization, imply that there is a polynomial time algorithm to solve the matrix factorization problem. In order to find a global optimum of Problem 2.7, we first compute the

Algorithm 3: Polynomial time solver for Problem 2.7

Input: Matrix $\mathrm{M} \in \mathbb{R}^{d_0 \times d_0}$ to be factorized, regularization parameter λ

1: $r \leftarrow \mathrm{Rank}(\mathrm{M})$
2: $\rho \leftarrow \max\{j \in [r] : \ \lambda_j(\mathrm{M}) > \frac{\lambda j \kappa_j}{1+\lambda j}\}$,
where $\kappa_j = \frac{1}{j}\sum_{i=1}^{j} \lambda_i(\mathrm{M})$ for $j \in [r]$.
3: $\bar{\mathrm{M}} \leftarrow \mathcal{S}_{\frac{\lambda \rho \kappa_\rho}{1+\lambda\rho}}(\mathrm{M})$
4: $(\mathrm{U}, \Sigma, \mathrm{V}) \leftarrow \mathtt{svd}(\bar{\mathrm{M}})$
5: $\tilde{\mathrm{U}} \leftarrow \mathrm{U}\Sigma^{\frac{1}{2}}$, $\tilde{\mathrm{V}} \leftarrow \mathrm{V}\Sigma^{\frac{1}{2}}$
6: $\mathrm{Q} \leftarrow \mathtt{EQZ}(\tilde{\mathrm{U}})$ {Algorithm 2}
7: $\bar{\mathrm{U}} \leftarrow \tilde{\mathrm{U}}\mathrm{Q}$, $\bar{\mathrm{V}} \leftarrow \tilde{\mathrm{V}}\mathrm{Q}$

Output: $\bar{\mathrm{U}}, \bar{\mathrm{V}}$ {global optimum of Problem 2.7}

optimal $\bar{\mathrm{M}} = \tilde{\mathrm{U}}\tilde{\mathrm{V}}^\top$ using shrinkage-thresholding operation (see Theorem 6). A global optimum $(\bar{\mathrm{U}}, \bar{\mathrm{V}})$ is then obtained by joint equalization of $(\tilde{\mathrm{U}}, \tilde{\mathrm{V}})$ (see Theorem 4) using Algorithm 2. The whole procedure is described in Algorithm 3. Few remarks are in order.

Remark 4 (Computational cost of Algorithm 3)**.** *It is easy to check that computing $\rho, \bar{M}, \tilde{U}$ and $\tilde{V}$ requires computing a rank-r SVD of M, which costs $O(d^2 r)$, where $d = \max\{d_2, d_0\}$. Algorithm 2 entails computing $G_U = U^\top U$, which costs $O(r^2 d)$ and the cost of each iterate of Algorithm 2 is dominated by computing the eigendecomposition which is $O(r^3)$. Overall, the computational cost of Algorithm 3 is $O(d^2 r + d r^2 + r^4)$.*

Remark 5 (Universal Equalizer)**.** *While Algorithm 2 is efficient (only linear in the dimension) for any rank r, there is a more effective equalization procedure when r is a power of 2. In this case, we can give a universal equalizer which works simultaneously for all matrices in $\mathbb{R}^{d\times r}$. Let $U \in \mathbb{R}^{d\times r}$, $r = 2^k$, $k \in \mathbb{N}$ and let $U = W\Sigma V^\top$ be its full SVD. The matrix $\tilde{U} = UQ$ is equalized, where $Q = VZ_k$ and*

$$Z_k := \begin{cases} 1 & k = 1 \\ 2^{\frac{-k+1}{2}} \begin{bmatrix} Z_{k-1} & Z_{k-1} \\ -Z_{k-1} & Z_{k-1} \end{bmatrix} & k > 1 \end{cases}.$$

Finally, we note that Problem 2.7 is an instance of regularized matrix factorization which has recently received considerable attention in the machine learning literature [GLM16, GJZ17, HV17]. These works show that the saddle points of a class of regularized matrix factorization problems have certain "nice" properties (i.e. escape directions characterized by negative curvature around saddle points) which allow variants of first-order methods such as perturbed gradient descent [GHJY15, JGN+17] to converge to a local optimum. Distinct from that line of research, we completely characterize the set of global optima of Problem 2.7, and provide a polynomial time algorithm to find a global optimum.

2.4.1 Comparison with Previous Work

The work most similar to the matrix factorization problem we consider in this section is that of [CHL+18], with respect to which we make several important contributions:

1. [CHL+18] characterize optimal solutions only in terms of the product of the factors, and not in terms of the factors themselves, whereas we provide globally optimal solutions in terms of the factors;

2. [CHL+18] require the rank r of the desired factorization to be variable and above some threshold, whereas we consider fixed rank-r factorization for any r;

3. [CHL+18] can only find low rank solutions using an adaptive dropout rate, which is not how dropout is used in practice, whereas we consider any fixed dropout rate;

4. We give an efficient polynomial time algorithm to find optimal factors.

2.5 Proofs

We first provide a proof for Lemma 2, which extracts the explicit regularizer.

Lemma 2. *Let* $\mathrm{x} \in \mathbb{R}^{d_0}$ *be distributed according to distribution* $\mathcal{D}$ *with* $\mathbb{E}_{\mathrm{x}}[\mathrm{x}\mathrm{x}^\top] = I$. *Then, for* $L(U, V) := \mathbb{E}_{\mathrm{x}}[\|\mathrm{y} - UV^\top \mathrm{x}\|^2]$ *and* $L_\theta(U, V) := \mathbb{E}_{b,\mathrm{x}}[\|\mathrm{y} - \frac{1}{\theta} U \operatorname{diag}(b) V^\top \mathrm{x}\|^2]$, *it holds that*

$$L_\theta(U, V) = L(U, V) + \lambda \sum_{i=1}^{r} \|u_i\|^2 \|v_i\|^2. \tag{2.8}$$

Furthermore, $L(U, V) = \|M - UV^\top\|_F^2$.

Proof of Lemma 2. The proof closely follows [CHL+18]. Recall that $\mathrm{y} = \mathrm{M}\mathrm{x}$, for some unknown $\mathrm{M} \in \mathbb{R}^{d_0 \times d_2}$. Observe that

$$\begin{aligned} L_\theta(\mathrm{U}, \mathrm{V}) = \mathbb{E}_{\mathrm{x}}[\|\mathrm{y}\|^2] + \frac{1}{\theta^2}\mathbb{E}_{\mathrm{b},\mathrm{x}}[\|\mathrm{U}\operatorname{diag}(\mathrm{b})\mathrm{V}^\top \mathrm{x}\|^2] \\ - \frac{2}{\theta}\mathbb{E}_{\mathrm{x}}[\langle \mathrm{M}\mathrm{x}, \mathbb{E}_{\mathrm{b}}[\mathrm{U}\operatorname{diag}(\mathrm{b})\mathrm{V}^\top]\mathrm{x}\rangle] \end{aligned} \tag{2.9}$$

where we used the fact that $\mathrm{y} = \mathrm{Mx}$. We have the following set of equalities for the second term on the right hand side of Equation (2.9):

$$\begin{aligned}
\mathbb{E}_{\mathrm{b,x}}[\|\mathrm{U}\,\mathrm{diag}(\mathrm{b})\mathrm{V}^\top\mathrm{x}\|^2] &= \mathbb{E}_{\mathrm{x}}\sum_{i=1}^{d_0}\mathbb{E}_{\mathrm{b}}\left(\sum_{j=1}^{r}u_{ij}b_j\mathrm{v}_j^\top\mathrm{x}\right)^2 \\
&= \mathbb{E}_{\mathrm{x}}\sum_{i=1}^{d_0}\mathbb{E}_{\mathrm{b}}[\sum_{j,k=1}^{r}u_{ij}u_{ik}b_jb_k(\mathrm{v}_j^\top\mathrm{x})(\mathrm{v}_k^\top\mathrm{x})] \\
&= \mathbb{E}_{\mathrm{x}}\sum_{i=1}^{d_0}\sum_{j,k=1}^{r}u_{ij}u_{ik}(\theta^2 1_{j\neq k}+\theta 1_{j=k})(\mathrm{v}_j^\top\mathrm{x})(\mathrm{v}_k^\top\mathrm{x}) \\
&= \theta^2\mathbb{E}_{\mathrm{x}}[\|\mathrm{UV}^\top\mathrm{x}\|^2]+(\theta-\theta^2)\mathbb{E}_{\mathrm{x}}\sum_{i=1}^{d_0}\sum_{j=1}^{r}u_{ij}^2(\mathrm{v}_j^\top\mathrm{x})^2 \\
&= \theta^2\mathbb{E}_{\mathrm{x}}[\|\mathrm{UV}^\top\mathrm{x}\|^2]+(\theta-\theta^2)\sum_{j=1}^{r}\|\mathrm{v}_j\|^2\sum_{i=1}^{d_0}u_{ij}^2 \\
&= \theta^2\mathbb{E}_{\mathrm{x}}[\|\mathrm{UV}^\top\mathrm{x}\|^2]+(\theta-\theta^2)\sum_{j=1}^{r}\|\mathrm{v}_j\|^2\|\mathrm{u}_j\|^2, \qquad (2.10)
\end{aligned}$$

where the second to last equality follows because $\mathbb{E}_{\mathrm{x}}[(\mathrm{v}_j^\top\mathrm{x})^2] = \mathrm{v}_j^\top\mathbb{E}_{\mathrm{x}}[\mathrm{xx}^\top]\mathrm{v}_j = \|\mathrm{v}_j\|^2$. For the third term in Equation (2.9) we have:

$$\langle \mathrm{Mx}, \mathbb{E}_{\mathrm{b}}[\mathrm{U}\,\mathrm{diag}(\mathrm{b})\mathrm{V}^\top]\mathrm{x}\rangle = \theta\langle \mathrm{Mx}, \mathrm{UV}^\top\mathrm{x}\rangle \qquad (2.11)$$

Plugging Equations (2.10) and (2.11) into (2.9), we get

$$\begin{aligned}
L_\theta(\mathrm{U},\mathrm{V}) = \mathbb{E}_{\mathrm{x}}[\|\mathrm{y}\|^2] + \mathbb{E}_{\mathrm{x}}[\|\mathrm{UV}^\top\mathrm{x}\|^2] - 2\mathbb{E}_{\mathrm{x}}\langle \mathrm{Mx}, \mathrm{UV}^\top\mathrm{x}\rangle \\
+ \frac{1-\theta}{\theta}\sum_{i=1}^{r}\|\mathrm{u}_i\|^2\|\mathrm{v}_i\|^2 \qquad (2.12)
\end{aligned}$$

It is easy to check that the first three terms in Equation (2.12) sum to $L(\mathrm{U},\mathrm{V})$. Furthermore, since for any $\mathrm{A}\in\mathbb{R}^{d_0\times d_2}$ it holds that $\|\mathrm{Ax}\|^2 = \|\mathrm{A}\|_F^2$, we should have $L(\mathrm{U},\mathrm{V}) = \|\mathrm{M}-\mathrm{UV}^\top\|_F^2$. □

2.5.1 Proofs of Theorems in Section 2.1

Theorem 1 states that any weight matrix can be equalized. In the following, we provide a constructive proof for this theorem, which is the basis of our analysis for the main results of this section.

Proof of Theorem 1. Consider the matrix $\mathrm{G}_1 := \mathrm{G}_\mathrm{U} - \frac{\mathrm{Trace}\,\mathrm{G}_\mathrm{U}}{r}\mathrm{I}_r$. We exhibit an orthogonal transformation Q, such that $\mathrm{Q}^\top \mathrm{G}_1 \mathrm{Q}$ is zero on its diagonal. Observe that

$$\mathrm{Q}^\top \mathrm{G}_\mathrm{U} \mathrm{Q} = \mathrm{Q}^\top \mathrm{G}_1 \mathrm{Q} + \frac{\mathrm{Trace}\,\mathrm{G}_\mathrm{U}}{r}\mathrm{I}_r,$$

so that all diagonal elements of G_U are equal to $\frac{\mathrm{Trace}\,\mathrm{G}_\mathrm{U}}{r}$, i.e. G_U is equalized.

Our construction closely follows the proof of a classical theorem in matrix analysis, which states that any trace zero matrix is a commutator [AM57, Kah99]. For the zero trace matrix G_1, we first show that there exists a unit vector w_{11} such that $\mathrm{w}_{11}^\top \mathrm{G}_1 \mathrm{w}_{11} = 0$.

Claim 1. *Assume G is a zero trace matrix and let $G = \sum_{i=1}^r \lambda_i u_i u_i^\top$ be an eigendecomposition of G. Then $\mathrm{w} = \frac{1}{\sqrt{r}} \sum_{i=1}^r u_i$ has a vanishing Rayleigh quotient, that is, $\mathrm{w}^\top G \mathrm{w} = 0$, and $\|\mathrm{w}\| = 1$.*

Proof of Claim 1. First, we notice that w has unit norm

$$\|\mathrm{w}\|^2 = \|\frac{1}{\sqrt{r}} \sum_{i=1}^r \mathrm{u}_i\|^2 = \frac{1}{r}\|\sum_{i=1}^r \mathrm{u}_i\|^2 = \frac{1}{r}\sum_{i=1}^r \|\mathrm{u}_i\|^2 = 1.$$

It is easy to see that w has a zero Rayleigh quotient

$$\begin{aligned}\mathrm{w}^\top \mathrm{G} \mathrm{w} &= (\frac{1}{\sqrt{r}} \sum_{i=1}^r \mathrm{u}_i)^\top \mathrm{G} (\frac{1}{\sqrt{r}} \sum_{i=1}^r \mathrm{u}_i) \\ &= \frac{1}{r} \sum_{i,j=1}^r \mathrm{u}_i \mathrm{G} \mathrm{u}_j = \frac{1}{r} \sum_{i=1}^r \lambda_j \mathrm{u}_i^\top \mathrm{u}_j = \frac{1}{r} \sum_{i=1}^r \lambda_i = 0.\end{aligned}$$

□

Let $\mathrm{W}_1 := [\mathrm{w}_{11}, \mathrm{w}_{12}, \cdots, \mathrm{w}_{1d}]$ be such that $\mathrm{W}_1^\top \mathrm{W}_1 = \mathrm{W}_1 \mathrm{W}_1^\top = \mathrm{I}_d$. Observe that $\mathrm{W}_1^\top \mathrm{G}_1 \mathrm{W}_1$ has zero on its first diagonal elements

$$\mathrm{W}_1^\top \mathrm{G}_1 \mathrm{W}_1 = \begin{bmatrix} 0 & \mathrm{b}_1^\top \\ \mathrm{b}_1 & \mathrm{G}_2 \end{bmatrix}$$

The principal submatrix G_2 also has a zero trace. With a similar argument, let $\mathrm{w}_{22} \in \mathbb{R}^{d-1}$ be such that $\|\mathrm{w}_{22}\| = 1$ and $\mathrm{w}_{22}^\top \mathrm{G}_2 \mathrm{w}_{22} = 0$ and define $\mathrm{W}_2 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \mathrm{w}_{22} & \mathrm{w}_{23} & \cdots & \mathrm{w}_{2d} \end{bmatrix} \in$

$\mathbb{R}^{d\times d}$ such that $\mathrm{W}_2^\top \mathrm{W}_2 = \mathrm{W}_2 \mathrm{W}_2^\top = \mathrm{I}_d$, and observe that

$$(\mathrm{W}_1\mathrm{W}_2)^\top \mathrm{G}_1 (\mathrm{W}_1\mathrm{W}_2) = \begin{bmatrix} 0 & \cdot & \cdots \\ \cdot & 0 & \cdots \\ \vdots & \vdots & \mathrm{G}_2 \end{bmatrix}.$$

This procedure can be applied recursively so that for the *equalizer* $\mathrm{Q} = \mathrm{W}_1\mathrm{W}_2\cdots\mathrm{W}_d$ we have

$$\mathrm{Q}^\top \mathrm{G}_1 \mathrm{Q} = \begin{bmatrix} 0 & \cdot & \cdots & \cdot \\ \cdot & 0 & \cdots & \cdot \\ \vdots & \vdots & \ddots & \vdots \\ \cdot & \cdot & \cdot & 0 \end{bmatrix}.$$

□

Proof of Theorem 2. Let us denote the squared column norms of U by $\mathrm{n_u} = (\|\mathrm{u}_1\|^2, \ldots, \|\mathrm{u}_r\|^2)$. Observe that for any weight matrix U:

$$\begin{aligned} R(\mathrm{U},\mathrm{U}) &= \lambda \sum_{i=1}^r \|\mathrm{u}_i\|^4 = \frac{\lambda}{r}\|1_r\|^2 \|\mathrm{n_u}\|^2 \\ &\geq \frac{\lambda}{r}\langle 1_r, \mathrm{n_u}\rangle^2 = \frac{\lambda}{r}\left(\sum_{i=1}^r \|\mathrm{u}_i\|^2\right)^2 = \frac{\lambda}{r}\|\mathrm{U}\|_F^4, \end{aligned}$$

where $1_r \in \mathbb{R}^r$ is the vector of all ones and the inequality is due to Cauchy-Schwartz. Hence, the regularizer is lower bounded by $\frac{\lambda}{r}\|\mathrm{U}\|_F^4$, with equality if and only if $\mathrm{n_u}$ is parallel to 1_r, i.e. when U is equalized. Now, if U is not equalized, by Theorem 1 there exist a rotation matrix Q such that UQ is equalized, which implies $R(\mathrm{UQ},\mathrm{UQ}) < R(\mathrm{U},\mathrm{U})$. Together with rotational invariance of the loss function, this gives a contradiction with global optimality U. Hence, if U is a global optimum then it is equalized and we have $R(\mathrm{U},\mathrm{U}) = \lambda \sum_{i=1}^r \|\mathrm{u}_i\|^4 = \frac{\lambda}{r}\|\mathrm{U}\|_F^4$. □

If a weight matrix W minimizes the dropout objective, then it should also minimize the explicit regularizer over all weight matrices U such that $\mathrm{WW}^\top = \mathrm{UU}^\top$. In light of Theorem 1, we can characterize such weight matrices, which is the key idea in the proof of Theorem 3.

Proof of Theorem 3. By Theorem 2, if W is an optimum of Problem 2.4, then it holds that $\lambda \sum_{i=1}^{r} \|\mathrm{w}_i\|^4 = \frac{\lambda}{r}\|\mathrm{W}\|_F^4$. Also, by Theorem 1, it is always possible to equalize any given weight matrix. Hence, Problem 2.4 reduces to the following problem:

$$\min_{\mathrm{W}\in\mathbb{R}^{d\times r}} \|\mathrm{M} - \mathrm{W}\mathrm{W}^\top\|_F^2 + \frac{\lambda}{r}\|\mathrm{W}\|_F^4 \tag{2.13}$$

Let $\mathrm{M} = \mathrm{U}_\mathrm{M}\Lambda_\mathrm{M}\mathrm{U}_\mathrm{M}^\top$ and $\mathrm{W} = \mathrm{U}_\mathrm{W}\Sigma_\mathrm{W}\mathrm{V}_\mathrm{W}^\top$ be an eigendecomposition of M and a full SVD of W respectively, such that $\lambda_i(\mathrm{M}) \geq \lambda_{i+1}(\mathrm{M})$ and $\sigma_i(\mathrm{W}) \geq \sigma_{i+1}(\mathrm{W})$ for all $i \in [d-1]$. Rewriting objective of Problem 2.13 in terms of these decompositions gives:

$$\begin{aligned}
&\|\mathrm{M} - \mathrm{W}\mathrm{W}^\top\|_F^2 + \frac{\lambda}{r}\|\mathrm{W}\|_F^4 \\
&= \|\mathrm{U}_\mathrm{M}\Lambda_\mathrm{M}\mathrm{U}_\mathrm{M}^\top - \mathrm{U}_\mathrm{W}\Sigma_\mathrm{W}\Sigma_\mathrm{W}^\top\mathrm{U}_\mathrm{W}^\top\|_F^2 + \frac{\lambda}{r}\|\mathrm{U}_\mathrm{W}\Sigma_\mathrm{W}\mathrm{V}_\mathrm{W}^\top\|_F^4 \\
&= \|\Lambda_\mathrm{M} - \mathrm{U}'\Sigma_\mathrm{W}\Sigma_\mathrm{W}^\top\mathrm{U}'^\top\|_F^2 + \frac{\lambda}{r}\|\Sigma_\mathrm{W}\|_F^4 \\
&= \|\Lambda_\mathrm{M}\|_F^2 + \|\Lambda_\mathrm{W}\|_F^2 - 2\langle\Lambda_\mathrm{M}, \mathrm{U}'\Lambda_\mathrm{W}\mathrm{U}'^\top\rangle + \frac{\lambda}{r}\left(\mathrm{Trace}(\Lambda_\mathrm{W})\right)^2
\end{aligned}$$

where $\Lambda_\mathrm{W} := \Sigma_\mathrm{W}\Sigma_\mathrm{W}^\top$ and $\mathrm{U}' = \mathrm{U}_\mathrm{M}^\top\mathrm{U}_\mathrm{W}$. By Von Neumann's trace inequality, for a fixed Σ_W we have that

$$\langle\Lambda_\mathrm{M}, \mathrm{U}'\Lambda_W\mathrm{U}'^\top\rangle \leq \sum_{i=1}^{d}\lambda_i(\mathrm{M})\lambda_i(\mathrm{W}),$$

where the equality is achieved when $\Lambda_i(\mathrm{W})$ have the same ordering as $\Lambda_i(\mathrm{M})$ and $\mathrm{U}' = \mathrm{I}$, i.e. $\mathrm{U}_\mathrm{M} = \mathrm{U}_\mathrm{W}$. Now, Problem 2.13 is reduced to

$$\begin{aligned}
&\min_{\substack{\|\Lambda_\mathrm{W}\|_0\leq r,\\ \Lambda_\mathrm{W}\geq 0}} \|\Lambda_\mathrm{M} - \Lambda_\mathrm{W}\|_F^2 + \frac{\lambda}{r}\left(\mathrm{Trace}(\Lambda_\mathrm{W})\right)^2 \\
&= \min_{\bar{\lambda}\in\mathbb{R}_+^r} \sum_{i=1}^{r}\left(\lambda_i(\mathrm{M}) - \bar{\lambda}_i\right)^2 + \sum_{i=r+1}^{d}\lambda_i^2(\mathrm{M}) + \frac{\lambda}{r}\left(\sum_{i=1}^{r}\bar{\lambda}_i\right)^2
\end{aligned}$$

The Lagrangian is given by

$$\begin{aligned}
L(\bar{\lambda}, \alpha) = &\sum_{i=1}^{r}\left(\lambda_i(\mathrm{M}) - \bar{\lambda}_i\right)^2 + \sum_{i=r+1}^{d}\lambda_i^2(\mathrm{M}) \\
&+ \frac{\lambda}{r}\left(\sum_{i=1}^{r}\bar{\lambda}_i\right)^2 - \sum_{i=1}^{r}\alpha_i\bar{\lambda}_i
\end{aligned}$$

The KKT conditions ensures that at the optima it holds for all $i \in [r]$ that

$$\begin{aligned}
&\bar{\lambda}_i \geq 0,\ \alpha_i \geq 0,\ \bar{\lambda}_i \alpha_i = 0 \\
&2(\bar{\lambda}_i - \lambda_i(\mathrm{M})) + \frac{2\lambda}{r}\left(\sum_{i=1}^{r} \bar{\lambda}_i\right) - \alpha_i = 0
\end{aligned}$$

Let $\rho = |i : \bar{\lambda}_i > 0| \leq r$ be the number of nonzero $\bar{\lambda}_i$. For $i = 1, \ldots, \rho$ we have $\alpha_i = 0$, hence

$$\begin{aligned}
&\bar{\lambda}_i + \frac{\lambda}{r}\left(\sum_{i=1}^{\rho} \bar{\lambda}_i\right) = \lambda_i(\mathrm{M}) \\
&\implies (\mathrm{I}_\rho + \frac{\lambda}{r} 1 1^\top)\bar{\lambda}_{1:\rho} = \lambda_{1:\rho}(\mathrm{M}) \\
&\implies \bar{\lambda}_{1:\rho} = (\mathrm{I}_\rho - \frac{\lambda}{r + \lambda\rho} 1 1^\top)\lambda_{1:\rho}(\mathrm{M}) \\
&\implies \bar{\lambda}_{1:\rho} = \lambda_{1:\rho}(\mathrm{M}) - \frac{\lambda\rho\kappa_\rho}{r + \lambda\rho} 1_\rho \\
&\implies \Lambda_{\mathrm{W}} = (\Lambda_{\mathrm{M}} - \frac{\lambda\rho\kappa_\rho}{r + \lambda\rho}\mathrm{I}_d)_+
\end{aligned}$$

where $\kappa_\rho := \frac{1}{\rho}\sum_{i=1}^{\rho} \lambda_i(\mathrm{M})$ and the second implication is due to Lemma 20. It only remains to find the optimal ρ. Let's define the function

$$\begin{aligned}
g(\rho) &:= \sum_{i=1}^{\rho}\left(\lambda_i(\mathrm{M}) - \bar{\lambda}_i\right)^2 + \sum_{i=\rho+1}^{d} \lambda_i^2(\mathrm{M}) + \frac{\lambda}{r}\left(\sum_{i=1}^{\rho} \bar{\lambda}_i\right)^2 \\
&= \sum_{i=1}^{\rho}\left(\frac{\lambda \sum_{k=1}^{\rho} \lambda_k(\mathrm{M})}{r + \lambda\rho}\right)^2 + \sum_{i=\rho+1}^{d} \lambda_i(\mathrm{M})^2 \\
&+ \frac{\lambda}{r}\left(\sum_{i=1}^{\rho}\left(\lambda_i(\mathrm{M}) - \frac{\lambda \sum_{k=1}^{\rho} \lambda_k(\mathrm{M})}{r + \lambda\rho}\right)\right)^2.
\end{aligned}$$

By Lemma 21, $g(\rho)$ is monotonically non-increasing in ρ, hence ρ should be the largest *feasible* integer, i.e.

$$\rho = \max\{j :\ \lambda_j > \frac{\lambda j \kappa}{r + \lambda j}\}.$$

□

We conclude this section by a proof of Remark 5, which gives a universal equalizer when r is a power of 2.

Proof of Remark 5. For $\tilde{\mathrm{U}}$ to have equal column norms, it suffices to show that $\tilde{\mathrm{U}}^\top\tilde{\mathrm{U}}$ is constant on its diagonal. Next, we note that

$$\begin{aligned}\tilde{\mathrm{U}}^\top\tilde{\mathrm{U}} &= \mathrm{Q}^\top\mathrm{U}^\top\mathrm{U}\mathrm{Q} \\ &= (\mathrm{V}\mathrm{Z}_k)^\top(\mathrm{W}\Sigma\mathrm{V}^\top)^\top(\mathrm{W}\Sigma\mathrm{V}^\top)(\mathrm{V}\mathrm{Z}_k) \\ &= \mathrm{Z}_k^\top\mathrm{V}^\top\mathrm{V}\Sigma\mathrm{W}^\top\mathrm{W}\Sigma\mathrm{V}^\top\mathrm{V}\mathrm{Z}_k \\ &= \mathrm{Z}_k^\top\Sigma^2\mathrm{Z}_k\end{aligned}$$

It remains to show that for any diagonal matrix D, $\mathrm{Z}_k^\top\mathrm{D}\mathrm{Z}_k$ is diagonalized. First note that

$$\mathrm{Z}_2\mathrm{Z}_2^\top = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = \mathrm{I}_2$$

so that Z_2 is indeed a rotation. By induction, it is easy to see that Z_k is a rotation for all k. Now, we show that Z_k equalizes any diagonal matrix D. Observe that

$$[\mathrm{Z}_k^\top\mathrm{D}\mathrm{Z}_k]_{ii} = \sum_{i=1}^{2^{k-1}} D_{ii}z_{ji}^2 = \sum_{i=1}^{2^{k-1}} D_{ii}2^{-k+1} = 2^{1-k}\mathrm{Trace}\,\mathrm{D}$$

so that all the diagonal elements are identically equal to the average of the diagonal elements of D. □

2.5.2 Proofs of Theorems in Section 2.2

In this section we prove the main optimality results in the general case of two-layer linear networks.

Proof of Theorem 4. Let $\mathrm{U}\mathrm{V}^\top = \mathrm{W}\Sigma\mathrm{Y}^\top$ be a compact SVD of $\mathrm{U}\mathrm{V}^\top$. Define $\tilde{\mathrm{U}} := \mathrm{W}\Sigma^{1/2}$ and $\tilde{\mathrm{V}} := \mathrm{Y}\Sigma^{1/2}$ and observe that $\tilde{\mathrm{U}}\tilde{\mathrm{V}}^\top = \mathrm{U}\mathrm{V}^\top$. Furthermore, let $\mathrm{G}_{\tilde{\mathrm{U}}} = \tilde{\mathrm{U}}^\top\tilde{\mathrm{U}}$ and $\mathrm{G}_{\tilde{\mathrm{V}}} = \tilde{\mathrm{V}}^\top\tilde{\mathrm{V}}$ be their Gram matrices. Observe that $\mathrm{G}_{\tilde{\mathrm{U}}} = \mathrm{G}_{\tilde{\mathrm{V}}} = \Sigma$. Hence, by Theorem 1, there exists a rotation Q such that $\bar{\mathrm{V}} := \tilde{\mathrm{V}}\mathrm{Q}$ and $\bar{\mathrm{U}} := \tilde{\mathrm{U}}\mathrm{Q}$ are equalized, with $\|\bar{\mathrm{u}}_i\|^2 = \|\bar{\mathrm{v}}_i\|^2 = \frac{1}{r}\,\mathrm{Trace}\,\Sigma$. □

Proof of Theorem 5. Define

$$\mathrm{n}_{\mathrm{u,v}} = (\|\mathrm{u}_1\|\|\mathrm{v}_1\|, \ldots, \|\mathrm{u}_r\|\|\mathrm{v}_r\|)$$

and observe that

$$\begin{aligned} R(\mathrm{U},\mathrm{V}) &= \lambda \sum_{i=1}^{r} \|\mathrm{u}_i\|^2 \|\mathrm{v}_i\|^2 \\ &= \frac{\lambda}{r} \|\mathrm{n}_{\mathrm{u,v}}\|^2 \|1_r\|^2 \geq \frac{\lambda}{r} \langle \mathrm{n}_{\mathrm{u,v}}, 1_r \rangle^2 \\ &= \frac{\lambda}{r} \left(\sum_{i=1}^{r} \|\mathrm{u}_i\| \|\mathrm{v}_i\| \right)^2 \qquad (2.14) \\ &= \frac{\lambda}{r} \left(\sum_{i=1}^{r} \|\mathrm{u}_i \mathrm{v}_i^\top\|_* \right)^2 \\ &\geq \frac{\lambda}{r} \left(\| \sum_{i=1}^{r} \mathrm{u}_i \mathrm{v}_i^\top \|_* \right)^2 \\ &= \frac{\lambda}{r} \|\mathrm{U}\mathrm{V}^\top\|_*^2 \qquad (2.15) \end{aligned}$$

where the first inequality is due to Cauchy-Schwartz, and it holds with equality if and only if $\mathrm{n}_{\mathrm{u,v}}$ is parallel to 1_r. The second inequality is a simple application of the triangle inequality. Let $(\bar{\mathrm{U}}, \bar{\mathrm{V}})$ be a global optima of Problem 2.6. The inequality above together with Theorem 4 imply that $\bar{\mathrm{U}}$ and $\bar{\mathrm{V}}$ should be jointly equalized up to dilation transformations, hence the first equality claimed by the theorem.

To see the second equality, note that if U and V are jointly equalized, then

$$\|\mathrm{u}_i\|^2 = \|\mathrm{v}_i\|^2 = \frac{1}{r} \operatorname{Trace} \Sigma,$$

where Σ is the matrix of singular values of $\mathrm{U}\mathrm{V}^\top$. Hence,

$$\begin{aligned} R(\mathrm{U},\mathrm{V}) = \frac{\lambda}{r} \left(\sum_{i=1}^{r} \|\mathrm{u}_i\| \|\mathrm{v}_i\| \right)^2 &= \frac{\lambda}{r} \left(\frac{1}{r} \sum_{i=1}^{r} \operatorname{Trace} \Sigma \right)^2 \\ &= \frac{\lambda}{r} \left(\operatorname{Trace} \Sigma \right)^2 \end{aligned}$$

which is equal to $\frac{\lambda}{r} \|\bar{\mathrm{U}}\bar{\mathrm{V}}^\top\|_*^2$ as claimed. □

Proof of Theorem 6. By Theorem 5, if (X, Y) is an optimum of Problem 2.6, then it holds that

$$\lambda \sum_{i=1}^{r} \|\mathrm{x}_i\|^2 \|\mathrm{y}_i\|^2 = \frac{\lambda}{r} \|\mathrm{X}\mathrm{Y}^\top\|_*^2.$$

Hence, Problem 2.6 reduces to the following problem:

$$\min_{\mathrm{X} \in \mathbb{R}^{d_2 \times r}, \mathrm{Y} \in \mathbb{R}^{d_0 \times r}} \|\mathrm{M} - \mathrm{X}\mathrm{Y}^\top\|_F^2 + \frac{\lambda}{r} \|\mathrm{X}\mathrm{Y}^\top\|_*^2 \tag{2.16}$$

Let $\mathrm{M} = \mathrm{U}_\mathrm{M} \Sigma_\mathrm{M} \mathrm{V}_\mathrm{M}^\top$ and $\mathrm{W} := \mathrm{X}\mathrm{Y}^\top = \mathrm{U}_\mathrm{W} \Sigma_\mathrm{W} \mathrm{V}_\mathrm{W}^\top$ be full SVDs of M and W respectively, such that $\sigma_i(\mathrm{M}) \geq \sigma_{i+1}(\mathrm{M})$ and $\sigma_i(\mathrm{W}) \geq \sigma_{i+1}(\mathrm{W})$ for all $i \in [d-1]$ where $d = \min\{d_2, d_0\}$. Rewriting objective of Problem 2.16 in terms of these decompositions,

$$\begin{aligned}
&\|\mathrm{M} - \mathrm{X}\mathrm{Y}^\top\|_F^2 + \frac{\lambda}{r} \|\mathrm{X}\mathrm{Y}^\top\|_*^2 \\
&= \|\mathrm{U}_\mathrm{M} \Sigma_\mathrm{M} \mathrm{V}_\mathrm{M}^\top - \mathrm{U}_\mathrm{W} \Sigma_\mathrm{W} \mathrm{V}_\mathrm{W}^\top\|_F^2 + \frac{\lambda}{r} \|\mathrm{U}_\mathrm{W} \Sigma_\mathrm{W} \mathrm{V}_\mathrm{W}^\top\|_*^2 \\
&= \|\Sigma_\mathrm{M} - \mathrm{U}' \Sigma_\mathrm{W} \mathrm{V}'^\top\|_F^2 + \frac{\lambda}{r} \|\Sigma_\mathrm{W}\|_*^2 \\
&= \|\Sigma_\mathrm{M}\|_F^2 + \|\Sigma_\mathrm{W}\|_F^2 - 2\langle \Sigma_\mathrm{M}, \mathrm{U}' \Sigma_\mathrm{W} \mathrm{U}'^\top \rangle + \frac{\lambda}{r} \|\Sigma_\mathrm{W}\|_*^2
\end{aligned}$$

where $\mathrm{U}' = \mathrm{U}_\mathrm{M}^\top \mathrm{U}_\mathrm{W}$. By Von Neumann's trace inequality, for a fixed Σ_W we have that $\langle \Sigma_\mathrm{M}, \mathrm{U}' \Sigma_\mathrm{W} \mathrm{U}'^\top \rangle \leq \sum_{i=1}^{d} \sigma_i(\mathrm{M}) \sigma_i(\mathrm{W})$, where the equality is achieved when $\Sigma_i(\mathrm{W})$ have the same ordering as $\Sigma_i(\mathrm{M})$ and $\mathrm{U}' = \mathrm{I}$, i.e. $\mathrm{U}_\mathrm{M} = \mathrm{U}_\mathrm{W}$. Now, Problem 2.16 is reduced to

$$\begin{aligned}
&\min_{\substack{\|\Sigma_\mathrm{W}\|_0 \leq r, \\ \Sigma_\mathrm{W} \geq 0}} \|\Sigma_\mathrm{M} - \Sigma_\mathrm{W}\|_F^2 + \frac{\lambda}{r} \|\Sigma_\mathrm{W}\|_*^2 \\
&= \min_{\bar{\sigma} \in \mathbb{R}_+^r} \sum_{i=1}^{r} (\sigma_i(\mathrm{M}) - \bar{\sigma}_i)^2 + \sum_{i=r+1}^{d} \sigma_i^2(\mathrm{M}) + \frac{\lambda}{r} \left(\sum_{i=1}^{r} \bar{\sigma}_i \right)^2
\end{aligned}$$

The Lagrangian is given by

$$\begin{aligned}
L(\bar{\lambda}, \alpha) = &\sum_{i=1}^{r} (\sigma_i(\mathrm{M}) - \bar{\sigma}_i)^2 + \sum_{i=r+1}^{d} \sigma_i^2(\mathrm{M}) \\
&+ \frac{\lambda}{r} \left(\sum_{i=1}^{r} \bar{\sigma}_i \right)^2 - \sum_{i=1}^{r} \alpha_i \bar{\sigma}_i
\end{aligned}$$

The KKT conditions ensures that $\forall i = 1, \ldots, r$,

$$\begin{aligned}
&\bar{\sigma}_i \geq 0,\ \alpha_i \geq 0,\ \bar{\sigma}_i \alpha_i = 0 \\
&2(\bar{\sigma}_i - \sigma_i(\mathrm{M})) + \frac{2\lambda}{r}\left(\sum_{i=1}^{r} \bar{\sigma}_i\right) - \alpha_i = 0
\end{aligned}$$

Let $\rho = |i : \bar{\sigma}_i > 0| \leq r$ be the number of nonzero $\bar{\sigma}_i$. For $i = 1, \ldots, \rho$ we have $\alpha_i = 0$, hence

$$\begin{aligned}
&\bar{\sigma}_i + \frac{\lambda}{r}\left(\sum_{i=1}^{\rho} \bar{\sigma}_i\right) = \sigma_i(\mathrm{M}) \\
&\implies (\mathrm{I}_\rho + \frac{\lambda}{r} 1 1^\top)\bar{\sigma}_{1:\rho} = \sigma_{1:\rho}(\mathrm{M}) \\
&\implies \bar{\sigma}_{1:\rho} = (\mathrm{I}_\rho - \frac{\lambda}{r + \lambda\rho} 1 1^\top)\sigma_{1:\rho}(\mathrm{M}) \\
&\implies \bar{\sigma}_{1:\rho} = \sigma_{1:\rho}(\mathrm{M}) - \frac{\lambda\rho\kappa_\rho}{r + \lambda\rho} 1_\rho \\
&\implies \Sigma_{\mathrm{W}} = (\Sigma_{\mathrm{M}} - \frac{\lambda\rho\kappa_\rho}{r + \lambda\rho}\mathrm{I}_d)_+
\end{aligned}$$

where $\kappa_\rho = \frac{1}{\rho}\sum_{i=1}^{\rho} \sigma_i(\mathrm{M})$ and the second implication holds since $(\mathrm{I}_\rho + \frac{\lambda}{r} 1 1^\top)^{-1} = \mathrm{I}_\rho - \frac{\lambda}{r+\lambda\rho} 1 1^\top$. It only remains to find the optimal ρ. Let's define the function

$$\begin{aligned}
g(\rho) :=& \sum_{i=1}^{\rho} (\sigma_i(\mathrm{M}) - \bar{\sigma}_i)^2 + \sum_{i=\rho+1}^{d} \sigma_i^2(\mathrm{M}) + \frac{\lambda}{r}\left(\sum_{i=1}^{\rho} \bar{\sigma}_i\right)^2 \\
=& \sum_{i=1}^{\rho}\left(\frac{\lambda \sum_{k=1}^{\rho} \sigma_k(\mathrm{M})}{r + \lambda\rho}\right)^2 + \sum_{i=\rho+1}^{d} \sigma_i(\mathrm{M})^2 \\
&+ \frac{\lambda}{r}\left(\sum_{i=1}^{\rho}\left(\sigma_i(\mathrm{M}) - \frac{\lambda \sum_{k=1}^{\rho} \sigma_k(\mathrm{M})}{r + \lambda\rho}\right)\right)^2.
\end{aligned}$$

By Lemma 21, $g(\rho)$ is monotonically non-increasing in ρ, hence ρ should be the largest *feasible* integer, i.e.

$$\rho = \max\{j :\ \sigma_j > \frac{\lambda j \kappa_j}{r + \lambda j}\}.$$

□

2.5.3 Proofs of Theorems in Sections 2.3

In this section for ease of notation we let λ_i denote $\lambda_i(\mathrm{M})$. Furthermore, with slight abuse of notation we let $f(\mathrm{U})$, $\ell(\mathrm{U})$ and $R(\mathrm{U})$ denote the objective, the loss function

and the regularizer, respectively.

It is easy to see that the gradient of the objective of Problem 2.4 is given by

$$\nabla f(\mathrm{U}) = 4(\mathrm{U}\mathrm{U}^\top - \mathrm{M})\mathrm{U} + 4\lambda \mathrm{U}\,\mathrm{diag}(\mathrm{U}^\top \mathrm{U}).$$

We first make the following important observation about the critical points of Problem 2.4.

Lemma 3. *If U is a critical point of Problem 2.4, then it holds that $UU^\top \preceq M$.*

Proof of Lemma 3. Since $\nabla f(\mathrm{U}) = 0$, we have that

$$(\mathrm{M} - \mathrm{U}\mathrm{U}^\top)\mathrm{U} = \lambda \mathrm{U}\,\mathrm{diag}(\mathrm{U}^\top \mathrm{U})$$

multiply both sides from right by $\mathrm{U}^\top$ and rearrange to get

$$\mathrm{M}\mathrm{U}\mathrm{U}^\top = \mathrm{U}\mathrm{U}^\top \mathrm{U}\mathrm{U}^\top + \lambda \mathrm{U}\,\mathrm{diag}(\mathrm{U}^\top \mathrm{U})\mathrm{U}^\top \tag{2.17}$$

Note that the right hand side is symmetric, which implies that the left hand side must be symmetric as well, i.e.

$$\mathrm{M}\mathrm{U}\mathrm{U}^\top = (\mathrm{M}\mathrm{U}\mathrm{U}^\top)^\top = \mathrm{U}\mathrm{U}^\top \mathrm{M},$$

so that M and $\mathrm{U}\mathrm{U}^\top$ commute. Note that in Equation (2.17), $\mathrm{U}\,\mathrm{diag}(\mathrm{U}^\top \mathrm{U})\mathrm{U}^\top \succeq 0$. Thus, $\mathrm{M}\mathrm{U}\mathrm{U}^\top \succeq \mathrm{U}\mathrm{U}^\top \mathrm{U}\mathrm{U}^\top$. Let $\mathrm{U}\mathrm{U}^\top = \mathrm{W}\Gamma\mathrm{W}^\top$ be a compact eigendecomposition of $\mathrm{U}\mathrm{U}^\top$. We get

$$\mathrm{M}\mathrm{U}\mathrm{U}^\top = \mathrm{M}\mathrm{W}\Gamma\mathrm{W}^\top \succeq \mathrm{U}\mathrm{U}^\top \mathrm{U}\mathrm{U}^\top = \mathrm{W}\Gamma^2\mathrm{W}^\top.$$

Multiplying from right and left by $\mathrm{W}\Gamma^{-1}$ and $\mathrm{W}^\top$ respectively, we have that

$$\mathrm{W}^\top \mathrm{M}\mathrm{W} \succeq \Gamma$$

which completes the proof. □

Lemma 3 allows us to bound different norms of the critical points, as will be seen later in the proofs.

To explore the landscape properties of Problem 2.4, we first focus on the non-equalized critical points in Lemma 4. We show that the set of non-equalized critical points does not include any local optima. Furthermore, all such points are strict saddles. Therefore, we turn our focus to the equalized critical points in Lemma 5. We show all such points inherit the eigenspace of the input matrix M. This allows us to give a closed-form characterization of all the equalized critical points in terms of the eigendecompostion of M. We then show that if λ is chosen appropriately, all such critical points that are not global optima, are strict saddle points.

Lemma 4. *All local minima of Problem 2.4 are equalized. Moreover, all critical points that are not equalized, are strict saddle points.*

Proof of Lemma 4. We show that if U is not equalized, then any ϵ-neighborhood of U contains a point with objective strictly smaller than $f(\mathrm{U})$. More formally, for any $\epsilon > 0$, we exhibit a rotation Q_ϵ such that $\|\mathrm{U} - \mathrm{U}\mathrm{Q}_\epsilon\|_F \leq \epsilon$ and $f(\mathrm{U}\mathrm{Q}_\epsilon) < f(\mathrm{U})$. Let U be a critical point of Problem 2.4 that is not equalized, i.e. there exists two columns of U with different norms. Without loss of generality, let $\|\mathrm{u}_1\| > \|\mathrm{u}_2\|$. We design a rotation matrix Q such that it is almost an isometry, but it moves mass from u_1 to u_2. Consequently, the new factor becomes "less un-equalized" and achieves a smaller regularizer, while preserving the value of the loss. To that end, define

$$\mathrm{Q}_\delta := \begin{bmatrix} \sqrt{1-\delta^2} & -\delta & 0 \\ \delta & \sqrt{1-\delta^2} & 0 \\ 0 & 0 & \mathrm{I}_{r-2} \end{bmatrix}$$

and let $\widehat{\mathrm{U}} := \mathrm{U}\mathrm{Q}_\delta$. It is easy to verify that Q_ϵ is indeed a rotation. First, we show that

for any ϵ, as long as $\delta^2 \leq \frac{\epsilon^2}{2\,\mathrm{Trace(M)}}$, we have $\widehat{\mathrm{U}} \in \mathcal{B}_\epsilon(\mathrm{U})$:

$$\begin{aligned}
\|\mathrm{U} - \widehat{\mathrm{U}}\|_F^2 &= \sum_{i=1}^{r} \|\mathrm{u}_i - \widehat{\mathrm{u}}_i\|^2 \\
&= \|\mathrm{u}_1 - \sqrt{1-\delta^2}\mathrm{u}_1 - \delta \mathrm{u}_2\ \|^2 \\
&+ \|\mathrm{u}_2 - \sqrt{1-\delta^2}\mathrm{u}_2 + \delta \mathrm{u}_1\ \|^2 \\
&= 2(1-\sqrt{1-\delta^2})(\|\mathrm{u}_1\|^2 + \|\mathrm{u}_2\|^2) \\
&\leq 2\delta^2\,\mathrm{Trace(M)} \leq \epsilon^2
\end{aligned}$$

where the second to last inequality follows from Lemma 3, because $\|\mathrm{u}_1\|^2 + \|\mathrm{u}_2\|^2 \leq \|\mathrm{U}\|_F^2 = \mathrm{Trace}(\mathrm{U}\mathrm{U}^\top) \leq \mathrm{Trace(M)}$, and also the fact that $1 - \sqrt{1-\delta^2} = \frac{1-1+\delta^2}{1+\sqrt{1-\delta^2}} \leq \delta^2$.

Next, we show that for small enough δ, the value of the function at $\widehat{\mathrm{U}}$ is strictly smaller than that of U. Observe that

$$\begin{aligned}
\|\widehat{\mathrm{u}}_1\|^2 &= (1-\delta^2)\|\mathrm{u}_1\|^2 + \delta^2\|\mathrm{u}_2\|^2 + 2\delta\sqrt{1-\delta^2}\mathrm{u}_1^\top \mathrm{u}_2 \\
\|\widehat{\mathrm{u}}_2\|^2 &= (1-\delta^2)\|\mathrm{u}_2\|^2 + \delta^2\|\mathrm{u}_1\|^2 - 2\delta\sqrt{1-\delta^2}\mathrm{u}_1^\top \mathrm{u}_2
\end{aligned}$$

and the remaining columns will not change, i.e. for $i = 3, \cdots, r$, $\widehat{\mathrm{u}}_i = \mathrm{u}_i$. Together with the fact that Q_δ preserves the norms, i.e. $\|\mathrm{U}\|_F = \|\mathrm{U}\mathrm{Q}_\delta\|_F$, we get

$$\|\widehat{\mathrm{u}}_1\|^2 + \|\widehat{\mathrm{u}}_2\|^2 = \|\mathrm{u}_1\|^2 + \|\mathrm{u}_2\|^2. \tag{2.18}$$

Let $\delta = -c \cdot \mathrm{sgn}(\mathrm{u}_1^\top \mathrm{u}_2)$ for a small enough $c > 0$ such that $\|\mathrm{u}_2\| < \|\widehat{\mathrm{u}}_2\| \leq \|\widehat{\mathrm{u}}_1\| < \|\mathrm{u}_1\|$. Using Equation (2.18), This implies that $\|\widehat{\mathrm{u}}_1\|^4 + \|\widehat{\mathrm{u}}_2\|^4 < \|\mathrm{u}_1\|^4 + \|\mathrm{u}_2\|^4$, which in turn gives us $R(\widehat{\mathrm{U}}) < R(\mathrm{U})$ and hence $f(\widehat{\mathrm{U}}) < f(\mathrm{U})$. Therefore, a non-equalized critical point cannot be local minimum, hence the first claim of the lemma.

We now prove the second part of the lemma. Let U be a critical point that is not equalized. To show that U is a strict saddle point, it suffices to show that the Hessian has a negative eigenvalue. In here, we exhibit a curve along which the second directional derivative is negative. Assume, without loss of generality that $\|\mathrm{u}_1\| > \|\mathrm{u}_2\|$ and consider the curve

$$\Delta(t) := [(\sqrt{1-t^2}-1)\mathrm{u}_1 + t\mathrm{u}_2, (\sqrt{1-t^2}-1)\mathrm{u}_2 - t\mathrm{u}_1, 0_{d,r-2}]$$

It is easy to check that for any $t \in \mathbb{R}$, $\ell(\mathrm{U} + \Delta(t)) = \ell(\mathrm{U})$ since $\mathrm{U} + \Delta(t)$ is essentially a rotation on U and ℓ is invariant under rotations. Observe that

$$\begin{aligned}
g(t) &:= f(\mathrm{U} + \Delta(t)) \\
&= f(\mathrm{U}) + \|\sqrt{1-t^2}\mathrm{u}_1 + t\mathrm{u}_2\|^4 - \|\mathrm{u}_1\|^4 \\
&+ \|\sqrt{1-t^2}\mathrm{u}_2 - t\mathrm{u}_1\|^4 - \|\mathrm{u}_2\|^4 \\
&= f(\mathrm{U}) - 2t^2(\|\mathrm{u}_1\|^4 + \|\mathrm{u}_2\|^4) + 8t^2(\mathrm{u}_1\mathrm{u}_2)^2 \\
&+4t^2\|\mathrm{u}_1\|^2\|\mathrm{u}_2\|^2 + 4t\sqrt{1-t^2}\mathrm{u}_1^\top\mathrm{u}_2(\|\mathrm{u}_1\|^2 - \|\mathrm{u}_2\|^2) + O(t^3).
\end{aligned}$$

The derivative of g then is given as

$$\begin{aligned}
g'(t) = &-4t(\|\mathrm{u}_1\|^4 + \|\mathrm{u}_2\|^4) + 16t(\mathrm{u}_1\mathrm{u}_2)^2 + 8t\|\mathrm{u}_1\|^2\|\mathrm{u}_2\|^2 \\
&+4(\sqrt{1-t^2} - \frac{t^2}{\sqrt{1-t^2}})(\mathrm{u}_1^\top\mathrm{u}_2)(\|\mathrm{u}_1\|^2 - \|\mathrm{u}_2\|^2) + O(t^2).
\end{aligned}$$

Since U is a critical point and f is continuously differentiable, it should hold that $g'(0) = 4(\mathrm{u}_1^\top\mathrm{u}_2)(\|\mathrm{u}_1\|^2 - \|\mathrm{u}_2\|^2) = 0$. Since by assumption $\|\mathrm{u}_1\|^2 - \|\mathrm{u}_2\|^2 > 0$, it should be the case that $\mathrm{u}_1^\top\mathrm{u}_2 = 0$. We now consider the second order directional derivative:

$$\begin{aligned}
g''(0) &= -4(\|\mathrm{u}_1\|^4 + \|\mathrm{u}_2\|^4) + 16(\mathrm{u}_1\mathrm{u}_2)^2 + 8\|\mathrm{u}_1\|^2\|\mathrm{u}_2\|^2 \\
&= -4(\|\mathrm{u}_1\|^2 - \|\mathrm{u}_2\|^2)^2 < 0
\end{aligned}$$

which completes the proof. □

We now focus on the critical points that are equalized, i.e. points U such that $\nabla f(\mathrm{U}) = 0$ and $\mathrm{diag}(\mathrm{U}^\top\mathrm{U}) = \frac{\|\mathrm{U}\|_F^2}{r}\mathrm{I}$.

Lemma 5. *Assume that* $\lambda < \frac{r\lambda_r}{\sum_{i=1}^r \lambda_i - r\lambda_r}$. *Then all equalized local minima are global. All other equalized critical points are strict saddle points.*

Proof of Lemma 5. Let $\mathrm{U} = \mathrm{W}\Sigma\mathrm{V}^\top$ be a compact SVD of the rank-r' weight matrix

U. We have:

$$\begin{aligned}
&\nabla f(\mathrm{U}) = 4(\mathrm{U}\mathrm{U}^\top - \mathrm{M})\mathrm{U} + 4\lambda \mathrm{U}\,\mathrm{diag}(\mathrm{U}^\top \mathrm{U}) = 0 \\
\implies &\mathrm{U}\mathrm{U}^\top \mathrm{U} + \frac{\lambda \|\mathrm{U}\|_F^2}{r}\mathrm{U} = \mathrm{M}\mathrm{U} \\
\implies &\mathrm{W}\Sigma^3 \mathrm{V}^\top + \frac{\lambda \|\Sigma\|_F^2}{r}\mathrm{W}\Sigma\mathrm{V}^\top = \mathrm{M}\mathrm{W}\Sigma\mathrm{V}^\top \\
\implies &\Sigma^2 + \frac{\lambda \|\Sigma\|_F^2}{r}\mathrm{I} = \mathrm{W}^\top \mathrm{M}\mathrm{W}
\end{aligned}$$

Since the left hand side of the above equality is diagonal, it implies that $\mathrm{W} \in \mathbb{R}^{d \times r'}$ corresponds to some r' eigenvectors of M. Let $\mathcal{E} \subseteq [d],\ |\mathcal{E}| = r'$ denote the set of eigenvectors of M that are present in W. Note that the above is equivalent of the following system of linear equations:

$$(\mathrm{I} + \frac{\lambda}{r} 1 1^\top)\sigma^2 = \vec{\lambda},$$

where $\sigma^2 := \mathrm{diag}(\Sigma^2)$ and $\vec{\lambda} = \mathrm{diag}(\mathrm{W}^\top \mathrm{M}\mathrm{W})$. By Lemma 20, the solution to this linear system is given by

$$\sigma^2 = (\mathrm{I} - \frac{\lambda}{r + \lambda r'})\vec{\lambda}. \tag{2.19}$$

The set $\mathcal{E}$ belongs to one of the following categories:

1. $\mathcal{E} = [r'],\ r' = \rho$
2. $\mathcal{E} = [r'],\ r' < \rho$
3. $\mathcal{E} \neq [r']$

The case $\mathcal{E} = [r'],\ r' > \rho$ is excluded from the above partition, since whenever $\mathcal{E} = [r']$, it should hold that $r' \leq \rho$. To see this, note that due to $\mathrm{U} = \mathrm{W}\Sigma\mathrm{V}^\top$ being a compact SVD of M, it holds that $\sigma_j > 0$ for all $j \in [r']$. Specifically for $j = r'$, plugging $\sigma_{r'} > 0$ back to Equation (2.19), we get

$$\lambda_{r'} > \frac{\lambda \sum_{i=1}^{r'} \lambda_i}{r + \lambda r'} = \frac{\lambda r' \kappa_{r'}}{r + \lambda r'}.$$

Then it follows from definition of ρ in Theorem 3 that $r' \leq \rho$. We provide a case by case analysis for the above partition here.

Case 1. $[\mathcal{E} = [r'],\ r' = \rho]$ When W corresponds to the top-ρ eigenvectors of M, we retrieve the global optimal solution described by Theorem 3. Therefore, all such critical points are global minima.

Case 2. $[\mathcal{E} = [r'],\ r' < \rho]$ Let $\mathrm{W}_r := [\mathrm{W}, \mathrm{W}_\perp]$ be the top-r eigenvectors of M and $\mathrm{V}_\perp$ span the orthogonal subspace of V, i.e. $\mathrm{V}_r := [\mathrm{V}, \mathrm{V}_\perp]$ be an orthonormal basis for $\mathbb{R}^r$. Define $\mathrm{U}(t) = \mathrm{W}_r \Sigma' \mathrm{V}_r^\top$ where $\sigma_i' = \sqrt{\sigma_i^2 + t^2}$ for $i \leq r$. Observe that

$$\mathrm{U}(t)^\top \mathrm{U}(t) = \mathrm{V}\Sigma\mathrm{V}^\top + t^2 \mathrm{V}_r^\top \mathrm{V}_r = \mathrm{U}^\top \mathrm{U} + t^2 \mathrm{I}_r$$

so that for all t, the parametric curve $\mathrm{U}(t)$ is equalized. The value of the loss function at $\mathrm{U}(t)$ is given by:

$$\begin{aligned}
\ell(\mathrm{U}(t)) &= \sum_{i=1}^{r} (\lambda_i - \sigma_i^2 - t^2)^2 + \sum_{i=r+1}^{d} (\lambda_i)^2 \\
&= \ell(\mathrm{U}) + rt^4 - 2t^2 \sum_{i=1}^{r} (\lambda_i - \sigma_i^2).
\end{aligned}$$

Furthermore, since $\mathrm{U}(t)$ is equalized, we obtain the following form for the regularizer:

$$\begin{aligned}
R(\mathrm{U}(t)) &= \frac{\lambda}{r} \|\mathrm{U}(t)\|_F^4 = \frac{\lambda}{r} \left(\|\mathrm{U}\|_F^2 + rt^2 \right)^2 \\
&= \ell(\mathrm{U}) + \lambda r t^4 + 2\lambda t^2 \|\mathrm{U}\|_F^2.
\end{aligned}$$

Now define $g(t) := \ell(\mathrm{U}(t)) + R(\mathrm{U}(t))$ and observe

$$\begin{aligned}
g(t) = \ell(\mathrm{U}) + R(\mathrm{U}) + rt^4 - 2t^2 \sum_{i=1}^{r} (\lambda_i - \sigma_i^2) \\
+ \lambda r t^4 + 2\lambda t^2 \|\mathrm{U}\|_F^2.
\end{aligned}$$

It is easy to verify that $g'(0) = 0$. Moreover, the second derivative of g at the origin is

given as:

$$\begin{aligned} g''(0) &= -4\sum_{i=1}^{r}(\lambda_i - \sigma_i^2) + 4\lambda\|\mathrm{U}\|_F^2 \\ &= -4\sum_{i=1}^{r}\lambda_i + 4(1+\lambda)\|\mathrm{U}\|_F^2 \\ &= -4\sum_{i=1}^{r}\lambda_i + 4\frac{r + r\lambda}{r + \lambda r'}\sum_{i=1}^{r'}\lambda_i \end{aligned}$$

where the last equality follows from the fact Equation (2.19) and the fact that $\|\mathrm{U}\|_F^2 = \sum_{i=1}^{r'}\sigma_i^2$. To get a sufficient condition for U to be a strict saddle point, we set $g''(0) < 0$:

$$\begin{aligned} & -4\sum_{i=r'+1}^{r}\lambda_i + 4\frac{(r-r')\lambda}{r+\lambda r'}\sum_{i=1}^{r'}\lambda_i < 0 \\ \implies & \frac{(r-r')\lambda}{r+\lambda r'}\sum_{i=1}^{r'}\lambda_i < \sum_{i=r'+1}^{r}\lambda_i \\ \implies & \lambda < \frac{(r+\lambda r')\sum_{i=r'+1}^{r}\lambda_i}{(r-r')\sum_{i=1}^{r'}\lambda_i} \\ \implies & \lambda(1 - \frac{r'\sum_{i=r'+1}^{r}\lambda_i}{(r-r')\sum_{i=1}^{r'}\lambda_i}) < \frac{r\sum_{i=r'+1}^{r}\lambda_i}{(r-r')\sum_{i=1}^{r'}\lambda_i} \\ \implies & \lambda < \frac{r\sum_{i=r'+1}^{r}\lambda_i}{(r-r')\sum_{i=1}^{r'}\lambda_i - r'\sum_{i=r'+1}^{r}\lambda_i} \\ \implies & \lambda < \frac{rh(r')}{\sum_{i=1}^{r'}(\lambda_i - h(r'))} \end{aligned}$$

where $h(r') := \frac{\sum_{i=r'+1}^{r}\lambda_i}{r-r'}$ is the average of the eigenvalues $\lambda_{r'+1}, \cdots, \lambda_r$. It is easy to see that the right hand side is monotonically decreasing with r', since $h(r')$ monotonically decrease with r'. Hence, it suffices to make sure that λ is smaller than the right hand side for the choice of $r' = r - 1$, i.e. $\lambda < \frac{r\lambda_r}{\sum_{i=1}^{r}(\lambda_i - \lambda_r)}$.

Case 3. $[\mathcal{E} \neq [r']]$ We show that all such critical points are strict saddle points. Let w' be one of the top r' eigenvectors that are missing in W. Let $j \in \mathcal{E}$ be such that w_j is not among the top r' eigenvectors of M. For any $t \in [0, 1]$, let $\mathrm{W}(t)$ be identical to W in all the columns but the j^{th} one, where $\mathrm{w}_j(t) = \sqrt{1-t^2}\mathrm{w}_j + t\mathrm{w}'$. Note that $\mathrm{W}(t)$ is still an orthogonal matrix for all values of t. Define the parametrized curve

$\mathrm{U}(t) := \mathrm{W}(t)\Sigma\mathrm{V}^\top$ for $t \in [0,1]$ and observe that:

$$\begin{aligned}\|\mathrm{U} - \mathrm{U}(t)\|_F^2 &= \sigma_j^2 \|\mathrm{w}_j - \mathrm{w}_j(t)\|^2 \\ &= 2\sigma_j^2(1 - \sqrt{1-t^2}) \leq t^2 \operatorname{Trace} \mathrm{M}\end{aligned}$$

That is, for any $\epsilon > 0$, there exist a $t > 0$ such that $\mathrm{U}(t)$ belongs to the ϵ-ball around U. We show that $f(\mathrm{U}(t))$ is strictly smaller than $f(\mathrm{U})$, which means U cannot be a local minimum. Note that this construction of $\mathrm{U}(t)$ guarantees that $R(\mathrm{U}') = R(\mathrm{U})$. In particular, it is easy to see that $\mathrm{U}(t)^\top \mathrm{U}(t) = \mathrm{U}^\top \mathrm{U}$, so that $\mathrm{U}(t)$ remains equalized for all values of t. Moreover, we have that

$$\begin{aligned}f(\mathrm{U}(t)) - f(\mathrm{U}) &= \|\mathrm{M} - \mathrm{U}(t)\mathrm{U}(t)^\top\|_F^2 - \|\mathrm{M} - \mathrm{U}\mathrm{U}^\top\|_F^2 \\ &= -2 \operatorname{Trace}(\Sigma^2 \mathrm{W}(t)^\top \mathrm{M} \mathrm{W}(t)) + 2 \operatorname{Trace}(\Sigma^2 \mathrm{W}^\top \mathrm{M} \mathrm{W}) \\ &= -2\sigma_j^2 t^2 (\mathrm{w}_j(t)^\top \mathrm{M} \mathrm{w}_j(t) - \mathrm{w}_j^\top \mathrm{M} \mathrm{w}_j) < 0,\end{aligned}$$

where the last inequality follows because by construction $\mathrm{w}_j(t)^\top \mathrm{M} \mathrm{w}_j(t) > \mathrm{w}_j^\top \mathrm{M} \mathrm{w}_j$. Define $g(t) := f(\mathrm{U}(t)) = \ell(\mathrm{U}(t)) + R(\mathrm{U}(t))$. To see that such saddle points are non-degenerate, it suffices to show $g''(0) < 0$. It is easy to check that the second directional derivative at the origin is given by

$$g''(0) = -4\sigma_j^2 (\mathrm{w}_j(t)^\top \mathrm{M} \mathrm{w}_j(t) - \mathrm{w}_j^\top \mathrm{M} \mathrm{w}_j) < 0,$$

which completes the proof. □

Proof of Lemma 1. Follows from Lemma 4 □

Proof of Theorem 7. Follows from Lemma 4 and Lemma 5. □

2.6 Empirical Results

Dropout is a popular algorithmic technique used for avoiding overfitting when training large deep neural networks. The goal of this section is not to attest to the already

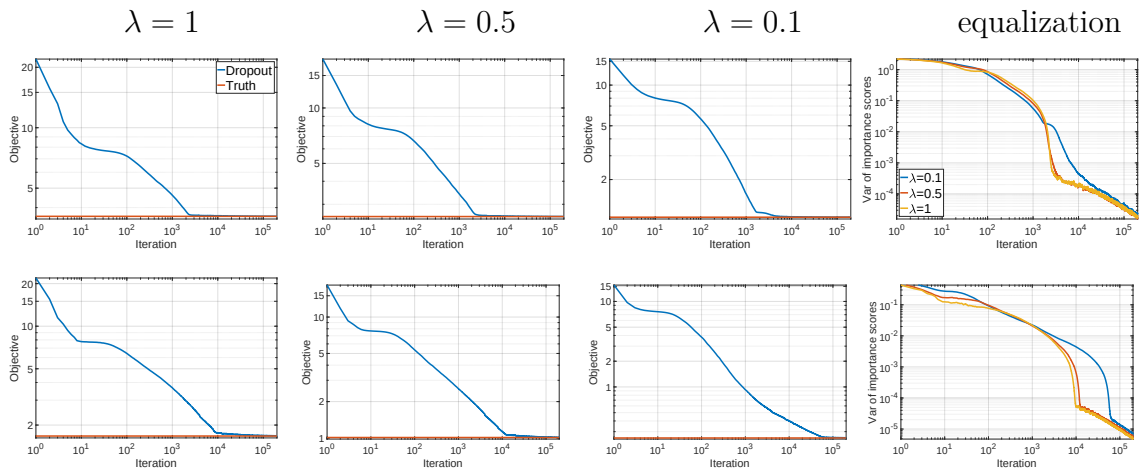

Figure 2-3. Dropout converges to global optima for different values of $\lambda \in \{0.1, 0.5, 1\}$ and different widths of the hidden layer $r = 20$ (top) and $r = 80$ (bottom). The right column shows the variance of the product of column-wise norms for each of the weight matrices. As can be seen, the weight matrices become equalized very quickly since variance goes to zero.

well-established success of dropout. Instead, the purpose of this section is to simply confirm the theoretical results we showed in the previous section, as a proof of concept.

We begin with a toy example in order to visually illustrate the optimization landscape. We use dropout to learn a simple linear auto-encoder with one-dimensional input and output (i.e. a network represented by a scalar $\mathrm{M} = 2$) and a single hidden layer of width $r = 2$. The input features are sampled for a standard normal distribution. Figure 2-2 shows the optimization landscape along with the contours of the level sets, and a trace of iterates of dropout (Algorithm 1). The initial iterates and global optima (given by Theorem 3) are shown by red and green dots, respectively. Since at any global optimum the weights are equalized, the optimal weight vector in this case is parallel to the vector $(\pm 1, \pm 1)$. We see that dropout converges to a global minimum.

For a second illustrative experiment, we use Algorithm 1 to train a shallow linear network, where the input $\mathrm{x} \in \mathbb{R}^{80}$ is distributed according to the standard Normal distribution. The output $\mathrm{y} \in \mathbb{R}^{120}$ is generated as $\mathrm{y} = \mathrm{Mx}$, where $\mathrm{M} \in \mathbb{R}^{120 \times 80}$ is drawn randomly by uniformly sampling the right and left singular subspaces and with

a spectrum decaying exponentially. Figure 2-3 illustrates the behavior of Algorithm 1 for different values of the regularization parameter ($\lambda \in \{0.1, 0.5, 1\}$), and for different sizes of factors ($r \in \{20, 80\}$). The curve in blue shows the objective value for the iterates of dropout, and the line in red shows the optimal value of the objective (i.e. objective for a global optimum found using Theorem 6). All plots are averaged over 50 runs of Algorithm 1 (averaged over different random initializations, random realizations of Bernoulli dropout, as well as random draws of training examples).

To verify that the solution found by dropout actually has equalized factors, we consider the following measure. At each iteration, we compute the "importance scores", $\alpha_t^{(i)} = \|\mathrm{u}_{ti}\|\|\mathrm{v}_{ti}\|,\ i \in [r]$, where u_{ti} and v_{ti} are the i-th columns of U_t and V_t, respectively. The rightmost panel of Figure 2-3 shows the variance of $\alpha_t^{(i)}$'s, over the hidden nodes $i \in [r]$, at each iterate t. Note that a high variance in α_t corresponds to large variation in the values of $\|\mathrm{u}_{ti}\|\|\mathrm{v}_{ti}\|$. When the variance is equal to zero, all importance scores are equal, thus the factors are equalized. We see that iterations of Algorithm 1 decrease this measure monotonically, and the larger the value of λ, the faster the weights become equalized.

2.7 Discussion

In this chapter, we study the inductive bias of dropout in shallow linear networks. We show that dropout prefers solutions with minimal path regularization which yield strong capacity control guarantees in deep learning. Despite being a non-convex optimization problem, we are able to fully characterize the global optima of the dropout objective. Our analysis shows that dropout favors low-rank weight matrices that are equalized. This theoretical finding confirms that dropout as a procedure uniformly allocates weights to different subnetworks, which is akin to preventing co-adaptation.

We characterize the optimization landscape of learning autoencoders with dropout. We first show that the local optima inherit the same implicit bias as global optimal, i.e. all local optima are equalized. Then, we show that for sufficiently small dropout rates, there are no spurious local minima in the landscape, and all saddle points are non-degenerate. These properties suggest that dropout – as an optimization procedure – can efficiently converge to a globally optimal solution specified by our theorems. Extending (or rejecting) these benign landscape properties beyond shallow autoencoders is an interesting open problem that we leave for the future work.

Chapter 3

Dropout Regularizer: Deep Linear Networks

In Chapter 2, we studied dropout in linear regression, focusing on two-layer linear networks, under the assumption that the input marginals are isotropic. We showed that, learning such shallow models with dropout amounts to regularizing the objective with ℓ_2-path norm of the network [NTS15], and completely characterized the optima of the resulting regularized risk minimization problem. We discovered that under dropout – irrespective of the dropout rate or the width of the network – the optima of the regularized objective obey a fundamental structural property, that we called *equalization*. In particular, in an equalized network, the product of the norms of the weights incoming and outgoing to/from hidden nodes are equal across all neurons. We argued that equalized networks have minimal co-adaptation among hidden neurons, a property which is one of the main motivations behind the invention of dropout [HSK+12, SHK+14]. We further analyzed the non-convex landscape of the dropout objective for the case of linear auto-encoders with tied weights, and showed that it enjoys benign landscape properties that allow dropout to efficiently escape saddle points and find a global optimum.

In this chapter, we completely lift the isotropic assumption that we made in the previous chapter, and analyze dropout for deep linear networks of any architecture, i.e.,

any number of layers, and any number of hidden nodes at each layer [GBCB16]. While the overall function is linear, the representation in factored form makes the optimization landscape non-convex and hence, challenging to analyze. More importantly, we argue that the fact we choose to represent a linear map in a factored form has important implications to the learning problem, akin in many ways to the implicit bias due to stochastic optimization algorithms and various algorithmic heuristics used in deep learning [GWB+17, LMZ18, AH19].

Several recent works have investigated the optimization landscape properties of deep linear networks [BH89, SMG13, Kaw16, HM16, LB18], as well as the implicit bias due to first-order optimization algorithms for training such networks [GLSS18b, JT18], and the convergence rates of such algorithms [BHL18, ACGH18]. The focus here is to have a similar development for dropout when training deep linear networks.

Formally, we consider the hypotheses class of multilayer feed-forward linear networks with input dimension d_0, output dimension d_{k+1}, k hidden layers of widths $d_1, \ldots, d_k$, and linear transformations $\mathrm{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}$, for $i = 1, \ldots, k+1$:

$$\mathcal{L}_{\{d_i\}} = \{g : \mathrm{x} \mapsto \mathrm{W}_{k+1} \cdots \mathrm{W}_1 \mathrm{x},\ \mathrm{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}\}.$$

We refer to the set of $k+1$ integers $\{d_i\}_{i=0}^{k+1}$ specifying the width of each layer as the *architecture* of the function class, the set of the weight parameters $\{\mathrm{W}_i\}_{i=1}^{k+1}$ as an *implementation*, or an element of the function class, and $\mathrm{W}_{k+1\to 1} := \mathrm{W}_{k+1}\mathrm{W}_k \cdots \mathrm{W}_1$ as the *network map*[1].

The focus here is on *deep regression* with dropout under ℓ_2 loss, which is widely used in computer vision tasks, including human pose estimation [TS14], facial landmark detection, and age estimation [LMAPH19]. More formally, we study the following learning problem for deep linear networks. Let $\mathcal{X} \subseteq \mathbb{R}^{d_0}$ and $\mathcal{Y} \subseteq \mathbb{R}^{d_{k+1}}$ denote the input feature space and the output label space, respectively. Let $\mathcal{D}$ denote

[1]Similarly $\mathrm{W}_{i\to j} := \mathrm{W}_i \mathrm{W}_{i-1} \cdots \mathrm{W}_j$ denotes the linear map given by product of the layers $i, i-1, \ldots, j$.

the joint probability distribution on $\mathcal{X} \times \mathcal{Y}$. We assume that $\mathbb{E}[\mathrm{x}\mathrm{x}^\top]$ has full rank. Given a training set $\{\mathrm{x}_i, \mathrm{y}_i\}_{i=1}^n$ drawn i.i.d. from the distribution $\mathcal{D}$, the goal of the learning problem is to minimize the *population risk* under the squared loss $L(\{\mathrm{W}_i\}) := \mathbb{E}_{\mathrm{x},\mathrm{y}}[\|\mathrm{y} - \mathrm{W}_{k+1\to 1}\mathrm{x}\|^2]$. Note that the population risk L depends only on the network map and not the specific implementations of it, i.e. $L(\{\mathrm{W}_i\}) = L(\{\mathrm{W}'_i\})$ for all $\mathrm{W}_{k+1}\cdots\mathrm{W}_1 = \mathrm{W}'_{k+1}\cdots\mathrm{W}'_1$. For that reason, with a slight abuse of notation we write $L(\mathrm{W}_{k+1\to 1}) := L(\{\mathrm{W}_i\})$.

Dropout is an iterative procedure wherein at each iteration each node in the network is dropped independently and identically according to a Bernoulli random variable with parameter θ. Equivalently, we can view dropout, algorithmically, as an instance of stochastic gradient descent for minimizing the following objective over $\{\mathrm{W}_i\}$:

$$L_\theta(\{\mathrm{W}_i\}) := \mathbb{E}_{(\mathrm{x},\mathrm{y},\{\mathrm{b}_i\})}[\|\mathrm{y} - \bar{\mathrm{W}}_{k+1\to 1}\mathrm{x}\|^2], \tag{3.1}$$

where $\bar{\mathrm{W}}_{i\to j} := \frac{1}{\theta^k}\mathrm{W}_i\mathrm{B}_{i-1}\mathrm{W}_{i-1}\cdots\mathrm{B}_j\mathrm{W}_j$, and $\mathrm{B}_l = \mathrm{diag}\,[\mathrm{b}_l(1), \ldots, \mathrm{b}_l(d_l)]$ represents the dropout pattern in the l^{th} layer with Bernoulli random variables on the diagonal; if $\mathrm{B}_l(i,i) = 0$ then all paths from the input to the output that pass through the i^{th} hidden node in the l^{th} layer are turned "off", i.e., those paths have no contribution in determining the output of the network for that instance of the dropout pattern; we refer to the parameter $1-\theta$ as the *dropout rate*. $\bar{\mathrm{W}}_{i\to j}$ is an unbiased estimator of $\mathrm{W}_{i\to j}$, i.e. $\mathbb{E}_{\{\mathrm{b}_i\}}[\bar{\mathrm{W}}_{i\to j}] = \mathrm{W}_{i\to j}$.

We say that the dropout algorithm *succeeds* in training a network if it returns a map $\mathrm{W}_{k+1\to 1}$ that (approximately) minimizes L_θ. In this paper, the central question under investigation is to understand *which network maps/architectures is a successful dropout training biased towards.*

To answer this question, we begin with the following simple observation that

$$L_\theta(\{\mathrm{W}_i\}) = L(\{\mathrm{W}_i\}) + \mathbb{E}_{(\mathrm{x},\{\mathrm{b}_i\})}\|\mathrm{W}_{k+1\to 1}\mathrm{x} - \bar{\mathrm{W}}_{k+1\to 1}\mathrm{x}\|^2$$

In other words, the dropout objective is composed of the population risk $L(\{\mathrm{W}_i\})$ plus an *explicit regularizer* $R(\{\mathrm{W}_i\}) := \mathbb{E}_{(\mathrm{x},\mathrm{y},\{\mathrm{b}_i\})}[\|\mathrm{W}_{k+1\to 1}\mathrm{x} - \bar{\mathrm{W}}_{k+1\to 1}\mathrm{x}\|^2]$ induced by dropout. Denoting the second moment of x by $\mathrm{C} := \mathbb{E}[\mathrm{x}\mathrm{x}^\top]$, we note that $R(\{\mathrm{W}_i\}) = \mathbb{E}_{\{\mathrm{b}_i\}}[\|\mathrm{W}_{k+1\to 1} - \bar{\mathrm{W}}_{k+1\to 1}\|_{\mathrm{C}}^2]$. Since any stochastic network map specified by $\bar{\mathrm{W}}_{k+1\to 1}$ is an unbiased estimator of the network map specified by $\mathrm{W}_{k+1\to 1}$, the explicit regularizer captures the variance of the network implemented by the weights $\{\mathrm{W}_i\}$ under Bernoulli perturbations. By minimizing this variance term, dropout training aims at *breaking co-adaptation between hidden nodes* – it biases towards networks whose random sub-networks yield similar outputs [SHK+14].

If $\{\mathrm{W}_i^*\}$ is an infimum of (3.1), then it minimizes the explicit regularizer among all implementations of the network map,

$$\mathrm{M} = \mathrm{W}_{k+1}^* \cdots \mathrm{W}_1^*, \quad \text{i.e. ,} \quad R(\{\mathrm{W}_i^*\}) = \inf_{\mathrm{W}_{k+1}\cdots\mathrm{W}_1=\mathrm{M}} R(\{\mathrm{W}_i\}).$$

We refer to the infimum of the explicit regularizer over all implementations of a given network map M as the *induced regularizer*:

$$\Theta(\mathrm{M}) := \inf_{\mathrm{W}_{k+1}\cdots\mathrm{W}_1=\mathrm{M}} R(\{\mathrm{W}_i\}) \tag{3.2}$$

The domain of the induced regularizer Θ is the linear maps implementable by the network, i.e., the set $\{\mathrm{M} : \operatorname{Rank}\mathrm{M} \leq \min_i d_i\}$. Since the infimum of the induced regularizer is always attained (see Lemma 6 in the appendix), we can equivalently study the following problem to understand the solution to Problem 3.1 in terms of the network map:

$$\min_{\mathrm{M}} \; L(\mathrm{M}) + \Theta(\mathrm{M}), \quad \operatorname{Rank}\mathrm{M} \leq \min_{i\in[k+1]} d_i. \tag{3.3}$$

To characterize which networks are preferable by dropout training, one needs to understand the explicit regularizer R, understand the induced regularizer Θ, and explore the global minima of Problem 3.3. In this regard, we make several important contributions summarized as follows.

1. We derive the closed form expression for the explicit regularizer $R(\{\mathrm{W}_i\})$ induced by dropout training in deep linear networks. The explicit regularizer is comprised of the ℓ_2-path regularizer as well as other rescaling invariant sub-regularizers.

2. We show that the convex envelope of the induced regularizer is proportional to the squared nuclear norm of the network map, generalizing a similar result for matrix factorization [CHL+18] and single hidden layer linear networks [MAV18]. Furthermore, we show that the induced regularizer equals its convex envelope if and only if the network is *equalized*, a notion that quantitatively measures *co-adaptation* between hidden nodes [MAV18].

3. We completely characterize the global minima of the dropout objective L_θ in Problem 3.1 despite the objective being non-convex, under a simple eigengap condition (see Theorem 10). This gap condition depends on the model, the data distribution, the network architecture and the dropout rate, and is always satisfied by deep linear network architectures with one output neuron.

The rest of this chapter is organized as follows. In Section 3.1, we present the explicit regularizer due to dropout, and in Section 3.2, we analyze the induced regularizer. In Section 3.3, we identify a simple sufficient condition under which we completely characterize the global optima of the dropout objective. Finally, in Section 3.4, we conclude this chapter by providing experimental results confirming our theoretical finding.

3.1 The explicit regularizer

In this section, we give the closed form expression for the explicit regularizer $R(\{\mathrm{W}_i\})$, and discuss some of its important properties.

Proposition 8. *The explicit regularizer is composed of k sub-regularizers:* $R(\{W_i\}) =$

$\sum_{l\in[k]} \lambda^l R_l(\{W_i\})$, *where* $\lambda := \frac{1-\theta}{\theta}$. *Each of the sub-regularizers has the form:*

$$R_l(\{W_i\}) = \sum_{\substack{(j_l,\ldots,j_1)\\ \in\binom{[k]}{l}}} \sum_{\substack{(i_l,\ldots,i_1)\\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}} \alpha^2_{j_1,i_1} \prod_{p=1\cdots l-1} \beta_p^2 \gamma^2_{j_l,i_l}$$

where $\alpha_{j_1,i_1} := \|W_{j_1\to 1}(i_1,:)\|_{\mathrm{C}}$, $\beta_p := W_{j_{p+1}\to j_p+1}(i_{p+1},i_p)$, $\gamma_{j_l,i_l} := \|W_{k+1\to j_l+1}(:,i_l)\|$.

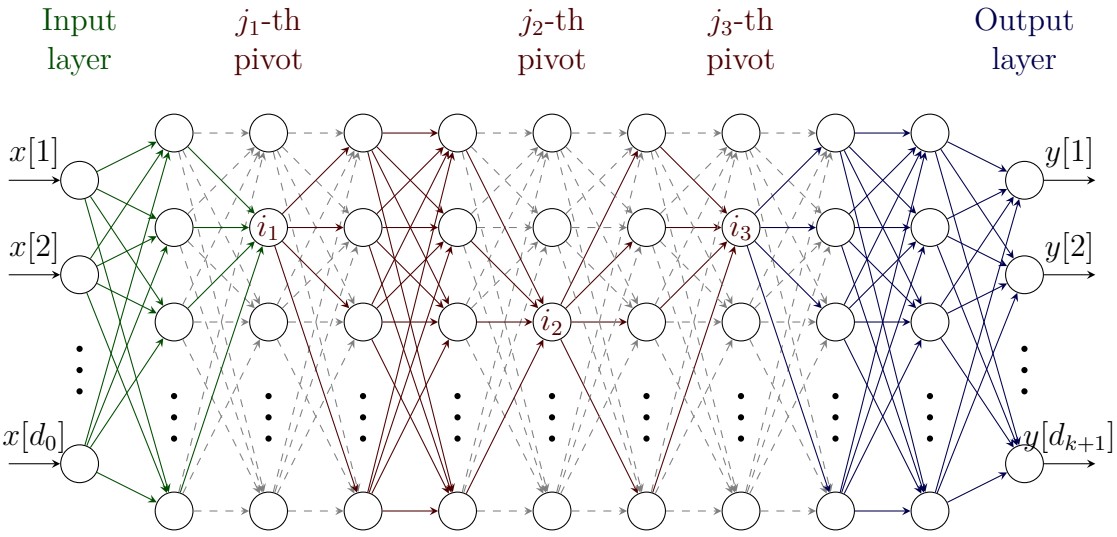

Figure 3-1. Illustration of the explicit regularizer as given in Proposition 8 for $k = 9$, $l = 3$, $(i_1, i_2, i_3) = (2, 3, 2)$ and $(j_1, j_2, j_3) = (2, 5, 7)$. The head term $\alpha^2_{j_1,i_1}$ corresponds to the summation over the product of the weights on any pairs of path from an input node to i_1-th node in the j_1-th hidden layer. Similarly, the tail term $\gamma^2_{j_l,i_l}$ accounts for the product of the weights along any pair of path between the output and the i_l-th node in the j_l-th hidden layer. Each of the middle terms β_p^2, accumulates the product of of the weights along pair of path between i_p-th node in the $(j_p + 1)$-th hidden layer and the i_{p+1}-th node in the j_{p+1}-th hidden layer.

Proof of Proposition 8. A quantity that shows up when analyzing dropout training under squared error is $\mathbb{E}[\|\mathrm{U}\,\mathrm{diag}\,\mathrm{bVx}\|^2]$, where b is a Bernoulli random vector with parameter θ. As we show in Proposition 8, it holds that:

$$\mathbb{E}[\|\mathrm{U}\,\mathrm{diag}\,\mathrm{bVx}\|^2] = \theta^2\mathbb{E}[\|\mathrm{UVx}\|^2] + (\theta-\theta^2)\sum_{j=1}^{r}\|\mathrm{u}_{:j}\|^2\|\mathrm{C}^{\frac{1}{2}}\mathrm{v}_{j:}\|^2. \tag{3.4}$$

We start by expanding the squared error:

$$\begin{aligned} L_\theta(\{\mathrm{W}_i\}_{i=1}^{k+1}) &= \mathop{\mathbb{E}_{\mathrm{b}_i\sim\mathrm{Bern}(\theta)}}_{(\mathrm{x},\mathrm{y})\sim\mathcal{D}}[\|\mathrm{y}-\bar{\mathrm{W}}_{k+1\to 1}\mathrm{x}\|^2] \\ &= \mathbb{E}[\|\mathrm{y}\|^2] - 2\mathbb{E}[\langle\bar{\mathrm{W}}_{k+1\to 1}\mathrm{x},\mathrm{y}\rangle] + \mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to 1}\mathrm{x}\|^2] \\ &= \mathbb{E}[\|\mathrm{y}\|^2] - 2\mathbb{E}\langle\mathrm{W}_{k+1\to 1}\mathrm{x},\mathrm{y}\rangle + \frac{1}{\theta^{2k}}\mathbb{E}[\|\mathrm{W}_{k+1}\,\mathrm{diag}\,\mathrm{b}_k\mathrm{W}_k\ldots\mathrm{diag}\,\mathrm{b}_1\mathrm{W}_1\mathrm{x}\|^2] \end{aligned} \tag{3.5}$$

We now focus on the last term in the right hand side of Equation (3.5).

$$\mathbb{E}[\|\mathrm{W}_{k+1}\,\mathrm{diag}\,\mathrm{b}_k\mathrm{W}_k\ldots\mathrm{diag}\,\mathrm{b}_1\mathrm{W}_1\mathrm{x}\|^2] = \mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to 2}\,\mathrm{diag}\,\mathrm{b}_1\mathrm{W}_1\mathrm{x}\|^2]$$
$$= \theta^2\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to 2}\mathrm{W}_1\mathrm{x}\|^2] + (\theta-\theta^2)\sum_{j=1}^{d_1}\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to 2}(:,j)\|^2]\|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_1(j,:)\|^2 \tag{3.6}$$

The second equality follows from Equation (3.4). Similarly, the first term on the right hand side of Equation (3.6) can be expressed as:

$$\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to 2}\mathrm{W}_1\mathrm{x}\|^2] = \mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to 3}\,\mathrm{diag}\,\mathrm{b}_2\mathrm{W}_{2\to 1}\mathrm{x}\|^2]$$
$$= \theta^2\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to 3}\mathrm{W}_{2\to 1}\mathrm{x}\|^2]$$
$$+ (\theta-\theta^2)\sum_{j=1}^{d_2}\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to 3}(:,j)\|^2]\|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{2\to 1}(j,:)\|^2$$

By recursive application of the above identity and plugging the result into Equation (3.6), we obtain:

$$\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to 1}\mathrm{x}\|^2] = \theta^{2k}\mathbb{E}[\|\mathrm{W}_{k+1\to 1}\mathrm{x}\|^2] \tag{3.7}$$
$$+ (1-\theta)\sum_{i=1}^{k}\sum_{j=1}^{d_i}\theta^{2i-1}\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to i+1}(:,j)\|^2]\|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{i\to 1}(j,:)\|^2 \tag{3.8}$$

Plugging back the above equality into Equation (3.5), we get

$$L_\theta(\{\mathrm{W}_i\}) = \|\mathrm{y}\|^2 - 2\mathbb{E}\langle W_{k+1\to 1}\mathrm{x},\mathrm{y}\rangle + \mathbb{E}[\|\mathrm{W}_{k+1\to 1}\mathrm{x}\|^2] \tag{3.9}$$
$$+ \frac{1-\theta}{\theta^{2k}}\sum_{i=1}^{k}\sum_{j=1}^{d_i}\theta^{2i-1}\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to i+1}(:,j)\|^2]\|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{i\to 1}(j,:)\|^2$$
$$= \mathbb{E}_{\mathrm{x}}[\|\mathrm{y}-\mathrm{W}_{k+1\to 1}\mathrm{x}\|^2] \tag{3.10}$$
$$+ (1-\theta)\sum_{i=1}^{k}\sum_{j=1}^{d_i}\theta^{2(i-k)-1}\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to i+1}(:,j)\|^2]\|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{i\to 1}(j,:)\|^2. \tag{3.11}$$

It remains to calculate the terms of the form $\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to i+1}(:,j)\|^2]$ in the right hand side of Equation (3.9). We introduce the variable $x\sim\mathcal{N}(0,1)$ so that we can use Equation (3.4) again:

$$\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to i+1}(:,j)\|^2] = \mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to i+2}\,\mathrm{diag}\,\mathrm{b}_{i+1}\mathrm{W}_{i+1}(:,j)x\|^2]$$
$$= \theta^2\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to i+2}\mathrm{W}_{i+1}(:,j)\|^2] + (\theta-\theta^2)\mathbb{E}\sum_{l=1}^{d_{i+1}}\|\bar{\mathrm{W}}_{k+1\to i+2}(:,l)\|^2\mathrm{W}_{i+1}(l,j)^2. \tag{3.12}$$

The first term on the right hand side of Equation (3.12) can be expanded as:

$$\begin{aligned}\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to i+2}\mathrm{W}_{i+1}(:,j)\|^2] &= \mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to i+3}\operatorname{diag}\mathrm{b}_{i+2}\mathrm{W}_{i+2\to i+1}(:,j)x\|^2] \\ &= \theta^2\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to i+3}\mathrm{W}_{i+2\to i+1}(:,j)\|^2] \\ &+ (\theta-\theta^2)\mathbb{E}\sum_{l=1}^{d_{i+2}}\|\bar{\mathrm{W}}_{k+1\to i+3}(:,l)\|^2\mathrm{W}_{i+2\to i+1}(l,j)^2\end{aligned}$$

By recursive application of the above equality and plugging the results into Equation (3.12), we get

$$\begin{aligned}\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to i+1}(:,j)\|^2] &= \theta^{2(k-i)}\|\mathrm{W}_{k+1\to i+1}(:,j)\|^2 \\ &+ (1-\theta)\sum_{m=1}^{k-i}\theta^{2m-1}\mathbb{E}\sum_{l=1}^{d_{i+m}}\|\bar{\mathrm{W}}_{k+1\to i+1+m}(:,l)\|^2\mathrm{W}_{i+m\to i+1}(l,j)^2\end{aligned}$$

Plugging back the above identity into Equation (3.9) we get

$$\begin{aligned}&R(\{\mathrm{W}_i\}) = (1-\theta)\sum_{i=1}^{k}\sum_{j=1}^{d_i}\theta^{2(i-k)-1}\|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{i\to 1}(j,:)\|^2\mathbb{E}[\|\bar{\mathrm{W}}_{k+1\to i+1}(:,j)\|^2] \\ &= \frac{1-\theta}{\theta}\sum_{i=1}^{k}\sum_{j=1}^{d_i}\|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{i\to 1}(j,:)\|^2\|\mathrm{W}_{k+1\to i+1}(:,j)\|^2 + \\ &(\frac{1-\theta}{\theta})^2\sum_{i=1}^{k}\sum_{j=1}^{d_i}\|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{i\to 1}(j,:)\|^2\sum_{m=1}^{k-i}\theta^{2(i+m-k)}\mathbb{E}\sum_{l=1}^{d_{i+m}}\mathrm{W}_{i+m\to i+1}(l,j)^2\|\bar{\mathrm{W}}_{k+1\to i+m+1}(:,l)\|^2 \\ &= \frac{1-\theta}{\theta}\sum_{i=1}^{k}\sum_{j=1}^{d_i}\|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{i\to 1}(j,:)\|^2\|\mathrm{W}_{k+1\to i+1}(:,j)\|^2 \\ &+ (\frac{1-\theta}{\theta})^2\sum_{i=1}^{k}\sum_{j=1}^{d_i}\|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{i\to 1}(j,:)\|^2\sum_{m=1}^{k-i}\sum_{l=1}^{d_{i+m}}\mathrm{W}_{i+m\to i+1}(l,j)^2\|\mathrm{W}_{k+1\to i+m+1}(:,l)\|^2 \\ &+ (\frac{1-\theta}{\theta})^3\sum_{i=1}^{k}\sum_{j=1}^{d_i}\|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{i\to 1}(j,:)\|^2\sum_{m=1}^{k-i}\sum_{l=1}^{d_{i+m}}\mathrm{W}_{i+m\to i+1}(l,j)^2\Bigg(\sum_{mm=1}^{k-i-m}\theta^{2(i+m+mm)} \\ &\sum_{ll=1}^{d_{i+m+mm}}\mathrm{W}_{i+m+mm\to i+m+1}(ll,l)^2\mathbb{E}\|\bar{\mathrm{W}}_{k+1\to i+1+m+mm}(:,ll)\|^2\Bigg) \\ &= \cdots \\ &= \sum_{l=1}^{k}\lambda^l\sum_{\substack{(j_l,\ldots,j_1)\\ \in\binom{[k]}{l}}}\sum_{\substack{(i_l,\ldots,i_1)\\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}}\|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{j_1\to 1}(i_1,:)\|^2 \\ &\qquad\prod_{p=1\cdots l-1}\mathrm{W}_{j_{p+1}\to j_p+1}(i_{p+1},i_p)^2\|\mathrm{W}_{k+1\to j_l+1}(:,i_l)\|^2,\end{aligned}$$

which completes the proof. □

Understanding the regularizer in Proposition 8. For simplicity, we assume here the case where the data distribution is whitened, i.e. $\mathrm{C} = \mathrm{I}$. This assumption is by no means restrictive, as we can always redefine $\mathrm{W}_1 \leftarrow \mathrm{W}_1\mathrm{C}^{\frac{1}{2}}$ to absorb the second moment the first layer. Moreover, it is a common practice to whiten the data as a preprocessing step. The l-th sub-regularizer, i.e. $R_l(\{\mathrm{W}_i\})$, partitions the network graph (see Figure 3-1) into $l+1$ subgraphs. This partitioning is done via the choice of l *pivot layers*, a set of l distinct hidden layers, indexed by $(j_1, \ldots, j_l) \in \binom{[k]}{l}$. The sub-regularizer enumerates over all such combinations of pivot layers, and *pivot nodes* within them indexed by $(i_1, \ldots, i_l) \in [d_{j_1}] \times \cdots \times [d_{j_l}]$. For a given set of pivot layers and pivot nodes, the corresponding summand in the sub-regularizer is a product of three types of terms: a "head" term α_{j_1,i_1}, "middle" terms $\beta_p,\ p \in [l-1]$, and "tail" terms γ_{j_l,i_l}. It is easy to see that each of the head, middle and tail terms computes a summation over product of the weights along certain walks on the (undirected) graph associated with the network (see Figure 3-1). For instance, a head term

$$\alpha_{j_1,i_1} = \sum_{i_0 \in [d_0]} \sum_{\substack{i'_1, i'_2, \ldots, i'_{j_1-1} \\ i''_{j_1-1}, \ldots, i''_2, i''_1}} \mathrm{W}_1(i'_1, i_0)\mathrm{W}_2(i'_2, i'_1)\cdots \\ \mathrm{W}_{j_1}(i_1, i'_{j_l-1})\mathrm{W}_{j_1}(i_1, i''_{j_l-1})\cdots \mathrm{W}_2(i''_2, i''_1)\mathrm{W}_1(i''_1, i_0),$$

is precisely the sum of the product of all weights along all walks from i_0 in the input layer to i_1 in layer j_1 and back to i_0, i.e., walks from $i_0 \xrightarrow{i'_1, i'_2, \ldots, i'_{j_1-1}} i_1 \xrightarrow{i''_{j_1-1}, \ldots, i''_2, i''_1} i_0$. Similarly, middle terms are the sum of the product of the weights along $i_p \xrightarrow{i'_1, i'_2, \ldots, i'_{j_1-1}} i_{p+1} \xrightarrow{i''_{j_1-1}, \ldots, i''_2, i''_1} i_p$.

A few remarks are in order.

Remark 6. *For $k = 1$, the explicit regularizer reduces to*

$$R(\mathrm{W}_2, \mathrm{W}_1) = \lambda \sum_{i=1}^{d_1} \|\mathrm{W}_1(:, i)\|^2 \|\mathrm{W}_2(i, :)\|^2,$$

which recovers the explicit regularizer for shallow networks in Proposition 8.

Remark 7. *All sub-regularizers, and consequently the explicit regularizer itself are* rescaling invariant. *That is, for any given implementation* $\{W_i\}$*, and any sequence of scalars* $\alpha_1, \ldots, \alpha_{k+1}$ *such that* $\prod_i \alpha_i = 1$*, it holds that* $R_l(\{W_i\}) = R_l(\{\alpha_i W_i\})$*. In particular,* R_k *equals*

$$R_k(\{W_i\}) = \sum_{i_k,\ldots,i_1} \|W_1(i_1,:)\|^2 W_2(i_2,i_1)^2 \\ W_3(i_3,i_2)^2 \cdots W_k(i_k,i_{k-1})^2 \|W_{k+1}(:,i_k)\|^2.$$

Note that $R_k(\{W_i\}) = \psi_2^2(W_{k+1}, \ldots, W_1)$*, the* ℓ_2-path regularizer*, which which has been recently studied in [NTS15] and [NSS15].*

3.2 The induced regularizer

In this section, we study the induced regularizer as given by the optimization problem in Equation (3.2). We show that the convex envelope of Θ factors into a product of two terms: a term that only depends on the network architecture and the dropout rate, and a term that only depends on the network map. These two factors are captured by the following definitions.

Definition 4 (effective regularization parameter)**.** *For given* $\{d_i\}$ *and* λ*, we refer to the following quantity as the effective regularization parameter:*

$$\nu_{\{d_i\}} := \sum_{l\in[k]} \sum_{(j_l,\ldots,j_1)\in\binom{[k]}{l}} \frac{\lambda^l}{\prod_{i\in[l]} d_{j_i}}.$$

We drop the subscript $\{d_i\}$ *whenever it is clear from the context.*

The effective regularization parameter naturally arises when we lowerbound the explicit regularizer (see Equation (3.14)). It is only a function of the network architecture and the dropout rate and does not depend on the weights – it increases with the dropout rate and the depth of the network, but decreases with the width.

Definition 5 (equalized network)**.** *A network implemented by* $\{W_i\}_{i=1}^{k+1}$ *is said to be* equalized *if* $\|W_{k+1}\cdots W_1 \mathrm{C}^{\frac{1}{2}}\|_*$ *is equally distributed among all the summands in Proposition 8, i.e. for any* $l \in [k]$, $(j_l,\dots,j_1) \in \binom{[k]}{l}$, *and* $(i_l,\dots,i_1) \in [d_{j_l}]\times\cdots\times[d_{j_1}]$ *it holds that*

$$|\alpha_{j_1,i_1}\beta_1\cdots\beta_{l-1}\gamma_{j_l,i_l}| = \frac{\|W_{k+1}\cdots W_1\mathrm{C}^{\frac{1}{2}}\|_*}{\Pi_l d_{j_l}}.$$

Before stating the main result of this section, we highlight three important properties of the dropout regularizer, which are essential in our analysis.

Lemma 6. *[Properties of* R *and* Θ*] The following statements hold true:*

1. *All sub-regularizers, and hence the explicit regularizer, are re-scaling invariant.*
2. *The infimum in Equation (3.2) is always attained.*
3. *If* $\mathrm{C} = I$*, then* $\Theta(M)$ *is a* spectral function*, i.e. if* M *and* M' *have the same singular values, then* $\Theta(M) = \Theta(M')$*.*

Proof of Lemma 6. First, it is easy to see that the explicit regularizer and the sub-regularizers are all *rescaling invariant*. For any sequence of scalars $\{\alpha_i\}$ such that such that $\prod_{i=1}^{k+1}\alpha_i = 1$, let $\bar{\mathrm{W}}_i := \alpha_i \mathrm{W}_i$. Then it holds that:

$$\begin{aligned}
&R_l(\{\bar{\mathrm{W}}_i\}) \\
&= \sum_{\substack{(j_l,\dots,j_1)\\ \in\binom{[k]}{l}}} \sum_{\substack{(i_l,\dots,i_1)\\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}} \|\prod_{q=1}^{j_1}\alpha_q \mathrm{W}_{j_1\to 1}(i_1,:)\|^2 \left(\prod_{p\in[l-1]}\prod_{q=j_p+1}^{j_{p+1}}\alpha_q^2 \mathrm{W}_{j_{p+1}\to j_p+1}(i_{p+1},i_p)^2\right.\\
&\left.\|\prod_{q=j_l+1}^{k+1}\alpha_q \mathrm{W}_{k+1\to j_l+1}(:,i_l)\|^2\right)\\
&= \prod_{q=1}^{k+1}\alpha_q^2 \sum_{\substack{(j_l,\dots,j_1)\\ \in\binom{[k]}{l}}} \sum_{\substack{(i_l,\dots,i_1)\\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}} \|\mathrm{W}_{j_1\to 1}(i_1,:)\|^2\\
&\qquad\qquad \prod_{p\in[l-1]} \mathrm{W}_{j_{p+1}\to j_p+1}(i_{p+1},i_p)^2 \|\mathrm{W}_{k+1\to j_l+1}(:,i_l)\|^2\\
&= R_l(\{\mathrm{W}_i\})
\end{aligned}$$

Therefore, without loss of generality, we can express the induced regularizer as follows:

$$\Theta(\mathrm{M}) := \inf_{\substack{\mathrm{W}_{k+1}\cdots\mathrm{W}_1=\mathrm{M} \\ \|\mathrm{W}_i\|_F \le \|\mathrm{M}\|_F}} R(\{\mathrm{W}_i\}) \tag{3.13}$$

Note that $R(\{\mathrm{W}_i\})$ is a continuous function and the feasible set $\mathcal{F} := \{(\mathrm{W}_i)_{i=1}^{k+1} : \mathrm{W}_{k+1}\cdots\mathrm{W}_1 = \mathrm{M},\ \|\mathrm{W}_i\|_F \le \|\mathrm{M}\|_F\}$ is compact. Hence, by Weierstrass extreme value theorem, the infimum is attained.

Now let $\mathrm{U} \in \mathbb{R}^{d_{k+1}\times d_{k+1}}$ and $\mathrm{V} \in \mathbb{R}^{d_0\times d_0}$ be a pair of rotation matrices, i.e. $\mathrm{U}^\top\mathrm{U} = \mathrm{U}\mathrm{U}^\top = \mathrm{I}$ and $\mathrm{V}^\top\mathrm{V} = \mathrm{V}\mathrm{V}^\top = \mathrm{I}$. When the data is isotropic, i.e. $\mathrm{C} = \mathrm{I}$, the following equalities hold

$$\begin{aligned}
R(\{\mathrm{W}_i\}) &= \sum_{l=1}^{k} \lambda^l \sum_{\substack{(j_l,\ldots,j_1) \\ \in\binom{[k]}{l}}} \sum_{\substack{(i_l,\ldots,i_1) \\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}} \|\mathrm{W}_{j_1\to 1}(i_1,:)\|^2 \left(\prod_{p=1\cdots l-1} \mathrm{W}_{j_{p+1}\to j_p+1}(i_{p+1},i_p)^2 \right. \\
&\quad \left. \|\mathrm{W}_{k+1\to j_l+1}(:,i_l)\|^2 \right) \\
&= \sum_{l=1}^{k} \lambda^l \sum_{\substack{(j_l,\ldots,j_1) \\ \in\binom{[k]}{l}}} \sum_{\substack{(i_l,\ldots,i_1) \\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}} \|\mathrm{W}_{j_1\to 1}(i_1,:)^\top\mathrm{V}\|^2 \left(\prod_{p=1\cdots l-1} \mathrm{W}_{j_{p+1}\to j_p+1}(i_{p+1},i_p)^2 \right. \\
&\quad \left. \|\mathrm{U}^\top\mathrm{W}_{k+1\to j_l+1}(:,i_l)\|^2 \right) \\
&= R(\mathrm{U}^\top\mathrm{W}_{k+1},\mathrm{W}_k,\cdots,\mathrm{W}_2,\mathrm{W}_1\mathrm{V})
\end{aligned}$$

That is, $R(\mathrm{U}^\top\mathrm{W}_{k+1},\mathrm{W}_k,\ldots,\mathrm{W}_2,\mathrm{W}_1\mathrm{V}) = R(\mathrm{W}_{k+1},\mathrm{W}_k,\ldots,\mathrm{W}_2,\mathrm{W}_1)$ for all rotation matrices U and V. In particular, let U,V be the left and right singular vectors of M, i.e. $\mathrm{M} = \mathrm{U}\Sigma\mathrm{V}^\top$. To prove that Θ is a spectral function, we need to show that $\Theta(\mathrm{M}) = \Theta(\Sigma)$. Let $\{\bar{\mathrm{W}}_i\},\{\tilde{\mathrm{W}}_i\}$ be such that $\Theta(\mathrm{M}) = R(\{\bar{\mathrm{W}}_i\}), \Theta(\Sigma) = R(\{\tilde{\mathrm{W}}_i\})$. Note that such weight matrices always exist since the infimum is always attained. Then

$$\begin{aligned}
\Theta(\Sigma) = \Theta(\mathrm{U}^\top\mathrm{M}\mathrm{V}) &\le R(\mathrm{U}^\top\bar{\mathrm{W}}_{k+1},\bar{\mathrm{W}}_k,\ldots,\bar{\mathrm{W}}_2,\bar{\mathrm{W}}_1\mathrm{V}) \\
&= R(\bar{\mathrm{W}}_{k+1},\bar{\mathrm{W}}_k,\ldots,\bar{\mathrm{W}}_2,\bar{\mathrm{W}}_1) = \Theta(\mathrm{M}).
\end{aligned}$$

Similarly, we have that

$$\Theta(\mathrm{M}) \leq R(\mathrm{U}^\top \tilde{\mathrm{W}}_{k+1}, \tilde{\mathrm{W}}_k, \ldots, \tilde{\mathrm{W}}_2, \bar{\mathrm{W}}_1 \mathrm{V}) = R(\tilde{\mathrm{W}}_{k+1}, \bar{\mathrm{W}}_k, \ldots, \tilde{\mathrm{W}}_2, \tilde{\mathrm{W}}_1) = \Theta(\Sigma),$$

which completes the proof. □

We are now ready to state the main result of this section. Recall that the convex envelope of a function is the largest convex under-estimator of that function. We show that irrespective of the architecture, the convex envelope of the induced regularizer is proportional to the squared nuclear norm of the network map times the principal root of the second moment.

Theorem 9 (Convex Envelope)**.** *For any architecture* $\{d_i\}$ *and any network map* $M \in \mathbb{R}^{d_{k+1} \times d_0}$ *implementable by that architecture, it holds that:*

$$\Theta^{**}(M) = \nu_{\{d_i\}} \|M \mathrm{C}^{\frac{1}{2}}\|_*^2$$

Furthermore, $\Theta(M) = \Theta^{**}(M)$ *if and only if the network is equalized.*

This result is particularly interesting because it connects dropout, an algorithmic heuristic to avoid overfitting, to nuclear norm regularization, which is a classical regularization method with strong theoretical foundations. We remark that a result similar to Theorem 9 was recently established for matrix factorization [CHL+18].

The key steps in the proof of Theorem 9 are as follows:

1. First, in Lemma 7, we show that for any set of weights $\{\mathrm{W}_i\}$, it holds that $R(\{\mathrm{W}_i\}) \geq \nu_{\{d_i\}} \|\mathrm{W}_{k+1 \to 1} \mathrm{C}^{\frac{1}{2}}\|_*^2$. In particular, this implies that $\Theta(\mathrm{M}) \geq \nu_{\{d_i\}} \|\mathrm{M}\mathrm{C}^{\frac{1}{2}}\|_*^2$ holds for any M.

2. Next, in Lemma 8, we show that $\Theta^{**}(\mathrm{M}) \leq \nu_{\{d_i\}} \|\mathrm{M}\mathrm{C}^{\frac{1}{2}}\|_*^2$ holds for all M.

3. The claim is implied by Lemmas 7 and 8, and the fact that $\|\cdot\|_*^2$ is a convex function.

Despite the complex form of the explicit regularizer given in Proposition 8, we can show that it is always lower bounded by *effective regularization parameter* times $\|\mathrm{MC}^{\frac{1}{2}}\|_*^2$. This result is given by Lemma 7.

Lemma 7. *Let $\{W_i\}$ be an arbitrary set of weights. The explicit regularizer $R(\{W_i\})$ satisfies*

$$R(\{W_i\}) \geq \nu_{\{d_i\}} \| W_{k+1} W_k \cdots W_1 \mathrm{C}^{\frac{1}{2}} \|_*^2,$$

and the equality holds if and only if the network is equalized.

Proof of Lemma 7. Recall that the explicit regularizer $R(\{\mathrm{W}_i\})$ is composed of k *sub-regularizers*

$$R(\{\mathrm{W}_i\}) = R_1(\{\mathrm{W}_i\}) + R_2(\{\mathrm{W}_i\}) + \cdots + R_k(\{\mathrm{W}_i\}).$$

The l-th sub-regularizer $R_l(\{\mathrm{W}_i\})$ can be written in the form of:

$$R_l(\{\mathrm{W}_i\}) = \lambda^l \sum_{(j_l,\ldots,j_1)\in\binom{[k]}{l}} R_{\{j_l,\ldots,j_1\}}(\{\mathrm{W}_i\})$$

where

$$R_{\{j_l,\ldots,j_1\}}(\{\mathrm{W}_i\}) := \|\mathrm{C}^{\frac{1}{2}} \mathrm{W}_{j_1\to 1}(i_1,:)\|^2 \prod_{p=1\cdots l-1} \mathrm{W}_{j_{p+1}\to j_p+1}(i_{p+1},i_p)^2 \|\mathrm{W}_{k+1\to j_l+1}(:,i_l)\|^2.$$

The following set of equalities hold true:

$$
\begin{aligned}
&R_{\{j_l,\ldots,j_1\}}(\{\mathrm{W}_i\}) \\
&= \sum_{i_l,\cdots,i_1} \|\mathrm{W}_{k+1\to j_l+1}(:,i_l)\|^2 \mathrm{W}_{j_l\to j_{l-1}+1}(i_l,i_{l-1})^2 \cdots \mathrm{W}_{j_2\to j_1+1}(i_2,i_1)^2 \|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{j_1\to 1}(i_1,:)\|^2 \\
&\geq \frac{\left(\sum_{i_l,\ldots,i_1} \|\mathrm{W}_{k+1\to j_l+1}(:,i_l)\| |\mathrm{W}_{j_l\to j_{l-1}+1}(i_l,i_{l-1})| \cdots |\mathrm{W}_{j_2\to j_1+1}(i_2,i_1)| \|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{j_1\to 1}(i_1,:)\|\right)^2}{\prod_{i\in[l]} d_{j_i}} \\
&= \frac{\left(\sum_{i_l,\ldots,i_1} \|\mathrm{W}_{k+1\to j_l+1}(:,i_l)\mathrm{W}_{j_l\to j_{l-1}+1}(i_l,i_{l-1}) \cdots \mathrm{W}_{j_2\to j_1+1}(i_2,i_1)\mathrm{W}_{j_1\to 1}(i_1,:)^\top \mathrm{C}^{\frac{1}{2}}\|_*\right)^2}{\prod_{i\in[l]} d_{j_i}} \\
&\geq \frac{\|\sum_{i_l,\ldots,i_1} \mathrm{W}_{k+1\to j_l+1}(:,i_l)\mathrm{W}_{j_l\to j_{l-1}+1}(i_l,i_{l-1}) \cdots \mathrm{W}_{j_2\to j_1+1}(i_2,i_1)\mathrm{W}_{j_1\to 1}(i_1,:)^\top \mathrm{C}^{\frac{1}{2}}\|_*^2}{\prod_{i\in[l]} d_{j_i}} \\
&= \frac{\|\sum_{i_l,i_1} \mathrm{W}_{k+1\to j_l+1}(:,i_l)\left(\sum_{i_{l-1},\ldots,i_2} \mathrm{W}_{j_l\to j_{l-1}+1}(i_l,i_{l-1}) \cdots \mathrm{W}_{j_2\to j_1+1}(i_2,i_1)\right)\mathrm{W}_{j_1\to 1}(i_1,:)^\top \mathrm{C}^{\frac{1}{2}}\|_*^2}{\prod_{i\in[l]} d_{j_i}} \\
&= \frac{\|\sum_{i_l,i_1} \mathrm{W}_{k+1\to j_l}(:,i_l)\mathrm{W}_{j_l-1\to j_1}(i_l,i_1)\mathrm{W}_{j_1\to 1}(i_1,:)^\top \mathrm{C}^{\frac{1}{2}}\|_*^2}{\prod_{i\in[l]} d_{j_i}} \\
&= \frac{\|\mathrm{W}_{k+1}\cdots\mathrm{W}_1\mathrm{C}^{\frac{1}{2}}\|_*^2}{\prod_{i\in[l]} d_{j_i}}
\end{aligned}
$$

where the first inequality follows due to the Cauchy-Schwartz inequality, and the second inequality follows from the triangle inequality for the matrix norms. The inequality holds with equality if and only if all the summands inside the summation are equal to each other, and sum up to $\frac{\|\mathrm{W}_{k+1\to 1}\mathrm{C}^{\frac{1}{2}}\|_*}{\prod_{i\in[l]} d_{j_i}}$, i.e. when

$$
\begin{aligned}
&\|\mathrm{W}_{k+1\to j_l+1}(:,i_l)\| |\mathrm{W}_{j_l\to j_{l-1}+1}(i_l,i_{l-1})| \cdots |\mathrm{W}_{j_2\to j_1+1}(i_2,i_1)| \|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{j_1\to 1}(i_1,:)\| \\
=&\frac{1}{\prod_{i\in[l]} d_{j_i}} \|\mathrm{W}_{k+1\to 1}\mathrm{C}^{\frac{1}{2}}\|_*
\end{aligned}
$$

for all $(i_l,\ldots,i_1) \in [d_{j_l}]\times\cdots\times[d_{j_1}]$. This lowerbound holds for all $l\in[k]$, and for all $(j_l,\ldots,j_1)\in\binom{[k]}{l}$. Thus, we get the following lowerbound on the regularizer:

$$
R(\{\mathrm{W}_i\}) \geq \sum_{l\in[k]} \lambda^l \sum_{(j_l,\ldots,j_1)\in\binom{[k]}{l}} \frac{1}{\prod_{i\in[l]} d_{j_i}} \|\mathrm{W}_{k+1\to 1}\mathrm{C}^{\frac{1}{2}}\|_*^2 = \nu_{\{\mathrm{d}_i\}} \|\mathrm{W}_{k+1\to 1}\mathrm{C}^{\frac{1}{2}}\|_*^2
$$

which completes the proof. □

Lemma 7 is central to our analysis for two reasons. First, it gives a sufficient and necessary condition for the induced regularizer to equal the square of the nuclear

norm of the network map. This also motivates the concept of equalized networks in Definition 5. We note that for the special case of single hidden layer linear networks, i.e., for k=1, this lower bound can always be achieved [MAV18]; it remains to be seen whether the lower bound can be achieved for deeper networks. Second, summing over $\{j_l, \dots, j_1\} \in \binom{[k]}{l}$, we conclude that

$$R_l(\{\mathrm{W}_i\}) \geq \sum_{j_l,\dots,j_1} \frac{\|\mathrm{W}_{k+1\to 1}\|_*^2}{\prod_l d_{j_l}} =: LB_l(\{W_i\}). \tag{3.14}$$

The right hand side above is the lowerbound for l-th subregularizer, denoted by LB_l. Summing over $l \in [k]$, we get the following lowerbound on the explicit regularizer

$$R(\{\mathrm{W}_i\}) \;\geq\; \|\mathrm{W}_{k+1\to 1}\|_*^2 \underbrace{\sum_l \sum_{j_l,\dots,j_1} \frac{\lambda^l}{\prod_l d_{j_l}}}_{\nu_{\{d_i\}}} \tag{3.15}$$

which motivates the notion of *effective regularization parameter* in Definition 4. As an immediate corollary of Lemma 7, it holds that for any matrix M we have that $\Theta(\mathrm{M}) \geq \nu_{\{d_i\}} \|\mathrm{M}\|_*^2$. We now focus on the biconjugate of the induced regularizer, and show that it is upper bounded by the same function, i.e. the effective regularization parameter times the square of the nuclear norm of the network map.

Not only $\nu_{\{d_i\}} \|\mathrm{M}\mathrm{C}^{\frac{1}{2}}\|_*^2$ is a lowerbound for the induced regularizer, but also is an upperbound for its convex envelope. We prove this result in Lemma 8.

Lemma 8. *For any architecture $\{d_i\}$ and any network map M, it holds that $\Theta^{**}(M) \leq \nu_{\{d_i\}} \|M\mathrm{C}^{\frac{1}{2}}\|_*^2$.*

Proof of Lemma 8. The induced regularizer is non-negative. Hence, the domain of the Fenchel dual of the induced regularizer is the whole $\mathbb{R}^{d_{k+1}\times d_0}$. The Fenchel dual of the induced regularizer $\Theta(\cdot)$ is given by:

$$\begin{aligned}
\Theta^*(\mathrm{M}) &= \max_{\mathrm{P}} \; \langle \mathrm{M}, \mathrm{P} \rangle - \Theta(\mathrm{P}) \\
&= \max_{\mathrm{P}} \; \langle \mathrm{M}, \mathrm{P} \rangle - \min_{\substack{\{\mathrm{W}_i\} \\ \mathrm{W}_{k+1\to 1}=\mathrm{P}}} R(\{\mathrm{W}_i\}) \\
&= \max_{\{\mathrm{W}_i\}} \; \langle \mathrm{M}, \mathrm{W}_{k+1\to 1} \rangle - R(\{\mathrm{W}_i\}).
\end{aligned} \tag{3.16}$$

Define $\Phi(\{\mathrm{W}_i\}) := \langle \mathrm{M}, \mathrm{W}_{k+1\to 1}\rangle - R(\{\mathrm{W}_i\})$ as the objective in the right hand side of Equation (3.16). Due to the complicated products of the norms of the weights in the regularizer, maximizing Φ with respect to $\{\mathrm{W}_i\}$ is a daunting task. Here, we find a lower bound on this maximum value. Let $\mathrm{W}_{k+1}^{\alpha} := \alpha \mathrm{u}_1 1_{d_k}^\top$ and $\mathrm{W}_1^{\alpha} := 1_{d_1} \mathrm{v}_1^\top \mathrm{C}^{-\frac{1}{2}}$, where $(\mathrm{u}_1, \mathrm{v}_1)$ is the top singular vectors of $\mathrm{MC}^{-\frac{1}{2}}$, and 1_d is the d-dimensional vector of all 1s. Furthermore, let $\mathrm{W}_i^{\alpha} := 1_{d_i} 1_{d_{i-1}}^\top$, for all $i \in \{2, \dots, k\}$. Note that

$$\Theta^*(\mathrm{M}) = \max_{\{\mathrm{W}_i\}} \Phi(\{\mathrm{W}_i\}) \geq \max_{\alpha} \Phi(\{\mathrm{W}_i^{\alpha}\}).$$

We now simplify $\Phi(\{\mathrm{W}_i^{\alpha}\})$. First, the following equalities hold for the $\langle \mathrm{M}, \mathrm{W}_{k+1\to 1}^{\alpha}\rangle$:

$$\begin{aligned}
\langle \mathrm{M}, \mathrm{W}_{k+1\to 1}^{\alpha}\rangle &= \sum_{(i_{k+1},\dots,i_1)\in[d_{k+1}]\times\cdots\times[d_1]} \langle \mathrm{M}, \mathrm{W}_{k+1}^{\alpha}(:, i_k) \prod_{j=\{k-1,\dots,1\}} \mathrm{W}_{j+1}^{\alpha}(i_{j+1}, i_j) \mathrm{W}_1^{\alpha}(i_1, :)^\top \rangle \\
&= \sum_{(i_{k+1},\dots,i_1)\in[d_{k+1}]\times\cdots\times[d_1]} \mathrm{W}_{k+1}^{\alpha}(:, i_k)^\top \mathrm{M} \mathrm{W}_1^{\alpha}(i_1, :) \\
&= \sum_{(i_{k+1},\dots,i_1)\in[d_{k+1}]\times\cdots\times[d_1]} \alpha \mathrm{u}_1^\top \mathrm{MC}^{-\frac{1}{2}} \mathrm{v}_1 \\
&= \sum_{(i_{k+1},\dots,i_1)\in[d_{k+1}]\times\cdots\times[d_1]} \alpha \|\mathrm{MC}^{-\frac{1}{2}}\|_2 \\
&= \alpha \|\mathrm{MC}^{-\frac{1}{2}}\|_2 \prod_{j\in[k]} d_j =: \alpha \|\mathrm{MC}^{-\frac{1}{2}}\|_2 D.
\end{aligned}$$

The following terms show up in the expansion of the regularizer:

$$\begin{aligned}
\mathrm{W}_{j_1\to 1}^{\alpha}(i_1, :)^\top &= \mathrm{W}_{j_1}^{\alpha}(i_1, :) \mathrm{W}_{j_1-1}^{\alpha} \cdots \mathrm{W}_2^{\alpha} \mathrm{W}_1^{\alpha} = 1_{d_{j_1-1}}^\top 1_{d_{j_1-1}} 1_{d_{j_1-2}}^\top \cdots 1_{d_2} 1_{d_1}^\top 1_{d_1} \mathrm{v}_1^\top \\
&= \prod_{i\in[j_1-1]} d_i \mathrm{v}_1^\top \mathrm{C}^{-\frac{1}{2}} \\
\mathrm{W}_{j_{p+1}\to j_p+1}^{\alpha}(i_{p+1}, i_p) &= \mathrm{W}_{j_{p+1}}^{\alpha}(i_{p+1}, :) \mathrm{W}_{j_{p+1}-1}^{\alpha} \cdots \mathrm{W}_{j_p+2}^{\alpha} \mathrm{W}_{j_p+1}^{\alpha}(:, i_p) \\
&= 1_{d_{j_{p+1}-1}}^\top 1_{d_{j_{p+1}-1}} 1_{d_{j_{p+1}-2}}^\top \cdots 1_{d_{j_p+2}} 1_{d_{j_p+1}}^\top 1_{d_{j_p+1}} = \prod_{i\in\{j_p+1,\cdots,j_{p+1}-1\}} d_i \\
\mathrm{W}_{k+1\to j_l+1}^{\alpha}(:, i_l) &= \alpha \mathrm{W}_{k+1}^{\alpha} \mathrm{W}_k^{\alpha} \cdots \mathrm{W}_{j_l+2}^{\alpha} \mathrm{W}_{j_l+1}^{\alpha}(:, i_l) = \alpha \mathrm{u}_1 1_{d_k}^\top 1_{d_k} 1_{d_{k-1}}^\top \cdots 1_{d_{j_l+2}} 1_{d_{j_l+1}}^\top 1_{d_{j_l+1}} \\
&= \alpha \prod_{i\in\{j_l+1,\cdots,k\}} d_i \mathrm{u}_1
\end{aligned}$$

With the above equalities, the explicit regularizer reduces to:

$$
\begin{aligned}
&R(\{\mathrm{W}_i^\alpha\}) \\
&= \sum_{l=1}^{k} \lambda^l \sum_{\substack{(j_l,\dots,j_1)\\ \in\binom{[k]}{l}}} \sum_{\substack{(i_l,\dots,i_1)\\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}} \|\mathrm{C}^{\frac{1}{2}}\mathrm{W}^\alpha_{j_1\to 1}(i_1,:)\|^2 \prod_{p=1\cdots l-1} \mathrm{W}^\alpha_{j_{p+1}\to j_p+1}(i_{p+1},i_p)^2 \|\mathrm{W}^\alpha_{k+1\to j_l+1}(:,i_l)\|^2 \\
&= \sum_{l=1}^{k} \lambda^l \sum_{\substack{(j_l,\dots,j_1)\\ \in\binom{[k]}{l}}} \sum_{\substack{(i_l,\dots,i_1)\\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}} \|\mathrm{C}^{\frac{1}{2}}\mathrm{C}^{-\frac{1}{2}}\mathrm{v}_1 \prod_{i\in[j_1-1]} d_i\|^2 \prod_{\substack{p=1\cdots l-1\\ i\in\{j_p+1,\cdots,j_{p+1}-1\}}} d_i^2 \|\alpha \mathrm{u}_1 \prod_{i\in\{j_l+1,\cdots,k\}} d_i\|^2 \\
&= \sum_{l=1}^{k} \lambda^l \sum_{\substack{(j_l,\dots,j_1)\\ \in\binom{[k]}{l}}} \sum_{\substack{(i_l,\dots,i_1)\\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}} \prod_{i\in[j_1-1]} d_i^2 \prod_{p=1\cdots l-1} \prod_{i\in\{j_p+1,\cdots,j_{p+1}-1\}} d_i^2\alpha^2 \prod_{i\in\{j_l+1,\cdots,k\}} d_i^2 \\
&= \alpha^2 \sum_{l=1}^{k} \lambda^l \sum_{(j_l,\dots,j_1)\in\binom{[k]}{l}} \sum_{(i_l,\dots,i_1)\in[d_{j_l}]\times\cdots\times[d_{j_1}]} \frac{\prod_{i\in[k]} d_i^2}{\prod_{i\in[l]} d_{j_i}^2} =: \alpha^2\rho
\end{aligned}
$$

Plugging back the above equalities into the definition of Φ, we arrive at $\Phi(\{\mathrm{W}_i^\alpha\}) = \alpha\|\mathrm{MC}^{-\frac{1}{2}}\|_2 D - \alpha^2\rho$. The maximum of $\Phi(\{\mathrm{W}_i^\alpha\})$ with respect to α is achieved when $\alpha^* = \frac{\|\mathrm{MC}^{-\frac{1}{2}}\|_2 D}{2\rho}$, in which case we have

$$
\Theta^*(\mathrm{M}) \geq \Phi(\{\mathrm{W}_i^{\alpha^*}\}) = \frac{D^2}{4\rho}\|\mathrm{MC}^{-\frac{1}{2}}\|_2^2 =: \Psi(\mathrm{M}).
$$

Since Fenchel dual is order reversing, we get

$$
\begin{aligned}
\Theta^{**}(\mathrm{M}) &\leq \Psi^*(\mathrm{M}) \\
&= \frac{\rho}{D^2}\|\mathrm{MC}^{\frac{1}{2}}\|_*^2 \\
&= \frac{\sum_{l=1}^{k} \lambda^l \sum_{(j_l,\dots,j_1)\in\binom{[k]}{l}} \sum_{(i_l,\dots,i_1)\in[d_{j_l}]\times\cdots\times[d_{j_1}]} \frac{\prod_{i\in[k]} d_i^2}{\prod_{i\in[l]} d_{j_i}^2}}{\prod_{j\in[k]} d_j^2} \|\mathrm{MC}^{\frac{1}{2}}\|_*^2 \\
&= \sum_{l=1}^{k} \lambda^l \sum_{(j_l,\dots,j_1)\in\binom{[k]}{l}} \sum_{(i_l,\dots,i_1)\in[d_{j_l}]\times\cdots\times[d_{j_1}]} \frac{1}{\prod_{i\in[l]} d_{j_i}^2} \|\mathrm{MC}^{\frac{1}{2}}\|_*^2 \\
&= \sum_{l=1}^{k} \lambda^l \sum_{(j_l,\dots,j_1)\in\binom{[k]}{l}} \frac{1}{\prod_{i\in[l]} d_{j_i}} \|\mathrm{MC}^{\frac{1}{2}}\|_*^2 \\
&= \nu_{\{d_i\}}\|\mathrm{MC}^{\frac{1}{2}}\|_*^2
\end{aligned}
$$

where the first equality follows from the fact that if $f(\mathrm{M}) = \beta\|\mathrm{MA}\|^2$ and $\mathrm{A} \succ 0$ then $f^*(\mathrm{M}) = \frac{1}{4\beta}\|\mathrm{MA}^{-1}\|_*^2$. This result is standard in the literature, but we prove it here

for completeness. Note that

$$\begin{aligned}\langle \mathrm{Y}, \mathrm{M}\rangle - \beta\|\mathrm{YA}\|^2 &= \langle \mathrm{YA}, \mathrm{MA}^{-1}\rangle - \beta\|\mathrm{YA}\|^2 \\ &\leq \|\mathrm{YA}\|\|\mathrm{MA}^{-1}\|_* - \beta\|\mathrm{YA}\|^2\end{aligned}$$

where the inequality is due to Holder's identity. The right hand side above is a quadratic in $\|\mathrm{YA}\|$ and is maximized when $\|\mathrm{YA}\| = \frac{1}{2\beta}\|\mathrm{MA}^{-1}\|_*$, in which case we have

$$\begin{aligned}f^*(\mathrm{M}) &= \sup_{\mathrm{Y}} \langle \mathrm{Y}, \mathrm{M}\rangle - \beta\|\mathrm{YA}\|^2 \\ &= \frac{1}{2\beta}\|\mathrm{MA}^{-1}\|_*\|\mathrm{MA}^{-1}\|_* - \beta(\frac{1}{2\beta}\|\mathrm{MA}^{-1}\|_*)^2 \\ &= \frac{1}{4\beta}\|\mathrm{MA}^{-1}\|_*^2.\end{aligned}$$

□

We now provide a proof of Theorem 9.

Proof of Theorem 9. By Lemma 7, for any architecture, any dropout rate, and any set of weights $\{\mathrm{W}_i\}$ that implements a network map $\mathrm{W}_{k+1\to 1}$, the explicit regularizer is lower bounded by the effective regularization parameter times the product of the squared nuclear norm of the network map and the principal squared root of the second moment of x, i.e. $R(\{\mathrm{W}_i\}) \geq \nu_{\{d_i\}}\|\mathrm{W}_{k+1\to 1}\mathrm{C}^{\frac{1}{2}}\|_*^2$. Consequently, the induced regularizer can also be lowerbounded as $\Theta(\mathrm{M}) \geq \nu_{\{d_i\}}\|\mathrm{MC}^{\frac{1}{2}}\|_*^2$. On the other hand, Lemma 8 establishes that $\Theta^{**}(\mathrm{M}) \leq \nu_{\{d_i\}}\|\mathrm{MC}^{\frac{1}{2}}\|_*^2$ holds for any network map M. Putting these two inequalities together, we arrive at

$$\Theta^{**}(\mathrm{M}) \leq \nu_{\{d_i\}}\|\mathrm{MC}^{\frac{1}{2}}\|_*^2 \leq \Theta(\mathrm{M}).$$

Since $\Theta^{**}(\mathrm{M})$ is the largest convex underestimator of $\Theta(\mathrm{M})$, and the squared nuclear norm is a convex function, we conclude that $\Theta^{**}(\mathrm{M}) = \nu_{\{d_i\}}\|\mathrm{M}\|_*^2$. □

3.3 Global optimality

Theorem 9 provides a sufficient and necessary condition under which the induced regularizer equals its convex envelope. If any network map can be implemented by an equalized network, then $\Theta(\mathrm{M}) = \Theta^{**}(\mathrm{M}) = \nu_{\{d_i\}}\|\mathrm{M}\mathrm{C}^{\frac{1}{2}}\|_*^2$, and the learning problem in Equation (3.3) is a convex program. In particular, for the case of linear networks with single hidden layer, [MAV18] show that any network map can be implemented by an equalized network, which enables them to characterize the set of global optima under the additional generative assumption $\mathrm{y} = \mathrm{M}\mathrm{x}$. However, it is not clear if the same holds for general deep linear networks since the regularizer here is more complex. Nonetheless, the following result provides a sufficient condition under which global optima of $L_\theta(\{\mathrm{W}_i\})$ are completely characterized.

Theorem 10. *Let* $\mathrm{C}_{\mathrm{yx}} := \mathbb{E}[\mathrm{y}\mathrm{x}^\top]$ *and* $\mathrm{C} := \mathbb{E}[\mathrm{x}\mathrm{x}^\top]$*, and denote* $\bar{M} := \mathrm{C}_{\mathrm{yx}}\mathrm{C}^{-\frac{1}{2}}$*. If* $\sigma_1(\bar{M}) - \sigma_2(\bar{M}) \geq \frac{1}{\nu}\sigma_2(\bar{M})$*, then* M^**, the global optimum of Problem 3.3, is given by*

$$W^*_{k+1\to 1} = \mathcal{S}_{\frac{\nu\sigma_1(\bar{M})}{1+\nu}}(\bar{M})\mathrm{C}^{-\frac{1}{2}},$$

where $\mathcal{S}_\alpha(\bar{M})$ *shrinks the spectrum of matrix* $\bar{M}$ *by* α *and thresholds it at zero. Furthermore, it is possible to implement* M^* *by an equalized network* $\{W_i^*\}$ *which is a global optimum of* $L_\theta(\{W_i\})$*.*

In light of Theorem 9, if the optimal network map $\mathrm{W}^*_{k+1\to 1}$, i.e. the optimum of the problem in Equation 3.3 can be implemented by an equalized network, then $\Theta(\mathrm{W}^*_{k+1\to 1}) = \Theta^{**}(\mathrm{W}^*_{k+1\to 1}) = \nu_{\{d_i\}}\|\mathrm{W}^*_{k+1\to 1}\mathrm{C}^{\frac{1}{2}}\|_*^2$. Thus, the learning problem essentially boils down to the following convex program:

$$\min_{\mathrm{W}} \mathbb{E}_{\mathrm{x},\mathrm{y}}[\|\mathrm{y} - \mathrm{W}\mathrm{x}\|^2] + \nu_{\{d_i\}}\|\mathrm{W}\mathrm{C}^{\frac{1}{2}}\|_*^2. \tag{3.17}$$

Following the previous work of [CHL+18, MAV18], we show that the solution to problem (3.17) can be given as $\mathrm{W}^* = \mathcal{S}_{\alpha_\rho}(\mathrm{C}_{\mathrm{yx}}\mathrm{C}^{-\frac{1}{2}})\mathrm{C}^{-\frac{1}{2}}$, where $\alpha_\rho := \frac{\nu\sum_{i=1}^{\rho}\sigma_i(\mathrm{C}_{\mathrm{yx}}\mathrm{C}^{-\frac{1}{2}})}{1+\rho\nu}$,

ρ is the rank of W^*, and $\mathcal{S}_{\alpha_\rho}(\mathrm{M})$ shrinks the spectrum of the input matrix M by α_ρ and thresholds them at zero. However, as mentioned above, it is not clear if any network map can be implemented by an equalized network. Nonetheless, it is easy to see that the equalization property is satisfied for rank-1 network maps.

Proposition 1. *Let $\{d_i\}_{i=0}^{k+1}$ be an architecture and $M \in \mathbb{R}^{d_{k+1} \times d_0}$ be a rank-1 network map. Then, there exists a set of weights $\{W_i\}_{i=1}^{k+1}$ that implements M, and is equalized.*

Proof of Proposition 1. When the network map has rank equal to one, it can be expressed as $\mathrm{uv}^\top$, where $\mathrm{u} \in \mathbb{R}^{d_{k+1}}$ and $\mathrm{v} \in \mathbb{R}^{d_0}$. We show that for any architecture $\{d_i\}$ and any network mapping $\mathrm{uv}^\top \in \mathbb{R}^{d_{k+1} \times d_0}$, it is always possible to represent $\mathrm{uv}^\top = \mathrm{W}_{k+1} \cdots \mathrm{W}_1$ such that the resulting network is equalized. One such factorization is when $\mathrm{W}_1 = \frac{1_{d_1} \mathrm{v}^\top}{\sqrt{d_1}}$, $\mathrm{W}_{k+1} = \frac{\mathrm{u} 1_{d_k}^\top}{\sqrt{d_k}}$, and $\mathrm{W}_i = \frac{1_{d_i} 1_{d_{i-1}}^\top}{\sqrt{d_i d_{i-1}}}$ for $i \in \{2, \ldots, k\}$. For these weight parameters, we have that

$$
\begin{aligned}
\mathrm{W}_{j_1 \to 1}(i_1, :)^\top &= \mathrm{W}_{j_1}(i_1, :)^\top \mathrm{W}_{j_1 - 1} \cdots \mathrm{W}_2 \mathrm{W}_1 \\
&= \frac{1_{d_{j_1 - 1}}^\top}{\sqrt{d_{j_1} d_{j_1 - 1}}} \frac{1_{d_{j_1 - 1}} 1_{d_{j_1 - 2}}^\top}{\sqrt{d_{j_1 - 1} d_{j_1 - 2}}} \cdots \frac{1_{d_2} 1_{d_1}^\top}{\sqrt{d_2 d_1}} \frac{1_{d_1} \mathrm{v}^\top}{\sqrt{d_1}} \\
&= \frac{\mathrm{v}^\top}{\sqrt{d_{j_1}}} \\
\mathrm{W}_{j_{p+1} \to j_p + 1}(i_{p+1}, i_p) &= \mathrm{W}_{j_{p+1}}(i_{p+1}, :)^\top \mathrm{W}_{j_{p+1} - 1} \cdots \mathrm{W}_{j_p + 2} \mathrm{W}_{j_p + 1}(:, i_p) \\
&= \frac{1_{d_{j_{p+1} - 1}}^\top}{\sqrt{d_{j_{p+1}} d_{j_{p+1} - 1}}} \frac{1_{d_{j_{p+1} - 1}} 1_{d_{j_{p+1} - 2}}^\top}{\sqrt{d_{j_{p+1} - 1} d_{j_{p+1} - 2}}} \cdots \frac{1_{d_{j_p + 2}} 1_{d_{j_{p+1}}}^\top}{\sqrt{d_{j_p + 2} d_{j_{p+1}}}} \frac{1_{d_{j_{p+1}}}}{\sqrt{d_{j_{p+1}} d_{j_p}}} \\
&= \frac{1}{\sqrt{d_{j_{p+1}} d_{j_p}}} \\
\mathrm{W}_{k+1 \to j_l + 1}(:, i_l) &= \mathrm{W}_{k+1} \mathrm{W}_k \cdots \mathrm{W}_{j_l + 2} \mathrm{W}_{j_l + 1}(:, i_l) \\
&= \frac{\mathrm{u} 1_{d_k}^\top}{\sqrt{d_k}} \frac{1_{d_k} 1_{d_{k-1}}^\top}{\sqrt{d_k d_{k-1}}} \cdots \frac{1_{d_{j_l + 2}} 1_{d_{j_l + 1}}^\top}{\sqrt{d_{j_l + 2} d_{j_l + 1}}} \frac{1_{d_{j_l + 1}}}{\sqrt{d_{j_l + 1} d_{j_l}}} \\
&= \frac{\mathrm{u}}{\sqrt{d_{j_l}}}
\end{aligned}
$$

With the above equalities, the regularizer reduces to:

$$
\begin{aligned}
R(\{\mathrm{W}_i\}) &= \sum_{l=1}^{k} \lambda^l \sum_{\substack{(j_l,\ldots,j_1)\\ \in\binom{[k]}{l}}} \sum_{\substack{(i_l,\ldots,i_1)\\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}} \|\mathrm{C}^{\frac{1}{2}}\mathrm{W}_{j_1\to 1}(i_1,:)\|^2 \left(\prod_{p=1\cdots l-1} \mathrm{W}_{j_{p+1}\to j_p+1}(i_{p+1},i_p)^2 \right. \\
&\quad \left. \|\mathrm{W}_{k+1\to j_l+1}(:,i_l)\|^2 \right) \\
&= \sum_{l=1}^{k} \lambda^l \sum_{\substack{(j_l,\ldots,j_1)\\ \in\binom{[k]}{l}}} \sum_{\substack{(i_l,\ldots,i_1)\\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}} \|\mathrm{C}^{\frac{1}{2}}\frac{\mathrm{v}}{\sqrt{d_{j_1}}}\|^2 \prod_{p=1\cdots l-1} \frac{1}{d_{j_{p+1}}d_{j_p}} \|\frac{\mathrm{u}}{\sqrt{d_{j_l}}}\|^2 \\
&= \sum_{l=1}^{k} \lambda^l \sum_{\substack{(j_l,\ldots,j_1)\\ \in\binom{[k]}{l}}} \sum_{\substack{(i_l,\ldots,i_1)\\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}} \frac{\|\mathrm{C}^{\frac{1}{2}}\mathrm{v}\|^2\|\mathrm{u}\|^2}{\prod_{p\in[l]} d_{j_p}^2} \\
&= \sum_{l=1}^{k} \sum_{\substack{(j_l,\ldots,j_1)\\ \in\binom{[k]}{l}}} \frac{\lambda^l}{\prod_{p\in[l]} d_{j_p}} \|\mathrm{u}\mathrm{v}^\top\mathrm{C}^{\frac{1}{2}}\|_*^2 = \nu_{\{d_i\}}\|\mathrm{u}\mathrm{v}^\top\mathrm{C}^{\frac{1}{2}}\|_*^2
\end{aligned}
$$

where we used the fact that $\|\mathrm{u}\|\|\mathrm{C}^{\frac{1}{2}}\mathrm{v}\| = \|\mathrm{u}\mathrm{v}^\top\mathrm{C}^{\frac{1}{2}}\|_*$. Moreover, note that the network specified by the above weight matrices is equalized, since

$$
|\alpha_{j_1,i_1}| \prod_{p=1\cdots l-1} |\beta_p||\gamma_{j_l,i_l}| = \sqrt{\frac{\|\mathrm{u}\mathrm{v}^\top\mathrm{C}^{\frac{1}{2}}\|_*^2}{\prod_{p\in[l]} d_{j_p}^2}} = \frac{1}{\prod_{p\in[l]} d_{j_p}} \|\mathrm{u}\mathrm{v}^\top\mathrm{C}^{\frac{1}{2}}\|_*.
$$

□

For example, for deep networks with single output neuron, the weights $\mathrm{W}_1 = \frac{1_{d_1}\mathrm{w}^\top}{\sqrt{d_1}}$ and $\mathrm{W}_i = \frac{1_{d_i}1_{d_{i-1}}^\top}{\sqrt{d_i d_{i-1}}}$ for $i \neq 1$ implements the network map $\mathrm{w}^\top$, and are equalized.

Denote $\bar{\mathrm{M}} := \mathrm{C}_{\mathrm{yx}}\mathrm{C}^{-\frac{1}{2}}$. Equipped with Proposition 1, the key here is to ensure that $\mathcal{S}_\alpha(\bar{\mathrm{M}})$ has rank equal to one. In this case, W^* will also have rank at most one and can be implemented by a network that is equalized. To that end, it suffices to have $\alpha \geq \sigma_2(\bar{\mathrm{M}})$, which implies

$$
\frac{\nu\sigma_1(\bar{\mathrm{M}})}{1+\nu} \geq \sigma_2(\bar{\mathrm{M}}) \implies \sigma_1(\bar{\mathrm{M}}) - \sigma_2(\bar{\mathrm{M}}) \geq \frac{\sigma_2(\bar{\mathrm{M}})}{\nu}
$$

which gives the sufficient condition in Theorem 10. This discussion leads to the following Lemma.

Lemma 9. *Consider the following optimization problem where the induced regularizer in Problem 3.3 is replaced with its convex envelope:*

$$\min_{W \in \mathbb{R}^{d_{k+1} \times d_0}} \mathbb{E}[\|\mathrm{y} - W\mathrm{x}\|^2] + \Theta^{**}(W), \quad \text{Rank } W \leq \min_{i \in [k+1]} d_i =: r \tag{3.18}$$

Define the "model" $\bar{M} := \mathrm{C_{yx}} \mathrm{C}^{-\frac{1}{2}}$. *The global optimum of problem 3.18 is given as* $M^* = \mathcal{S}_{\alpha_\rho}(\bar{M}) \mathrm{C}^{-\frac{1}{2}}$, *where* $\alpha_\rho := \frac{\nu_{\{d_i\}} \sum_{j=1}^{\rho} \sigma_j(\bar{M})}{1 + \rho \nu_{\{d_i\}}}$, *and* $\rho \in [\min\{r, \text{Rank } \bar{M}\}]$ *is the largest integer such that for all* $i \in [\rho]$, *it holds that* $\sigma_i(\bar{M}) > \alpha_\rho$.

Proof of Lemma 9. Denote the objective in the optimization problem (3.18) as

$$\mathcal{E}_{\nu_{\{d_i\}}}(\mathrm{W}) := \mathbb{E}[\|\mathrm{y} - \mathrm{Wx}\|^2] + \nu_{\{d_i\}} \|\mathrm{WC}^{\frac{1}{2}}\|_*^2.$$

Let $\mathrm{C_y} := \mathbb{E}[\mathrm{yy}^\top]$ and $\mathrm{C_{xy}} := \mathbb{E}[\mathrm{xy}^\top]$. Note that

$$\begin{aligned}
\min_{\text{Rank W} \leq r} \mathcal{E}_{\nu_{\{d_i\}}}(\mathrm{W}) &= \min_{\text{Rank W} \leq r} \mathbb{E}[\|\mathrm{y}\|^2] + \mathbb{E}[\|\mathrm{Wx}\|^2] - 2\mathbb{E}[\langle \mathrm{y}, \mathrm{Wx} \rangle] + \nu_{\{d_i\}} \|\mathrm{WC}^{\frac{1}{2}}\|_*^2 \\
&\equiv \min_{\text{Rank W} \leq r} \ \mathrm{Tr}\left(\mathbb{E}[\mathrm{Wxx}^\top \mathrm{W}^\top]\right) - 2\,\mathrm{Tr}\left(\mathbb{E}[\mathrm{Wxy}^\top]\right) + \nu_{\{d_i\}} \|\mathrm{WC}^{\frac{1}{2}}\|_*^2 \\
&= \min_{\text{Rank W} \leq r} \ \mathrm{Tr}\left(\mathrm{WCW}^\top\right) - 2\,\mathrm{Tr}\left(\mathrm{WC_{xy}}\right) + \nu_{\{d_i\}} \|\mathrm{WC}^{\frac{1}{2}}\|_*^2
\end{aligned}$$

Make the change of variable $\bar{\mathrm{W}} \leftarrow \mathrm{WC}^{\frac{1}{2}}$ and denote $\bar{\mathrm{M}} := \mathrm{C_{yx}} \mathrm{C}^{-\frac{1}{2}}$, the goal is to solve the following problem

$$\min_{\text{Rank W} \leq r} \ \mathrm{Tr}\left(\bar{\mathrm{W}} \bar{\mathrm{W}}^\top\right) - 2\langle \bar{\mathrm{W}}, \bar{\mathrm{M}} \rangle + \nu_{\{d_i\}} \|\bar{\mathrm{W}}\|_*^2 \equiv \min_{\text{Rank } \bar{\mathrm{W}} \leq r} \|\bar{\mathrm{M}} - \bar{\mathrm{W}}\|_F^2 + \nu_{\{d_i\}} \|\bar{\mathrm{W}}\|_*^2 \tag{3.19}$$

If $\bar{\mathrm{W}}$ is a solution to the above problem, then a solution to the original problem in Equation (3.18) is given as $\bar{\mathrm{W}} \mathrm{C}^{-\frac{1}{2}}$. Following [CHL+18, MAV18], we show that the global optimum of Problem 3.19 is given in terms of an appropriate shrinkage-thresholding on the spectrum of $\bar{\mathrm{M}}$. Define $r' := \max\{\text{Rank } \bar{\mathrm{M}}, r\}$. Let $\bar{\mathrm{M}} = \mathrm{U}_{\bar{\mathrm{M}}} \Sigma_{\bar{\mathrm{M}}} \mathrm{V}_{\bar{\mathrm{M}}}^\top$ and $\bar{\mathrm{W}} = \mathrm{U}_{\bar{\mathrm{W}}} \Sigma_{\bar{\mathrm{W}}} \mathrm{V}_{\bar{\mathrm{W}}}^\top$ be rank-r' SVDs of $\bar{\mathrm{M}}$ and $\bar{\mathrm{W}}$ respectively, such that $\sigma_i(\bar{\mathrm{M}}) \geq \sigma_{i+1}(\bar{\mathrm{M}})$ and $\sigma_i(\bar{\mathrm{W}}) \geq \sigma_{i+1}(\bar{\mathrm{W}})$ for all $i \in [r' - 1]$. Rewriting objective of Problem 3.19

in terms of these decompositions gives:

$$\begin{aligned}
&\|\bar{\mathrm{M}} - \bar{\mathrm{W}}\|_F^2 + \nu_{\{d_i\}}\|\bar{\mathrm{W}}\|_*^2 = \|\mathrm{U}_{\bar{\mathrm{M}}}\Sigma_{\bar{\mathrm{M}}}\mathrm{V}_{\bar{\mathrm{M}}}^\top - \mathrm{U}_{\bar{\mathrm{W}}}\Sigma_{\bar{\mathrm{W}}}\mathrm{V}_{\bar{\mathrm{W}}}^\top\|_F^2 + \nu_{\{d_i\}}\|\mathrm{U}_{\bar{\mathrm{W}}}\Sigma_{\bar{\mathrm{W}}}\mathrm{V}_{\bar{\mathrm{W}}}^\top\|_*^2 \\
&= \|\Sigma_{\bar{\mathrm{M}}} - \mathrm{U}'\Sigma_{\bar{\mathrm{W}}}\mathrm{V}'^\top\|_F^2 + \nu_{\{d_i\}}\|\Sigma_{\bar{\mathrm{W}}}\|_*^2 \\
&= \|\Sigma_{\bar{\mathrm{M}}}\|_F^2 + \|\Sigma_{\bar{\mathrm{W}}}\|_F^2 - 2\langle\Sigma_{\bar{\mathrm{M}}}, \mathrm{U}'\Sigma_{\bar{\mathrm{W}}}\mathrm{V}'^\top\rangle + \nu_{\{d_i\}}\|\Sigma_{\bar{\mathrm{W}}}\|_*^2
\end{aligned}$$

where $\mathrm{U}' = \mathrm{U}_{\bar{\mathrm{M}}}^\top\mathrm{U}_{\bar{\mathrm{W}}}$ and $\mathrm{V}' = \mathrm{V}_{\bar{\mathrm{M}}}^\top\mathrm{V}_{\bar{\mathrm{W}}}$. By Von Neumann's trace inequality, for a fixed $\Sigma_{\bar{\mathrm{W}}}$ we have that

$$\langle\Sigma_{\bar{\mathrm{M}}}, \mathrm{U}'\Sigma_{\bar{\mathrm{W}}}\mathrm{V}'^\top\rangle \leq \sum_{i=1}^{r'}\sigma_i(\bar{\mathrm{M}})\sigma_i(\bar{\mathrm{W}}),$$

where the equality is achieved when $\mathrm{U}_{\bar{\mathrm{M}}} = \mathrm{U}_{\bar{\mathrm{W}}}$ and $\mathrm{V}_{\bar{\mathrm{M}}} = \mathrm{V}_{\bar{\mathrm{W}}}$. Hence, problem 3.19 is reduced to

$$\min_{\substack{\|\Sigma_{\bar{\mathrm{W}}}\|_0 \leq r, \\ \Sigma_{\bar{\mathrm{W}}} \geq 0}} \|\Sigma_{\bar{\mathrm{M}}} - \Sigma_{\bar{\mathrm{W}}}\|_F^2 + \nu_{\{d_i\}}\left(\mathrm{Trace}(\Sigma_{\bar{\mathrm{W}}})\right)^2 = \min_{\bar{\sigma}\in\mathbb{R}_+^r} \sum_{i=1}^{r}\left(\lambda_i(\bar{\mathrm{M}}) - \bar{\sigma}_i\right)^2 + \nu_{\{d_i\}}\left(\sum_{i=1}^{r}\bar{\sigma}_i\right)^2$$

The Lagrangian is given by

$$L(\bar{\sigma}, \alpha) = \sum_{i=1}^{r}\left(\lambda_i(\bar{\mathrm{M}}) - \bar{\sigma}_i\right)^2 + \nu_{\{d_i\}}\left(\sum_{i=1}^{r}\bar{\sigma}_i\right)^2 - \sum_{i=1}^{r}\alpha_i\bar{\sigma}_i$$

The KKT conditions ensures that at the optima it holds for all $i \in [r]$ that

$$\bar{\sigma}_i \geq 0,\ \alpha_i \geq 0,\ \bar{\sigma}_i\alpha_i = 0, \quad 2(\bar{\sigma}_i - \sigma_i(\bar{\mathrm{M}})) + 2\nu_{\{d_i\}}\sum_{j=1}^{r}\bar{\sigma}_j - \alpha_i = 0$$

Let $\rho = |i : \bar{\sigma}_i > 0| \leq r$ be the number of nonzero $\bar{\sigma}_i$, i.e. rank of the global optimum $\bar{\mathrm{W}}$. For $i \in [\rho]$, we have $\alpha_i = 0$. Therefore, we have that:

$$\begin{aligned}
&\bar{\sigma}_i + \nu_{\{d_i\}}\sum_{j=1}^{r}\bar{\sigma}_j = \sigma_i(\bar{\mathrm{M}}) \\
\implies &(\mathrm{I}_\rho + \nu_{\{d_i\}}1_\rho 1_\rho^\top)\bar{\sigma}_{1:\rho} = \sigma_{1:\rho}(\bar{\mathrm{M}}) \\
\implies &\bar{\sigma}_{1:\rho} = (\mathrm{I}_\rho - \frac{\nu_{\{d_i\}}}{1+\rho\nu_{\{d_i\}}}1_\rho 1_\rho^\top)\sigma_{1:\rho}(\bar{\mathrm{M}}) = \sigma_{1:\rho}(\bar{\mathrm{M}}) - \frac{\nu_{\{d_i\}}\rho\kappa_\rho}{1+\rho\nu_{\{d_i\}}}1_\rho
\end{aligned}$$

where $\kappa_j := \frac{1}{j}\sum_{i=1}^{j}\sigma_i(\bar{\mathrm{M}})$. Also, in the second implication we use an instance of the Woodbury's matrix identity. In particular, for any integer r, and for any $\nu \in \mathbb{R}_+$, it

holds that

$$(\mathrm{I}_r + \nu 1_r 1_r^\top)^{-1} = \mathrm{I}_r - \frac{\nu}{1+r\nu} 1_r 1_r^\top. \tag{3.20}$$

The proof simply follows from the following set of equations.

$$\begin{aligned}(\mathrm{I}_r + \nu 1_r 1_r^\top)(\mathrm{I}_r - \frac{\nu}{1+r\nu} 1_r 1_r^\top) &= \mathrm{I}_r + \nu 1_r 1_r^\top - \frac{\nu}{1+r\nu} 1_r 1_r^\top - \frac{\nu^2}{1+r\nu} 1_r 1_r^\top 1_r 1_r^\top \\ &= \mathrm{I}_r + \left(\nu - \frac{\nu}{1+r\nu} - \frac{\nu^2 r}{1+r\nu}\right) 1_r 1_r^\top = \mathrm{I}_r\end{aligned}$$

The equation above tell us that for $i \in [\rho]$, the singular values of $\bar{\mathrm{W}}$ are just a shrinkage of the singular values of $\bar{\mathrm{M}}$. In particular, it means that $\rho \leq \operatorname{Rank} \bar{\mathrm{M}}$. Therefore, without loss of generality, we assume that $r \leq \operatorname{Rank} \bar{\mathrm{M}}$. Also, since $\bar{\sigma}_i > 0$ for all $i \in [\rho]$, it holds that $\sigma_i(\bar{\mathrm{M}}) > \frac{\nu_{\{d_i\}}\rho\kappa_\rho}{1+\rho\nu_{\{d_i\}}}$ for all $i \in [\rho]$. For $i \in \{\rho+1, \dots, r\}$, on the other hand, $\bar{\sigma}_i = 0$ and we have

$$\begin{aligned}\frac{1}{2}\alpha_i &= \bar{\sigma}_i + \nu_{\{d_i\}} \sum_{j=1}^{r} \bar{\sigma}_j - \sigma_i(\bar{\mathrm{M}}) \\ &= 0 + \frac{\nu_{\{d_i\}}}{1+\rho\nu_{\{d_i\}}} \sum_{j=1}^{\rho} \sigma_j(\bar{\mathrm{M}}) - \sigma_i(\bar{\mathrm{M}}) \\ &= -\sigma_i(\bar{\mathrm{M}}) + \frac{\nu_{\{d_i\}}\rho\kappa_\rho}{1+\rho\nu_{\{d_i\}}},\end{aligned}$$

where we used the fact that

$$\sum_{i=1}^{r} \bar{\sigma}_i = 1_\rho^\top \bar{\sigma}_{1:\rho} = \sum_{i=1}^{\rho} \sigma_i(\bar{\mathrm{M}}) - \frac{\nu_{\{d_i\}}\rho^2\kappa_\rho}{1+\rho\nu_{\{d_i\}}} = (1 - \frac{\nu_{\{d_i\}}\rho}{1+\rho\nu_{\{d_i\}}})\kappa_\rho = \frac{\rho\kappa_\rho}{1+\rho\nu_{\{d_i\}}}.$$

By dual feasibility, we conclude that $\sigma_i(\bar{\mathrm{M}}) \leq \frac{\nu_{\{d_i\}}\rho\kappa_\rho}{1+\rho\nu_{\{d_i\}}}$ for all $i \in \{\rho+1, \dots, r\}$, which completes the proof. □

In light of the above discussions, we can finally provide a proof for Theorem 10.

Proof of Theorem 10. Consider W^*, a global optimum of problem 3.3. If all such global optima can be implemented by equalized networks, then by Theorem 9 it holds that $\Theta(\mathrm{W}^*) = \Theta^{**}(\mathrm{W}^*) = \nu_{\{d_i\}} \|\mathrm{W}^* \mathrm{C}^{\frac{1}{2}}\|_*^2$. In this case, the lifted problem in Equation 3.3 boils down to the following convex problem

$$\min_{\mathrm{W} \in \mathbb{R}^{d_{k+1} \times d_0}} \mathbb{E}[\|\mathrm{y} - \mathrm{W}\mathrm{x}\|^2] + \nu_{\{d_i\}} \|\mathrm{W}\mathrm{C}^{\frac{1}{2}}\|_*^2, \quad \operatorname{Rank} \mathrm{W} \leq \min_{i \in [k+1]} d_i =: r. \tag{3.21}$$

Proposition 1, on the other hand, states that any rank-1 network map can be implemented by an equalized network. Therefore, the key idea of the proof is to make sure that the global optimum of problem 3.21 has rank equal to one. It suffices to notice that under the assumption $\sigma_1(\bar{\mathrm{M}}) - \sigma_2(\bar{\mathrm{M}}) \geq \frac{1}{\nu_{\{d_i\}}}\sigma_2(\bar{\mathrm{M}})$, it holds that $\sigma_1(\bar{\mathrm{M}}) > \frac{\nu_{\{d_i\}}\sigma_1(\bar{\mathrm{M}})}{1+\nu_{\{d_i\}}}$ and $\sigma_j(\bar{\mathrm{M}}) \leq \frac{\nu_{\{d_i\}}\sigma_1(\bar{\mathrm{M}})}{1+\nu_{\{d_i\}}}$ for all $j > 1$. In this case, using Lemma 9, the solution $\mathcal{S}_{\alpha_1}(\bar{\mathrm{M}})\mathrm{C}^{-\frac{1}{2}}$ has rank equal to one, which completes the proof. □

The gap condition in the theorem above can always be satisfied (e.g. by increasing the dropout rate or the depth, or decreasing the width) as long as there exists a gap between the first and the second singular values of $\bar{\mathrm{M}}$. Moreover, for the special case of deep linear networks with one output neuron [JT18, NLG+18], this condition is always satisfied since $\bar{\mathrm{M}} \in \mathbb{R}^{1\times d_0}$ and $\sigma_2(\bar{\mathrm{M}}) = 0$.

Corollary 1. *Consider the class of deep linear networks with a single output neuron. Let $\{W_i^*\}$ be a minimizer of L_θ. For any architecture $\{d_i\}$ and any network map $W_{k+1\to 1}$, it holds that: (1) $\Theta(W_{k+1\to 1}) = \nu\|W_{k+1\to 1}\|_{\mathrm{C}}^2$, (2) $W^*_{k+1\to 1} = \frac{1}{1+\nu}\mathrm{C}_{\mathrm{yx}}$, (3) the network specified by $\{W_i^*\}$ is equalized.*

We conclude this section with a remark. We know from the early work of [SHK+14] that feature dropout in linear regression is closely related to ridge regression. Corollary 1 generalizes the results of [SHK+14] to deep linear networks, and establishes a similar connection between dropout training and ridge regression.

3.4 Experimental Results

Dropout is widely used for training modern deep learning architectures resulting in the state-of-the-art performance in numerous machine learning tasks [SHK+14, KSH12, SLJ+15, TS14]. The purpose of this section is not to make a case for (or against) dropout when training deep networks, but rather verify empirically the theoretical

results from the previous section.[2]

For simplicity, the training data $\{x_i\}$ is sampled from a standard Gaussian distribution which in particular ensures that $C = I$. The labels $\{y_i\}$ are generated as $y_i \leftarrow Nx_i$, where $N \in \mathbb{R}^{d_{k+1} \times d_0}$. N is composed of $UV^\top + \texttt{noise}$, where $U \in \mathbb{R}^{d_{k+1} \times r}$, $V \in \mathbb{R}^{d_0 \times r}$ are sampled from a standard Gaussian and the entries of `noise` are sampled independently from a Gaussian distribution with small standard deviation. At each step of the dropout training, we use a minibatch of size 1000 to train the network. The learning rate is tuned over the set $\{1, 0.1, 0.01\}$. All experiments are repeated 50 times, the curves correspond to the average of the runs, and the grey region shows the standard deviation.

The experiments are organized as follows. First, since the convex envelope of the induced regularizer equals the squared nuclear norm of the network map (Theorem 9), it is natural to expect that dropout training performs a shrinkage-thresholding on the spectrum of $C_{yx}C^{-\frac{1}{2}} = M$ (see Lemma 9 in the appendix). We experimentally verify this in Section 3.4.1. Second, in Section 3.4.2, we focus on the equalization property. We attest Theorem 10 by showing that dropout training equalizes deep networks with one output neuron.

3.4.1 Spectral shrinkage and rank control

Note that the induced regularizer $\Theta(M)$ is a *spectral function* (see Lemma 6 in the appendix). On the other hand, by Theorem 9, $\Theta^{**}(M) = \nu_{\{d_i\}}\|M\|_*^2$. Therefore, if dropout training succeeds in finding an (approximate) minimizer of L_θ, it minimizes an upperbound on the squared of the nuclear norm of the network map. Hence, it is natural to expect that the dropout training performs a shrinkage-thresholding on the spectrum of the model, much like nuclear norm regularization. Figure 3-2 confirms this intuition. Here, we plot the singular value distribution of the final network map

[2]The code for the experiments can be found at: https://github.com/r3831/dln_dropout

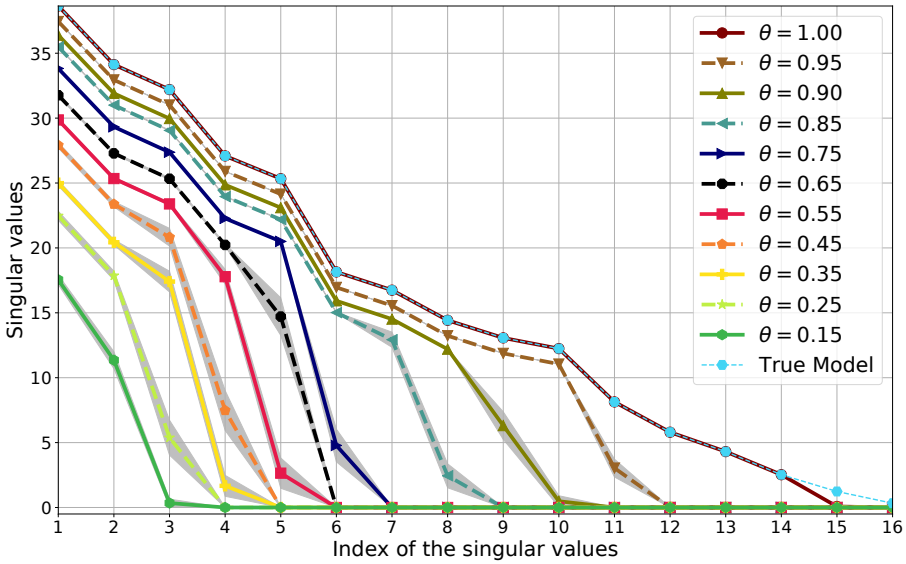

Figure 3-2. Distribution of the singular values of the trained network for different values of the dropout rate $1 - \theta$. It can be seen that the dropout training performs a more sophisticated form of shrinkage and thresholding on the spectrum of the model matrix M.

trained by dropout, for different values of the dropout rate.

As can be seen in the figure, dropout training indeed shrinks the spectrum of the model and thresholds it at zero. However, unlike the nuclear norm regularization, the shrinkage is not uniform across the singular values that are not thresholded. Moreover, note that the shrinkage parameter in Theorem 10 is governed by the effective regularization parameter $\nu_{\{d_i\}}$, which strictly increases with the dropout rate. This suggests that as we increase the dropout rate (decrease θ), the spectrum should be shrunk more severely, and the resulting network map should have a smaller rank. This is indeed the case as can be seen in Figure 3-2.

3.4.2 Convergence to equalized networks

One of the key concepts behind our analysis is the notion of equalized networks. In particular, in Lemma 7 we see that if a network map can be implemented by an equalized network, then there is no gap between the induced regularizer and its convex envelope. It is natural to ask if dropout training indeed finds such equalized networks.

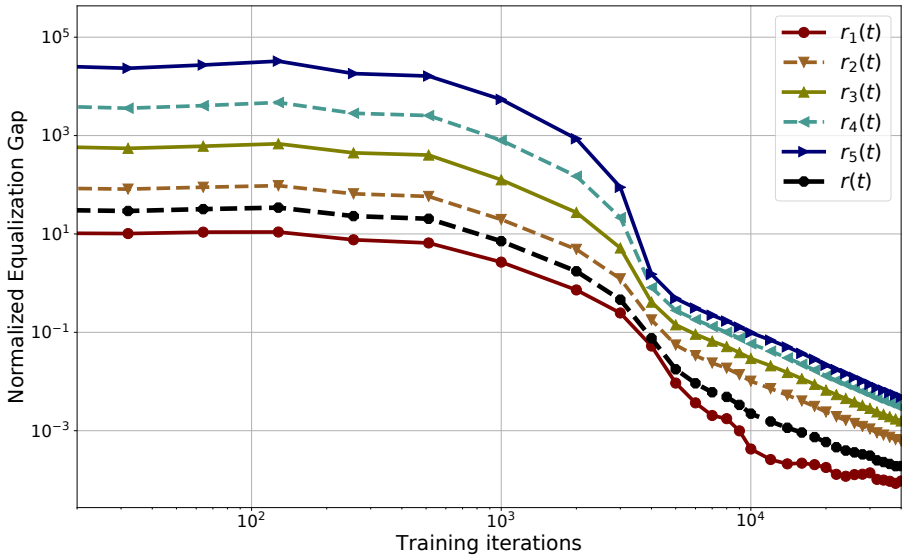

Figure 3-3. The normalized equalization gap $r_\ell^{(t)}$, which captures the gap between the sub-regularizers and their respective lower bounds, is plotted as a function of the number of iterations. Dropout converges to the set of equalized networks.

As we will discuss, Figure 3-3 answers this question affirmatively.

Recall that a network is equalized if and only if each and every sub-regularizer achieves its respective lowerbound in Equation 3.14, i.e. $R_l(\{\mathrm{W}_i\}) = LB_l(\{\mathrm{W}_i\})$ for all $l \in [k]$. Figure 3-3 illustrates that dropout training consistently decreases the gap between the sub-regularizers and their respective lowerbounds. Here, the network has one output neuron, five hidden layers each of width 5, and input dimensionality of $d_0 = 5$. In Figure 3-3 we plot the *normalized equalization gap* $r_\ell^{(t)} := \frac{R_\ell(\{\mathrm{W}_i^{(t)}\})}{LB_\ell(\{\mathrm{W}_i^{(t)}\})} - 1$ of the network under dropout training as a function of the iteration number. Similarly, we define the normalized equalization gap for the explicit regularizer $r^{(t)} = \frac{R(\{\mathrm{W}_i\})}{\Theta^{**}(\mathrm{W}_{k+1\to 1})} - 1$. The network quickly becomes (approximately) equalized, and thereafter the trajectory of dropout training stays close to the equalized networks. We believe that this observation can be helpful in analyzing the dynamics of dropout training, which we leave for future work.

3.5 Discussion

Previous work of [ZZ15, HLL+16, CHL+18] study dropout training with ℓ_2-loss in matrix factorization. In the previous Chapter, we also study dropout training with ℓ_2-loss in shallow linear networks. The work that is most relevant to us is that of [CHL+18], whose results are extended to the case of deep linear networks in this paper. In particular, we derive the *explicit regularizer* induced by dropout, which happens to be composed of the ℓ_2-path regularizer and other rescaling invariant regularizers. Furthermore, we show that the convex envelope of the induced regularizer factors into an *effective regularization parameter* and the square of the nuclear norm of network map multiplied with the principal root of the second moment of the input distribution. We further highlight *equalization* as a key network property under which the induced regularizer equals its convex envelope. We specify a subclass of problems satisfying the equalization property, for which we completely characterize the optimal networks that dropout training is biased towards.

Chapter 4

Statistical Guarantees for Dropout

In the previous two chapters, we laid out a foundation for a formal understanding of how dropout explicitly regularizes the learning objective. We focused on linear regression with shallow and deep linear networks (in Chapter 2 and Chapter 3, respectively), and provided a range of theoretical and empirical results suggesting that dropout explicitly biases the learning algorithm towards low-rank solutions.

In Chapter 3, we demonstrated the explicit form of the regularizer due to dropout for general deep linear networks of any architecture, which recovers our results for two-layer linear networks in Chapter 2 as a special case. We showed that the regularizer is a data-dependent quantity which includes (is equal to, for shallow networks) the ℓ_2-path norm of the network as well as other rescaling invariant terms that can be seen as product of the weights along certain circles in the graph of the network. We then analyzed the induced regularizer, i.e., the minimum of the explicit regularizer across all possible reparameterization of the network which compute the same function. We characterized a sufficient condition – always satisfied by shallow networks, and arbitrary deep networks with a single output neuron – under which, the induced regularizer reduces to a nuclear norm penalty.

The theoretical results presented in the previous chapters give precise characterization of the inductive bias due to dropout for factored models used in lin-

ear regression. In particular, the nuclear norm penalty, which is explicitly induced by dropout in deep regression, is the canonical regularizer in low-rank matrix learning problems, and is known to yield a rich inductive bias in such problems [SRJ04, Bac08, RFP10, SS10, CT10, KLT11]. However, neither of the previous chapters discuss how the induced regularizer provides capacity control, or equivalently, help us establish generalization bounds for dropout. It is thus natural to take a step forward and ask *how does the regularizer induced by dropout help generalization?*

In this chapter, we provide an answer to this question. First, we give explicit forms of the regularizers induced by dropout for the matrix sensing problem and two-layer neural networks with ReLU activations. Further, we leverage tools from statistical learning theory and establish capacity control due to dropout by giving precise generalization bounds. Our key contributions are as follows.

1. Our generalization bounds are solely in terms of the value of the explicit regularizer due to dropout. This is a significant departure from most of the prior work wherein dropout is analyzed in conjunction with additional norm-based capacity control, e.g., max-norm [WZZ+13, GZ16], or ℓ_p norm on the weights of the model [ZW18].

2. Our generalization bounds are data-dependent. We identify a simple distributional property (a notion we refer to as *retentivity*) that yields tight generalization bounds as evidenced by matching lower and upper bounds. We believe that this property may be useful more generally; see [ZDK+21] for another application.

3. Our results emphasize the role of parametrization, i.e., the choice of model architecture. We find that dropout does not yield useful capacity control when training a two-layer linear networks (unless we further assume that the covariance matrix of input features satisfies certain isotropic assumption). On the other hand, dropout for training a network with convolutional topology or a non-

linearity imparts useful inductive bias (see Section 4.4 for more details).

4. We provide extensive numerical evaluations for validating our theory including verifying that the proposed theoretical bound on the Rademacher complexity is predictive of the observed generalization gap as well as highlighting how dropout breaks "co-adaptation", a notion that was the main motivation behind the invention of dropout [HSK+12].

The rest of this chapter is organized as follows. In Section 4.1, we survey the related work. In Section 4.2, we study dropout for matrix completion, wherein, the matrix factors are dropped randomly during training. We show that this algorithmic procedure induces a data-dependent regularizer that in expectation behaves similar to the weighted trace-norm which has been shown to yield strong generalization guarantees for matrix completion [FSSS11]. In Section 4.3, we study dropout in two-layer ReLU networks. We show that the regularizer induced by dropout is a data-dependent measure that in expectation behaves as ℓ_2-path norm [NSS15], and establish distribution-dependent generalization bounds. We prove the main results in Section 4.5. In Section 4.6, we present empirical evaluations that confirm our theoretical findings for matrix completion and deep regression on real world datasets including the MovieLens data, as well as the MNIST and Fashion MNIST datasets.

4.1 Related Work

There has been several studies in recent years aimed at establishing theoretical underpinnings of why and how dropout helps with generalization. Of particular interest to the focus of this chapter, the works of [ZW18], [GZ16], and [WZZ+13] bound the Rademacher complexity of deep neural networks trained using dropout. In particular, [GZ16] show that the Rademacher complexity of the target class decreases polynomially or exponentially, for shallow and deep networks, respectively, albeit

they assume additional norm bounds on the weight vectors. Similarly, the works of [WZZ+13] and [ZW18] assume that certain norms of the weights are bounded, and show that the Rademacher complexity of the target class decreases with dropout rates.

We argue in this paper that dropout alone does not directly control the norms of the weight vectors; therefore, each of the works above fail to capture the practice. We emphasize that none of the previous works provide a generalization guarantee, i.e., a bound on the gap between the population risk and the empirical risk, merely in terms of the value of the explicit regularizer due to dropout. We give a first such result for dropout in the context of matrix completion and for a single hidden layer ReLU network.

4.2 Matrix Sensing

We begin with understanding dropout for matrix sensing, a problem which arguably is an important instance of a matrix learning problem with lots of applications, and is well understood from a theoretical perspective. The problem setup is the following. Let $\mathrm{M}_* \in \mathbb{R}^{d_2 \times d_0}$ be a matrix with rank $r_* := \mathrm{Rank}(\mathrm{M}_*)$. Let $\mathrm{A}^{(1)}, \ldots, \mathrm{A}^{(n)}$ be a set of measurement matrices of the same size as M_*. The goal of matrix sensing is to recover the matrix M_* from n observations of the form $y_i = \langle \mathrm{M}_*, \mathrm{A}^{(i)} \rangle$ such that $n \ll d_2 d_0$. A natural approach is to represent the matrix in terms of factors and solve the following *empirical risk* minimization problem:

$$\min_{\mathrm{U},\mathrm{V}} \widehat{L}(\mathrm{U}, \mathrm{V}) := \widehat{\mathbb{E}}_i (y_i - \langle \mathrm{U}\mathrm{V}^\top, \mathrm{A}^{(i)} \rangle)^2 \tag{4.1}$$

where $\mathrm{U} = [\mathrm{u}_1, \ldots, \mathrm{u}_{d_1}] \in \mathbb{R}^{d_2 \times d_1}, \mathrm{V} = [\mathrm{v}_1, \ldots, \mathrm{v}_{d_1}] \in \mathbb{R}^{d_0 \times d_1}$. When the number of factors is unconstrained, i.e., when $d_1 \gg r_*$, there exist many "bad" empirical minimizers, i.e., those with a large *true risk* $L(\mathrm{U}, \mathrm{V}) := \mathbb{E}(y - \langle \mathrm{U}\mathrm{V}^\top, \mathrm{A} \rangle)^2$. Interestingly, [LMZ18] showed recently that under a restricted isometry property (RIP), despite the existence of such poor ERM solutions, gradient descent with proper initialization is

implicitly biased towards finding solutions with minimum nuclear norm – this is an important result which was first conjectured and empirically verified by [GWB+17]. We do not make an RIP assumption here. Further, we argue that for the most part, modern machine learning systems employ *explicit* regularization techniques. In fact, as we show in the experimental section, the *implicit* bias due to (stochastic) gradient descent does not prevent it from blatant overfitting in the matrix completion problem.

We propose solving the ERM problem (4.1) using dropout, where at training time, corresponding columns of U and V are dropped uniformly at random. As opposed to an *implicit* effect of gradient descent, dropout *explicitly* regularizes the empirical objective. It is then natural to ask, in the case of matrix sensing, if dropout also biases the ERM towards certain low norm solutions. To answer this, we begin with the observation that dropout can be viewed as an instance of SGD on the following objective [WM13, SHK+14] $\widehat{L}_{\text{drop}}(\mathrm{U},\mathrm{V}) = \widehat{\mathbb{E}}_j \mathbb{E}_{\mathrm{B}}(y_j - \langle \mathrm{UBV}^\top, \mathrm{A}^{(j)} \rangle)^2$, where $\mathrm{B} \in \mathbb{R}^{d_1 \times d_1}$ is a diagonal matrix whose diagonal elements are Bernoulli random variables distributed as $\mathrm{B}_{ii} \sim \frac{1}{1-p}\mathrm{Ber}(1-p)$. It is easy to show that for $p \in [0,1)$:

$$\widehat{L}_{\text{drop}}(\mathrm{U},\mathrm{V}) = \widehat{L}(\mathrm{U},\mathrm{V}) + \frac{p}{1-p}\widehat{R}(\mathrm{U},\mathrm{V}), \tag{4.2}$$

where $\widehat{R}(\mathrm{U},\mathrm{V}) := \sum_{i=1}^{d_1} \widehat{\mathbb{E}}_j (\mathrm{u}_i^\top \mathrm{A}^{(j)} \mathrm{v}_i)^2$ is a data-dependent term that captures the *explicit* regularizer due to dropout. A similar result was shown by [MAV18], but we provide a proof for completeness (see Proposition 3 in Section 4.5).

Furthermore, given that we seek a minimum of $\widehat{L}_{\text{drop}}$, it suffices to consider the factors with the minimal value of the regularizer among all that yield the same empirical loss. This motivates studying the following distribution-dependent *induced* regularizer:

$$\Theta(\mathrm{M}) := \min_{\mathrm{UV}^\top = \mathrm{M}} R(\mathrm{U},\mathrm{V}), \text{ where } R(\mathrm{U},\mathrm{V}) := \mathbb{E}_{\mathrm{A}}[\widehat{R}(\mathrm{U},\mathrm{V})].$$

We instantiate induced regularizer for two instances of random measurements (See Proposition 4 in Section 4.5).

Gaussian Measurements. For all $j \in [n]$, let $\mathrm{A}^{(j)}$ be standard Gaussian matrices. In this case, it is easy to see that $\mathrm{L}(\mathrm{U}, \mathrm{V}) = \|\mathrm{M}_* - \mathrm{U}\mathrm{V}^\top\|_F^2$ and we recover the matrix factorization problem. Furthermore, we know from [MA19] that dropout regularizer acts as trace-norm regularization, i.e., $\Theta(\mathrm{M}) = \frac{1}{d_1}\|\mathrm{M}\|_*^2$.

Matrix Completion. For all $j \in [n]$, let $\mathrm{A}^{(j)}$ be an indicator matrix drawn from a product distribution over the rows and columns. That is, the probability of choosing the indicator of the (i,k)-th element is $p(i)q(k)$, where $p(i)$ and $q(k)$ denote the probability of choosing the i-th row and the k-th column, respectively. Then, $\Theta(\mathrm{M}) = \frac{1}{d_1}\|\operatorname{diag}(\sqrt{p})\mathrm{U}\mathrm{V}^\top \operatorname{diag}(\sqrt{q})\|_*^2$ is the *weighted trace-norm* studied by [SS10] and [FSSS11].

These observations are specifically important because they connect dropout, an algorithmic heuristic in deep learning, to strong complexity measures that are empirically effective as well as theoretically well understood. To illustrate, here we give a generalization bound for matrix completion using dropout in terms of the value of the *explicit* regularizer at the minimizer.

Theorem 11. Assume that $d_2 \geq d_0$ and $\|\mathrm{M}_*\| \leq 1$. Furthermore, assume that $\min_{i,k} p(i)q(k) \geq \frac{\log(d_2)}{n\sqrt{d_2 d_0}}$. Let Let (U, V) be the output of ERM with dropout with $R(\mathrm{U}, \mathrm{V}) \leq \alpha/d_1$. Then, for any $\delta \in (0,1)$, the following generalization bounds holds with probability at least $1-\delta$ over a sample of size n:

$$L(g(\mathrm{U}\mathrm{V}^\top)) \leq \widehat{L}(\mathrm{U}, \mathrm{V}) + 8\sqrt{\frac{2\alpha d_2 \log(d_2) + \frac{1}{4}\log(2/\delta)}{n}}$$

where $g(\mathrm{M})$ thresholds M at ± 1, i.e. $g(\mathrm{M})(i,j) = \max\{-1, \min\{1, \mathrm{M}(i,j)\}\}$, and $L(g(\mathrm{U}\mathrm{V}^\top)) := \mathbb{E}(y - \langle g(\mathrm{U}\mathrm{V}^\top), \mathrm{A}\rangle)^2$ is the *true risk* of $g(\mathrm{U}\mathrm{V}^\top)$.

The proof of Theorem 11 follows from standard generalization bounds for ℓ_2 loss [MRT18] based on the Rademacher complexity [BM02] of the class of functions with weighted trace-norm bounded by $\sqrt{\alpha}$, i.e. $\mathcal{M}_\alpha := \{\mathrm{M} : \|\operatorname{diag}(\sqrt{\mathrm{p}})\mathrm{M}\operatorname{diag}(\sqrt{\mathrm{q}})\|_*^2 \leq \alpha\}$. The non-degeneracy condition $\min_{i,j} p(i)q(j) \geq \frac{\log(d_2)}{n\sqrt{d_2 d_0}}$ is required to obtain a bound

on the Rademacher complexity of $\mathcal{M}_\alpha$, as established by [FSSS11]. Furthermore, since the induced regularizer is scaled as $1/d_1$ compared to the squared weighted trace-norm, i.e. $\Theta(\mathrm{U}\mathrm{V}^\top) = \frac{1}{d_1}\|\operatorname{diag}(\sqrt{p})\mathrm{U}\mathrm{V}^\top \operatorname{diag}(\sqrt{\mathrm{q}})\|_*^2$, we scale α accordingly by letting $R(\mathrm{U},\mathrm{V}) \leq \alpha/d_1$.

In practice, for models that are trained with dropout, the training error $\widehat{L}(\mathrm{U},\mathrm{V})$ is negligible (see Figure 4-1 for experiments on the MovieLens dataset). Moreover, given that the sample size is large enough, the third term can be made arbitrarily small. Having said that, the second term, which is $\tilde{O}(\sqrt{\alpha d_2/n})$, dominates the right hand side of generalization error bound in Theorem 20. In Appendix, we also give optimistic generalization bounds that decay as $\tilde{O}(ad_2/n)$.

Finally, the required sample size depends on the value of the explicit regularizer (i.e., α/d_1), and hence, on the dropout rate p. In particular, increasing the dropout rate increases the regularization parameter $\lambda := \frac{p}{1-p}$, thereby intensifying the penalty due to the explicit regularizer. Intuitively, a larger dropout rate p results in a smaller α, thereby a tighter generalization gap can be guaranteed. We show through experiments that that is indeed the case in practice.

4.2.1 Comparison with Previous Work

Our study of dropout in this section is motivated in part by recent works of [CHL+18], as well as our own results in Chapter 2 and Chapter 3. This line of work was initiated by [CHL+18], who studied dropout for low-rank matrix factorization without constraining the rank of the factors or adding an explicit regularizer to the objective. They show that dropout in the context of matrix factorization yields an explicit regularizer whose convex envelope is given by nuclear norm. We further strengthened this result in Chapter 3, where we show that induced regularizer is indeed nuclear norm.

While matrix factorization is not a learning problem per se (for instance, what is training versus test data), as we showed in Chapter 2, training deep linear networks

with ℓ_2-loss using dropout reduces to the matrix factorization problem if the marginal distribution of the input feature vectors is assumed to be isotropic, i.e., $\mathbb{E}[\mathrm{x}\mathrm{x}^\top] = \mathrm{I}$. We note that this is a strong assumption. If we do not assume isotropy, we show that dropout induces a data-dependent regularizer which amounts to a simple scaling of the parameters and, therefore, does not control capacity in any meaningful way. We revisit this discussion in Section 4.4. To summarize, while we are motivated by [CHL+18], the problem setup, the nature of statements in this chapter, and the tools we use are different from that in [CHL+18].

We note that, different from our results in Chapter 2, in this chapter, we rigorously argue for dropout in matrix completion by 1) showing that the induced regularizer is equal to weighted trace-norm, which as far as we know, is a novel result, 2) giving strong generalization bounds, and 3) providing extensive experimental evidence that dropout provides state of the art performance on one of the largest datasets in recommendation systems research. Beyond that, in the next section, we rigorously extend our results to two layer ReLU networks, describe the explicit regularizer, bound the Rademacher complexity of the hypothesis class controlled by dropout, show precise generalization bounds, and support them with empirical results.

4.3 Non-linear Networks

Next, we focus on neural networks with a single hidden layer. Let $\mathcal{X} \subseteq \mathbb{R}^{d_0}$ and $\mathcal{Y} \subseteq [-1, 1]^{d_2}$ denote the input and output spaces, respectively. Let $\mathcal{D}$ denote the joint probability distribution on $\mathcal{X} \times \mathcal{Y}$. Given n examples $\{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^n \sim \mathcal{D}^n$ drawn i.i.d. from the joint distribution and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, the goal of learning is to find a hypothesis $f_\mathrm{w} : \mathcal{X} \to \mathcal{Y}$, parameterized by w, that has a small *population risk* $L(f_\mathrm{w}) := \mathbb{E}_\mathcal{D}[\ell(f_\mathrm{w}(\mathrm{x}), \mathrm{y})]$.

We focus on the squared ℓ_2 loss, i.e., $\ell(\mathrm{y}, \mathrm{y}') = \|\mathrm{y} - \mathrm{y}'\|^2$, and study the gen-

eralization properties of the dropout algorithm for minimizing the *empirical risk* $\widehat{L}(f_{\mathrm{w}}) := \widehat{\mathbb{E}}_i[\|\mathrm{y}_i - f_{\mathrm{w}}(\mathrm{x}_i)\|^2]$. We consider the hypothesis class associated with feed-forward neural networks with 2 layers, i.e., functions of the form $f_{\mathrm{w}}(\mathrm{x}) = \mathrm{U}\sigma(\mathrm{V}^\top \mathrm{x})$, where $\mathrm{U} = [\mathrm{u}_1, \ldots, \mathrm{u}_{d_1}] \in \mathbb{R}^{d_2 \times d_1}, \mathrm{V} = [\mathrm{v}_1, \ldots, \mathrm{v}_{d_1}] \in \mathbb{R}^{d_0 \times d_1}$ are the weight matrices. The parameter w is the collection of weight matrices $\{\mathrm{U}, \mathrm{V}\}$ and $\sigma : \mathbb{R} \to \mathbb{R}$ is the ReLU activation function applied entrywise to an input vector. As in Section 4.2, we view dropout as an instance of stochastic gradient descent on the following *dropout objective*:

$$\widehat{L}_{\mathrm{drop}}(\mathrm{w}) := \widehat{\mathbb{E}}_i \mathbb{E}_{\mathrm{B}} \|\mathrm{y}_i - \mathrm{UB}\sigma(\mathrm{V}^\top \mathrm{x}_i)\|^2, \tag{4.3}$$

where B is a diagonal random matrix with diagonal elements distributed i.i.d. as $\mathrm{B}_{ii} \sim \frac{1}{1-p}\mathrm{Bern}(1-p),\ i \in [d_1]$, for some *dropout rate* p. We seek to understand the *explicit regularizer* due to dropout:

$$\widehat{R}(\mathrm{w}) := \widehat{L}_{\mathrm{drop}}(\mathrm{w}) - \widehat{L}(f_{\mathrm{w}}). \tag{4.4}$$

We denote the output of the i-th hidden node on an input vector x by $a_i(\mathrm{x}) \in \mathbb{R}$; for example, $a_2(\mathrm{x}) = \sigma(\mathrm{v}_2^\top \mathrm{x})$. Similarly, the vector $\mathrm{a}(\mathrm{x}) \in \mathbb{R}^{d_1}$ denotes the activation of the hidden layer on input x. Using this notation, we can rewrite the objective in (4.3) as $\widehat{L}_{\mathrm{drop}}(\mathrm{w}) := \mathbb{E}_i \mathbb{E}_{\mathrm{B}} \|\mathrm{y}_i - \mathrm{UBa}(\mathrm{x}_i)\|^2$. It is then easy to show that the regularizer due to dropout in (4.4) is given as (See Proposition 5 in Section 4.5):

$$\widehat{R}(\mathrm{w}) = \frac{p}{1-p} \sum_{j=1}^{d_1} \|\mathrm{u}_j\|^2 \widehat{a}_j^2, \text{ where } \widehat{a}_j = \sqrt{\widehat{\mathbb{E}}_i a_j(\mathrm{x}_i)^2}.$$

The explicit regularizer $\widehat{R}(\mathrm{w})$ is a summation over hidden nodes, of the product of the squared norm of the outgoing weights with the empirical second moment of the output of the corresponding neuron. We should view it as a data-dependent variant of the ℓ_2 path-norm of the network, studied recently by [NTS15] and shown to yield capacity control in deep learning. Indeed, if we consider ReLU activations and input distributions that are symmetric and isotropic [MAV18], the expected regularizer is

equal to the sum over all paths from input to output of the product of the squares of weights along the paths, i.e., $R(\mathrm{w}) := \mathbb{E}[\widehat{R}(\mathrm{w})] = \frac{1}{2}\sum_{i_0,i_1,i_2=1}^{d_0,d_1,d_2} \mathrm{U}(i_2,i_1)^2\mathrm{V}(i_0,i_1)^2$, which is precisely the squared ℓ_2 path-norm of the network. We refer the reader to Proposition 6 in the Appendix for a formal statement and proof.

Generalization Bounds. To understand the generalization properties of dropout, we focus on the following distribution-dependent hypothesis class

$$\mathcal{F}_\alpha := \{f_\mathrm{w} : \mathrm{x} \mapsto \mathrm{u}^\top\sigma(\mathrm{V}^\top\mathrm{x}),\ \sum_{i=1}^{d_1}|u_i|a_i \le \alpha\}, \tag{4.5}$$

where $\mathrm{u} \in \mathbb{R}^{d_1}$ is the top layer weight vector, u_i denotes the i-th entry of u, and $a_i^2 := \mathbb{E}_\mathrm{x}[\widehat{a}_i^2] = \mathbb{E}_\mathrm{x}[a_i(\mathrm{x})^2]$ is the expected squared activation of the i-th hidden node. For simplicity, we focus on networks with one output neuron ($d_2 = 1$); extension to multiple output neurons is straightforward.

We argue that networks trained with dropout belong to the class $\mathcal{F}_\alpha$, for a small value of α. In particular, by Cauchy-Schwartz inequality, it is easy to to see that $\sum_{i=1}^{d_1}|u_i|a_i \le \sqrt{d_1 R(\mathrm{w})}$. Thus, for a fixed width, dropout implicitly controls the function class $\mathcal{F}_\alpha$. More importantly, this inequality is loose if a small subset of hidden nodes $\mathcal{J} \subset [d_1]$ *"co-adapt"* in a way that for all $j \in [d_1] \setminus \mathcal{J}$, the other hidden nodes are almost inactive, i.e. $u_j a_j \approx 0$. In other words, by minimizing the expected regularizer, dropout is biased towards networks where gap between $R(\mathrm{w})$ and $(\sum_{i=1}^{d_1}|u_i|a_i)^2/d_1$ is small, which in turn happens if $|u_i|a_i \approx |u_j|a_j, \forall i,j \in [d_1]$. In this sense, dropout breaks *"co-adaptation"* between neurons by promoting solutions with nearly equal contribution from hidden neurons.

As we mentioned in the introduction, a bound on the dropout regularizer is not sufficient to guarantee a bound on a norm-based complexity measures that are common in the deep learning literature (see, e.g., [GRS18] and the references therein), whereas a norm bound on the weight vector would imply a bound on the explicit regularizer due to dropout. Formally, we show the following.

Proposition 2. *For any $C > 0$, there exists a distribution on the unit Euclidean sphere, and a network $f_{\mathrm{w}} : \mathrm{x} \mapsto \sigma(\mathrm{w}^\top \mathrm{x})$, such that $R(\mathrm{w}) = \sqrt{\mathbb{E}\sigma(\mathrm{w}^\top \mathrm{x})^2} \leq 1$, while $\|\mathrm{w}\| > C$.*

Proof of Proposition 2. For $\delta \in (0, \frac{1}{2})$, consider the following random variable:

$$\mathrm{x} = \begin{cases} [1; 0] & \text{with probability } \delta \\ [\frac{-\delta}{1-\delta}; \frac{\sqrt{1-2\delta}}{1-\delta}] & \text{with probability } \frac{1-\delta}{2} \\ [\frac{-\delta}{1-\delta}; -\frac{\sqrt{1-2\delta}}{1-\delta}] & \text{with probability } \frac{1-\delta}{2} \end{cases}$$

It is easy to check that the x has zero mean and is supported on the unit sphere. Consider the vector $\mathrm{w} = [\frac{1}{\sqrt{\delta}}; 0]$. It is easy to check that x satisfies $R(\mathrm{w}) = \sqrt{\mathbb{E}\sigma(\mathrm{w}^\top \mathrm{x})^2} = 1$; however, for any given C, it holds that $\|\mathrm{w}\| \geq C$ as long as we let $\delta = C^2$. □

In other words, even though we connect the dropout regularizer to path-norm, the data-dependent nature of the regularizer prevents us from leveraging that connection in data-independent manner (i.e., for all distributions). At the same time, making strong distributional assumptions (as in Proposition 6) would be impractical. Instead, we argue for the following milder condition on the input distribution which we show as sufficient to ensure generalization.

Assumption 1 (β-retentive)**.** The marginal input distribution is β-retentive for some $\beta \in (0, 1/2]$, if for any non-zero vector $\mathrm{v} \in \mathbb{R}^d$, it holds that $\mathbb{E}\sigma(\mathrm{v}^\top \mathrm{x})^2 \geq \beta \mathbb{E}(\mathrm{v}^\top \mathrm{x})^2$.

Intuitively, what the assumption implies is that the variance (aka, the information or signal in the data) in the pre-activation at any node in the network is not quashed considerably due to the non-linearity. In fact, no reasonable training algorithm should learn weights where β is small. However, we steer clear from algorithmic aspects of dropout training, and make the assumption above for every weight vector as we need to take a union bound. We now present the first main result of this section, which bounds the Rademacher complexity of $\mathcal{F}_\alpha$ in terms of α, the retentiveness coefficient

β, and Mahalanobis norm of data w.r.t. the pseudo-inverse of the second moment matrix, i.e. $\|\mathrm{X}\|_{\mathrm{C}^\dagger}^2 = \sum_{i=1}^n \mathrm{x}_i^\top \mathrm{C}^\dagger \mathrm{x}_i$.

Theorem 12. *For any sample $S = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^n$ of size n, $\mathfrak{R}_{\mathcal{S}}(\mathcal{F}_\alpha) \leq \frac{2\alpha\|\mathrm{X}\|_{\mathrm{C}^\dagger}}{n\sqrt{\beta}}$. Furthermore, it holds for the expected Rademacher complexity that $\mathfrak{R}_n(\mathcal{F}_\alpha) \leq 2\alpha\sqrt{\frac{\mathrm{Rank(C)}}{\beta n}}$.*

First, note that the bound depends on the quantity $\|\mathrm{X}\|_{\mathrm{C}^\dagger}$ which can be in the same order as $\|\mathrm{X}\|_F$ with both scaling as $\asymp \sqrt{nd_0}$; the latter is more common in the literature [NLB+18, BFT17, NBS17, GRS18, NTS15]. This is unfortunately unavoidable, unless one makes stronger distributional assumptions.

Second, as we discussed earlier, the dropout regularizer directly controls the value of α, thereby controlling the Rademacher complexity in Theorem 12. This bound also gives us a bound on the Rademacher complexity of the networks trained using dropout. To see that, consider the following class of networks with bounded explicit regularizer, i.e., $\mathcal{H}_r := \{h_{\mathrm{w}} : \mathrm{x} \mapsto \mathrm{u}^\top \sigma(\mathrm{V}^\top \mathrm{x}),\ R(\mathrm{u}, \mathrm{V}) \leq r\}$. Then, Theorem 12 yields $\mathfrak{R}_{\mathcal{S}}(\mathcal{H}_r) \leq \frac{2\sqrt{d_1 r}\|\mathrm{X}\|_{\mathrm{C}^\dagger}}{n\sqrt{\beta}}$. In fact, we can show that this bound is tight up to $1/\sqrt{\beta}$ by a reduction to the linear case. Formally, we show the following.

Theorem 13 (Lowerbound)**.** *There is a constant c such that for any $r > 0$, $\mathfrak{R}_{\mathcal{S}}(\mathcal{H}_r) \geq \frac{c\sqrt{d_1 r}\|\mathrm{X}\|_{\mathrm{C}^\dagger}}{n}$.*

Moreover, it is easy to give a generalization bound based on Theorem 12 that depends only on the distribution dependent quantities α and β. Let $g_{\mathrm{w}}(\cdot) := \max\{-1, \min\{1, f_{\mathrm{w}}(\cdot)\}\}$ project the network output f_{w} onto the range $[-1, 1]$. We have the following generalization gurantees for g_{w}.

Corollary 2. *For any $\mathrm{w} \in \mathcal{F}_\alpha$, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over a sample $\mathcal{S}$ of size n, we have $L(g_{\mathrm{w}}) \leq \widehat{L}(g_{\mathrm{w}}) + \frac{16\alpha\|\mathrm{X}\|_{\mathrm{C}^\dagger}}{\sqrt{\beta}n} + 12\sqrt{\frac{\log(2/\delta)}{2n}}$.*

Proof of Corollary 2. We use the standard generalization bound in Theorem 19 for

class $\mathcal{G}_\alpha$:

$$\begin{aligned} L_{\mathcal{D}}(g_{\mathrm{w}}) &\leq \widehat{L}_{\mathcal{S}}(g_{\mathrm{w}}) + 4M\mathfrak{R}_{\mathcal{S}}(\mathcal{G}_\alpha) + 3M^2\sqrt{\frac{\log(2/\delta)}{2n}} \\ &\leq \widehat{L}_{\mathcal{S}}(g_{\mathrm{w}}) + 8\mathfrak{R}_{\mathcal{S}}(\mathcal{F}_\alpha) + 12\sqrt{\frac{\log(2/\delta)}{2n}} && \text{(Lemma 10)} \\ &\leq \widehat{L}_{\mathcal{S}}(g_{\mathrm{w}}) + \frac{16\alpha\|\mathrm{X}\|_{\mathrm{C}^\dagger}}{\sqrt{\beta}n} + 12\sqrt{\frac{\log(2/\delta)}{2n}} && \text{(Theorem 12)} \end{aligned}$$

where second inequality follows because the maximum deviation parameter M in Theorem 19 is bounded as

$$M = \sup_{\mathrm{w}\in\mathcal{W}} \sup_{(\mathrm{x},y)\in\mathcal{X}\times\mathcal{Y}} |y - g_{\mathrm{w}}(\mathrm{x})| \leq \sup_{\mathrm{w}\in\mathcal{W}} \sup_{(\mathrm{x},y)\in\mathcal{X}\times\mathcal{Y}} |y| + |g_{\mathrm{w}}(\mathrm{x})| \leq 2.$$

□

We would like to remark that the focus here is on *understanding* how the expected explicit regularizer alone – without any additional norm-bounds on the weights – can provide generalization. If one is interested in *predicting* the generalization gap, then one can estimate the (empirical) explicit regularizer on a held-out dataset, and appeal to simple concentration arguments, just as we do in our experiments.

β-independent Bounds. Geometrically, β-retentiveness requires that for any hyperplane passing through the origin, both halfspaces contribute significantly to the second moment of the data in the direction of the normal vector. It is not clear, however, if β can be estimated efficiently on a dataset. Nonetheless, when $\mathcal{X} \subseteq \mathbb{R}^{d_0}_+$, which is the case for image datasets, a simple *symmetrization* technique, described below, allows us to give bounds that are β-independent. We propose the following randomized symmetrization. *Given a training sample $\mathcal{S} = \{(\mathrm{x}_i, y_i), i \in [n]\}$, consider the randomly perturbed dataset, $\mathcal{S}' = \{(\zeta_i \mathrm{x}_i, y_i),\ i \in [n]\}$, where ζ_i's are i.i.d. Rademacher random variables.* We give a generalization bound (w.r.t. the original data distribution) for the hypothesis class with bounded regularizer w.r.t. perturbed data distribution.

Corollary 3. *Given an i.i.d. sample* $\mathcal{S} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^n$, *let* $\mathcal{F}'_\alpha := \{f_\mathrm{w} : \mathrm{x} \mapsto u^\top \sigma(V^\top \mathrm{x}),\ \sum_{i=1}^{d_1} |u_i| a'_i \leq \alpha\}$, *where* $a_i'^2 := \mathbb{E}_{\mathrm{x},\zeta}[a_i(\zeta \mathrm{x})^2]$. *For any* $\mathrm{w} \in \mathcal{F}'_\alpha$, *for any* $\delta \in (0,1)$, *with probability at least* $1-\delta$ *over a sample of size* n *and the randomization in symmetrization, we have that* $L(g_\mathrm{w}) \leq 2\widehat{L}(g_\mathrm{w}) + \frac{46\alpha \|\mathrm{X}\|_{\mathrm{C}\dagger}}{n} + 24\sqrt{\frac{\log(2/\delta)}{2n}}$, *where* $\widehat{L}$ *is evaluated on the symmetrized sample* $\mathcal{S}'$.

Note that the population risk of the clipped predictor $g_\mathrm{w}(\cdot) := \max\{-1, \min\{1, f_\mathrm{w}(\cdot)\}\}$ is bounded in terms of empirical risk on $\mathcal{S}'$. Finally, we verify in Section 4.6 that symmetrization of the training set, on MNIST and FashionMNIST datasets, does not have an effect on performance of the trained models.

4.3.1 Comparison with Previous Work

We would like to make a remark regarding the previous work of [MZGW18]. They consider a variant of dropout, which they call "truthful" dropout, that ensures that the output of the randomly perturbed network is unbiased.

Note that the results of this section, and in particular Corollary 2, bounds the generalization gap, i.e., $L(\cdot) - \widehat{L}(\cdot)$. However, rather than bound generalization gap, [MZGW18] bound the gap between the population risk and the dropout objective, i.e., the empirical risk plus the explicit regularizer. That is, [MZGW18] bound $L(\cdot) - \widehat{L}_{\mathrm{drop}}(\cdot)$, where $\widehat{L}_{\mathrm{drop}}(\mathrm{w}) = \widehat{L}(f_\mathrm{w}) + \widehat{R}(\mathrm{w})$, as in Equation (4.4).

The explicit regularizer $\widehat{R}(\cdot)$ is a positive quantity that does not vanish with the sample size. Therefore, the bound of [MZGW18] can guarantee that the generalization gap decays as $1/\sqrt{n}$ only if the dropout rate decreases as $1/\sqrt{n}$ (to ensure that $\widehat{R}(\cdot) = O(1/\sqrt{n})$). This is a stringent requirement on the dropout rate – in practice, dropout rate is treated as a hyperparameter that is tuned over a validation set, or otherwise is simply set to a constant, which does not decay with the sample size [HSK+12, SHK+14]. In sharp contrast, our analysis here is valid for *any* dropout rate.

4.4 Role of Parametrization

In this section, we argue that parametrization plays an important role in determining the nature of the inductive bias. We begin by considering matrix sensing in non-factorized form, which entails minimizing $\widehat{L}(\mathrm{M}) := \widehat{\mathbb{E}}_i(y_i - \langle \mathrm{vec}\,(\mathrm{M})\,, \mathrm{vec}\big(\mathrm{A}^{(i)}\big)\rangle)^2$, where $\mathrm{vec}\,(\mathrm{M})$ denotes the column vectorization of M. Then, the expected explicit regularizer due to dropout equals $R(\mathrm{M}) = \frac{p}{1-p}\|\,\mathrm{vec}\,(\mathrm{M})\,\|^2_{\mathrm{diag(C)}}$, where $\mathrm{C} = \mathbb{E}[\mathrm{vec}\,(\mathrm{A})\,\mathrm{vec}\,(\mathrm{A})^\top]$ is the second moment of the measurement matrices. For instance, with Gaussian measurements, the second moment equals the identity matrix, in which case, the regularizer reduces to the Frobenius norm of the parameters $R(\mathrm{M}) = \frac{p}{1-p}\|\mathrm{M}\|_F^2$. While such a ridge penalty yields a useful inductive bias in linear regression, it is not *"rich"* enough to capture the kind of inductive bias that provides rank control in matrix sensing.

However, simply representing the hypotheses in a factored form alone is not sufficient in terms of imparting a rich inductive bias to the learning problem. Recall that in linear regression, dropout, when applied on the input features, yields ridge regularization. However, if we were to represent the linear predictor in terms of a deep linear network, then we argue that the effect of dropout is markedly different. Consider a deep linear network, $f_{\mathrm{w}} : \mathrm{x} \mapsto \mathrm{W}_k \cdots \mathrm{W}_1\mathrm{x}$ with a single output neuron. In this case, [MA19] show that $\nu\|f\|^2_{\widehat{\mathrm{C}}} = \min_{f_{\mathrm{w}}=f} \widehat{R}(\mathrm{w})$, where ν is a regularization parameter independent of the parameters w. Consequently, in deep linear networks with a single output neuron, dropout reduces to solving

$$\min_{\mathrm{u}\in\mathbb{R}^{d_0}} \widehat{\mathbb{E}}_i(y_i - \mathrm{u}^\top \mathrm{x}_i)^2 + \nu\|\mathrm{u}\|^2_{\widehat{\mathrm{C}}}.$$

All the minimizers of the above problem are solutions to the system of linear equations $(1 + \frac{\nu}{n})\mathrm{X}\mathrm{X}^\top\mathrm{u} = \mathrm{X}\mathrm{y}$, where $\mathrm{X} = [\mathrm{x}_1, \ldots, \mathrm{x}_n] \in \mathbb{R}^{d_0\times n}, \mathrm{y} = [y_1; \ldots; y_n] \in \mathbb{R}^n$ are the design matrix and the response vector, respectively. In other words, the dropout regularizer manifests itself merely as a scaling of the parameters.

What we argue above may at first seem to contradict the results of Section 4.2 on matrix sensing, which is arguably an instance of regression with a two-layer linear network. Note though that casting matrix sensing in a factored form as a linear regression problem requires us to use a convolutional structure. This is easy to check since

$$\begin{aligned}\langle \mathrm{U}\mathrm{V}^\top, \mathrm{A}\rangle &= \langle \mathrm{vec}\left(\mathrm{U}^\top\right), \mathrm{vec}\left(\mathrm{V}^\top \mathrm{A}^\top\right)\rangle \\ &= \langle \mathrm{vec}\left(\mathrm{U}^\top\right), (\mathrm{I}_{d_2} \otimes \mathrm{V}^\top)\, \mathrm{vec}\left(\mathrm{A}^\top\right)\rangle,\end{aligned}$$

where $\otimes$ is the Kronecker product, and we used the fact that $\mathrm{vec}\,(\mathrm{AB}) = (\mathrm{I} \otimes \mathrm{A})\,\mathrm{vec}\,(\mathrm{B})$ for any pair of matrices A, B. The expression $(\mathrm{I} \otimes \mathrm{V}^\top)$ represents a fully connected convolutional layer with d_1 filters specified by columns of V. The convolutional structure in addition to dropout is what imparts the problem of matrix sensing the nuclear norm regularization. For nonlinear networks, however, a simple feed-forward structure suffices as we saw in Section 4.3.

4.5 Proofs

4.5.1 Matrix Sensing

The following Proposition gives the explicit regularizer due to dropout in matrix sensing.

Proposition 3 (Dropout regularizer in matrix sensing)**.** *The following holds for any* $p \in [0, 1)$*:*

$$\widehat{L}_{drop}(U, V) = \widehat{L}(U, V) + \lambda \widehat{R}(U, V), \tag{4.6}$$

where $\widehat{R}(U, V) = \sum_{i=1}^{d_1} \widehat{\mathbb{E}}_j (u_i^\top A^{(j)} v_i)^2$ *and* $\lambda = \frac{p}{1-p}$ *is the regularization parameter.*

Proof of Proposition 3. Similar statements and proofs can be found in several previous works [SHK+14, WM13, CHL+18, MAV18]. For completeness, we include a proof

here. The following equality follows from the definition of variance:

$$\mathbb{E}_{\mathrm{b}}[(y_i - \langle \mathrm{UBV}^\top, \mathrm{A}^{(i)} \rangle)^2] = \Big(\mathbb{E}_{\mathrm{b}}[y_i - \langle \mathrm{UBV}^\top, \mathrm{A}^{(i)} \rangle]\Big)^2 + \mathrm{Var}(y_i - \langle \mathrm{UBV}^\top, \mathrm{A}^{(i)} \rangle) \tag{4.7}$$

Recall that for a Bernoulli random variable B_{ii}, we have $\mathbb{E}[\mathrm{B}_{ii}] = 1$ and $\mathrm{Var}(\mathrm{B}_{ii}) = \frac{p}{1-p}$. Thus, the first term on right hand side is equal to $(y_i - \langle \mathrm{UV}^\top, \mathrm{A}^{(i)} \rangle)^2$. For the second term we have

$$\begin{aligned}
\mathrm{Var}(y_i - \langle \mathrm{UBV}^\top, \mathrm{A}^{(i)} \rangle) &= \mathrm{Var}(\sum_{j=1}^{d_1} \mathrm{B}_{jj} \mathrm{u}_j^\top \mathrm{A}^{(i)} \mathrm{v}_j) \\
&= \sum_{j=1}^{d_1} (\mathrm{u}_j^\top \mathrm{A}^{(i)} \mathrm{v}_j)^2 \, \mathrm{Var}(\mathrm{B}_{jj}) \\
&= \frac{p}{1-p} \sum_{j=1}^{d_1} (\mathrm{u}_j^\top \mathrm{A}^{(i)} \mathrm{v}_j)^2
\end{aligned}$$

Plugging the above into Equation (4.7) and averaging over samples we get

$$\begin{aligned}
\widehat{L}_{\mathrm{drop}}(\mathrm{U}, \mathrm{V}) &= \widehat{\mathbb{E}}_i \mathbb{E}_{\mathrm{b}}[(y_i - \langle \mathrm{UBV}^\top, \mathrm{A}^{(i)} \rangle)^2] \\
&= \widehat{\mathbb{E}}_i (y_i - \langle \mathrm{UV}^\top, \mathrm{A}^{(i)} \rangle)^2 + \widehat{\mathbb{E}}_i \frac{p}{1-p} \sum_{j=1}^{d_1} (\mathrm{u}_j^\top \mathrm{A}^{(i)} \mathrm{v}_j)^2 \\
&= \widehat{L}(\mathrm{U}, \mathrm{V}) + \frac{p}{1-p} \widehat{R}(\mathrm{U}, \mathrm{V}).
\end{aligned}$$

which completes the proof. □

The following Proposition gives the induced regularizer in matrix completion.

Proposition 4. *[Induced regularizer] For $j \in [n]$, let $A^{(j)}$ be an indicator matrix whose (i,k)-th element is selected randomly with probability $p(i)q(k)$, where $p(i)$ and $q(k)$ denote the probability of choosing the i-th row and the k-th column. Then* $\Theta(M) = \frac{1}{d_1} \| \operatorname{diag}(\sqrt{p})\, UV^\top \operatorname{diag}(\sqrt{q}) \|_*^2$.

Proof of Proposition 4. For any pair of factors (U, V) it holds that

$$\begin{aligned}
R(\mathrm{U}, \mathrm{V}) &= \sum_{i=1}^{d_1} \mathbb{E}(\mathrm{u}_i^\top \mathrm{A} \mathrm{v}_i)^2 = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \sum_{k=1}^{d_0} p(j)q(k) (\mathrm{u}_i^\top \mathrm{e}_j \mathrm{e}_k^\top \mathrm{v}_i)^2 \\
&= \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \sum_{k=1}^{d_0} p(j)q(k) \mathrm{U}(j,i)^2 \mathrm{V}(k,i)^2 = \sum_{i=1}^{d_1} \| \operatorname{diag}(\sqrt{p}) \mathrm{u}_i \|^2 \| \operatorname{diag}(\sqrt{q}) \mathrm{v}_i \|^2
\end{aligned}$$

We can now lower bound the right hand side above as follows:

$$\begin{aligned} R(\mathrm{U},\mathrm{V}) &\geq \frac{1}{d_1}\left(\sum_{i=1}^{d_1} \|\operatorname{diag}(\sqrt{p})\mathrm{u}_i\|\|\operatorname{diag}(\sqrt{q})\mathrm{v}_i\|\right)^2 \\ &= \frac{1}{d_1}\left(\sum_{i=1}^{d_1} \|\operatorname{diag}(\sqrt{p})\mathrm{u}_i\mathrm{v}_i^\top \operatorname{diag}(\sqrt{q})\|_*\right)^2 \\ &\geq \frac{1}{d_1}\left(\|\operatorname{diag}(\sqrt{p})\sum_{i=1}^{d_1}\mathrm{u}_i\mathrm{v}_i^\top \operatorname{diag}(\sqrt{q})\|_*\right)^2 = \frac{1}{d_1}\|\operatorname{diag}(\sqrt{p})\mathrm{U}\mathrm{V}^\top \operatorname{diag}(\sqrt{q})\|_*^2 \end{aligned}$$

where the first inequality is due to Cauchy-Schwartz and the second inequality follows from the triangle inequality. The equality right after the first inequality follows from the fact that for any two vectors a,b, $\|\mathrm{a}\mathrm{b}^\top\|_* = \|\mathrm{a}\mathrm{b}^\top\| = \|\mathrm{a}\|\|\mathrm{b}\|$. Since the inequalities hold for any U,V, it implies that

$$\Theta(\mathrm{U}\mathrm{V}^\top) \geq \frac{1}{d_1}\|\operatorname{diag}(\sqrt{p})\mathrm{U}\mathrm{V}^\top \operatorname{diag}(\sqrt{q})\|_*^2.$$

Applying Theorem 4 on $(\operatorname{diag}(\sqrt{p})\mathrm{U}, \operatorname{diag}(\sqrt{p})\mathrm{V})$, there exist a rotation matrix Q such that

$$\|\operatorname{diag}(\sqrt{p})\mathrm{U}\mathrm{q}_i\|\|\operatorname{diag}(\sqrt{q})\mathrm{V}\mathrm{q}_i\| = \frac{1}{d_1}\|\operatorname{diag}(\sqrt{p})\mathrm{U}\mathrm{V}^\top \operatorname{diag}(\sqrt{q})\|_*$$

We evaluate the expected dropout regularizer at UQ,VQ:

$$\begin{aligned} R(\mathrm{UQ},\mathrm{VQ}) &= \sum_{i=1}^{d_1}\|\operatorname{diag}(\sqrt{p})\mathrm{U}\mathrm{q}_i\|^2\|\operatorname{diag}(\sqrt{q})\mathrm{V}\mathrm{q}_i\|^2 \\ &= \sum_{i=1}^{d_1}\frac{1}{d_1^2}\|\operatorname{diag}(\sqrt{p})\mathrm{U}\mathrm{V}^\top \operatorname{diag}(\sqrt{q})\|_*^2 \\ &= \frac{1}{d_1}\|\operatorname{diag}(\sqrt{p})\mathrm{U}\mathrm{V}^\top \operatorname{diag}(\sqrt{q})\|_*^2 \leq \Theta(\mathrm{U}\mathrm{V}^\top) \end{aligned}$$

which completes the proof of the first part. □

We now provide a proof for Theorem 11.

Proof of Theorem 11. We use Theorem 19 to bound the population risk in terms of the Rademacher complexity of the target class. Define the class of predictors with weighted trace-norm bounded by $\sqrt{\alpha}$, i.e.

$$\mathcal{M}_\alpha = \{\mathrm{M} : \ \|\operatorname{diag}(\sqrt{\mathrm{p}})\mathrm{M}\operatorname{diag}(\sqrt{\mathrm{q}})\|_*^2 \leq \alpha\}.$$

In particular dropout empirical risk minimizers U, V belong to this class:

$$\| \operatorname{diag}(\sqrt{p}) \mathrm{U}\mathrm{V}^\top \operatorname{diag}(\sqrt{q}) \|_*^2 = d_1 \Theta(\mathrm{U}\mathrm{V}^\top) \leq d_1 R(\mathrm{U}, \mathrm{V}) \leq \alpha$$

where the first inequality holds by definition of the induced regularizer, and the second inequality follows from the assumption of the theorem. Since g is a contraction, by Talagrand's lemma and Theorem 20, we have that $\mathfrak{R}_n(g \circ \mathcal{M}_\alpha) \leq \mathfrak{R}_n(\mathcal{M}_\alpha) \leq \sqrt{\frac{\alpha d_2 \log(d_2)}{n}}$. To obtain the maximum deviation parameter M in Theorem 19, we note that the assumption $\|\mathrm{M}_*\| \leq 1$ implies that $|\mathrm{M}_*(i,j)| \leq 1$ for all i, j, so that $g(\mathrm{M}_*) = \mathrm{M}_*$. We have that:

$$\begin{aligned}
\max_{\mathrm{A}} |\langle \mathrm{M}_* - g(\mathrm{U}\mathrm{V}^\top), \mathrm{A} \rangle| &= \max_{i,j} |\langle \mathrm{M}_* - g(\mathrm{U}\mathrm{V}^\top), \mathrm{e}_i \mathrm{e}_j^\top \rangle| \\
&\leq \max_{i,j} |\mathrm{M}_*(i,j)| + \max_{i,j} |\langle \mathrm{U}\mathrm{V}^\top, \mathrm{e}_i \mathrm{e}_j^\top \rangle| \\
&\leq \|\mathrm{M}_*\| + 1 \leq 2
\end{aligned}$$

Let $L(g(\mathrm{U}\mathrm{V}^\top)) := \mathbb{E}(y - \langle g(\mathrm{U}\mathrm{V}^\top), \mathrm{A} \rangle)^2$ and $\widehat{L}(g(\mathrm{U}\mathrm{V}^\top)) := \widehat{\mathbb{E}}_i(y_i - \langle g(\mathrm{U}\mathrm{V}^\top), \mathrm{A}^{(i)} \rangle)^2$ denote the *true risk* and the *empirical risk* of $g(\mathrm{U}\mathrm{V}^\top)$, respectively. Plugging the above results in Theorem 19, we get

$$\begin{aligned}
L(g(\mathrm{U}, \mathrm{V})) &\leq \widehat{L}(g(\mathrm{U}, \mathrm{V})) + 8\mathfrak{R}_n(g \circ \mathcal{M}_\alpha) + 4\sqrt{\frac{\log(2/\delta)}{2n}} \\
&\leq \widehat{L}(\mathrm{U}, \mathrm{V}) + 8\sqrt{\frac{\alpha d_2 \log(d_2)}{n}} + 4\sqrt{\frac{\log(2/\delta)}{2n}} \\
&\leq \widehat{L}(\mathrm{U}, \mathrm{V}) + 8\sqrt{\frac{2\alpha d_2 \log(d_2) + \frac{1}{4}\log(2/\delta)}{n}}
\end{aligned}$$

where the second inequality holds since $\widehat{L}(g(\mathrm{U}, \mathrm{V})) \leq \widehat{L}(\mathrm{U}, \mathrm{V})$. □

4.5.2 Non-linear Neural Networks

We begin this section by giving the dropout regularizer in deep regression.

Proposition 5 (Dropout regularizer in deep regression)**.**

$$\widehat{L}_{drop}(\mathrm{w}) = \widehat{L}(\mathrm{w}) + \widehat{R}(\mathrm{w}), \quad \textit{where} \quad \widehat{R}(\mathrm{w}) = \lambda \sum_{j=1}^{d_1} \|u_j\|^2 \widehat{a}_j^2.$$

where $\widehat{a}_j = \sqrt{\widehat{\mathbb{E}}_i a_j(\mathrm{x}_i)^2}$ *and* $\lambda = \frac{p}{1-p}$ *is the regularization parameter.*

Proof of Proposition 5. Similar statements and proofs can be found in several previous works [SHK+14, WM13, CHL+18, MAV18]. Here we include a proof for completeness. Recall that $\mathbb{E}[\mathrm{B}_{ii}] = 1$ and $\mathrm{Var}(\mathrm{B}_{ii}) = \frac{p}{1-p}$. Conditioned on x, y in the current mini-batch, we have that:

$$\mathbb{E}_{\mathrm{B}}\|\mathrm{y} - \mathrm{U}^\top \mathrm{B}a(\mathrm{x})\|^2 = \sum_{i=1}^{d_2} \Big(\mathbb{E}_{\mathrm{B}}[y_i - \mathrm{u}_i^\top \mathrm{B}a(\mathrm{x})]\Big)^2 + \sum_{i=1}^{d_2} \mathrm{Var}(y_i - \mathrm{u}_i^\top \mathrm{B}a(\mathrm{x}))$$

Since $\mathbb{E}[\mathrm{B}] = \mathrm{I}$, the first term on right hand side is equal to $\|\mathrm{y} - \mathrm{U}^\top a(\mathrm{x})\|^2$. For the second term we have

$$\begin{aligned}
\sum_{i=1}^{d_2} \mathrm{Var}(y_i - \mathrm{u}_i^\top \mathrm{B}a(\mathrm{x})) &= \sum_{i=1}^{d_2} \mathrm{Var}(\sum_{j=1}^{d_1} \mathrm{U}_{j,i}\mathrm{B}_{jj}a_j(\mathrm{x})) \\
&= \sum_{i=1}^{d_2}\sum_{j=1}^{d_1} (\mathrm{U}_{j,i}a_j(\mathrm{x}))^2 \,\mathrm{Var}(\mathrm{B}_{jj}) \\
&= \frac{p}{1-p}\sum_{j=1}^{d_1} \|\mathrm{u}_j\|^2 a_j(\mathrm{x})^2
\end{aligned}$$

Thus, conditioned on the sample (x, y), we have that

$$\mathbb{E}_{\mathrm{B}}[\|\mathrm{y} - \mathrm{U}^\top \mathrm{B}a(\mathrm{x})\|^2] = \|\mathrm{y} - \mathrm{U}^\top a(\mathrm{x})\|^2 + \frac{p}{1-p}\sum_{j=1}^{d_1} \|\mathrm{u}_j\|^2 a_j(\mathrm{x})^2$$

Now taking the empirical average with respect to x, y, we get

$$\widehat{L}_{\mathrm{drop}}(\mathrm{w}) = \widehat{L}(\mathrm{w}) + \frac{p}{1-p}\sum_{j=1}^{d_1} \|\mathrm{u}_j\|^2 \widehat{a}_j^2 = \widehat{L}(\mathrm{w}) + \widehat{R}(\mathrm{w})$$

which completes the proof. □

We then focus on two layer networks and characterize the regularizer when the distribution is symmetric and isotropic.

Proposition 6. *Consider a two layer neural network* $f_{\mathrm{w}}(\cdot)$ *with ReLU activation functions in the hidden layer. Furthermore, assume that the marginal input distribution*

$\mathbb{P}_{\mathcal{X}}(\mathrm{x})$ *is symmetric and isotropic, i.e.,* $\mathbb{P}_{\mathcal{X}}(\mathrm{x}) = \mathbb{P}_{\mathcal{X}}(-\mathrm{x})$ *and* $\mathbb{E}[\mathrm{x}\mathrm{x}^\top] = I$*. Then the following holds for the expected explicit regularizer due to dropout:*

$$R(\mathrm{w}) := \mathbb{E}[\widehat{R}(\mathrm{w})] = \frac{\lambda}{2} \sum_{i_0,i_1,i_2=1}^{d_0,d_1,d_2} U(i_1,i_2)^2\, V(i_1,i_0)^2, \tag{4.8}$$

Proof of Proposition 6. Using Proposition 5, we have that:

$$R(\mathrm{w}) = \mathbb{E}[\widehat{R}(\mathrm{w})] = \lambda \sum_{j=1}^{d_1} \|\mathrm{u}_j\|^2 \mathbb{E}[\sigma(\mathrm{V}(j,:)^\top \mathrm{x})^2]$$

It remains to calculate the quantity $\mathbb{E}_{\mathrm{x}}[\sigma(\mathrm{V}(j,:)^\top \mathrm{x})^2]$. By symmetry assumption, we have that $\mathbb{P}_{\mathcal{X}}(\mathrm{x}) = \mathbb{P}_{\mathcal{X}}(-\mathrm{x})$. As a result, for any $\mathrm{v} \in \mathbb{R}^{d_0}$, we have that $\mathbb{P}(\mathrm{v}^\top \mathrm{x}) = \mathbb{P}(-\mathrm{v}^\top \mathrm{x})$ as well. That is, the random variable $z_j := \mathrm{W}_1(j,:)^\top \mathrm{x}$ is also symmetric about the origin. It is easy to see that $\mathbb{E}_z[\sigma(z)^2] = \frac{1}{2}\mathbb{E}_z[z^2]$.

$$\begin{aligned}
\mathbb{E}_z[\sigma(z)^2] &= \int_{-\infty}^{\infty} \sigma(z)^2 d\mu(z) = \int_0^{\infty} \sigma(z)^2 d\mu(z) \\
&= \int_0^{\infty} z^2 d\mu(z) = \frac{1}{2}\int_{\infty}^{\infty} z^2 d\mu(z) = \frac{1}{2}\mathbb{E}_z[z^2].
\end{aligned}$$

Plugging back the above identity in the expression of $R(\mathrm{w})$, we get that

$$R(\mathrm{w}) = \lambda \sum_{j=1}^{d_1} \|\mathrm{u}_j\|^2 \mathbb{E}[(\mathrm{V}(j,:)^\top \mathrm{x})^2] = \frac{\lambda}{2} \sum_{j=1}^{d_1} \|\mathrm{u}_j\|^2 \|\mathrm{V}(j,:)\|^2$$

where the second equality follows from the assumption that the distribution is isotropic.

□

Next, we define some function classes that will be used frequently in the proofs.

Definition 6. *For any closed subset* $[a,b] \subset \mathbb{R}$*, let* $\Pi_{[a,b]}(y) := \max\{a, \min\{b, y\}\}$*. For* $z := (\mathrm{x}, y)$ *and* $f : \mathcal{X} \to \mathcal{Y}$*, define the squared loss* $\ell_2(f,z) := (1 - yf(\mathrm{x}))^2$*. For a given value* $\alpha > 0$*, consider the following classes*

$$\begin{aligned}
\mathcal{W}_\alpha &:= \{\mathrm{w} = (u, V) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_0 \times d_1},\ \sum_{i=1}^{d_1} |u_i| \sqrt{\mathbb{E}\sigma(v_i^\top \mathrm{x})^2} \le \alpha\} \\
\mathcal{F}_\alpha &:= \{f_{\mathrm{w}} : \mathrm{x} \mapsto u^\top \sigma(V^\top \mathrm{x}),\ \mathrm{w} \in \mathcal{W}_\alpha\}, \\
\mathcal{G}_\alpha &:= \Pi_{[-1,1]} \circ \mathcal{F}_\alpha = \{g_{\mathrm{w}} = \Pi_{[-1,-1]} \circ f_{\mathrm{w}},\ f_{\mathrm{w}} \in \mathcal{F}_\alpha\} \\
\mathcal{L}_\alpha &:= \{\ell_2 : (g_{\mathrm{w}}, z) \mapsto (y - g_{\mathrm{w}}(\mathrm{x}))^2,\ g_{\mathrm{w}} \in \mathcal{G}_\alpha\}
\end{aligned}$$

Lemma 10. *Let $\mathcal{W}_\alpha, \mathcal{F}_\alpha, \mathcal{G}_\alpha, \mathcal{L}_\alpha$ be as defined in Definition 6. Then the following holds true:*

1. $\mathfrak{R}_{\mathcal{S}}(\mathcal{G}_\alpha) \le \mathfrak{R}_{\mathcal{S}}(\mathcal{F}_\alpha)$.

2. *If $\mathcal{Y} = \{-1, +1\}$ (binary classification), then it holds that $\mathfrak{R}_{\mathcal{S}}(\mathcal{L}_\alpha) \le 2\mathfrak{R}_{\mathcal{S}}(\mathcal{G}_\alpha)$.*

Proof. Since $\Pi_{[-1,-1]}(\cdot)$ is 1-Lipschitz, by Talagrand's contraction lemma, we have that $\mathfrak{R}_{\mathcal{S}}(\mathcal{G}_\alpha) \le \mathfrak{R}_{\mathcal{S}}(\mathcal{F}_\alpha)$. The second claim follows from

$$\begin{aligned}
\mathfrak{R}_{\mathcal{S}}(\mathcal{L}_\alpha) &= \mathbb{E}_\zeta \sup_{\mathrm{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^n \zeta_i (y_i - g_{\mathrm{w}}(\mathrm{x}_i))^2 \\
&= \mathbb{E}_\zeta \sup_{\mathrm{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^n \zeta_i (1 - y_i g_{\mathrm{w}}(\mathrm{x}_i))^2 \qquad (y_i \in \{-1,+1\}) \\
&\le 2\mathbb{E}_\zeta \sup_{\mathrm{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^n \zeta_i y_i g_{\mathrm{w}}(\mathrm{x}_i) \\
&= 2\mathbb{E}_\zeta \sup_{\mathrm{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^n \zeta_i g_{\mathrm{w}}(\mathrm{x}_i) = 2\mathfrak{R}_{\mathcal{S}}(\mathcal{G}_\alpha)
\end{aligned}$$

where the first inequality follows from Talagrand's contraction lemma due to the fact that $h(z) = (1-z)^2$ is 2-Lipschitz for $z \in [-1, 1]$, and the penultimate holds true since for any fixed $(y_i)_{i=1}^n \in \{-1,+1\}^n$, the distribution of $(\zeta_1 y_1, \ldots, \zeta_n y_n)$ is the same as that of $(\zeta_1, \ldots, \zeta_n)$. □

We now prove the Rademacher complexity upper bound in Theorem 12.

Proof of Theorem 12. For any $j \in [h]$, let $a_j^2 := \mathbb{E}[\sigma(\mathrm{v}_j^\top \mathrm{x})^2]$ denote the average squared activation of the j-th node with respect to the input distribution. Given n i.i.d. samples $\mathcal{S} = \{\mathrm{x}_1, \cdots, \mathrm{x}_n\}$, the empirical Rademahcer complexity is bounded as follows:

$$\begin{aligned}
\mathfrak{R}_{\mathcal{S}}(\mathcal{F}_\alpha) &= \mathbb{E}_\zeta \sup_{f_{\{\mathrm{u},\mathrm{V}\}}\in\mathcal{F}_\alpha} \frac{1}{n}\sum_{j=1}^h u_j a_j \sum_{i=1}^n \zeta_i \frac{\sigma(\mathrm{v}_j^\top \mathrm{x}_i)}{a_j} \\
&\le \mathbb{E}_\zeta \sup_{f_{\{\mathrm{u},\mathrm{V}\}}\in\mathcal{F}_\alpha} \frac{1}{n}\sum_{j=1}^h |u_j a_j| \, |\sum_{i=1}^n \zeta_i \frac{\sigma(\mathrm{v}_j^\top \mathrm{x}_i)}{a_j}| \\
&\le \mathbb{E}_\zeta \left[\left(\sup_{f_{\{\mathrm{u},\mathrm{V}\}}\in\mathcal{F}_\alpha} \sum_{j=1}^h |u_j a_j| \right) \left(\sup_{\mathrm{V}} \max_{j\in[h]} |\frac{1}{n}\sum_{i=1}^n \zeta_i \frac{\sigma(\mathrm{v}_j^\top \mathrm{x}_i)}{a_j}| \right) \right]
\end{aligned}$$

where we used the fact that the supremum of product of positive functions is upper-bounded by the product of the supremums. By definition of $\mathcal{F}_\alpha$, the first term on the right hand side is bounded by α. To bound the second term in the right hand side, we note that the maximum over rows of $\mathrm{V}^\top$ can be absorbed into the supremum.

$$\begin{aligned}
\frac{1}{n}\mathbb{E}_\zeta \sup_{\mathrm{v}} |\sum_{i=1}^n \zeta_i \frac{\sigma(\mathrm{v}^\top \mathrm{x}_i)}{\sqrt{\mathbb{E}[\sigma(\mathrm{v}^\top \mathrm{x})^2]}}| &= \frac{1}{n}\mathbb{E}_\zeta \sup_{\mathbb{E}[\sigma(\mathrm{v}^\top \mathrm{x})^2]\le 1} |\sum_{i=1}^n \zeta_i \sigma(\mathrm{v}^\top \mathrm{x}_i)| \\
&\le \frac{2}{n}\mathbb{E}_\zeta \sup_{\mathbb{E}[\sigma(\mathrm{v}^\top \mathrm{x})^2]\le 1} \sum_{i=1}^n \zeta_i \sigma(\mathrm{v}^\top \mathrm{x}_i) \\
&\le \frac{2}{n}\mathbb{E}_\zeta \sup_{\beta\mathbb{E}(\mathrm{v}^\top \mathrm{x})^2\le 1} \sum_{i=1}^n \zeta_i \sigma(\mathrm{v}^\top \mathrm{x}_i) \qquad (\beta\text{-retentiveness})
\end{aligned}$$

Let $\mathrm{C}^\dagger$ be the pseudo-inverse of C. We perform the following change the variable: $\mathrm{w} \leftarrow \mathrm{C}^{-\dagger/2}\mathrm{v}$.

$$\begin{aligned}
\text{R.H.S.} &\le \frac{2}{n}\mathbb{E}_\zeta \sup_{\mathbb{E}[(\mathrm{w}^\top \mathrm{C}^{\dagger/2}\mathrm{x})^2]\le 1/\beta} \sum_{i=1}^n \zeta_i \mathrm{w}^\top \mathrm{C}^{\dagger/2}\mathrm{x}_i \\
&= \frac{2}{n}\mathbb{E}_\zeta \sup_{\|\mathrm{w}\|^2\le 1/\beta} \langle \mathrm{w}, \sum_{i=1}^n \zeta_i \mathrm{C}^{\dagger/2}\mathrm{x}_i\rangle \\
&= \frac{2}{n\sqrt{\beta}}\mathbb{E}_\zeta \|\sum_{i=1}^n \zeta_i \mathrm{C}^{\dagger/2}\mathrm{x}_i\| \\
&\le \frac{2}{n\sqrt{\beta}}\sqrt{\mathbb{E}_\zeta \|\sum_{i=1}^n \zeta_i \mathrm{C}^{\dagger/2}\mathrm{x}_i\|^2} = \frac{2}{n\sqrt{\beta}}\sqrt{\sum_{i=1}^n \mathrm{x}_i^\top \mathrm{C}^\dagger \mathrm{x}_i}
\end{aligned}$$

where the last inequality holds due to Jensen's inequality. To bound the expected Rademacher complexity, we take the expected value of both sides with respect to sample $\mathcal{S}$, which gives the following:

$$\mathfrak{R}_n(\mathcal{F}_\alpha) = \mathbb{E}_\mathrm{x}[\mathfrak{R}_\mathcal{S}(\mathcal{F}_\alpha)] \le \frac{2}{n\sqrt{\beta}}\mathbb{E}_\mathcal{S}\sqrt{\sum_{i=1}^n \mathrm{x}_i^\top \mathrm{C}^\dagger \mathrm{x}_i} \le \frac{2}{n\sqrt{\beta}}\sqrt{\sum_{i=1}^n \mathbb{E}_{\mathrm{x}_i}[\mathrm{x}_i^\top \mathrm{C}^\dagger \mathrm{x}_i]},$$

where the last inequality holds again due to Jensen's inequality. Finally, we have that $\mathbb{E}_{\mathrm{x}_i}\mathrm{x}_i^\top \mathrm{C}^\dagger \mathrm{x}_i = \mathbb{E}_{\mathrm{x}_i}\langle \mathrm{x}_i\mathrm{x}_i^\top, \mathrm{C}^\dagger\rangle = \langle \mathrm{C}, \mathrm{C}^\dagger\rangle = \mathrm{Rank}(\mathrm{C})$, which completes the proof of the Theorem. □

Next, we give the Rademacher complexity lower bound in Theorem 13.

Proof of Theorem 13. For simplicity, assume that the width of the hidden layer is even. Consider the linear function class:

$$\mathcal{G}_r := \{g_{\mathrm{w}} : \mathrm{x} \mapsto \mathrm{w}^\top \mathrm{x},\ \mathbb{E}(\mathrm{w}^\top \mathrm{x})^2 \le d_1 r/2\}.$$

Recall that $\mathcal{H}_r := \{h_{\mathrm{w}} : \mathrm{x} \mapsto \mathrm{u}^\top \sigma(\mathrm{V}^\top \mathrm{x}),\ R(\mathrm{u}, \mathrm{V}) \le r\}$. First, we argue that $\mathcal{G}_r \subset \mathcal{H}_r$. Let $g_{\mathrm{w}} \in \mathcal{G}_r$; we show that there exist u, V such that $g_{\mathrm{w}} = f_{\mathrm{u},\mathrm{V}}$ and $f_{\mathrm{u},\mathrm{V}} \in \mathcal{H}_r$. Define $\mathrm{u} := \frac{2}{d_1}[1; -1; \cdots 1; -1] \in \mathbb{R}^{d_1}$, and let $\mathrm{V} = \mathrm{w}(\mathrm{e}_1 - \mathrm{e}_2 + \mathrm{e}_3 - \mathrm{e}_4 + \cdots + \mathrm{e}_{d_1 - 1} - \mathrm{e}_{d_1})^\top$, where $\mathrm{e}_i \in \mathbb{R}^{d_1}$ is the i-th standard basis vector. It's easy to see that

$$\begin{aligned}
f_{\mathrm{u},\mathrm{V}}(\mathrm{x}) = \mathrm{u}^\top \sigma(\mathrm{V}^\top \mathrm{x}) &= \sum_{i=1}^{d_1} u_i \sigma(\mathrm{v}_i^\top \mathrm{x}) \\
&= \sum_{i=1}^{d_1} \frac{2}{d_1} (-1)^{i-1} \sigma(\mathrm{v}_i^\top \mathrm{x}) \\
&= \sum_{i=1}^{d_1/2} \frac{2}{d_1} (\sigma(\mathrm{v}_{2i-1}^\top \mathrm{x}) - \sigma(\mathrm{v}_{2i}^\top \mathrm{x})) \\
&= \sum_{i=1}^{d_1/2} \frac{2}{d_1} (\sigma(\mathrm{w}^\top \mathrm{x}) - \sigma(-\mathrm{w}^\top \mathrm{x})) = \mathrm{w}^\top \mathrm{x} = g_{\mathrm{w}}.
\end{aligned}$$

Furthermore, it holds for the explicit regularizer that

$$\begin{aligned}
R(\mathrm{u}, \mathrm{V}) = \sum_{i=1}^{d_1} u_i^2 \mathbb{E}\sigma(\mathrm{v}_i^\top \mathrm{x})^2 &= \sum_{i=1}^{d_1/2} \frac{4}{d_1^2} \left(\mathbb{E}\sigma(\mathrm{v}_{2i-1}^\top \mathrm{x})^2 + \mathbb{E}\sigma(\mathrm{v}_{2i}^\top \mathrm{x})^2\right) \\
&= \sum_{i=1}^{d_1/2} \frac{4}{d_1^2} \mathbb{E}[\sigma(\mathrm{w}^\top \mathrm{x})^2 + \sigma(-\mathrm{w}^\top \mathrm{x})^2] \\
&= \frac{2}{d_1} \mathbb{E}(\mathrm{w}^\top \mathrm{x})^2 \le r
\end{aligned}$$

Thus, we have that $\mathcal{G}_r \subset \mathcal{H}_r$, and the following inequalities follow.

$$\begin{aligned}
\mathfrak{R}_{\mathcal{S}}(\mathcal{H}_r) \geq \mathfrak{R}_{\mathcal{S}}(\mathcal{G}_r) &= \mathbb{E}_{\epsilon_i} \sup_{g_{\mathrm{w}} \in \mathcal{G}_r} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i g_{\mathrm{w}}(\mathrm{x}_i) \\
&= \mathbb{E}_{\epsilon_i} \sup_{\mathbb{E}(\mathrm{w}^\top \mathrm{x})^2 \leq d_1 r/2} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \mathrm{w}^\top \mathrm{x}_i \\
&= \mathbb{E}_{\epsilon_i} \sup_{\mathrm{w}^\top \mathrm{C} \mathrm{w} \leq d_1 r/2} \frac{1}{n} \langle \mathrm{w}, \sum_{i=1}^{n} \epsilon_i \mathrm{x}_i \rangle \\
&= \mathbb{E}_{\epsilon_i} \sup_{\|\mathrm{C}^{1/2}\mathrm{w}\|^2 \leq d_1 r/2} \frac{1}{n} \langle \mathrm{C}^{1/2}\mathrm{w}, \sum_{i=1}^{n} \epsilon_i \mathrm{C}^{-\dagger/2} \mathrm{x}_i \rangle \\
&= \frac{\sqrt{d_1 r}}{\sqrt{2}n} \mathbb{E}_{\epsilon_i} \| \sum_{i=1}^{n} \epsilon_i \mathrm{C}^{\dagger/2} \mathrm{x}_i \| \\
&\geq \frac{c\sqrt{d_1 r}}{\sqrt{2}n} \sqrt{\sum_{i=1}^{n} \|\mathrm{C}^{\dagger/2} \mathrm{x}_i\|^2} = \frac{c\sqrt{d_1 r} \|\mathrm{X}\|_{\mathrm{C}^\dagger}}{\sqrt{2}n}
\end{aligned}$$

where the last inequality follows from Khintchine-Kahane inequality in Lemma 22. □

Proof of Corollary 3. Recall that the input is jointly distributed as $(\mathrm{x}, y) \sim \mathcal{D}$. For $\mathcal{X} \subseteq \mathbb{R}_+^{d_0}$, let $\mathcal{X}' = \mathcal{X} \cup -\mathcal{X}$ be the *symmetrized input domain*. Let ζ be a Rademacher random variable. Denote the *symmetrized input* by $\mathrm{x}' = \zeta \mathrm{x}$, and the joint distribution of (x', y) by $\mathcal{D}'$. By construction, $\mathcal{D}'$ is centrally symmetric w.r.t. x', i.e., it holds for all $(\mathrm{x}, y) \in \mathcal{X} \times \mathcal{Y}$ that $\mathcal{D}'(\mathrm{x}, y) = \mathcal{D}'(-\mathrm{x}, y) = \frac{1}{2}\mathcal{D}(\mathrm{x}, y)$. As a result, population risk with respect to the original distribution $\mathcal{D}$ can be bounded in terms of the population risk with respect to the *symmetrized distribution* $\mathcal{D}'$ as follows:

$$\begin{aligned}
L_{\mathcal{D}}(f) &:= \mathbb{E}_{\mathcal{D}}[\ell(f(\mathrm{x}), y)] \\
&\leq \mathbb{E}_{\mathcal{D}}[\ell(f(\mathrm{x}), y) + \ell(f(-\mathrm{x}), y)] \\
&= 2\mathbb{E}_{\mathcal{D}}[\frac{1}{2}\ell(f(\mathrm{x}), y) + \frac{1}{2}\ell(f(-\mathrm{x}), y)] \\
&= 2\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\zeta}[\ell(f(\zeta \mathrm{x}), y) \mid \mathrm{x}, y] \\
&= 2\mathbb{E}_{\mathcal{D}'}[\ell(f(\mathrm{x}'), y)] = 2L_{\mathcal{D}'}(f)
\end{aligned} \tag{4.9}$$

Moreover, since $\mathcal{D}'$ is centrally symmetric, Assumption 1 holds with $\beta = \frac{1}{2}$. The proof of Corollary 3 follows by doubling the right hand side of inequalities in Corollary 2, and substituting $\beta = \frac{1}{2}$. □

	plain SGD		dropout			
width	last iterate	best iterate	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$
$d_1 = 30$	0.8041	0.7938	0.7805	0.785	0.7991	0.8186
$d_1 = 70$	0.8315	0.7897	0.7899	0.7771	0.7763	0.7833
$d_1 = 110$	0.8431	0.7873	0.7988	0.7813	0.7742	0.7743
$d_1 = 150$	0.8472	0.7858	0.8042	0.7852	0.7756	0.7722
$d_1 = 190$	0.8473	0.7844	0.8069	0.7879	0.7772	0.772

Table 4-I. MovieLens dataset: Test RMSE of plain SGD as well as the dropout algorithm with various dropout rates for various factorization sizes. The grey cells shows the best performance(s) in each row.

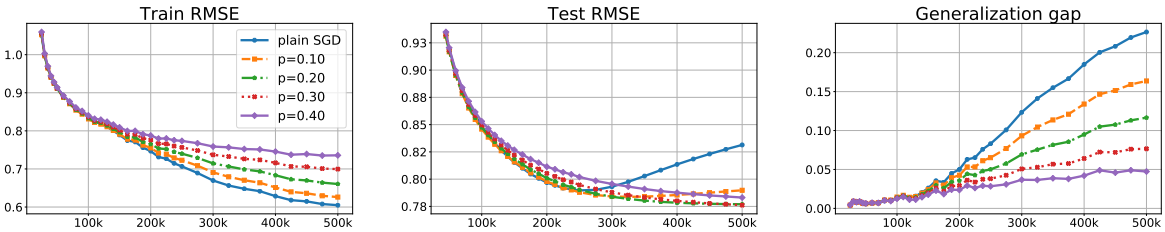

Figure 4-1. MovieLens dataset: training error (**left**), test error (**middle**), and generalization gap (**right**) for plain SGD and dropout with $p \in \{0.1, 0.2, 0.3, 0.4\}$ as a function of number of iterations; factorization size, $d_1 = 70$.

4.6 Experimental Results

In this section, we report our empirical findings on real world datasets. All results are averaged over 50 independent runs with random initialization.

Matrix Completion. We evaluate dropout on the MovieLens dataset [HK16], a publicly available collaborative filtering dataset that contains 10M ratings for 11K movies by 72K users of the online movie recommender service MovieLens. We initialize the factors using the standard He initialization scheme [HZRS15]. We train the model for 100 epochs over the training data, where we use a fixed learning rate of $\texttt{lr} = 1$, and a batch size of 2000. We report the results for plain SGD ($p = 0.0$) as well as the dropout algorithm with $p \in \{0.1, 0.2, 0.3, 0.4\}$.

Figure 4-1 shows the progress in terms of the training and test error as well as the gap between them as a function of the number of iterations for $d_1 = 70$. It can be seen that plain SGD is the fastest in minimizing the empirical risk. The dropout rate clearly determines the trade-off between the goodness of fit and the model complexity:

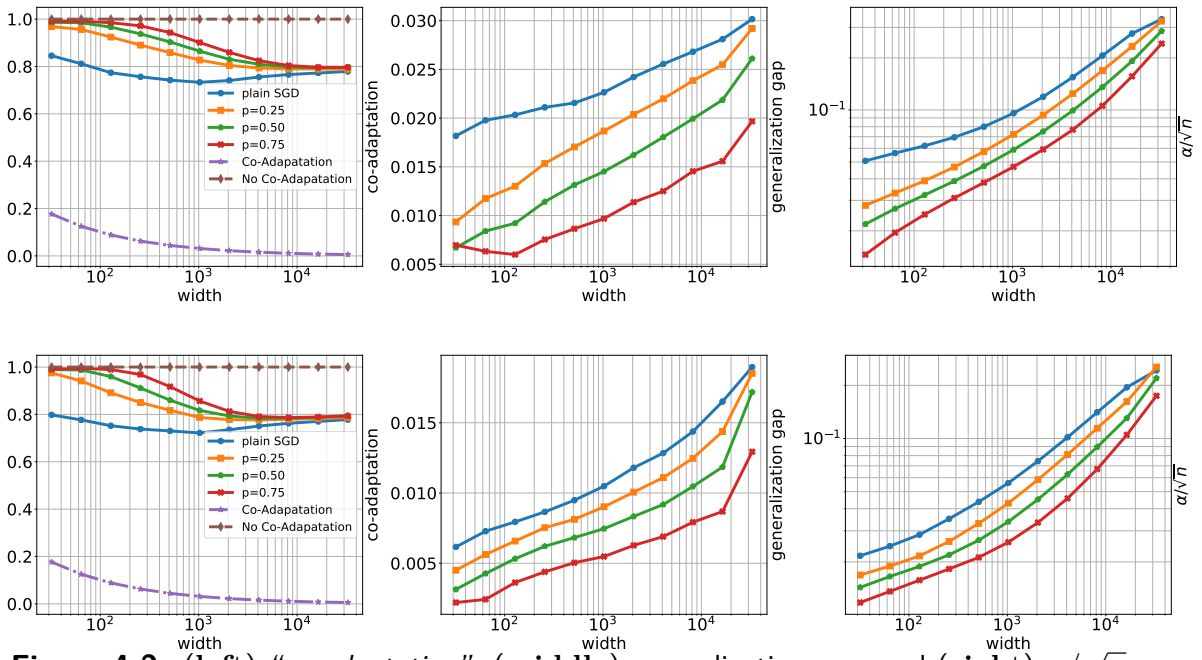

Figure 4-2. (**left**) *"co-adaptation"*; (**middle**) generalization gap; and (**right**) $\alpha/\sqrt{n}$ as a function of the width of networks trained with dropout on MNIST, with symmetrization (**top**) and without symmetrization (**bottom**). In left figure, the dashed brown and dotted purple lines represent minimal and maximal co-adaptations, respectively.

as the dropout rate p increases, the algorithm favors less complex solutions that suffer larger empirical error (left figure) but enjoy smaller generalization gap (right figure). The best trade-off here seems to be achieved by a moderate dropout rate of $p = 0.3$. We observe similar behaviour for different factorization sizes; please see the Appendix for additional plots with factorization sizes $d_1 \in \{30, 110, 150, 190\}$.

It is remarkable, how even in the "simple" problem of matrix completion, plain SGD lacks a proper inductive bias. As it is clearly depicted in the middle plot, without *explicit* regularization – in particular early stopping or dropout in this figure – SGD suffers from gross overfitting. We further illustrate this fact in Table 4-I, where we compare the test root-mean-squared-error (RMSE) of plain SGD with the dropout algorithm, for various factorization sizes. To show the superiority of dropout over SGD with early stopping, we give SGD the advantage of having access to the *test set* (and not a separate validation set), and report the best iterate in the third column.

Even with this impractical privilege, dropout performs significantly better (> 0.01 difference in test RMSE).

Neural Networks. We train 2-layer neural networks with and without dropout, on MNIST dataset of handwritten digits and Fashion MNIST dataset of Zalando's article images, each of which contains 60K training examples and 10K test examples, where each example is a 28×28 grayscale image, associated with a label from 10 classes. We extract two classes $\{4, 7\}$ and label them as $\{-1, +1\}$. We observe similar results across other choices of target classes. The learning rate in all experiments is set to $\texttt{lr} = 1e-3$. We train the models for 30 epochs over the training set. We run the experiments *both with and without symmetrization*. Here we only report the results with symmetrization, and on the MNIST dataset. We remark that *under the above experimental setting, trained networks achieve* 100% *training accuracy.*

For any node $i \in [d_1]$, define its *flow* as $\psi_i := |u_i|a_i$ (respectively $\psi_i := |u_i|a'_i$ for symmetrized data), which measures the overall contribution of a node to the output of the network. Co-adaptation occurs when a small subset of nodes dominate the overall function of the network. We argue that $\phi(\mathrm{w}) = \frac{\|\psi\|_1}{\sqrt{d_1}\|\psi\|_2}$ is a suitable measure of co-adaptation (or lack thereof) in a network parameterized by w. In case of high co-adaptation, only a few nodes have a high flow, which implies $\phi(\mathrm{w}) \approx \frac{1}{\sqrt{d_1}}$. At the other end of the spectrum, all nodes are equally active, in which case $\phi(\mathrm{w}) \approx 1$. Figure 4-2 (left) illustrates this measure as a function of the network width for several dropout rates $p \in \{0, 0.25, 0.5, 0.75\}$. In particular, we observe that a higher dropout rate corresponds to less co-adaptation. More interestingly, even plain SGD is *implicitly* biased towards networks with less co-adaptation. Moreover, for a fixed dropout rate, the regularization effect due to dropout decreases as we increase the width. Thus, it is natural to expect more co-adaptation as the network becomes wider, which is what we observe in the plots.

The generalization gap is plotted in Figure 4-2 (middle). As expected, increasing

dropout rate decreases the generalization gap. In our experiments, the generalization gap increases with the width of the network. The figure on the right shows the quantity $\alpha/\sqrt{n}$ that shows up in the Rademacher complexity bounds in Section 4.3. We note that, the bound on the Rademacher complexity is predictive of the generalization gap, in the sense that a smaller bound corresponds to a curve with smaller generalization gap.

4.7 Discussion

In this chapter, we studied the capacity control provided by dropout in matrix completion as well as two-layer neural networks. We gave generalization bounds that are solely in terms of the value of the dropout regularizer. In sharp contrast, in most of the prior work, dropout is analyzed in conjunction with additional norm-based capacity control. The generalization bounds presented in this chapter are data-dependent. In particular, we identify *retentiveness* as a simple distributional property that yields tight generalization bounds as evidenced by matching lower and upper bounds.

The focus here has been on *understanding* how the expected explicit regularizer alone – without any additional norm-bounds on the weights – can provide generalization. If one is interested in *predicting* the generalization gap, then one can estimate the (empirical) explicit regularizer on a held-out dataset, and appeal to simple concentration arguments, just as we do in our experiments.

Next, we list few natural research directions for future work.

Dropout Regularizer in more General Settings. In this chapter, as well as Chapters 2, and 3 of this thesis, we considered simpler linear models and shallow non-linear models; it would be interesting to extend these results to deep non-linear neural networks. Furthermore, these results were obtained for the particular choice of squared loss; it is important to understand dropout in networks trained with other

loss functions, especially those that are popular for various classification tasks.

Matrix Sensing: General Sampling Distributions. Our characterization of the induced regularizer due to dropout in the matrix sensing problem, and the corresponding generalization error bound, are under the assumption that the observations are drawn from a product distribution, i.e., row and column indices are selected independently. While we show that dropout induces a rich inductive bias even under this restricted distributional setting, it is important to analyze the induced regularizer in more general distributional settings.

ReLU Networks: Beyond β-retentivity. The generalization error bound presented in this chapter assumes that the distribution is β-retentive: for any hyperplane passing through the origin, both halfspaces contribute significantly to the second moment of the data. We also give a simple randomized symmetrization technique, with no additional computational overhead, which allows us to give bounds that are β-independent. Our empirical results confirm that this technique does not hurt the performance of the learned models. Regardless, it would be interesting to see if such an assumption can be avoided without any algorithmic tricks, like the symmetrization technique we introduced here, or show otherwise that it is required.

Chapter 5

Computational Guarantees for Dropout

In Chapter 2 and Chapter 3 of this dissertation, we analyzed the regularizer induced by dropout in deep regression. We showed that dropout in linear regression with deep linear networks induces a nuclear norm penalty on the learning objective. We therefore completely characterized the global optima of the resulting regularized objective – despite the non-convexity of the problem – and showed that dropout favors low-rank solutions. We also provided empirical evidence to verify this theoretical finding.

In Chapter 4, we turned our focus towards the learning theoretic aspects of dropout. We built on our analysis on Chapter 2 and gave explicit forms of the regularizers induced by dropout in two important learning problems: matrix sensing, and regression with two-layer ReLU neural networks. For each problem, we bounded the Rademacher complexity of the class of corresponding models with a *small* dropout regularizer, which enabled us to give precise generalization error bounds for dropout training in those problems.

Our results in previous chapters provide several rigorous theoretical explanations for the success of dropout in deep learning; however, we emphasize that so far in this dissertation, we have steered clear from the algorithmic and computational learning aspects of dropout. In fact, none of the prior work, before the current chapter of the

thesis, yields insights into the runtime of learning using dropout on non-linear neural networks. Here, we initiate a study into the iteration complexity of dropout training for achieving ϵ-suboptimality on true error – the misclassification error with respect to the underlying population – in two-layer neural networks with ReLU activations.

We leverage recent advances in the theory of deep learning in over-parameterized settings with extremely (or infinitely) wide networks [JGH18, LXS+19]. Analyzing two-layer ReLU networks in such a regime has led to a series of exciting results recently establishing that gradient descent (GD) or stochastic gradient descent (SGD) can successfully minimize the empirical error and the true error [LL18, DZPS19, Dan17, ZCZG18, AZLL19, SY19, ADH+19, CG19, OS20]. In a related line of research, several works attribute generalization in over-parametrized settings to the implicit inductive bias of optimization algorithms (through the geometry of local search methods) [NTSS17]. However, many real-world state-of-the-art systems employ various explicit regularizers, and there is growing evidence that implicit bias may be unable to explain generalization even in a simpler setting of stochastic convex optimization [DFKL20]. Here, we extend convergence guarantees and generalization bounds for GD-based methods with explicit regularization due to dropout. We show that the key insights from analysis of GD-based methods in over-parameterized settings carry over to dropout training.

We summarize the key contributions of this Chapter as follows.

1. We give precise non-asymptotic convergence rates for achieving ϵ-subotimality in the test error via dropout training in two-layer ReLU networks. This is the first of its kind result in the literature.

2. We show that dropout training implicitly compresses the network. In particular, we show that there exists a sub-network, i.e., one of the iterates of dropout training, that can generalize as well as any complete network.

3. Our results contributes to a growing body of work geared toward a theoretical understanding of GD-based methods for regularized risk minimization in over-parameterized settings.

The rest of this chapter is organized as follows. In Section 5.1, we survey the related work. In Section 5.2, we formally introduce the problem setup and dropout training, state the key assumptions, and introduce the notation. In Section 5.3, we give the main results of the paper. In Section 5.4, we present the proofs of our main results. We conclude this chapter by providing empirical evidence for our theoretical results in Section 5.5.

5.1 Related Work

Empirical success of dropout has inspired a series of works aimed at understanding its theoretical underpinnings. As we discussed in the previous chapters, most of these works have either focused on explaining the explicit regularization due to dropout in terms of *conventional* regularizers [SHK+14, BS13]; or bounding the generalization gap in dropout training, leveraging tools from uniform convergence [WFWL14, WZZ+13, ZW18, GZ16, MZGW18].

Despite the crucial insights provided by the previous art, there is not much known about the non-asymptotic convergence behaviour of dropout training in the literature. A very recent work by [SCS20] shows for deep neural networks with polynomially bounded activations with continuous derivatives, under squared loss, that the network weights converge to a stationary set of system of ODEs. In contrast, our results leverages over-parameterization in two-layer networks with non-differentiable ReLU activations, works with logistic loss, and establishes ϵ-suboptimality in the true misclassification error.

Our results are inspired by the recent advances in over-parameterized settings. A

large body of literature has focused on deriving optimization theoretic guarantees for (S)GD in this setting. In particular, [LL18, DZPS19] were among the first to provide convergence rates for empirical risk minimization using GD. Several subsequent works extended those results beyond two-layers, for smooth activation functions [DLL+18], and general activation functions [AZLS18, ZCZG18].

Learning theoretic aspects of GD-based methods have been studied for several important target concept classes. Under linear-separability assumption, via a compression scheme, [BGMSS18] showed that SGD can efficiently learn a two-layer ReLU network. [LL18] further showed that SGD enjoys small generalization error on two-layer ReLU networks if the data follows a well-separated mixture of distributions. [AZLL19] showed generalization error bounds for SGD in two- and three-layer networks with smooth activations where the concept class has fewer parameters. [ADH+19] proved data-dependent generalization error bounds based on the neural tangent kernel by analyzing the Rademacher complexity of the class of networks reachable by GD.

When the data distribution can be well-classified in the *random feature space* induced by the gradient of the network at initialization, [CG19] provide generalization guarantees for SGD in networks with arbitrary depth. [NS19] studied convergence of GD in two-layer networks with smooth activations, when the data distribution is further *separable* in the infinite-width limit of the random feature space. [JT19b] adopted the same margin assumption and improved the convergence rate as well as the over-parameterization size for non-smooth ReLU activations. Here, we generalize the margin assumption in [NS19] to take into account the randomness injected by dropout into the gradient of the network at initialization, or equivalently, the scale of the corresponding random feature. Our work is most closely related to and inspired by [JT19b]; however, we analyze dropout training as opposed to plain SGD, give generalization bounds in expectation, and show the compression benefits of dropout training.

We emphasize that all of the results above focus on (S)GD in absence of any explicit regularization. We summarize a few papers that study regularization in the over-parameterized setting. The work of [WLLM19] showed that even simple explicit regularizers such as weight decay can indeed provably improve the sample complexity of training using GD in the Neural Tangent Kernel (NTK) regime, appealing to a margin-maximization argument in homogeneous networks. We also note the recent works by [LSO19] and [HLY20], which studied the robustness of GD to noisy labels, with explicit regularization in forms of early stopping; and squared norm of the distance from initialization, respectively.

5.2 Poblem Setup

Let $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{\pm 1\}$ denote the input and label spaces, respectively. We assume that the data is jointly distributed according to an unknown distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$. Given T i.i.d. examples $\mathcal{S}_T = \{(\mathrm{x}_t, \mathrm{y}_t)\}_{t=1}^T \sim \mathcal{D}^T$, the goal of learning is to find a hypothesis $f(\cdot; \Theta) : \mathcal{X} \to \mathbb{R}$, parameterized by Θ, that has a small *misclassification error* $\mathcal{R}(\Theta) := \mathbb{P}\{yf(\mathrm{x}; \Theta) < 0\}$. Given a convex surrogate loss function $\ell : \mathbb{R} \to \mathbb{R}_{\geq 0}$, a common approach to the above learning problem is to solve the stochastic optimization problem $\min_\Theta L(\Theta) := \mathbb{E}_{\mathcal{D}}[\ell(yf(\mathrm{x}; \Theta))]$.

In this paper, we focus on logistic loss $\ell(z) = \log(1 + e^{-z})$, which is one of the most popular loss functions for classification tasks. We consider two-layer ReLU networks of width m, parameterized by the "weights" $\Theta = (\mathrm{W}, \mathrm{a}) \in \mathbb{R}^{m \times d} \times \mathbb{R}^m$, computing the function $f(\cdot; \Theta) : \mathrm{x} \mapsto \frac{1}{\sqrt{m}} \mathrm{a}^\top \sigma \mathrm{W}\mathrm{x}$. We initialize the network with $a_r \sim \mathrm{Unif}(\{+1, -1\})$ and $\mathrm{w}_{r,1} \sim \mathcal{N}(0, \mathrm{I})$, for all hidden nodes $r \in [m]$. We then fix the top layer weights and train the hidden layer W using the dropout algorithm. We denote the weight matrix at time t by W_t, and $\mathrm{w}_{r,t}$ represents its r-th column. For the sake of simplicity of the presentation, we drop non-trainable arguments from all functions, e.g., we use $f(\cdot; \mathrm{W})$ in lieu of $f(\cdot; \Theta)$.

Algorithm 4: Dropout in Two-Layer Networks

Input: data $\mathcal{S}_T = \{(\mathrm{x}_t, y_t)\}_{t=1}^T \sim \mathcal{D}^T$; Bernoulli masks $\mathcal{B}_T = \{\mathrm{B}_t\}_{t=1}^T$; dropout rate $1-q$; max-norm constraint parameter c; learning rate η

1: *initialize:* $\mathrm{w}_{r,1} \sim \mathcal{N}(0, \mathrm{I})$ and $a_r \sim \mathrm{Unif}(\{+1, -1\})$, $r \in [m]$
2: **for** $t = 1, \ldots, T-1$ **do**
3: *forward:* $g(\mathrm{W}_t; \mathrm{x}_t, \mathrm{B}_t) = \frac{1}{\sqrt{m}} \mathrm{a}^\top \mathrm{B}_t \sigma(\mathrm{W}_t \mathrm{x}_t)$
4: *backward:* $\nabla L_t(\mathrm{W}_t) = \nabla \ell(y_t g(\mathrm{W}_t; \mathrm{x}_t, \mathrm{B}_t) = \ell'(y_t g(\mathrm{W}_t; \mathrm{x}_t, \mathrm{B}_t)) \cdot y_t \nabla g(\mathrm{W}_t; \mathrm{x}_t, \mathrm{B}_t)$
5: *update:* $\mathrm{W}_{t+\frac{1}{2}} \leftarrow \mathrm{W}_t - \eta \nabla L_t(\mathrm{W}_t)$
6: *max-norm:* $\mathrm{W}_{t+1} \leftarrow \Pi_c(\mathrm{W}_{t+\frac{1}{2}})$
7: **end for**

Output: re-scale the weights as $\mathrm{W}_t \leftarrow q\mathrm{W}_t$

Let $\mathrm{B}_t \in \mathbb{R}^{m \times m}$, $t \in [T]$, be a random diagonal matrix with diagonal entries drawn independently and identically from a Bernoulli distribution with parameter q, i.e., $b_{r,t} \sim \mathrm{Bern}(q)$, where $b_{r,t}$ is the r-th diagonal entry of B_t. At the t-th iterate, dropout entails a SGD step on (the parameters of) the sub-network $g(\mathrm{W}; \mathrm{x}, \mathrm{B}_t) = \frac{1}{\sqrt{m}} \mathrm{a}^\top \mathrm{B}_t \sigma(\mathrm{W}\mathrm{x})$, yielding updates of the form $\mathrm{W}_{t+\frac{1}{2}} \leftarrow \mathrm{W}_t - \eta \nabla \ell(y_t g(\mathrm{W}_t; \mathrm{x}_t, \mathrm{B}_t))$. The iteration concludes with projecting the incoming weights – i.e. rows of $\mathrm{W}_{t+\frac{1}{2}}$ – onto a pre-specified Euclidean norm ball. We note that such max-norm constraints are standard in the practice of deep learning, and has been a staple to dropout training since it was proposed in [SHK+14][1]. Finally, at test time, the weights are multiplied by q so as to make sure that the output at test time is on par with the expected output at training time. The pseudo-code for dropout training is given in Algorithm 4[2].

Our analysis is motivated by recent developments in understanding the dynamics of (S)GD in the so-called *lazy regime*. Under certain initialization, learning rate, and network width requirements, these results show that the iterates of (S)GD tend to stay close to initialization; therefore, a first-order Taylor expansion of the t-th iterate around

[1] Quote from [SHK+14]: "One particular form of regularization was found to be especially useful for dropout— constraining the norm of the incoming weight vector at each hidden unit to be upper bounded by a fixed constant c"

[2] In a popular variant that is used in machine learning frameworks such as PyTorch, known as inverted dropout, (inverse) scaling is applied at the training time instead of the test time. The inverted dropout is equivalent to the method we study here, and can be analyzed in a similar manner.

initialization, i.e. $f(\mathrm{x};\mathrm{W}_t) \approx f(\mathrm{x};\mathrm{W}_1)+\langle \nabla f(\mathrm{x};\mathrm{W}_1), \mathrm{W}_t-\mathrm{W}_1\rangle$, can be used as a proxy to track the evolution of the network predictions [LL18, COB18, DZPS19, LXS+19]. In other words, training in lazy regime reduces to finding a linear predictor in the reproducing kernel Hilbert space (RKHS) associated with the gradient of the network at initialization, $\nabla f(\cdot;\mathrm{W}_1)$. In this work, following [NS19, JT19b], we assume that the data distribution is separable by a positive margin in the limiting RKHS:

Assumption 2 ((q,γ)-Margin)**.** *Let $z \sim \mathcal{N}(0, I_d)$ and $b \sim$ Bern(q) be a d-dimensional standard normal random vector, and a Bernoulli random variable with parameter q, respectively. There exists a* margin parameter $\gamma > 0$*, and a linear transformation $\psi : \mathbb{R}^d \to \mathbb{R}^d$ satisfying A) $\mathbb{E}_z[\|\psi(z)\|^2] < \infty$; B) $\|\psi(z)\|_2 \le 1$ for all $z \in \mathbb{R}^d$; and C) $\mathbb{E}_{z,b}[y\langle \psi(z), b\mathrm{x}\mathbb{I}[z^\top \mathrm{x} \ge 0]\rangle] \ge \gamma$ for almost all $(\mathrm{x}, y) \sim \mathcal{D}$.*

The above assumption provides an infinite-width extension to the separability of data in the RKHS induced by $\nabla g(\mathrm{W}_1;\cdot,\mathrm{B}_1)$. To see that, define $\mathrm{V} := [\mathrm{v}_1,\dots,\mathrm{v}_m]^\top \in \mathbb{R}^{m\times d}$, where $\mathrm{v}_r = \frac{1}{\sqrt{m}} a_r \psi(\mathrm{w}_{r,1})$ for all $r \in [m]$, satisfying $\|\mathrm{V}\|_F \le 1$. For any given point $(\mathrm{x}, y) \in \mathcal{X}\times\mathcal{Y}$, the margin attained by V is at least $y\langle \nabla g(\mathrm{W}_1;\mathrm{x},\mathrm{B}_1), \mathrm{V}\rangle = \frac{1}{m}\sum_{r=1}^m y\langle \psi(\mathrm{w}_{r,1}), b_{r,1}\mathrm{x}\mathbb{I}\{\mathrm{w}_{r,1}^\top \mathrm{x} > 0\}\rangle$, which is a finite-width approximation of the quantity $\mathbb{E}[y\langle \psi(\mathrm{z}), b\mathrm{x}\mathbb{I}\{\mathrm{z}^\top\mathrm{x} > 0\}\rangle]$ in Assumption 2.

We remark that when $q = 1$ (pure SGD – no dropout), with probability one it holds that $b = 1$, so that Assumption 2 boils down to that of [NS19] and [JT19b]. When $q < 1$, this assumption translates to a margin of γ/q on the *full* features $\nabla f(\cdot;\mathrm{W}_1)$, which is the appropriate scaling given that $\nabla f(\cdot;\mathrm{W}_1) = \frac{1}{q}\mathbb{E}_\mathrm{B}[\nabla g(\mathrm{W}_1;\cdot,\mathrm{B})]$. Alternatively, dropout training eventually outputs a network with weights scaled down as $q\mathrm{W}_t$, which (in expectation) corresponds to the shrinkage caused by the Bernoulli mask in $b\mathrm{x}\mathbb{I}\{\mathrm{z}^\top\mathrm{x} > 0\}$. Regardless, we note that *our analysis can be carried over even without this scaling*; however, new polynomial factors of $1/q$ will be introduced in the required width in our results in Section 5.3.

5.2.1 Notation

The r-th entry of vector y, and the r-th row of matrix Y, are denoted by y_i and y_i, respectively. For a sequence of matrices $\mathrm{W}_t, t \in \mathbb{N}$, the r-th row of the t-th matrix is denoted by $\mathrm{w}_{r,t}$. Let $\mathbb{I}$ denote the indicator of an event, i.e., $\mathbb{I}\{y \in \mathcal{Y}\}$ is one if $y \in \mathcal{Y}$, and zero otherwise. For any integer d, we represent the set $\{1, \ldots, d\}$ by $[d]$. For a matrix $\mathrm{W} \in \mathbb{R}^{m \times d}$, and a scalar $c > 0$, $\Pi_c(\mathrm{W})$ projects the rows of W onto the Euclidean ball of radius c with respect to the ℓ_2-norm.

For any $t \in [T]$ and any W, let $f_t(\mathrm{W}) := f(\mathrm{x}_t; \mathrm{W})$ denote the network output given input x_t, and let $g_t(\mathrm{W}) := g(\mathrm{W}; \mathrm{x}_t, \mathrm{B}_t)$ denote the corresponding output of the sub-network associated with the dropout pattern B_t. Let $L_t(\mathrm{W}) = \ell(y_t g_t(\mathrm{W}))$ and $Q_t(\mathrm{W}) = -\ell'(y_t g_t(\mathrm{W}))$ be the associated instantaneous loss and its negative derivative. The partial derivative of g_t with respect to the r-th hidden weight vector is given by $\frac{\partial g_t(\mathrm{W})}{\partial \mathrm{w}_r} = \frac{1}{\sqrt{m}} a_r b_{r,t} \mathbb{I}\{\mathrm{w}_r^\top \mathrm{x}_t \geq 0\} \mathrm{x}_t$. We denote the linearization of $g_t(\mathrm{W})$ based on features at time t by $g_t^{(k)}(\mathrm{W}) := \langle \nabla g_t(\mathrm{W}_k), \mathrm{W} \rangle$; and its corresponding instantaneous loss and its negative derivative by $L_t^{(k)}(\mathrm{W}) := \ell(y_t g_t^{(k)}(\mathrm{W}))$ and $Q_t^{(k)}(\mathrm{W}) := -\ell'(y_t g_t^{(k)}(\mathrm{W}))$, respectively. Q plays an important role in deriving generalization bounds for dropout sub-networks $g(\mathrm{W}_t; \mathrm{x}, \mathrm{B}_t)$; it has been recently used in [CG19, JT19b] for analyzing the convergence of SGD and bounding its generalization error.

We conclude this section by listing a few useful identities that are used throughout the paper. First, due to homogeneity of the ReLU, it holds that $g_t^{(t)}(\mathrm{W}_t) = \langle \nabla g_t(\mathrm{W}_t), \mathrm{W}_t \rangle = g_t(\mathrm{W}_t)$. Moreover, the norm of the network gradient, and the norm of the the gradient of the instantaneous loss can be upper-bounded as $\|\nabla g_t(\mathrm{W})\|_F^2 = \sum_{r=1}^m \|\frac{\partial g_t(\mathrm{W})}{\partial \mathrm{w}_r}\|^2 \leq \frac{\|\mathrm{B}_t\|_F^2}{m} \leq 1$, and $\|\nabla L_t(\mathrm{W})\|_F = -\ell'(y_t g_t(\mathrm{W}))\|y_t \nabla g_t(\mathrm{W})\|_F \leq Q_t(\mathrm{W})$, respectively. Finally, the logistic loss satisfies $|\ell'(z)| \leq \ell(z)$, so that $Q_t(\mathrm{W}) \leq L_t(\mathrm{W})$.

5.3 Main Results

We begin with a simple observation that given the random initialization scheme in Algorithm 4, the ℓ_2-norm of the rows of W_1 are expected to be concentrated around $\sqrt{d}$. In fact, using Gaussian concentration inequality (Theorem 21 in the appendix), it holds with probability at least $1 - 1/m$, uniformly for all $r \in [m]$, that $\|\mathrm{w}_{r,1}\| \leq \sqrt{d} + 2\sqrt{\ln m}$. For the sake of the simplicity of the presentation, we assume that the event $\max_{r\in[m]} \|\mathrm{w}_{r,1}\| \leq 2\sqrt{\ln m}$ holds through the run of dropout training. Alternatively, we can re-initialize the weights until this condition is satisfied, or multiply the probability of success in our theorems by a factor of $1 - 1/m$.

Our first result establishes that the true misclassification error of dropout training vanishes as $\tilde{\mathcal{O}}(1/T)$.

Theorem 14 (Learning with Dropout)**.** *Let* $c = \sqrt{d} + \max\{\frac{1}{14\gamma^2}, 2\sqrt{\ln m}\} + 1$ *and* $\lambda := 5\gamma^{-1}\ln 2\eta T + \sqrt{44\gamma^{-2}\ln 24\eta c\sqrt{m}T^2}$*. Under Assumption 2, for any learning rate* $\eta \in (0, \ln 2]$ *and any network of width satisfying* $m \geq 2401\gamma^{-6}\lambda^2$*, with probability one over the randomization due to dropout, we have that*

$$\min_{t\in[T]} \mathbb{E}[\mathcal{R}(q\,W_t)] \leq \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\mathcal{R}(q\,W_t)] \leq \frac{4\lambda^2}{\eta T} = \mathcal{O}\left(\frac{\ln T^2 + \ln mdT}{T}\right),$$

where the expectation is with respect to the initialization and the training samples.

Theorem 14 shows that dropout successfully trains the complete network $f(\cdot;\mathrm{W}_t)$. Perhaps more interestingly, our next result shows that dropout successfully trains a potentially significantly narrower sub-network $g(\mathrm{W}_t;\cdot,\mathrm{B}_t)$. For this purpose, denote the misclassification error due to a network with weights W given a Bernoulli mask B as follows

$$\mathcal{R}(\mathrm{W};\mathrm{B}) := \mathbb{P}\{yg(\mathrm{W};\mathrm{x},\mathrm{B}) < 0\}.$$

Then the following result holds for the misclassification error of the iterates of dropout training.

Theorem 15 (Compression with Dropout)**.** *Under the setting of Theorem 14, with probability at least* $1-\delta$ *over initialization, the training data, and the randomization due to dropout, we have that*

$$\min_{t\in[T]} \mathcal{R}(\, W_t; B_t) \leq \frac{1}{T}\sum_{t=1}^{T} \mathcal{R}(W_t; B_t) \leq \frac{12\lambda^2}{\eta T} + \frac{6\ln 1/\delta}{T} = \mathcal{O}\left(\frac{\ln mT}{T}\right).$$

A few remarks are in order.

Theorem 14 gives a generalization error bound in expectation. A technical challenge here stems from the unboundedness of the logistic loss. In our analysis, the max-norm constraint in Algorithm 4 is essential to guarantee that the logistic loss remains bounded through the run of the algorithm, thereby controlling the loss of the iterates in expectation. However, akin to analysis of SGD in the lazy regime, the iterates of dropout training are not likely to leave a small proximity of the initialization whatsoever. Therefore, for the particular choice of c in the above theorems, the max-norm projections in Algorithm 4 will be virtually inactive for a typical run.

The expected width of the sub-networks in Theorem 15 is only qm. Using Hoeffding's inequality and a union bound argument, for any $\delta \in (0,1)$, with probability at least $1-\delta$, it holds for all $t \in [T]$ that $g(\mathrm{W}_t; \mathrm{x}, \mathrm{B}_t)$ has at most $qm + \sqrt{2m \ln T/\delta}$ active hidden neurons. That is, in a typical run of the dropout training, with high probability, there exists a sub-network of width $\approx qm + \tilde{\mathcal{O}}(\sqrt{m})$ whose generalization error is no larger than $\tilde{\mathcal{O}}(1/T)$. In Section 5.5, we further provide empirical evidence to verify this compression result. We note that dropout has long been considered as a means of network compression, improving post-hoc pruning [GZS+19], in Bayesian settings [MAV17], and in connection with the Lottery Ticket Hypothesis [FC19]. However, we are not aware of any theoretical result supporting that claim prior to our work.

Finally, the sub-optimality results in both Theorem 14 and Theorem 15 are agnostic to the dropout rate $1-q$. This is precisely due to the (q,γ)-Margin assumption: if it holds, then so does (q',γ)-Margin for any $q' \in [q,1]$. That is, these theorems hold for *any* dropout rate not exceeding $1-q$. Therefore, in light of the remark above, larger admissible dropout rates are preferable since they result in higher compression rates, while enjoying the same generalization error guarantees.

5.4 Proofs

We begin by bounding $\mathbb{E}_{\mathcal{S}_t}[\mathcal{R}(q\mathrm{W}_t)]$, the expected population error of the iterates, in terms of $\mathbb{E}_{\mathcal{S}_t,\mathcal{B}_t}[L_t(\mathrm{W}_t)]$, the expected instantaneous loss of the random sub-networks. In particular, using simple arguments including the smoothing property, the fact that W_t is independent from (x_t, y_t) given $\mathcal{S}_{t-1}$, and that logistic loss upper-bounds the zero-one loss, it is easy to bound the expected population risk as $\mathbb{E}_{\mathcal{S}_t}[\mathcal{R}(q\mathrm{W}_t)] \leq \mathbb{E}_{\mathcal{S}_t}[\ell(y_t f(\mathrm{x}_t; q\mathrm{W}_t))]$. Furthermore, using Jensen's inequality, we have that $\ell(y_t f_t(q\mathrm{W}_t)) \leq \mathbb{E}_{\mathrm{B}_t}[L_t(\mathrm{W}_t)]$. The upper bound then follows from these two inequalities.

Lemma 11. *For any $t \in [T]$, let $\mathcal{B}_t := \{B_1, \ldots, B_t\}$ denote the set of Bernoulli masks up to time t. Then it holds almost surely that:*

$$\sum_{t=1}^{T} \ell(y_t f_t(q\,W_t)) \leq \mathbb{E}_{\mathcal{B}_T}[\sum_{t=1}^{T} L_t(\,W_t)]. \tag{5.1}$$

Proof of Lemma 11. For any $a, b \in \mathbb{R}$, the function $\ell(z) = \log(1+\exp(az+b))$ is

convex in z. We have the following inequalities:

$$\begin{aligned}
\mathbb{E}_{\mathcal{B}_T}[\sum_{t=1}^{T} L_t(\mathrm{W}_t)] &= \sum_{t=1}^{T} \mathbb{E}_{\mathcal{B}_t}[\ell(y_t \cdot \frac{1}{\sqrt{m}} \mathrm{a}^\top \mathrm{B}_t \sigma(\mathrm{W}_t \mathrm{x}_t))] \\
&= \sum_{t=1}^{T} \mathbb{E}_{\mathcal{B}_{t-1}}[\mathbb{E}_{\mathrm{B}_t} \ell(y_t \cdot \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r b_{r,t} \sigma(\mathrm{w}_{r,t}^\top \mathrm{x}_t)) | \mathcal{B}_{t-1}] \\
&\qquad \text{(smoothing property)} \\
&\geq \sum_{t=1}^{T} \mathbb{E}_{\mathcal{B}_{t-1}}[\ell(y_t \cdot \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \mathbb{E}_{\mathrm{B}_t}[b_{r,t}] \sigma(\mathrm{w}_{r,t}^\top \mathrm{x}_t)) | \mathcal{B}_{t-1}] \\
&\qquad \text{(Jensen's inequality)} \\
&= \sum_{t=1}^{T} \ell(y_t \cdot \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma(q \mathrm{w}_{r,t}^\top \mathrm{x}_t)) \\
&\qquad (\mathbb{E}[b_{r,t}] = q, \text{ homogeneity of ReLU}) \\
&= \sum_{t=1}^{T} \ell(y_t f_t(q \mathrm{W}_t))
\end{aligned}$$

which completes the proof. □

In the following, we present the main ideas in bounding the average instantaneous loss of the iterates.

Under Algorithm 4, dropout iterates are guaranteed to remain in the set $\mathcal{W}_c := \{\mathrm{W} \in \mathbb{R}^{m \times d} : \ \|\mathrm{w}_r\| \leq c\}$. Using this property and the dropout update rule, we track the distance of consecutive iterates $(\mathrm{W}_{t+1}, \mathrm{W}_t)$ from any competitor $\mathrm{U} \in \mathcal{W}_c$, which leads to the following upper bound on the average instantaneous loss of iterates.

Lemma 12. *Let $W_1, \ldots, W_T$ be the sequence of dropout iterates with a learning rate satisfying $\eta \leq \ln 2$. Then, it holds for any $U \in \mathcal{W}_c$ that*

$$\frac{1}{T} \sum_{t=1}^{T} L_t(W_t) \leq \frac{\|W_1 - U\|_F^2}{\eta T} + \frac{2}{T} \sum_{t=1}^{T} L_t^{(t)}(U). \tag{5.2}$$

Proof of Lemma 12. Using the dropout update rule in Algorithm 4, we start by analyzing the distance of consecutive iterates from the reference point U, assuming

that $\Pi_c(\mathrm{U}) = \mathrm{U}$:

$$\begin{aligned}
\|\mathrm{W}_{t+1} - \mathrm{U}\|_F^2 &= \|\Pi_c(\mathrm{W}_{t+\frac{1}{2}}) - \mathrm{U}\|_F^2 \\
&\leq \|\mathrm{W}_{t+\frac{1}{2}} - \mathrm{U}\|_F^2 && (\mathrm{U} \in \mathcal{W}_c) \\
&= \|\mathrm{W}_t - \eta \nabla L_t(\mathrm{W}_t) - \mathrm{U}\|_F^2 \\
&= \|\mathrm{W}_t - \mathrm{U}\|_F^2 - 2\eta \langle \nabla L_t(\mathrm{W}_t), \mathrm{W}_t - \mathrm{U} \rangle + \eta^2 \|\nabla L_t(\mathrm{W}_t)\|_F^2
\end{aligned}$$

The last term on the right hand side above is bounded as follows:

$$\begin{aligned}
\eta^2 \|\nabla L_t(\mathrm{W}_t)\|_F^2 &= \eta^2 \|\ell'(y_t g_t(\mathrm{W}_t)) y_t \nabla g_t(\mathrm{W}_t)\|_F^2 \\
&= \eta^2 \left(-\ell'(y_t g_t(\mathrm{W}_t)) \|\nabla g_t(\mathrm{W}_t)\|_F\right)^2 \\
&= \eta^2 Q_t(\mathrm{W}_t)^2 \sum_{r=1}^{m} \|\frac{\partial g_t(\mathrm{W}_t)}{\partial \mathrm{w}_{r,t}}\|^2 \\
&\leq \eta^2 Q_t(\mathrm{W}_t)^2 && (\|\tfrac{\partial g_t(\mathrm{W}_t)}{\partial \mathrm{w}_{r,r}}\| \leq \tfrac{1}{\sqrt{m}}) \\
&\leq \frac{\eta^2}{\ln 2} Q_t(\mathrm{W}_t) && (Q_t(\cdot) \leq 1/\ln 2) \\
&\leq \eta Q_t(\mathrm{W}_t) && (\text{assumption } \eta \leq \ln 2) \\
&\leq \eta L_t(\mathrm{W}_t) && (Q_t(\cdot) \leq L_t(\cdot))
\end{aligned}$$

The second term can be bounded as follows:

$$\begin{aligned}
\langle \nabla L_t(\mathrm{W}_t), \mathrm{W}_t - \mathrm{U} \rangle &= \ell'(y_t g_t(\mathrm{W}_t)) \langle y_t \nabla g_t(\mathrm{W}_t), \mathrm{W}_t - \mathrm{U} \rangle \\
&= \ell'(y_t g_t(\mathrm{W}_t))(y_t g_t(\mathrm{W}_t) - y_t g_t^{(t)}(\mathrm{U})) \\
& && (\text{Homogeneity, definition of } g_t^{(t)}) \\
&\geq (\ell(y_t g_t(\mathrm{W}_t)) - \ell(y_t g_t^{(t)}(\mathrm{U}))) && (\text{convexity of } \ell(\cdot)) \\
&= L_t(\mathrm{W}_t) - L_t^{(t)}(\mathrm{U})
\end{aligned}$$

Plugging back the above inequalities we get

$$\begin{aligned}
\|\mathrm{W}_{t+1} - \mathrm{U}\|_F^2 \leq \|\mathrm{W}_{t+\frac{1}{2}} - \mathrm{U}\|_F^2 &\leq \|\mathrm{W}_t - \mathrm{U}\|_F^2 - 2\eta(L_t(\mathrm{W}_t) - L_t^{(t)}(\mathrm{U})) + \eta L_t(\mathrm{W}_t) \\
&= \|\mathrm{W}_t - \mathrm{U}\|_F^2 - \eta L_t(\mathrm{W}_t) + 2\eta L_t^{(t)}(\mathrm{U}) \qquad (5.3)
\end{aligned}$$

Rearranging, dividing both sides by η, and averaging over iterates we arrive at

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} L_t(\mathrm{W}_t) &\leq \sum_{t=1}^{T} \frac{\|\mathrm{W}_t - \mathrm{U}\|_F^2 - \|\mathrm{W}_{t+1} - \mathrm{U}\|_F^2}{\eta T} + \frac{2}{T}\sum_{t=1}^{T} L_t^{(t)}(\mathrm{U}) \\ &\leq \frac{\|\mathrm{W}_1 - \mathrm{U}\|_F^2}{\eta T} + \frac{2}{T}\sum_{t=1}^{T} L_t^{(t)}(\mathrm{U}) \qquad \text{(Telescopic sum)}\end{aligned}$$

□

Note that the upper bound in Equation (5.2) holds for *any* competitor $\mathrm{U} \in \mathcal{W}_c$; however, we seek to minimize the upper-bound on the right hand side of Equation (5.2) by finding a *sweet spot* that maintains a trade-off between 1) the distance from initialization, and 2) the average instantaneous loss for the linearized models. Following [JT19b], we represent such a competitor as an interpolation between the initial weights W_1 and the *max-margin* competitor V, i.e. $\mathrm{U} := \mathrm{W}_1 + \lambda \mathrm{V}$, where λ is the trade-off parameter. Recall that $\mathrm{V} := [\mathrm{v}_1, \cdots, \mathrm{v}_m] \in \mathbb{R}^{d \times m}$, where $\mathrm{v}_r = \frac{1}{\sqrt{m}} a_r \psi(\mathrm{w}_{r,1})$ for any $r \in [m]$, and ψ is given by assumption 2 . Thus, the first term on the right hand side above can be conveniently bounded as $\frac{\lambda^2}{\eta T}$; Lemma 13 bounds the second term as follows.

Lemma 13. *Under the setting of Theorem 14, it holds with probability at least* $1 - \delta$ *simultaneously for all iterates* $t \in [T]$ *that 1)* $\|\mathrm{w}_{r,t} - \mathrm{w}_{r,1}\| \leq \frac{7\lambda}{2\gamma\sqrt{m}}$*, for all* $r \in [m]$*; and 2)* $L_t^{(t)}(U) \leq \frac{\lambda^2}{2\eta T}$.

We now present the main ideas in proving Lemma 13, which closely follows [JT19b]. In order to prove Lemma 13, we need Lemma 16, Lemma 14, and Lemma 15 that we are going to present next. Since $L_t^{(t)}(\mathrm{U}) \leq e^{-y_t \langle \nabla g_t(\mathrm{W}_t), \mathrm{U} \rangle}$, the proof entails lower bounding $y_t \langle \nabla g_t(\mathrm{W}_t), \mathrm{U} \rangle$, which can be decomposed as follows

$$\begin{aligned}y_t \langle \nabla g_t(\mathrm{W}_t), \mathrm{U} \rangle = y_t \langle \nabla g_t(\mathrm{W}_1), \mathrm{W}_1 \rangle + y_t \langle \nabla g_t(\mathrm{W}_t) - \nabla g_t(\mathrm{W}_1), \mathrm{W}_1 \rangle \\ + \lambda y_t \langle \nabla g_t(\mathrm{W}_1), \mathrm{V} \rangle + \lambda y_t \langle \nabla g_t(\mathrm{W}_t) - \nabla g_i(\mathrm{B}_t \mathrm{W}_1), \mathrm{V} \rangle. \quad (5.4)\end{aligned}$$

By homogeneity of the ReLU activations, the first term in Equation (5.4) precisely computes $y_t g_t(\mathrm{W}_1)$, which cannot be too negative under the initialization scheme used in Algorithm 4, as we show in Lemma 14.

Lemma 14. *With probability at least $1-\delta/3$ it holds uniformly over all $t \in [T]$ that $|g_t(W_1)| \leq \sqrt{2\ln 6T/\delta}$, provided that $m \geq 25 \ln 6T/\delta$.*

Proof of Lemma 14. The proof is similar to the proof of Lemma A.1 in [JT19b], except for that we have to take into account the randomness due to dropout as well. In particular, there are four different sources of randomness in $g_t(\mathrm{W}_1) = g(\mathrm{W}_1; \mathrm{x}_t, \mathrm{B}_t)$: 1) the randomly initialized hidden layer weights W_1, 2) the randomly initialized top layer weights a, 3) the input vector $\mathrm{x}_t,\ t \in [T]$, and 4) the Bernoulli masks $\mathrm{B}_t,\ t \in [T]$. Given input x_t and the dropout mask B_t, let $\mathrm{h}_t(\mathrm{W}) = \frac{1}{\sqrt{m}}\mathrm{B}_t\sigma(\mathrm{W}\mathrm{x}_t) \in \mathbb{R}^m$ denote the (scaled) output of the dropout layer with hidden weights W. It is easy to see that the function $g : \mathrm{W} \mapsto \|\mathrm{h}_t(\mathrm{W})\|$ is 1-Lipschitz:

$$
\begin{aligned}
|g(\mathrm{W}) - g(\mathrm{W}')| &= |\|\mathrm{h}_t(\mathrm{W})\| - \|\mathrm{h}_t(\mathrm{W}')\|| \\
&\leq \|\mathrm{h}_t(\mathrm{W}) - \mathrm{h}_t(\mathrm{W}')\| && \text{(Reverse Triangle Inequality)} \\
&= \sqrt{\sum_{r=1}^{m}(\frac{1}{\sqrt{m}}b_i^{(t)}\sigma(\langle \mathrm{w}_{r,1}, \mathrm{x}_t\rangle) - \frac{1}{\sqrt{m}}b_i^{(t)}\sigma(\langle \mathrm{w}'_{r,1}, \mathrm{x}_t\rangle))^2} \\
&= \frac{\sqrt{\sum_{r=1}^{m}(\langle \mathrm{w}_{r,1}, \mathrm{x}_t\rangle - \langle \mathrm{w}'_{r,1}, \mathrm{x}_t\rangle)^2}}{\sqrt{m}} && \text{(1-Lipschitzness of ReLU)} \\
&\leq \frac{\sqrt{\sum_{r=1}^{m}\|\mathrm{w}_{r,1} - \mathrm{w}'_{r,1}\|^2\|\mathrm{x}_t\|^2}}{\sqrt{m}} && \text{(Cauchy-Schwarz)} \\
&= \frac{\|\mathrm{W} - \mathrm{W}'\|_F}{\sqrt{m}}
\end{aligned}
$$

Using Gaussian concentration (Lemma 21), we get that $\|\mathrm{h}_t(\mathrm{W}_1)\| - \mathbb{E}_{\mathrm{W}_1}[\|\mathrm{h}_t(\mathrm{W}_1)\|] \leq \sqrt{\frac{2\ln\frac{6T}{\delta}}{m}}$ with probability at least $1 - \frac{\delta}{6T}$. It also holds that:

$$
\begin{aligned}
\mathbb{E}_{\mathrm{W}_1}[\|\mathrm{h}_t(\mathrm{W}_1)\|] &\leq \sqrt{\mathbb{E}_{\mathrm{W}_1}[\|\mathrm{h}_t(\mathrm{W}_1)\|^2]} \\
&= \sqrt{\sum_{r=1}^{m}\mathbb{E}_{\mathrm{w}_{r,1}}(\frac{1}{\sqrt{m}}b_{r,t}\sigma(\mathrm{w}_{r,1}^\top \mathrm{x}_t))^2} \\
&\leq \sqrt{\frac{\sum_{r=1}^{m}\mathbb{E}_{\mathrm{w}_{r,1}}[\sigma(\mathrm{w}_{r,1}^\top \mathrm{x}_t)^2]}{m}} \\
&= \sqrt{\mathbb{E}_{z\sim\mathcal{N}(0,1)}[\sigma(z)^2]} = \frac{1}{\sqrt{2}}
\end{aligned}
$$

As a result, we have with probability at least $1-\frac{\delta}{6T}$ that $\|\mathrm{h}_t(\mathrm{W}_1)\| \leq \sqrt{\frac{2\ln 6T/\delta}{m}}+\frac{\sqrt{2}}{2} \leq 1$ whenever $m \geq 25 \ln 6T/\delta$. Now, taking a union bound over all $t \in [T]$, we get that $\|\mathrm{h}_t(\mathrm{W}_1)\| \leq 1$ holds simultaneously for all iterates. Conditioned on this event, the random variable $g_t(\mathrm{W}_1) = \langle \mathrm{a}, \mathrm{h}_t(\mathrm{W}_1)\rangle$ is zero mean and 1-sub-Gaussian, so that by the general Hoeffding's inequality, for any t, with probability at least $1-\frac{\delta}{6T}$, it holds that $|g_t(\mathrm{W}_1)| \leq \sqrt{2\ln 6T/\delta}$. Taking union bound over all $t \in [T]$, with probability $1-\delta/6$ it holds that $|g_t(\mathrm{W}_1)| \leq \sqrt{2\ln 6T/\delta}$ simultaneously for all $t \in [T]$. Finally, the probability that both of these events hold is no less than $(1-\delta/6)^2 \geq 1-\delta/3$, which completes the proof. □

On the other hand, we show in Lemma 15 that under Assumption 2, V has a good margin with respect to the randomly initialized weights W_1, so that the third term in Equation (5.4) is concentrated around the margin parameter γ. This Lemma provides a finite-width analogues to the Assumption 2.

Lemma 15. *Under Assumption 2, for any $\delta \in (0,1)$, with probability at least $1-\delta/3$ it holds uniformly for all $t \in [T]$ that:*

$$y_t g_t^{(1)}(\mathrm{V}) = y_t \langle \nabla g_t(\mathrm{W}_1), \mathrm{V}\rangle \geq \gamma - \sqrt{\frac{2\ln 3T/\delta}{m}}$$

Proof of Lemma 15. By Assumption 2, it holds that $\mathbb{E}_{\mathrm{z},b}[y\langle \psi(\mathrm{z}), b\mathrm{x}\mathbb{I}\{\mathrm{z}^\top \mathrm{x} > 0\}\rangle] \geq \gamma$ for all (x, y) in the domain of $\mathcal{D}$. We observe that $y_t g_t^{(1)}(\mathrm{V})$ is an empirical estimate of this quantity:

$$\begin{aligned} y_t g_t^{(1)}(\mathrm{V}) &= y_t \langle \nabla g_t(\mathrm{W}_1), \mathrm{V}\rangle \\ &= y_t \sum_{r=1}^{m} \langle \frac{1}{\sqrt{m}} a_r b_{r,t} \mathbb{I}\{\mathrm{x}_t^\top \mathrm{w}_{r,1} > 0\}\mathrm{x}_t, \frac{1}{\sqrt{m}} a_r \psi(\mathrm{w}_{r,1})\rangle \\ &= \frac{1}{m}\sum_{r=1}^{m} y_t \langle \psi(\mathrm{w}_{r,1}), b_{r,t}\mathrm{x}_t \mathbb{I}\{\mathrm{w}_{r,1}^\top \mathrm{x}_t > 0\}\rangle \end{aligned}$$

For $t, r \in [T]\times[m]$, let $\gamma_{t,r} := y_t\langle \psi(\mathrm{w}_{r,1}), b_{r,t}\mathrm{x}_t \mathbb{I}\{\mathrm{w}_{r,1}^\top \mathrm{x}_t > 0\}\rangle$. Note that $\mathbb{E}_{\mathrm{W}_1,\mathrm{B}_t}[\gamma_{t,r}] = \mathbb{E}_{\mathrm{z},b}[y_t\langle \psi(\mathrm{z}), b\mathrm{x}_t \mathbb{I}\{\mathrm{z}^\top \mathrm{x}_t > 0\}\rangle]$. Also, for any t, the random variable $\gamma_{t,r}$ is bounded

almost surely as follows:

$$|\gamma_{t,r}| \leq |y_t| \, \|\psi(\mathrm{w}_{r,1})\| \, |b_{r,t}| \, \|\mathrm{x}_t\| \left|\mathbb{I}\{\mathrm{w}_{r,1}^\top \mathrm{x}_t > 0\}\right| \leq 1.$$

Therefore by Hoeffding's inequality (Theorem 17), with probability at least $1 - \frac{\delta}{3T}$, it holds that:

$$y_t g_t^{(1)}(\mathrm{V}) - \gamma \geq y_t g_t^{(1)}(\mathrm{V}) - \mathbb{E}[y_t g_t^{(1)}(\mathrm{V})] \geq -\sqrt{\frac{2 \ln 3T/\delta}{m}}$$

Applying a union bound over t finishes the proof. □

The second and the fourth terms in Equation (5.4) can be bounded thanks to the lazy regime, where W_t remains close to W_1 at all times. In particular, provided $\|\mathrm{w}_{r,t} - \mathrm{w}_{r,1}\| \leq \frac{7\lambda}{2\gamma\sqrt{m}}$, we show in Lemma 16 that at most only $O(1/\sqrt{m})$-fraction of neural activations change, and thus $\nabla g_t(\mathrm{W}_t) - \nabla g_t(\mathrm{W}_1)$ has a small norm. The following Lemma bounds $|R_t|$, where $R_t := \{r \in [m] |\ \mathbb{I}\{\mathrm{w}_{r,t}^\top \mathrm{x}_t > 0\} \neq \mathbb{I}\{\mathrm{w}_{r,1}^\top \mathrm{x}_t > 0\}\}$ is the set of hidden nodes at time t whose activation on sample x_t is different from the initialization.

Lemma 16. *Assume that $\|\mathrm{w}_{r,1} - \mathrm{w}_{r,t}\| \leq D$ holds for all $r \in [m]$, where D is a positive constant. Then, with probability at least $1 - \frac{\delta}{3}$, we have that*

$$|R_t| \leq mD + \sqrt{\frac{m \ln 3T/\delta}{2}}, \text{ for all } t \in [T].$$

Proof of Lemma 16. Assume that $r \in R_t$. Then it holds that

$$\begin{aligned}
|\mathrm{w}_{r,1}^\top \mathrm{x}_t| &\leq |\mathrm{w}_{r,1}^\top \mathrm{x}_t| + |\mathrm{w}_{r,t}^\top \mathrm{x}_t| \\
&= |(\mathrm{w}_{r,1} - \mathrm{w}_{r,t})^\top \mathrm{x}_t| && (r \in R_t) \\
&\leq \|\mathrm{w}_{r,1} - \mathrm{w}_{r,t}\| \|\mathrm{x}_t\| && \text{(Cauchy-Schwarz)} \\
&= \|\mathrm{w}_{r,1} - \mathrm{w}_{r,t}\| \leq D && (\|\mathrm{x}_t\| = 1)
\end{aligned}$$

Since $\mathrm{w}_{r,1}^\top \mathrm{x}_t$ is a standard Gaussian random variable, by anti-concentration property of the Gaussian distribution, $\mathbb{E}[\mathbb{I}\{\left|\mathrm{w}_{r,1}^\top \mathrm{x}_t\right| \leq D\}] = \Pr\{\left|\mathrm{w}_{r,1}^\top \mathrm{x}_t\right| \leq D\} \leq \frac{2D}{\sqrt{2\pi}}$. On the

other hand, we have that

$$|R_t| = \left|\{r|\ \mathbb{I}\{\mathrm{w}_{r,t}^\top \mathrm{x}_t > 0\} \neq \mathbb{I}\{\mathrm{w}_{r,1}^\top \mathrm{x}_t > 0\}\}\right| \leq |\{r|\ \left|\mathrm{w}_{r,1}^\top \mathrm{x}_t\right| \leq D\}| = \sum_{r=1}^{m} \mathbb{I}\{\left|\mathrm{w}_{r,1}^\top \mathrm{x}_t\right| \leq D\}$$

By Hoeffding's inequality, we have the following with probability at least $1 - \frac{\delta}{3T}$:

$$\frac{1}{m}\sum_{r=1}^{m} \mathbb{I}\{\left|\mathrm{w}_{r,1}^\top \mathrm{x}_t\right| \leq D\} \leq \Pr\{\left|\mathrm{w}_{r,1}^\top \mathrm{x}_t\right| \leq D\} + \sqrt{\frac{\ln 3T/\delta}{2m}} \leq \frac{2D}{\sqrt{2\pi}} + \sqrt{\frac{\ln 3T/\delta}{2m}}.$$

Multiplying both sides by m and applying union bound on $t \in [T]$ completes the proof. □

In light of Lemma 16, Lemma 14, and Lemma 15, the result of Lemma 13 follows from carefully choosing λ and m such that the right hand side of Equation (5.4) is sufficiently positive.

Proof of Lemma 13. We adopt the proof of Theorem 2.2 in [JT19b] for dropout training. Assume that $\|\mathrm{w}_{r,t} - \mathrm{w}_{r,1}\| \leq \frac{7\lambda}{2\gamma\sqrt{m}}$ holds for the first T iterates of Algorithm 4. Then with probability at least $1 - (\frac{\delta}{3} + \frac{\delta}{3} + \frac{\delta}{3}) = 1 - \delta$, Lemma 16, Lemma 14, and Lemma 15 hold simultaneously. We first prove that $L_t^{(t)}(\mathrm{U}) \leq \frac{\lambda^2}{2\eta T}$ for all $t \in [T]$. Using the inequality $\log(1 + z) \leq z$, we get that

$$L_t^{(t)}(\mathrm{U}) = \log(1 + e^{-y_t\langle \nabla g_t(\mathrm{W}_t), \mathrm{U}\rangle}) \leq e^{-y_t\langle \nabla g_t(\mathrm{W}_t), \mathrm{U}\rangle}$$

To upper-bound the right hand side, we lower-bound $y_t\langle \nabla g_t(\mathrm{W}_t), \mathrm{U}\rangle$. By definition of U, we have

$$y_t\langle \nabla g_t(\mathrm{W}_t), \mathrm{U}\rangle = y_t\langle \nabla g_t(\mathrm{W}_t), \mathrm{W}_1\rangle + \lambda y_t\langle \nabla g_t(\mathrm{W}_t), \mathrm{V}\rangle \tag{5.5}$$

We bound each of the terms separately. The first term can be decomposed as follows:

$$\begin{aligned} y_t\langle \nabla g_t(\mathrm{W}_t), \mathrm{W}_1\rangle &= y_t\langle \nabla g_t(\mathrm{W}_1), \mathrm{W}_1\rangle + y_t\langle \nabla g_t(\mathrm{W}_t) - \nabla g_t(\mathrm{W}_1), \mathrm{W}_1\rangle \\ &\geq -|y_t g_t(\mathrm{W}_1)| - |y_t\langle \nabla g_t(\mathrm{W}_t) - \nabla g_t(\mathrm{W}_1), \mathrm{W}_1\rangle| \end{aligned} \tag{5.6}$$

By Lemma 14, the first term on right hand side is lower-bounded by $-|g_t(\mathrm{W}_1)| \geq -\sqrt{2\ln 6T/\delta}$. We bound the second term as follows:

$$
\begin{aligned}
|y_t\langle \nabla g_t(\mathrm{W}_t) - \nabla g_t(\mathrm{W}_1), \mathrm{W}_1\rangle| &= \left|\frac{y_t}{\sqrt{m}}\sum_{r=1}^{m} a_r b_{r,t}(\mathbb{I}\{\mathrm{w}_{r,t}^\top \mathrm{x}_t > 0\} - \mathbb{I}\{\mathrm{w}_{r,1}^\top \mathrm{x}_t > 0\})\mathrm{w}_{r,1}^\top \mathrm{x}_t\right| \\
&\leq \frac{1}{\sqrt{m}}\sum_{r\in R_t} |a_r b_{r,t}\langle \mathrm{w}_{r,1}, \mathrm{x}_t\rangle| && \text{(Triangle inequality)} \\
&\leq \frac{1}{\sqrt{m}}\sum_{r\in R_t} |\langle \mathrm{w}_{r,t} - \mathrm{w}_{r,1}, \mathrm{x}_t\rangle| && (r \in R_t) \\
&\leq \frac{|R_t|\,\|\mathrm{w}_{r,t} - \mathrm{w}_{r,1}\|}{\sqrt{m}} \\
&\leq \frac{49\lambda^2}{4\gamma^2\sqrt{m}} + \sqrt{\frac{49\lambda^2 \ln 3T/\delta}{8\gamma^2 m}} && \text{(Lemma 16)} \\
&\leq \frac{\lambda\gamma}{2} && (5.7)
\end{aligned}
$$

where the last inequality holds when $m \geq \max\{98\gamma^{-4}\ln 3T/\delta, 2401\gamma^{-6}\lambda^2\} = 2401\gamma^{-6}\lambda^2$. The second term in Equation 5.5 is bounded as follows:

$$
\begin{aligned}
y_t\langle \nabla g_t(\mathrm{W}_t), \mathrm{V}\rangle &= y_t\langle \nabla g_t(\mathrm{W}_1), \mathrm{V}\rangle + y_t\langle \nabla g_t(\mathrm{W}_t) - \nabla g_t(\mathrm{W}_1), \mathrm{V}\rangle \\
&\geq y_t\langle \nabla g_t(\mathrm{W}_1), \mathrm{V}\rangle - |y_t\langle \nabla g_t(\mathrm{W}_t) - \nabla g_t(\mathrm{W}_1), \mathrm{V}\rangle| \\
&= y_t g_t^{(1)}(\mathrm{V}) - \left|\frac{y_t}{\sqrt{m}}\sum_{r=1}^{m} a_r b_{r,t}(\mathbb{I}\{\mathrm{w}_{r,t}^\top \mathrm{x}_t > 0\} - \mathbb{I}\{\mathrm{w}_{r,1}^\top \mathrm{x}_t > 0\})\langle \frac{1}{\sqrt{m}} a_r \psi(\mathrm{w}_{r,1}), \mathrm{x}_t\rangle\right| \\
&\geq \gamma - \sqrt{\frac{2\ln 3T/\delta}{m}} - \frac{1}{m}\sum_{r\in R_t} |a_r b_{r,t}\langle \psi(\mathrm{w}_{r,1}), \mathrm{x}_t\rangle| && \text{(Lemma 15)} \\
&\geq \gamma - \sqrt{\frac{2\ln 3T/\delta}{m}} - \frac{|R_t|}{m} \\
&\geq \gamma - \sqrt{\frac{2\ln 3T/\delta}{m}} - \frac{7\lambda}{2\gamma\sqrt{m}} - \sqrt{\frac{\ln 3T/\delta}{2m}} && \text{(Lemma 16)} \\
&\geq \gamma - \frac{\gamma^2}{7} - \frac{\gamma^2}{14} - \frac{\gamma^2}{14} = \gamma - \frac{2\gamma^2}{7} \geq \frac{5\gamma}{7} && (5.8)
\end{aligned}
$$

where the penultimate inequality holds when $m \geq \max\{98\gamma^{-4}\ln 3T/\delta, 2401\gamma^{-6}\lambda^2\} = 2401\gamma^{-6}\lambda^2$. Plugging Equations (5.7) and (5.8) in Equation 5.5, we get that

$$
y_t\langle \nabla g_t(\mathrm{W}_t), \mathrm{U}\rangle \geq -\sqrt{2\ln 6T/\delta} + \frac{3\lambda\gamma}{14} \geq \ln\frac{2\eta T}{\lambda^2}, \tag{5.9}
$$

where the inequality hold true for $\lambda := 5\gamma^{-1}\ln 2\eta T + \sqrt{44\gamma^{-2}\ln 6T/\delta}$. Thus, we have that

$$L_t^{(t)}(\mathrm{U}) = \log(1 + e^{-y_t\langle \nabla g_t(\mathrm{W}_t), \mathrm{U}\rangle}) \leq \frac{\lambda^2}{2\eta T}.$$

We now prove by induction that $\|\mathrm{w}_{r,t} - \mathrm{w}_{r,1}\| \leq \frac{7\lambda}{2\gamma\sqrt{m}}$ holds throughout dropout training. First, we show that the claim holds for $t = 2$:

$$\begin{aligned}
\|\mathrm{w}_{r,2} - \mathrm{w}_{r,1}\| = \|\Pi_c(\eta\frac{\partial L_t(\mathrm{B}_1\mathrm{W}_1)}{\partial \mathrm{w}_{r,1}})\| &\leq \|\eta\frac{\partial L_t(\mathrm{B}_1\mathrm{W}_1)}{\partial \mathrm{w}_{r,1}}\| \\
&\leq \|\eta\ell'(y_t f_t(\mathrm{B}_1\mathrm{W}_1))y_i\frac{\partial f_t(\mathrm{B}_1\mathrm{W}_1)}{\partial \mathrm{w}_{r,1}}\| \\
&\leq \frac{\eta}{\ln 2\sqrt{m}} \leq \frac{7\lambda}{2\gamma\sqrt{m}}, \qquad (\eta \leq \ln 2)
\end{aligned}$$

which proves the basic step. Now by induction hypothesis, we assume that the claim holds for all $k \in [t]$, i.e., it holds that $\|\mathrm{w}_{r,k} - \mathrm{w}_{r,1}\| \leq \frac{7\lambda}{2\gamma\sqrt{m}}$. Therefore, it holds that $\|\mathrm{w}_{r,k}\| \leq \|\mathrm{w}_{r,1}\| + \|\mathrm{w}_{r,k} - \mathrm{w}_{r,1}\| \leq c - 1 + 1 \leq c$, where we used the triangle inequality, the fact that $\|\mathrm{w}_{r,1}\| \leq c - 1$, and that $m \geq 2401\gamma^{-6}\lambda^2$. This, in particular, means that all iterates $1 < k \leq t$ remain in $\mathcal{W}_c$:

$$\mathrm{W}_k = \Pi_c(\mathrm{W}_{k-\frac{1}{2}}) = \mathrm{W}_{k-\frac{1}{2}} \text{ for all } 1 < k \leq t. \tag{5.10}$$

For the $t+1$-th iterate, we first upper-bound the distance from initialization in terms of the Q function:

$$\begin{aligned}
\|\mathrm{w}_{r,t+1} - \mathrm{w}_{r,1}\| &= \|\Pi_c(\mathrm{w}_{r,t} - \eta\frac{\partial L_t(\mathrm{W}_t)}{\partial \mathrm{w}_{r,t}}) - \mathrm{w}_{r,1}\| \\
&\leq \|\mathrm{w}_{r,t} - \eta\frac{\partial L_t(\mathrm{W}_t)}{\partial \mathrm{w}_{r,t}} - \mathrm{w}_{r,1}\| \\
&\leq \|\eta\frac{\partial L_t(\mathrm{W}_t)}{\partial \mathrm{w}_{r,t}}\| + \|\mathrm{w}_{r,t} - \mathrm{w}_{r,1}\| \\
&\leq \sum_{k=1}^{t}\|\eta\frac{\partial L_k(\mathrm{W}_k)}{\partial \mathrm{w}_{r,k}}\| \\
&\leq \eta\sum_{k=1}^{t} -\ell'(y_k g_k(\mathrm{W}_k))\|y_k\frac{\partial g_k(\mathrm{W}_k)}{\partial \mathrm{w}_{r,k}}\| \\
&\leq \frac{\eta}{\sqrt{m}}\sum_{k=1}^{t} -\ell'(y_k g_k(\mathrm{W}_k))
\end{aligned}$$

The idea is to turn the right hand side above into a telescopic sum using the identity $\mathrm{W}_{k+1} - \mathrm{W}_k = \mathrm{W}_{k+\frac{1}{2}} - \mathrm{W}_k = \eta \ell'(y_k g_k(\mathrm{W}_k)) y_k \nabla g_k(\mathrm{W}_k),\ k \in [t-1]$. By induction hypothesis, for all $k \in [t]$, Equation (5.8) guarantees $y_k \langle \nabla g_k(\mathrm{W}_k), \mathrm{V} \rangle \geq \frac{5\gamma}{7}$. Thus, multiplying the right hand side of (5.11) by $\frac{7}{5\gamma} y_k \langle \nabla g_k(\mathrm{W}_k), \mathrm{V} \rangle$, we get that:

$$
\begin{aligned}
\|\mathrm{w}_{r,t+1} - \mathrm{w}_{r,1}\| &\leq \frac{7\eta}{5\gamma\sqrt{m}} \sum_{k=1}^{t} -\ell'(y_k g_k(\mathrm{W}_k)) y_k \langle \nabla g_k(\mathrm{W}_k), \mathrm{V} \rangle \\
&= \frac{7}{5\gamma\sqrt{m}} \sum_{k=1}^{t} \langle \eta \nabla L_k(\mathrm{W}_k), \mathrm{V} \rangle \\
&= \frac{7}{5\gamma\sqrt{m}} \langle \mathrm{W}_{t+\frac{1}{2}} - \mathrm{W}_1, \mathrm{V} \rangle && \text{(Equation (5.10))} \\
&= \frac{7\langle \mathrm{W}_{t+\frac{1}{2}} - \mathrm{U}, \mathrm{V} \rangle + 7\langle \mathrm{U} - \mathrm{W}_1, \mathrm{V} \rangle}{5\gamma\sqrt{m}} \\
&\leq \frac{7\|\mathrm{W}_{t+\frac{1}{2}} - \mathrm{U}\|_F \|\mathrm{V}\|_F + 7\langle \lambda \mathrm{V}, \mathrm{V} \rangle}{5\gamma\sqrt{m}} && \text{(Cauchy-Schwarz)} \\
&\leq \frac{7\|\mathrm{W}_{t+\frac{1}{2}} - \mathrm{U}\|_F + 7\lambda}{5\gamma\sqrt{m}} && (5.11)
\end{aligned}
$$

Again by induction hypothesis, Equation (5.3) and Equation (5.9) hold for all $k \in [t]$, which are used to bound $\|\mathrm{W}_{t+\frac{1}{2}} - \mathrm{U}\|_F$ as follows:

$$
\begin{aligned}
\|\mathrm{W}_{t+\frac{1}{2}} - \mathrm{U}\|_F^2 &\leq \|\mathrm{W}_t - \mathrm{U}\|_F^2 - 2\eta(L_t(\mathrm{W}_t) - L_t^{(t)}(\mathrm{U})) + \eta L_t(\mathrm{W}_t) \\
& && \text{(Equation (5.3))} \\
&\leq \|\mathrm{W}_t - \mathrm{U}\|_F^2 + 2\eta L_t^{(t)}(\mathrm{U})) \\
&\leq \|\mathrm{W}_1 - \mathrm{U}\|_F^2 + 2\eta \sum_{k=1}^{t} L_k^{(k)}(\mathrm{U}) \\
&\leq \|\lambda \mathrm{V}\|_F^2 + 2\eta t \frac{\lambda^2}{2\eta T} && \text{(Equation 5.9)} \\
&\leq \lambda^2 + \frac{\lambda^2 t}{T} && (\|\mathrm{V}\|_F \leq 1) \\
&\leq 2\lambda^2 \\
\implies \|\mathrm{W}_{t+\frac{1}{2}} - \mathrm{U}\|_F &\leq \sqrt{2}\lambda && (5.12)
\end{aligned}
$$

Plugging Equation (5.12) back in Equation (5.11), we arrive at:

$$
\|\mathrm{w}_{r,t+1} - \mathrm{w}_{r,1}\| \leq \frac{7\sqrt{2}\lambda + 7\lambda}{5\gamma\sqrt{m}} \leq \frac{7\lambda}{2\gamma\sqrt{m}}
$$

Which completes the induction step and the proof. □

Therefore, left hand side of Equation (5.2), i.e., the average instantaneous loss of the iterates, can be bounded with high probability as $\frac{2\lambda^2}{\eta T}$. To get the bound in expectation, as presented in Theorem 14, we need to control $L_t^{(t)}(\mathrm{U})$ in worst-case scenario. A crucial step in giving generalization bounds in expectation via upper-bounding the logistic loss is to control the maximum value the loss can take on any iterate of the algorithm. In particular, we need to upper-bound the instantaneous loss of $g_t^{(t)}(\mathrm{U})$, which appears in the right hand side of Lemma 12. To that end, we note that the logistic loss only grows linearly for $z < 0$. More formally, it holds for all $z < 0$ that:

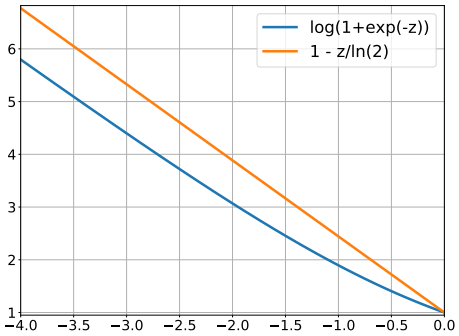

Figure 5-1. Linear upperbound on the logistic loss

$$\log(1 + e^{-z}) \leq \frac{-z}{\ln 2} + 1 \tag{5.13}$$

as depicted in Figure 5-1. We take advantage of the max-norm constraints in Algorithm 4, and show in Lemma 17 that with probability one all iterates satisfy $L_t^{(t)}(\mathrm{U}) \leq \frac{c\sqrt{m}}{\ln 2} + 1$.

Lemma 17. *Under Algorithm 4, it holds with probability one for all iterates that* $L_t^{(t)}(U) \leq \frac{c\sqrt{m}}{\ln 2} + 1$.

Proof of Lemma 17. Recall that $L_t^{(t)}(\mathrm{U}) = \ell(y_t g_t^{(t)}(\mathrm{U}))$. First we bound the argument

inside the loss function:

$$\begin{aligned}
\left|y_t g_t^{(t)}(\mathrm{W}_t)\right| &= |y_t \langle \nabla g_t(\mathrm{W}_t), \mathrm{U}\rangle| \\
&\leq \sum_{r=1}^{m} \left|\langle \frac{\partial g_t(\mathrm{W}_t)}{\partial \mathrm{w}_{r,t}}, \mathrm{u}_r\rangle\right| && \text{(triangle inequality)} \\
&\leq \sum_{r=1}^{m} \|\frac{\partial g_t(\mathrm{W}_t)}{\partial \mathrm{w}_{r,t}}\| \|\mathrm{w}_{r,1} + \lambda \mathrm{v}_r\| && \text{(Cauchy-Schwarz)} \\
&\leq \sum_{r=1}^{m} \frac{c - 1 + \lambda/\sqrt{m}}{\sqrt{m}} && (\|\mathrm{w}_{r,1}\| \leq c-1,\ \|\mathrm{v}_r\| \leq 1/\sqrt{m}) \\
&\leq c\sqrt{m} && (\lambda \leq \sqrt{m})
\end{aligned}$$

Now using Equation (5.13), we get that

$$L_t^{(t)}(\mathrm{U}) = \log(1 + e^{-y_t \langle \nabla g_t(\mathrm{W}_t), \mathrm{U}\rangle}) \leq \log(1 + \exp(c\sqrt{m})) \leq \frac{c\sqrt{m}}{\ln 2} + 1.$$

□

The proof of Theorem 14 then follows from carefully choosing δ, the confidence parameter.

Proof of Theorem 14. Note that W_t is conditionally independent from x_t given $\mathrm{x}_1, \ldots, \mathrm{x}_{t-1}$. Thus,

$$\mathbb{E}_{\mathcal{S}_T}[L_t(\mathrm{W}_t)] = \mathbb{E}_{\mathcal{S}_{t-1}}[\mathbb{E}_{(\mathrm{x}_t, y_t)} \ell(y_t f_t(\mathrm{W}_t)) | \mathcal{S}_{t-1}] = \mathbb{E}_{\mathcal{S}_{t-1}}[L(\mathrm{W}_t)]$$

Using the fact that logistic loss upper-bounds the zero-one loss, taking expectation over initialization, taking average over iterates, and using Lemma 11, we get that:

$$\begin{aligned}
\mathbb{E}_{\mathrm{W}_1, \mathrm{a}, \mathcal{S}_T}[\frac{1}{T}\sum_{t=1}^{T} \mathcal{R}(q\mathrm{W}_t)] &\leq \mathbb{E}_{\mathrm{W}_1, \mathrm{a}, \mathcal{S}_T}[\frac{1}{T}\sum_{t=1}^{T} \ell(y_t f_t(q\mathrm{W}_t))] && (\mathbb{I}\{z < 0\} \leq \ell(z)) \\
&\leq \mathbb{E}_{\mathrm{W}_1, \mathrm{a}, \mathcal{S}_T, \mathcal{B}_T}[\frac{1}{T}\sum_{t=1}^{T} L_t(\mathrm{W}_t)] && \text{(Lemma 11)} \\
&\leq \frac{\mathbb{E}_{\mathrm{W}_1}[\|\mathrm{W}_1 - \mathrm{U}\|_F^2]}{2\eta T} + \frac{2}{T}\sum_{t=1}^{T} \mathbb{E}_{\mathrm{W}_1, \mathrm{a}, \mathcal{S}_T, \mathcal{B}_T}[L_t^{(t)}(\mathrm{U})] \\
& && \text{(Lemma 12)}
\end{aligned}$$

The first term is upper-bounded by $\frac{\lambda^2}{2\eta T}$ since $\|\mathrm{W}_1 - \mathrm{U}\|_F^2 = \|\mathrm{W}_1 - \mathrm{W}_1 - \lambda \mathrm{V}\|_F^2 = \lambda^2 \|\mathrm{V}\|_F^2 \leq \lambda^2$. Bounding the second term is based on the following two facts:

1. By Lemma 13, with probability at least $1-\delta$, it holds that $L_t^{(t)}(\mathrm{U}) \leq \frac{\lambda^2}{2\eta T} =: u_1$.

2. By Lemma 17, it holds with probability one that $L_t^{(t)}(\mathrm{U}) \leq \frac{c\sqrt{m}}{\ln 2} + 1 \leq 2c\sqrt{m} =: u_2$.

Therefore, the expected value of $L_t^{(t)}(\mathrm{U})$ can be upper-bounded as:

$$\mathbb{E}[L_t^{(t)}(\mathrm{U})] \leq (1-\delta)u_1 + \delta u_2 \leq \frac{\lambda^2}{2\eta T} + 2\delta c\sqrt{m}$$

Choosing $\delta := \frac{1}{4\eta c\sqrt{m}T}$ guarantees that

$$\mathbb{E}[L_t^{(t)}(\mathrm{U})] \leq \frac{\lambda^2}{2\eta T} + \frac{1}{2\eta T} \leq \frac{\lambda^2}{\eta T},$$

where $\lambda := 5\gamma^{-1}\ln 2\eta T + \sqrt{44\gamma^{-2}\ln 24\eta c\sqrt{m}T^2}$. □

To prove the compression result in Theorem 15, we use the fact that the zero-one loss can be bounded in terms of the negative derivative of the logistic loss [CG19]. Therefore, we can bound $\mathcal{R}(\mathrm{W}_t;\mathrm{B}_t)$, the population risk of the sub-networks, in terms of $Q(\mathrm{W}_t;\mathrm{B}_t) = \mathbb{E}_{\mathcal{D}}[-\ell'(y_t g_t(\mathrm{W}_t))]$. Following [JT19b], the proof of Theorem 15 then entails showing that $\sum_{t=1}^T Q(\mathrm{W}_t;\mathrm{B}_t)$ is close to $\sum_{t=1}^T Q_t(\mathrm{W}_t)$, which itself is bounded by the average instantaneous loss of the iterates.

Proof of Theorem 15. First, recall the following property of the logistic loss:

$$\mathbb{I}\{z<0\} \leq -2\ln 2\ell'(z) \leq 2\ln 2\ell(z)$$

which implies that $\mathcal{R}(\mathrm{W}_t;\mathrm{B}_t) \leq 2\ln 2 Q(\mathrm{W}_t;\mathrm{B}_t)$, where $Q(\mathrm{W};\mathrm{B}) := \mathbb{E}_{\mathcal{D}}[-\ell'(yg(\mathrm{W};\mathrm{x},\mathrm{B})]$ is the expected value of the negative derivative of the logistic loss. On the other hand, taking the empirical average over the training data, and using Lemma 12 and

Lemma 13, we conclude that the following holds with probability at least $1-\delta$:

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T} Q_t(\mathrm{W}_t) &\le \frac{1}{T}\sum_{t=1}^{T} L_t(\mathrm{W}_t) \\
&\le \frac{\|\mathrm{W}_1 - \mathrm{U}\|_F^2}{\eta T} + \frac{2}{T}\sum_{t=1}^{T} L_t^{(t)}(\mathrm{U}) && \text{(Lemma 12)} \\
&\le \frac{\lambda^2}{\eta T} + \frac{2}{T}\sum_{t=1}^{T} \frac{\lambda^2}{2\eta T} && \text{(Lemma 13)} \\
&\le \frac{2\lambda^2}{\eta T}.
\end{aligned}$$

Given the dropout masks $\mathcal{B}_T$, since $Q(\mathrm{W}_t;\mathrm{B}_t) = \mathbb{E}_{\mathcal{D}}[Q_t(\mathrm{W}_t)]$, we know that

$$\sum_{t=1}^{T} Q(\mathrm{W}_t;\mathrm{B}_t) - \sum_{t=1}^{T} Q_t(\mathrm{W}_t)$$

is a martingale difference with respect to the past observations, $\mathcal{S}_{T-1}$. We next show that the average of $Q_t(\mathrm{W}_t)$ on the right hand side above is close to the average of $Q(\mathrm{W}_t;\mathrm{B}_t)$, using Theorem 22, similar to Lemma 4.3. of [JT19b]. First, this martingale difference sequence is bounded almost surely as $R := 1/\ln 2 \ge Q(\mathrm{W}_t;\mathrm{B}_t) - Q_t(\mathrm{W}_t)$, simply because $0 \le -\ell'(z) \le 1/\ln 2$. The conditional variance can be bounded as:

$$\begin{aligned}
V_t &:= \sum_{t=1}^{T} \mathbb{E}[(Q(\mathrm{W}_t;\mathrm{B}_t) - Q_t(\mathrm{W}_t))^2 | \mathcal{S}_{t-1}] \\
&= \sum_{t=1}^{T} Q(\mathrm{W}_t;\mathrm{B}_t)^2 - 2Q(\mathrm{W}_t;\mathrm{B}_t)\mathbb{E}[Q_t(\mathrm{W}_t)|\mathcal{S}_{t-1}] + \mathbb{E}[Q_t(\mathrm{W}_t)^2|\mathcal{S}_{t-1}] \\
&\le \sum_{t=1}^{T} \mathbb{E}[Q_t(\mathrm{W}_t)^2|\mathcal{S}_{t-1}] && (\mathbb{E}[Q_t(\mathrm{W}_t)|\mathcal{S}_{t-1}] = Q(\mathrm{W}_t;\mathrm{B}_t)) \\
&\le \frac{1}{\ln 2}\sum_{t=1}^{T} \mathbb{E}[Q_t(\mathrm{W}_t)|\mathcal{S}_{t-1}] && (0 \le Q_t(\mathrm{W}_t) \le 1/\ln 2) \\
&= \frac{1}{\ln 2}\sum_{t=1}^{T} Q(\mathrm{W}_t;\mathrm{B}_t)
\end{aligned}$$

Thus, using Theorem 22 with $R \le 1/\ln 2$ and $V_t \le \sum_{t=1}^{T} Q(\mathrm{W}_t;\mathrm{B}_t)/\ln 2$, we conclude that with probability at least $1-\delta$ it holds that

$$\begin{aligned}
&\sum_{t=1}^{T} Q(\mathrm{W}_t;\mathrm{B}_t) - \sum_{t=1}^{T} Q_t(\mathrm{W}_t) \le (e-2)\sum_{t=1}^{T} Q(\mathrm{W}_t;\mathrm{B}_t) + \frac{\ln 1/\delta}{\ln 2} \\
\implies &\frac{1}{T}\sum_{t=1}^{T} Q(\mathrm{W}_t;\mathrm{B}_t) \le \frac{4}{T}\sum_{t=1}^{T} Q_t(\mathrm{W}_t) + \frac{4\log(1/\delta)}{T}
\end{aligned}$$

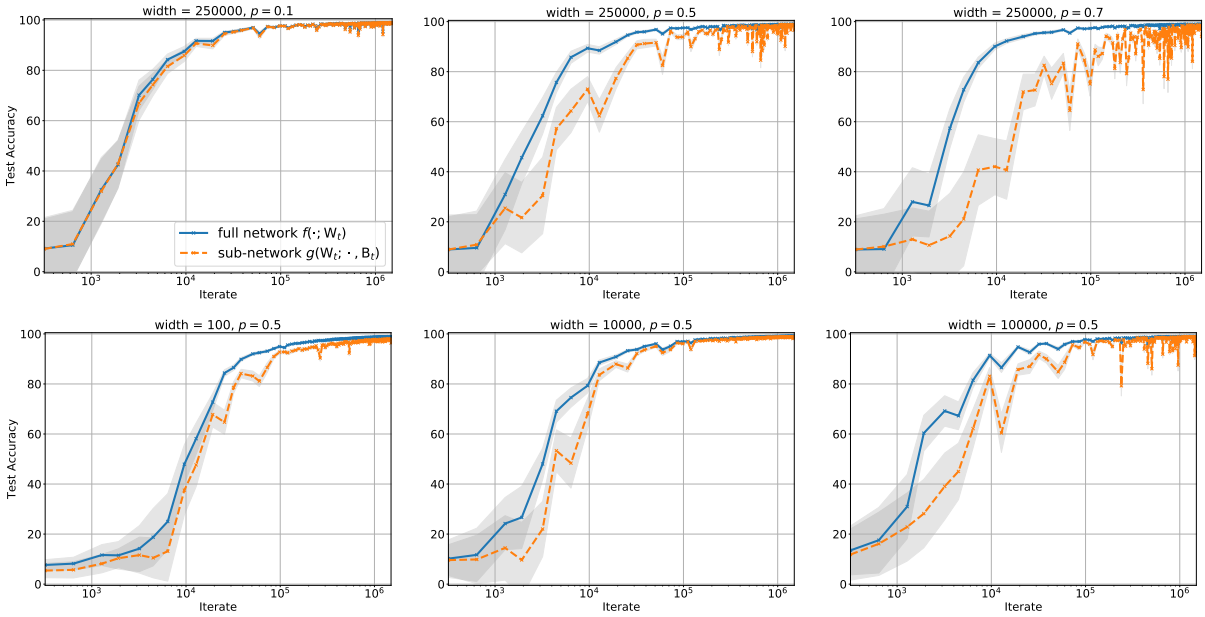

Figure 5-2. Test accuracy of the full network $f(\cdot; \mathrm{W}_t)$ as well as the sub-networks $g(\mathrm{W}_t; \cdot, \mathrm{B}_t)$ drawn by dropout iterates, as a function of number of iterations t, for **(top)** fixed width $m = 250K$ and several dropout rates $1 - p \in \{0.1, 0.5, 0.7\}$; and **(bottom)** fixed dropout rate $1 - p = 0.5$ and several widths $m \in \{100, 10K, 100K\}$.

Plugging the above back in $\mathcal{R}(\mathrm{W}_t; \mathrm{B}_t) \leq 2 \ln 2 Q(\mathrm{W}_t; \mathrm{B}_t)$, and averaging over iterates we have:

$$\frac{1}{T} \sum_{t=1}^{T} \mathcal{R}(\mathrm{W}_t; \mathrm{B}_t) \leq \frac{16 \ln 2 \lambda^2}{\eta T} + \frac{8 \ln 2 \ln 1/\delta}{T}$$

which completes the proof. □

5.5 Experimental Results

The goal of this section is to investigate if dropout indeed compresses the model, as predicted by Theorem 15. In particular, we seek to understand if the (sparse) dropout sub-networks $g(\mathrm{W}; \cdot, \mathrm{B})$ – regardless of being visited by dropout during the training – are comparable to the full network $f(\cdot; \mathrm{W})$, in terms of the test accuracy.

We train a convolutional neural network with a dropout layer on the top hidden layer, using cross-entropy loss, on the MNIST dataset. The network consists of two convolutional layers with max-pooling, followed by three fully-connected layers. All

the activations are ReLU. We use a constant learning rate $\eta = 0.01$ and batch-size equal to 64 for all the experiments. We train several networks where except for the top layer widths ($m \in \{100, 500, 1K, 5K, 10K, 50K, 100K, 250K\}$), all other architectural parameters are fixed. We track the test accuracy over 25 epochs as a function of number of iterations, for the full network, the sub-networks visited by dropout, as well as random but fixed sub-networks that are drawn independently, using the same dropout rate. We run the experiments for several values of the dropout rate, $1 - p \in \{0.1, 0.2, 0.3, \ldots, 0.9\}$.

Figure 5-2 shows the test accuracy of the full network $f(\cdot; \mathrm{W}_t)$ (blue, solid curve) as well as the dropout iterates $g(\mathrm{W}_t; \cdot, \mathrm{B}_t)$ (orange, dashed curve), as a function of the number of iterations. Both curves are averaged over 50 independent runs of the experiment; the grey region captures the standard deviation. It can be seen that the (sparse) sub-networks drawn by dropout during the training, are indeed comparable to the full network in terms of the generalization error. As expected, the gap between the full network and the sparse sub-networks is higher for narrower networks, and for higher dropout rates. This figure verifies our compression result in Theorem 15.

Next, we show that dropout also generalizes to sub-networks that were not observed during the training. In other words, random sub-networks drawn from the same Bernoulli distribution, also performed well. We run the following experiment on the same convolutional network architecture described above with widths $m \in \{100, 1K, 10K, 100K\}$. We draw 100 sub-networks $g(\mathrm{W}; \cdot, \mathrm{B}_1), \ldots, g(\mathrm{W}; \cdot, \mathrm{B}_{100})$, corresponding to diagonal Bernoulli matrices $\mathrm{B}_1, \ldots, \mathrm{B}_{100}$, all generated by the same Bernoulli distribution used at training (Bernoulli parameter $p = 0.2$, i.e., dropout rate $1 - p = 0.8$). In Figure 5-3, we plot the generalization error of these sub-networks as well as the full network as a function of iteration number, as orange and blue curves, respectively. We observe that, as the width increases, the sub-networks become increasingly more competitive; it is remarkable that the effective width of these *typical*

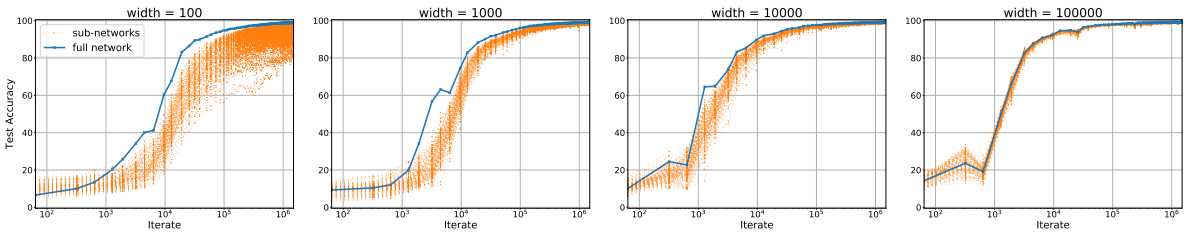

Figure 5-3. Test accuracy of the full network $f(\cdot; \mathrm{W}_t)$ as well as 100 random sub-networks $g(\mathrm{W}_t; \cdot, \mathrm{B}_1), \ldots, g(\mathrm{W}_t; \cdot, \mathrm{B}_{100})$ with dropout rate $1 - p = 0.8$, as a function of number of iterations t, for several width $m \in \{100, 1K, 10K, 100K\}$.

sub-networks are only $\approx 1/5$ of the full network.

5.6 Discussion

We leveraged recent advances in the theory of deep learning in over-parameterized settings and extended convergence guarantees and generalization bounds for GD-based methods with explicit regularization due to dropout. We gave precise non-asymptotic convergence rates for achieving ϵ-subotimality in the test error via dropout training in two-layer ReLU networks. We also showed that dropout training implicitly compresses the network – there exists a sub-network, i.e., one of the iterates of dropout training, that can generalize as well as any complete network.

Most of the results in the literature of over-parameterized neural networks focus on GD-based methods without any explicit regularization. On the other hand, recent theoretical investigations have challenged the view that *implicit bias* due to such GD-based methods can explain generalization in deep learning [DFKL20]. Therefore, it seems crucial to explore algorithmic regularization techniques in over-parameterized neural networks. In this chapter we take a step towards understanding a popular algorithmic regularization technique in deep learning. In particular, assuming that the data distribution is separable in the RKHS induced by the neural tangent kernel, we presents precise iteration complexity results for dropout training in two-layer ReLU networks using the logistic loss.

We see natural extensions of our results in two directions. First, our analysis holds in the lazy regime, where network weights stay close the initialization; it is important to investigate generalization and compression due to dropout beyond the lazy regime. Second, we show empirically, that compression is not limited to subnetworks that are visited at the time of training; typical subnetworks sampled according to the dropout pattern also generalize well. A formal proof of this observation can be an interesting direction for future work.

Chapter 6

Robustness Guarantees for Adversarial Training

Despite the tremendous success of local-search heuristics in deep learning – as we rigorously argued for in the previous chapters – they can result in models that are highly susceptible to adversarial examples; imperceptible perturbations of data that are incorrectly classified by the model [SZS+14]. Such vulnerability limits the deployment of neural networks-based systems, especially in safety-critical applications such as autonomous driving. In recent years, *robust learning* has been a central theme in machine learning, where the goal is to find models that yield reliable predictions on test data notwithstanding adversarial perturbations.

A principled approach to training models that are robust to adversarial examples that has emerged in recent years is that of *adversarial training* [MMS+18]. Adversarial training formulates learning as a min-max optimization problem wherein the 0-1 classification loss is replaced by a convex surrogate such as the cross-entropy loss, and alternating local-search heuristics are used to solve the resulting saddle point problem.

In this chapter, we turn our focus towards the *robustness* due to local-search heuristics used in adversarial training. Despite the empirical success of such heuristics in adversarial training [MMS+18, ACW18, CW17], our understanding of their theoretical underpinnings remains limited. From a practical standpoint, it is remarkable

that gradient based techniques can efficiently solve both (1) the inner maximization problem to find adversarial examples, and (2) the outer maximization problem to impart robust generalization. On the other hand, a theoretical analysis is challenging because (1) both the inner- and outer-level optimization problems are non-convex, and (2) it is unclear a-priori if solving the min-max optimization problem would even guarantee robust generalization.

Here, we take a step towards a better understanding of local-search heuristics in adversarial training. In particular, under a margin separability assumption, we provide robust generalization guarantees for two-layer neural networks with Leaky ReLU activation trained using adversarial training. Our key contributions are as follows.

1. We identify a disconnect between the robust learning objective and the min-max formulation of adversarial training. This observation inspires a simple modification of adversarial training – we propose *reflecting* the surrogate loss about the origin in the inner maximization phase when searching for an "optimal" perturbation vector to attack the current model.

2. We provide convergence guarantees for PGD attacks on two-layer neural networks with leaky ReLU activation. This is the first of its kind in the literature.

3. We give global convergence guarantees and establish learning rates for adversarial training for two-layer neural networks with Leaky ReLU activation function. Notably, our guarantees hold for *any bounded initialization* and *any width* – a property that is not present in the previous works in the neural tangent kernel (NTK) regime [GCL+19, ZPD+20].

4. We provide extensive empirical evidence showing that reflecting the surrogate loss in the inner loop does not have a significant impact on the test time performance of the adversarially trained models.

The rest of the chapter is organized as follows. In Section 6.1, we survey the related work. In Section 6.2, we formally introduce the problem setup and adversarial training, and introduce some additional notations. We give the main theoretical results of the chapter in Section 6.3, and present the proofs of the main results in Section 6.4. Finally, in Section 6.5, we provide supporting empirical evidence for our theoretical results.

6.1 Related Work

Adversarial training of linear models was recently studied by [CRWP19, LXXZ20, ZFG21]. In particular, [CRWP19, LXXZ20] give robust generalization error guarantees for adversarially trained linear models under a margin separability assumption. The hard margin assumption was relaxed by [ZFG21] who give robust generalization guarantees for distributions with agnostic label noise. We note that the optimal attack for linear models has a simple closed-form expression, which mitigates the challenge of analyzing the inner loop PGD attack. In contrast, one of our main contributions is to give convergence guarantees for the PGD attack. Nonetheless, as the Leaky ReLU activation function can also realize the identity map for $\alpha = 1$, our results also provide robust generalization error guarantees for training linear models.

Most related to our results are the works of [GCL+19] and [ZPD+20], which study the convergence of adversarial training in non-linear neural networks. Under specific initialization and width requirements, these works guarantee small robust training error with respect to the attack that is used in the inner-loop, without explicitly analyzing the convergence of the attack. [GCL+19] assume that the activation function is smooth and require that the width of the network, as well as the overall computational cost, is exponential in the input dimension. The work of [ZPD+20] partially addresses these issues. In particular, their results hold for ReLU neural networks, and they only require the width and the computational cost to be polynomial in the input

parameters.

In a parallel strand of research, a series of previous work study robust generalization in the adversarial settings through the lens of statistical learning theory. In particular, the works of [KL18] and [YKB19] study the adversarial Rademacher complexity of linear models under ℓ_∞- and general ℓ_p-threat models, respectively. The bounds were further improved later by [AFM20]. Furthermore, [AFM20] and [YKB19] also provide robust Rademacher complexity bounds for the adversarial Rademacher complexity of two-layer and general deep neural networks with ReLU activation.

6.2 Problem Setup

We focus on two-layer networks with m hidden nodes computing $f(\mathrm{x}; \mathrm{a}, \mathrm{W}) = \mathrm{a}^\top \sigma \mathrm{W}\mathrm{x}$, where $\mathrm{W} \in \mathbb{R}^{m \times d}$ and $\mathrm{a} \in \mathbb{R}^m$ are the weights of the first and the second layers, respectively, and $\sigma(z) = \max\{\alpha z, z\}$ is the Leaky ReLU activation function. We randomly initialize the weights a and W such that $\|\mathrm{a}\|_\infty \leq \kappa$ and $\|\mathrm{W}\|_F \leq \omega$. The top linear layer (i.e., weights a) is kept fixed, and the hidden layer (i.e., W) is trained using stochastic gradient descent (SGD).

For simplicity of notation, we represent the network as $f(\mathrm{x}; \mathrm{W})$, suppressing the dependence on the top layer weights. Further, with a slight abuse of notation, we denote the function by $f_{\mathrm{W}}(\mathrm{x})$ when optimizing over the input adversarial perturbations, and by $f_{\mathrm{x}}(\mathrm{W})$ when training the network weights.

Formally, adversarial learning is described as follows. Let $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{\pm 1\}$ denote the input feature space and the output label space, respectively. Let $\mathcal{D}$ be an unknown joint distribution on $\mathcal{X} \times \mathcal{Y}$. For any fixed $\mathrm{x} \in \mathcal{X}$, we consider norm-bounded adversarial perturbations in the set $\Delta(\mathrm{x}) := \{\delta : \ \|\delta - \mathrm{x}\| \leq \nu\}$, for some fixed noise budget ν.

Given a training sample $\mathcal{S} := \{(\mathrm{x}_i, y_i)\}_{i=1}^n \sim \mathcal{D}^n$ drawn independently and iden-

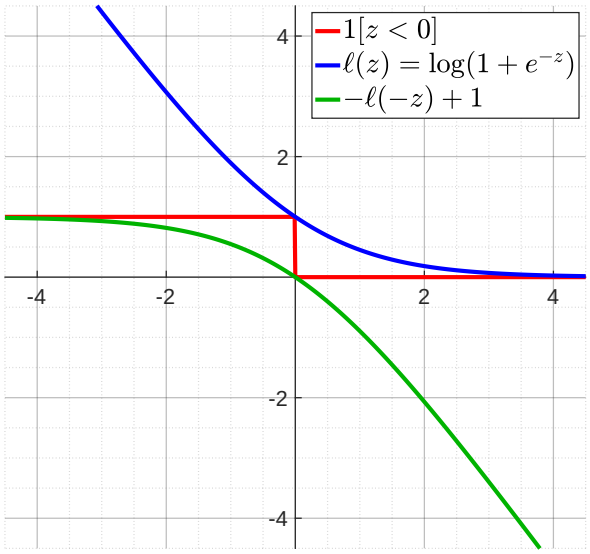

Figure 6-1. The 0-1 loss (red), its convex surrogate, the cross-entropy loss (blue), and the reflected cross-entropy loss (green).

tically from the underlying distribution $\mathcal{D}$, the goal is to find a network with small robust misclassification error

$$L_{\text{rob}}(\mathrm{W}) = \mathbb{E}_{\mathcal{D}} \max_{\delta \in \Delta(\mathrm{x})} \mathbb{I}[y f_{\bar{\mathrm{W}}}(\delta) < 0], \tag{6.1}$$

where $\bar{\mathrm{W}} := \mathrm{W}/\|\mathrm{W}\|_F$ is the weight matrix normalized to have unit Frobenius norm. Note that, due to the homogeneity of Leaky ReLU, such normalization has no effects on the robust error whatsoever.

In adversarial training, the $0-1$ loss inside the expectation is replaced with a convex surrogate such as cross entropy loss $\ell(z) = \log(1 + e^{-z})$, and the expected value is estimated using a sample average:

$$\widehat{L}_{\text{rob}}(\mathrm{W}) := \frac{1}{n} \sum_{i=1}^{n} \max_{\delta_i \in \Delta(\mathrm{x}_i)} \ell(y_i f_{\bar{\mathrm{W}}}(\delta_i)) \tag{6.2}$$

Notwithstanding the conventional wisdom, adversarial training entails *maximizing* an *upper bound* as opposed to a *lower bound* on the $0-1$ loss. In contrast, we propose using a *concave lowerbound* on the $0-1$ loss to solve the inner maximization problem.

Algorithm 5: `Atk` PGD Attack

Input: Sample (x, y), Weights W, Stepsize $\eta_{\texttt{atk}}$, # Iters $T_{\texttt{atk}}$

Initialize $\delta_1 \leftarrow \mathrm{x}$

for $t = 1$ to T **do**

  $\delta_{t+1} \leftarrow \Pi_{\Delta(\mathrm{x})}(\delta_t + \eta_{\texttt{atk}} \nabla_\delta \ell_-(y f_{\mathrm{W}}(\delta_t)))$

end for

Output: δ_τ, where $\tau \in \arg\max_{t \in [T]} \ell_-(y f_{\mathrm{W}}(\delta_t))$

Algorithm 6: `AdvTr` Adversarial Training

Input: Stepsize $\eta_{\texttt{tr}}$, # Iters $T_{\texttt{tr}}$

Initialize a and W_1 such that $\|\mathrm{a}\|_\infty \le \kappa$ and $\|\mathrm{W}_1\|_F \le \omega$

for $t = 1$ to T **do**

  Draw $(\mathrm{x}_t, y_t) \sim \mathcal{D}$

  $\delta_t \leftarrow \texttt{Atk}(\mathrm{W}_t, \mathrm{x}_t, y_t)$

  $\mathrm{W}_{t+1} \leftarrow \mathrm{W}_t - \eta_{\texttt{tr}} \nabla_{\mathrm{W}} \ell(y_t f_{\delta_t}(\mathrm{W}_t))$

end for

Let

$$\ell_-(z) = -\ell(-z) = -\log(1 + e^z)$$

denote the *reflected loss*. In Figure 6-1, we plot the 0-1 loss, the cross-entropy loss, and the reflected cross-entropy loss. Starting from $\delta_1 = \mathrm{x}$, the PGD attack updates iterates via

$$\delta_{t+1} = \Pi_{\Delta(\mathrm{x})}(\delta_t + \eta_{\texttt{atk}} \ell_-(y f_{\mathrm{W}}(\delta_t))),$$

as described in Algorithm 5. We emphasize that the only difference between standard adversarial training and what we propose in Algorithm 6 and Algorithm 5 is that we reflect the loss (about the origin) in Algorithm 5.

6.3 Main Results

We begin by stating the distributional assumption we need for our analysis.

Assumption 3. *Samples* (x, y) *are drawn i.i.d. from an unknown joint distribution* $\mathcal{D}$ *that satisfies the following:*

- $\|\mathrm{x}\| \le R$ *with probability* 1.

- *There exists a unit norm vector* $v_* \in \mathbb{R}^d$, $\|v_*\| = 1$, *such that for* $(x, y) \sim \mathcal{D}$, *we have with probability* 1 *that* $y(v_* \cdot \mathrm{x}) \geq \gamma > 0$.

The first assumption requires that the inputs are bounded, which is standard in the literature and is satisfied for most practical applications. The second assumption implies that $\mathcal{D}$ is linearly separable with margin $\gamma > 0$. While this assumption may seem restrictive, we note that even for training two-layer neural networks using SGD, the convergence guarantees in the hard margin setting were unknown until recently [BGMSS18], and we are not aware of any prior work analyzing adversarial training for two-layer neural networks in this setting.

We consider a slightly weaker version of the robust error. In particular, we are interested in adversarial attacks that can fool the learner with a margin – for some small, non-negative constant β, we define the *β-robust misclassification error* as:

$$L_\beta(\mathrm{W}) = \mathbb{P}\left\{\min_{\delta \in \Delta(\mathrm{x})} y f_{\bar{\mathrm{W}}}(\delta) < -\beta\right\}. \tag{6.3}$$

In particular, as β tends to zero, $L_\beta(\mathrm{W}) \to L_{\mathrm{rob}}(\mathrm{W})$. When β is a small positive constant bounded away from zero, $L_\beta(\mathrm{W})$ captures a more stringent notion of robustness: (x, y) contributes to $L_\beta(\mathrm{W})$ only if there exists an *attack* $\delta \in \Delta(\mathrm{x})$ such that $f_{\bar{\mathrm{W}}}$ *confidently* makes a wrong prediction on δ.

Our bounds depend on several important problem parameters. Before stating the main result of the paper, we remind the reader of these important quantities. R denotes the bound on the input. γ and ν denote the hard margin and the attack size. κ and ω are the bounds on the norm of the parameters a and W at the initialization. Finally, α is the Leaky-ReLU parameter. The following theorem establishes that adversarial training can efficiently find a network with small β-robust error.

Theorem 16 (Convergence of Algorithm 6)**.** *For any* $\epsilon > 0$, *in at most* $T_{tr} \leq \frac{64(R+\nu)^2(1+\omega\gamma\alpha\kappa\sqrt{m\epsilon})}{(\gamma-\nu)^2\alpha^2\epsilon^2}$ *iterations, Algorithm 6 with step-size* $\eta_{tr} \leq \frac{1}{m\kappa^2(R+\nu)^2}$ *finds an*

iterate τ *that, in expectation over* $\{(\mathrm{x}_t, y_t)\}_{t=1}^{T_{tr}}$*, satisfies:*

$$L_\beta(W_\tau) \leq 2\epsilon$$

for any $\beta \geq 2\nu(1-\alpha)\kappa\sqrt{m}$*, provided that for all* $t \in [T]$*,* $\eta_{atk} \leq \frac{1}{\kappa^2 m \|W_t\|_F^2}$ *and* $T_{atk} \geq \frac{8\nu^2}{\eta_{atk}\epsilon}$*.*

A few remarks are in order.

Beyond Neural Tangent Kernel. As opposed to the convergence results in the previous work [GCL+19, ZPD+20] which requires certain initialization and width requirements specific to the NTK regime, our results holds for *any bounded initialization* and *any width* m.

Role of the Robustness Parameter ν**.** Our guarantee holds only when the desired robustness parameter ν is smaller than the distribution margin γ. Furthermore, the iteration complexity increases gracefully as $O(\nu^2/(\gamma-\nu)^2)$ as the *attacks* become stronger, i.e., as the size of adversarial perturbations tends to the margin. Intuitively, as $\nu \to 0$, the attack becomes trivial, and the adversarial training reduces to the standard non-adversarial training. This is fully captured by our results — as $\nu \to 0$, the number of attack iterates $T_{\mathtt{atk}}$ goes to zero, and we recover the overall runtime of $O(\gamma^{-2}\epsilon^{-2})$ as in the previous work [BGMSS18, FCG21].

Computational Complexity. To guarantee ϵ-suboptimality in the β-robust misclassification error, we require $T_{\mathtt{tr}} = O((\gamma-\nu)^{-2}\epsilon^{-2})$ iterations of Algorithm 6. Each iteration invokes the PGD attack in Algorithm 5, which itself requires $T_{\mathtt{atk}} = O(\nu^2/\epsilon)$ gradient updates. Therefore, the overall computational cost of adversarial training to achieve ϵ-suboptimality is $O(\frac{\nu^2}{(\gamma-\nu)^2\epsilon^3})$. Note that $T_{\mathtt{atk}}$ is a purely computational requirement, and the statistical complexity of adversarial training is fully captured by $T_{\mathtt{tr}}$. Remarkably, there is only a mild $O(\gamma^2/(\gamma-\nu)^2)$ statistical overhead for β-robustness, and the computational cost increases gracefully by a multiplicative factor of $O\left(\frac{\nu^2\gamma^2}{(\gamma-\nu)^2\epsilon}\right)$.

Learning Robust Linear Halfspaces. When $\alpha = 1$, the Leaky ReLU activation equals the identity map, and the network reduces to a linear predictor. In this case, we retrieve strong robust generalization guarantees for learning halfspaces, as the lower bound required for β in Theorem 16 vanishes. The following corollary instantiates such a robust generalization guarantee.

Corollary 4. *Let $\kappa = 1/\sqrt{m}$, $\omega = 1/\gamma$, and $\eta_{tr} = (R+\nu)^{-2}$. For any $\epsilon > 0$, in at most $T_{tr} \leq \frac{128(R+\nu)^2}{(\gamma-\nu)^2\epsilon^2}$ iterations, Algorithm 6 finds an iterate τ, that in expectation over $\{(\mathrm{x}_t, y_t)\}_{t=1}^{T_{tr}}$, satisfies $L_{rob}(W_\tau) \leq 2\epsilon$, provided that for all $t \in [T]$, $\eta_{atk} \leq \|W_t\|_F^{-2}$ and $T_{atk} \geq \frac{8\nu^2}{\eta_{atk}\epsilon}$.*

Dependence on the Norm of Iterates. The iteration complexity of Algorithm 5 is inversely proportional to the learning rate $\eta_{\mathtt{atk}}$, and therefore increases with $\|\mathrm{W}_t\|_F^2$. Thus, when calculating the overall computational complexity, one needs to compute an upper bound on the norm of the iterates. As we show in Equation (6.7) in the appendix, it holds for all iterates that $\|\mathrm{W}_{t+1}\|_F^2 \leq \|\mathrm{W}_1\|_F^2 + 3\eta_{\mathtt{tr}}t$. Therefore, if we set $\kappa = 1/\sqrt{m}$ and $\omega^2 = 3/(R+\nu)^2$, we have the following worst-case weight-independent bound on the overall computational cost:

$$\begin{aligned} T &\leq \sum_{t=1}^{T_{\mathtt{tr}}} \frac{8\nu^2}{\eta_{\mathtt{atk}}\epsilon} \leq \sum_{t=1}^{T_{\mathtt{tr}}} \frac{8\nu^2\|\mathrm{W}_t\|_F^2}{\epsilon} \\ &\leq \sum_{t=1}^{T_{\mathtt{tr}}} \frac{8\nu^2(\omega^2 + 3\eta_{\mathtt{tr}}(t-1))}{\epsilon} \leq \sum_{t=1}^{T_{\mathtt{tr}}} \frac{24\nu^2 t}{(R+\nu)^2\epsilon} \\ &\leq \frac{12\nu^2 T_{\mathtt{tr}}^2}{(R+\nu)^2\epsilon} \leq \frac{196608\nu^2(R+\nu)^2}{(\gamma-\nu)^4\alpha^4\epsilon^5}. \end{aligned}$$

Therefore, the worst-case overall computational cost is of order $O(\frac{1}{(\gamma-\nu)^4\epsilon^5})$. We note again that this cost is purely computational – the statistical complexity is still in the order of $O\left(\frac{1}{(\gamma-\nu)^2\epsilon^2}\right)$.

Adversarial Robustness for any β. As we discussed earlier, as $\beta \to 0$, the β-robust error tends to the robust error, i.e., $L_\beta(\mathrm{W}) \to L_{\mathrm{rob}}(\mathrm{W})$. Although Theorem 16 does not hold for $\beta = 0$ (except for the linear case discussed above), it is possible to

guarantee robust generalization with arbitrarily small β, as stated in the following corollary.

Corollary 5. *For any desirable $\beta > 0$, let $\kappa = \frac{\beta}{2\nu(1-\alpha)\sqrt{m}}$. For any $\epsilon > 0$, in at most $T_{tr} \leq \frac{64(R+\nu)^2(1+\omega\gamma\alpha\beta\epsilon/(2\nu(1-\alpha)))}{(\gamma-\nu)^2\alpha^2\epsilon^2}$ iterations, Algorithm 6 with step-size $\eta_{tr} \leq \frac{4\nu^2(1-\alpha)^2}{\beta^2(R+\nu)^2}$ finds an iterate τ that, in expectation over $\{(\mathrm{x}_t, y_t)\}_{t=1}^{T_{tr}}$, satisfies:*

$$L_\beta(W_\tau) \leq 2\epsilon$$

provided that for all $t \in [T]$, $\eta_{atk} \leq \frac{4\nu^2(1-\alpha)^2}{\beta^2\|W_t\|_F^2}$ and $T_{atk} \geq \frac{2(1-\alpha)^2}{\beta^2\|W_t\|_F^2\epsilon}$.

6.3.1 Comparison with Previous Work

The results we present in this chapter are different from that of [GCL+19] and [ZPD+20] in several ways. Here we highlight three key differences.

1. First, while the prior work analyzes the convergence in the NTK setting with specific initialization and width requirements, our results hold for any initialization and width.

2. Second, none of the prior works studies computational aspects of finding an optimal attack vector in the inner loop. Instead, the prior work assumes oracle access to optimal attack vectors. We provide precise iteration complexity results for the projected gradient method (i.e., for the PGD attack) for finding near-optimal attack vectors.

3. Third, the prior works focus on minimizing the robust training loss, whereas we provide computational learning guarantees on the robust generalization error.

6.4 Proofs

In this section, we highlight the key ideas and insights based on our analysis, and give a sketch of the proof of the main result. We begin by recalling that we are interested

in bounding the β-robust misclassification error, which is the probability that for $(\mathrm{x}, y) \sim \mathcal{D}$, a β-*effective attack* exists.

Definition 7 (Effective Attacks)**.** *Given a neural networks with parameters* (a, W) *and a data point* (x, y) *and some constant* $\beta > 0$*, we say that* $\delta_* \in \Delta(\mathrm{x})$ *is a* β*-effective attack if*

$$y f_{\bar{W}}(\delta_*) \leq -\beta, \tag{6.4}$$

where $\bar{W} = W/\|W\|_F$.

Using the definition above, the proof of Theorem 16 crucially depends on the following two facts.

1. Whenever there exists a β-effective attack, Algorithm 5 will efficiently find an approximate attack (Lemma 18),

2. As long as the attack size ν is smaller than the margin γ, adversarial training is no harder than standard training.

The following Lemma gives convergence rates for Algorithm 5 in terms of the negated loss derivative $-\ell'(\cdot)$ under the assumption that an effective attack exists. The negative derivative, $-\ell'(\cdot)$, of the loss function has been used in several previous works to give an upper bound on the error [CG19]; here, we borrow similar ideas from [FCG21]. In particular, as it will become clear later, we will use positivity and monotonicity of $-\ell'(\cdot)$ to give an upper bound on the β-robust loss using Markov's inequality.

Lemma 18. *Let* δ_* *be a* β*-effective attack for a given network with weights* (a, W) *and a given example* (x, y)*, with* $\beta \geq 2\nu(1-\alpha)\kappa\sqrt{m}$*. Then, after* T_{atk} *iterations, PGD with step size* $\eta_{atk} \leq \dfrac{1}{\kappa^2 m \|W\|_F^2}$ *generates an attack* δ_{atk} *such that*

$$-\ell'(y f_W(\delta_*)) \leq -2\ell'(y f_W(\delta_{atk})) + \frac{4\nu^2}{\eta_{atk} T_{atk}}.$$

For simplicity of the notation, in our analysis of the PGD attack, we drop the subscripts denoting the iterates. That is, at the t-th iterate of the outer loop, the weight matrix and the sample point is denoted by W and x, y, instead of W_t and x_t, y_t. We can then use the same variable t to measure the progress of the attack in the inner loop. PGD updates are therefore given by $\Pi_{\Delta(\mathrm{x})}[\delta_t + \eta \nabla \ell_-(y f_{\mathrm{W}}(\delta_t))]$.

Proof of Lemma 18. Let $G := \|\mathrm{W}\|_F$ be the Frobenius norm of the iterate. We have the following bound on the gradient norm:

$$\begin{aligned}
\|\nabla f_{\mathrm{W}}(\delta_t)\| &= \|\sum_{r=1}^{m} a_r \sigma'(\mathrm{w}_r \cdot \delta_t)\mathrm{w}_r\| \\
&\le \sum_{r=1}^{m} \|a_r \sigma'(\mathrm{w}_r \cdot \delta_t)\mathrm{w}_r\| \\
&\le \kappa \sum_{r=1}^{m} \|\mathrm{w}_r\| \\
&\le \kappa\sqrt{m}\|\mathrm{W}\|_F =: G
\end{aligned}$$

We analyze the distance of iterates from δ_*:

$$\begin{aligned}
\|\delta_{t+1} - \delta_*\|^2 - \|\delta_t - \delta_*\|^2 &= \|\Pi[\delta_t + \eta_{\mathtt{att}} \nabla \ell_-(y f_{\mathrm{W}}(\delta_t))] - \delta_*\|^2 - \|\delta_t - \delta_*\|^2 \\
&\le \|\delta_t + \eta_{\mathtt{att}} \nabla \ell_-(y f_{\mathrm{W}}(\delta_t)) - \delta_*\|^2 - \|\delta_t - \delta_*\|^2 \\
&= 2\eta_{\mathtt{att}} \langle \nabla \ell_-(y f_{\mathrm{W}}(\delta_t)), \delta_t - \delta_* \rangle + \eta_{\mathtt{att}}^2 \|\nabla \ell_-(y f_{\mathrm{W}}(\delta_t))\|^2 \\
&= 2\eta_{\mathtt{att}} \ell_-'(y f_{\mathrm{W}}(\delta_t)) y \langle \nabla f_{\mathrm{W}}(\delta_t), \delta_t - \delta_* \rangle + \eta_{\mathtt{att}}^2 \ell_-'(y f_{\mathrm{W}}(\delta_t))^2 \|\nabla f_{\mathrm{W}}(\delta_t)\|^2 \\
&\le 2\eta_{\mathtt{att}} \ell_-'(y f_{\mathrm{W}}(\delta_t)) y \langle \nabla f_{\mathrm{W}}(\delta_t), \delta_t - \delta_* \rangle - \eta_{\mathtt{att}}^2 \ell_-(y f_{\mathrm{W}}(\delta_t)) G^2 \\
&\qquad\qquad (\left|\ell_-'(\cdot)\right| \le \max\{1, -\ell_-(\cdot)\}) \\
&\le 2\eta_{\mathtt{att}} \ell_-'(y f_{\mathrm{W}}(\delta_t)) \left(y f_{\mathrm{W}}(\delta_t) - y f_{\mathrm{W},t}(\delta_*)\right) - \eta_{\mathtt{att}} \ell_-(y f_{\mathrm{W}}(\delta_t)) \\
&\qquad\qquad (\eta_{\mathtt{att}} \le 1/G^2) \\
&\le 2\eta_{\mathtt{att}} \left(\ell_-(y f_{\mathrm{W}}(\delta_t)) - \ell_-(y f_{\mathrm{W},t}(\delta_*))\right) - \eta_{\mathtt{att}} \ell_-(y f_{\mathrm{W}}(\delta_t)) \\
&\qquad\qquad (\text{concavity}) \\
&\le \eta_{\mathtt{att}} \ell_-(y f_{\mathrm{W}}(\delta_t)) - 2\eta_{\mathtt{att}} \ell_-(y f_{\mathrm{W},t}(\delta_*))
\end{aligned}$$

where $f_{\mathrm{W},t}(\delta_*) := \langle \nabla f_{\mathrm{W}}(\delta_t), \delta_* \rangle$. Averaging over iterates, rearranging, and cancelling

telescopic terms, we arrive at:

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T} -\ell_-(yf_{\mathrm{W}}(\delta_t)) &\leq \sum_{t=1}^{T} \frac{\|\delta_t - \delta_*\|^2 - \|\delta_{t+1} - \delta_*\|^2}{\eta_{\texttt{att}} T} - \frac{2}{T}\sum_{t=1}^{T} \ell_-(yf_{\mathrm{W},t}(\delta_*)) \\
&\leq \frac{\|\delta_1 - \delta_*\|^2}{\eta_{\texttt{att}} T} - 2\min_t \ell_-(yf_{\mathrm{W},t}(\delta_*)) && \text{(Telescopic sum)} \\
&\leq \frac{\nu^2}{\eta_{\texttt{att}} T} - 2\min_t \ell_-(yf_{\mathrm{W},t}(\delta_*)) && (\delta_1 = \mathrm{x}, \delta_* \in \Delta(\mathrm{x})) \\
\implies -\ell_-(yf_{\mathrm{W}}(\delta_{\texttt{att}})) &\leq \frac{\nu^2}{\eta_{\texttt{att}} T} - 2\min_t \ell_-(yf_{\mathrm{W},t}(\delta_*))
\end{aligned}$$

Next, we show that if δ_* is an effective attack on f_{W}, then it's also effective on $f_{\mathrm{W},t}$. In other words, we have that:

$$\begin{aligned}
yf_{\mathrm{W},t}(\delta_*) &= y\langle \nabla f_{\mathrm{W}}(\delta_t), \delta_* \rangle \\
&= y\langle \nabla f_{\mathrm{W}}(\delta_*), \delta_* \rangle + y\langle \nabla f_{\mathrm{W}}(\delta_t) - \nabla f_{\mathrm{W}}(\delta_*), \delta_* \rangle \\
&= yf_{\mathrm{W}}(\delta_*) + y\langle \nabla f_{\mathrm{W}}(\delta_t) - \nabla f_{\mathrm{W}}(\delta_*), \delta_* \rangle
\end{aligned}$$

Let $\mathcal{I}_t := \{j \in [m] | \sigma'(\mathrm{w}_j \cdot \delta_t) \neq \sigma'(\mathrm{w}_j \cdot \delta_*)\}$ be the set of nodes whose pre-activation sign at time t is different from the optimal δ_*. The second term can be bounded as follows:

$$\begin{aligned}
y\langle \nabla f_{\mathrm{W}}(\delta_t) - \nabla f_{\mathrm{W}}(\delta_*), \delta_* \rangle &= y\sum_{j=1}^{m} a_j \left(\sigma'(\mathrm{w}_j \cdot \delta_t) - \sigma'(\mathrm{w}_j \cdot \delta_*)\right) \mathrm{w}_j \cdot \delta_* \\
&\leq |y| \sum_{j=1}^{m} \left| a_j \left(\sigma'(\mathrm{w}_j \cdot \delta_t) - \sigma'(\mathrm{w}_j \cdot \delta_*)\right) \mathrm{w}_j \cdot \delta_* \right| \\
&= \kappa \sum_{j \notin \mathcal{I}_t} \left|\sigma'(\mathrm{w}_j \cdot \delta_t) - \sigma'(\mathrm{w}_j \cdot \delta_*)\right| \left|\mathrm{w}_j \cdot \delta_*\right| \\
&\leq (1-\alpha)\kappa \sum_{j \notin \mathcal{I}} \left|\mathrm{w}_j \cdot \delta_* - \mathrm{w}_j \cdot \delta_t\right| && (j \notin \mathcal{I}_t) \\
&\leq (1-\alpha)\kappa \sum_{j \notin \mathcal{I}} \|\mathrm{w}_j\| \|\delta_* - \delta_t\| && \text{(Cauchy-Schwarz)} \\
&\leq (1-\alpha)\kappa\nu\sqrt{m}\|\mathrm{W}\|_F && \text{(Cauchy-Schwarz)}
\end{aligned}$$

Given that there exist a β-effective, i.e. there exist δ_* such that $yf_{\mathrm{W}}(\delta_*) \leq -\beta\|\mathrm{W}\|_F$,

we have:

$$\begin{aligned} y f_{\mathrm{W},t}(\delta_*) &\le y f_{\mathrm{W}}(\delta_*) + (1-\alpha)\kappa\nu\sqrt{m}\|\mathrm{W}\|_F \\ &\le y f_{\mathrm{W}}(\delta_*) + \frac{\beta}{2}\|\mathrm{W}\|_F && (\nu \le \tfrac{\beta}{2(1-\alpha)\kappa\sqrt{m}}) \\ &\le \frac{1}{2} y f_{\mathrm{W}}(\delta_*). \end{aligned}$$

Together with the previous inequality on the average of instantaneous loss, we arrive at:

$$\begin{aligned} -\ell_-(y f_{\mathrm{W}}(\delta_{\mathtt{att}})) &= \min_{t\in[T]} -\ell_-(y f_{\mathrm{W}}(\delta_t)) \\ &\le \frac{1}{T}\sum_{t=1}^{T} -\ell_-(y f_{\mathrm{W}}(\delta_t)) \\ &\le \frac{\nu^2}{\eta_{\mathtt{att}} T} - 2\ell_-(y f_{\mathrm{W}}(\delta_*)/2). \end{aligned}$$

Therefore, we have

$$2\ell_-(y f_{\mathrm{W}}(\delta_*/2)) \le \ell_-(y f_{\mathrm{W}}(\delta_{\mathtt{att}})) + \frac{\nu^2}{\eta_{\mathtt{att}} T}. \tag{6.5}$$

Next, we show that for any z, z', and $\epsilon > 0$, the inequality $2\ell_-(z/2) \le \ell_-(z') + \epsilon$ implies that $-\ell'(z) \le -2\ell'(z') + 4\epsilon$. The following inequalities hold true:

$$\begin{aligned} & 2\ell_-(z/2) \le \ell_-(z') + \epsilon' \\ \implies & -2\log(1+e^{z/2}) \le -\log(1+e^{z'}) + \epsilon' \\ \implies & 2\log(\frac{1}{1+e^{z/2}}) \le \log(\frac{e^{\epsilon'}}{1+e^{z'}}) \\ \implies & \frac{1}{1+e^{z}+2e^{z/2}} \le \frac{e^{\epsilon'}}{1+e^{z'}} \\ \implies & \frac{e^{-z}}{1+e^{-z}+2e^{-z/2}} \le \frac{e^{-z'}}{1+e^{-z'}}\cdot e^{\epsilon'} \\ \implies & \frac{1}{2}\frac{e^{-z}}{1+e^{-z}} \le \frac{e^{-z'}}{1+e^{-z'}}\cdot e^{\epsilon'} && (2e^{-z/2} \le 1+e^{-z}) \\ \implies & -\ell'(z) \le -\ell'(z')\cdot 2e^{\epsilon'} && (\text{definition of } \ell'(\cdot)) \\ \implies & -\ell'(z) \le -\ell'(z') 2(1+2\epsilon') && (e^z \le 1+2z \text{ for all } z\in[0,1]) \\ \implies & -\ell'(z) \le -2\ell'(z') + 4\epsilon' && (-\ell'(\cdot) \le 1) \end{aligned}$$

Let $z_* := yf_{\mathrm{W}}(\delta_*)$, $z_{\mathtt{att}} := yf_{\mathrm{W}}(\delta_{\mathtt{att}})$ and $\epsilon' := \frac{\nu^2}{\eta_{\mathrm{att}} T}$. Using the above inequality, sub-optimality in terms of $\ell_-(\cdot)$, as given in Equation (6.5), implies sub-optimality in terms of $-\ell'(\cdot)$:

$$2\ell_-(z_*/2) \leq \ell_-(z_{\mathtt{att}}) + \epsilon' \implies -\ell'(z_*) \leq -2\ell'(z_{\mathtt{att}}) + 4\epsilon'$$

That is, for *any* $(\mathrm{x}, y) \sim \mathcal{D}$, it holds with probability one that

$$-\ell'(yf_{\mathrm{W}}(\delta_*)) \leq -2\ell'(yf_{\mathrm{W}}(\delta_{\mathtt{att}})) + \frac{4\nu^2}{\eta_{\mathrm{att}} T}. \tag{6.6}$$

□

Finally, we show that robust training is not much harder than standard training if the attack size is smaller than the margin. In particular, the next Lemma establishes that the expected value of the negative loss derivative eventually becomes arbitrarily small.

Lemma 19. *For any $\epsilon > 0$, Algorithm 6 with stepsize $\eta_{tr} \leq m^{-1}\kappa^{-2}(R+\nu)^{-2}$ finds an iterate τ that, in expectation over $\{(\mathrm{x}_t, y_t)\}_{t=1}^{T_{tr}}$, satisfies:*

$$\mathbb{E}_{\mathcal{D}}[-\ell'(yf_{W_\tau}(\delta_{atk}(\mathrm{x})))] \leq \epsilon$$

in at most $T_{tr} \leq \dfrac{4(1 + \| W_1\|_F \gamma\alpha\kappa\sqrt{m\epsilon})}{\eta_{tr}(\gamma-\nu)^2\alpha^2\kappa^2 m\epsilon^2}$ iterations.

We remark that the result in Lemma 19 holds for *any* attack algorithm `Atk`, as long as it respects the condition $\delta_{\mathtt{atk}}(\mathrm{x}) \in \Delta(\mathrm{x})$ for all x.

We need the following additional notations for the proof of Lemma 19. Let $\widehat{G_t^2} = \|\mathrm{W}_t\|_F^2$ and $\widehat{H_t} := \langle \mathrm{W}_t, \mathrm{V}_* \rangle$ denote the squared-norm of the iterates and the correlation between the iterates and V_*, respectively. Let $\mathbb{E}_{\mathtt{AdvTr}}[\cdot]$ denote the expectation over a random draw of samples $(\mathrm{x}_i, y_i)_{i=1}^t$ for Adversarial Training given in Algorithm 6, and let $H_t := \mathbb{E}_{\mathtt{AdvTr}}[\widehat{H_t}]$ and $G_t^2 := \mathbb{E}_{\mathtt{AdvTr}}[\widehat{G_t^2}]$ be the corresponding population version of $\widehat{H_t}$ and $\widehat{G_t^2}$.

Proof of Lemma 19. Let $\mathrm{V}_* \in \mathbb{R}^{m\times d}$ be such that $\mathrm{v}_r = \frac{1}{\sqrt{m}}\operatorname{sgn}(a_r)\mathrm{v}_*$. $\widehat{H}_t$ evolves as:

$$\begin{aligned}
\widehat{H}_{t+1} &= \langle \mathrm{W}_{t+1}, \mathrm{V}_* \rangle \\
&= \langle \mathrm{W}_t - \eta_{\mathrm{tr}} \nabla \ell(y_t f_{\delta_t}(\mathrm{W}_t)), \mathrm{V}_* \rangle \\
&= \widehat{H}_t - \eta_{\mathrm{tr}} \ell'(y_t f_{\delta_t}(\mathrm{W}_t)) y_t \langle \nabla_{\mathrm{W}} f_{\delta_t}(\mathrm{W}_t), \mathrm{V}_* \rangle
\end{aligned}$$

Recall that $\frac{\partial}{\partial \mathrm{w}_r} f_\delta(\mathrm{W}) = a_r \sigma'(\mathrm{w}_r \cdot \delta)\delta$. Therefore, we have that:

$$\begin{aligned}
y_t \langle \nabla_{\mathrm{W}} f_{\delta_t}(\mathrm{W}_t), \mathrm{V}_* \rangle &= y_t \sum_{r=1}^{m} \langle a_r \sigma'(\mathrm{w}_r \cdot \delta)\delta, \frac{1}{\sqrt{m}} \operatorname{sgn}(a_r)\mathrm{v}_* \rangle \\
&= \frac{\kappa}{\sqrt{m}} \sum_{r=1}^{m} \sigma'(\mathrm{w}_r \cdot \delta) y_t \langle \delta, \mathrm{v}_* \rangle
\end{aligned}$$

Note that

$$\begin{aligned}
y_t \langle \delta, \mathrm{v}_* \rangle &= y_t \langle \mathrm{x}, \mathrm{v}_* \rangle + y_t \langle \delta - \mathrm{x}, \mathrm{v}_* \rangle \\
&\geq \gamma - |y_t \langle \delta - \mathrm{x}, \mathrm{v}_* \rangle| \\
&\geq \gamma - \|\delta - \mathrm{x}\| \|\mathrm{v}_*\| \\
&\geq \gamma - \nu
\end{aligned}$$

On the other hand, for leaky ReLU, it holds that $\sigma'(\cdot) \geq \alpha$. Therefore, we arrive at

$$\begin{aligned}
y_t \langle \nabla_{\mathrm{W}} f_{\delta_t}(\mathrm{W}_t), \mathrm{V}_* \rangle &\geq \frac{\kappa}{\sqrt{m}} \sum_{r=1}^{m} \alpha(\gamma - \nu) \\
&= \kappa \alpha \sqrt{m}(\gamma - \nu)
\end{aligned}$$

Therefore, we have that

$$\begin{aligned}
&\widehat{H}_{t+1} \geq \widehat{H}_t - \eta_{\mathrm{tr}} \ell'(y_t f_{\delta_t}(\mathrm{W}_t)) \alpha \kappa \sqrt{m}(\gamma - \nu) \\
\implies &H_{T+1} \geq H_1 - \eta_{\mathrm{tr}}(\gamma - \nu)\alpha\kappa\sqrt{m} \sum_{t=1}^{T} \mathbb{E}_{\texttt{AdvTr}} \ell'(y_t f_{\delta_t}(\mathrm{W}_t))
\end{aligned}$$

where the implication follows from taking expectation $\mathbb{E}_{\texttt{AdvTr}}[\cdot]$ on both sides. The gradient norm is bounded as follows:

$$\|\nabla_{\mathrm{W}} f_{\delta_t}(\mathrm{W}_t)\|^2 = \sum_{j=1}^{m} \|a_j \sigma'(\mathrm{w}_{j,t} \cdot \delta_t)\delta_t\|^2 \leq m\kappa^2(R + \nu)^2$$

Next, we analyze the norm of the iterates, i.e., $\widehat{G}_t^2 = \|\mathrm{W}_t\|_F^2$. It is also easy to verify that $-\ell'(z)z = \frac{ze^{-z}}{1+e^{-z}} = \frac{z}{1+e^z} \leq 1$. We have:

$$\begin{aligned}
\widehat{G}_{t+1}^2 &= \|\mathrm{W}_t - \eta_{\mathrm{tr}} \nabla \ell(y_t f_{\delta_t}(\mathrm{W}_t))\|^2 \\
&= \|\mathrm{W}_t\|_F^2 + \eta_{\mathrm{tr}}^2 \|\nabla \ell(y_t f_{\delta_t}(\mathrm{W}_t))\|^2 - 2\eta_{\mathrm{tr}} \ell'(y_t f_{\delta_t}(\mathrm{W}_t)) y_t \nabla_{\mathrm{W}} f_{\delta_t}(\mathrm{W}_t) \cdot \mathrm{W}_t \\
&= \widehat{G}_t^2 + \eta_{\mathrm{tr}}^2 \ell'(y_t f_{\delta_t}(\mathrm{W}_t))^2 \|\nabla_{\mathrm{W}} f_{\delta_t}(\mathrm{W}_t))\|^2 - 2\eta_{\mathrm{tr}} \ell'(y_t f_{\delta_t}(\mathrm{W}_t)) y_t f_{\delta_t}(\mathrm{W}_t) \\
&\leq \widehat{G}_t^2 + \eta_{\mathrm{tr}}^2 m \kappa^2 (R+\nu)^2 + 2\eta_{\mathrm{tr}} && (-\ell'(z)z \leq 1) \\
&\leq \widehat{G}_t^2 + 3\eta_{\mathrm{tr}} && (\eta_{\mathrm{tr}} \leq m^{-1}\kappa^{-2}(R+\nu)^{-2})
\end{aligned}$$

Therefore, taking expectation $\mathbb{E}_{\texttt{AdvTr}}$ on both sides, we have that

$$G_{T+1}^2 \leq G_1^2 + 3\eta_{\mathrm{tr}} T, \tag{6.7}$$

and using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we have $G_T \leq G_1 + \sqrt{3\eta_{\mathrm{tr}} T}$. Also, we have that $H_t^2 = \left(\mathbb{E}_{\texttt{AdvTr}} \langle \mathrm{W}_t, \mathrm{V}_* \rangle\right)^2 \leq \mathbb{E}_{\texttt{AdvTr}} \|\mathrm{W}_t\|_F^2 \|\mathrm{V}_*\|_F^2 \leq G_t^2$, so that $|H_t| \leq G_t$. Putting all together, we get:

$$\begin{aligned}
-G_1 - \eta_{\mathrm{tr}}(\gamma-\nu)\alpha\kappa\sqrt{m} \sum_{t=0}^{T-1} \mathbb{E}_{\texttt{AdvTr}} \ell'(y_t f_{\delta_t}(\mathrm{W}_t)) &\leq H_0 - \eta_{\mathrm{tr}}(\gamma-\nu)\alpha\kappa\sqrt{m} \sum_{t=0}^{T-1} \mathbb{E}_{\texttt{AdvTr}} \ell'(y_t f_{\delta_t}(\mathrm{W}_t)) \\
&\leq H_T \\
&\leq G_T \\
&\leq G_1 + \sqrt{3\eta_{\mathrm{tr}} T} \\
-\eta_{\mathrm{tr}}(\gamma-\nu)\alpha\kappa\sqrt{m} \sum_{t=0}^{T-1} \mathbb{E}_{\texttt{AdvTr}} \ell'(y_t f_{\delta_t}(\mathrm{W}_t)) &\leq 2G_1 + 2\sqrt{\eta_{\mathrm{tr}} T}
\end{aligned}$$

We now argue that for any ϵ, there exist an iterate t such that $-\mathbb{E}_{\texttt{AdvTr}} \ell'(y_t f_{\delta_t}(\mathrm{W}_t)) \leq \epsilon$.

Assume otherwise, then we get:

$$\begin{aligned}
\eta_{\text{tr}}(\gamma-\nu)\alpha\kappa\sqrt{m}\epsilon T &\leq -\eta_{\text{tr}}(\gamma-\nu)\alpha\kappa\sqrt{m}\sum_{t=0}^{T-1}\mathbb{E}_{\texttt{AdvTr}}\ell'(y_t f_{\delta_t}(\mathrm{W}_t)) \\
&\leq 2G_1 + 2\sqrt{\eta_{\text{tr}}T} \\
\implies \eta_{\text{tr}}(\gamma-\nu)\alpha\kappa\sqrt{m}\epsilon\tau^2 - 2\sqrt{\eta_{\text{tr}}}\tau - 2G_1 &\leq 0 \\
\implies \tau &\leq \frac{\sqrt{\eta_{\text{tr}}} + \sqrt{\eta_{\text{tr}} + 2G_1\eta_{\text{tr}}\gamma\alpha\kappa\sqrt{m}\epsilon}}{\eta_{\text{tr}}(\gamma-\nu)\alpha\kappa\sqrt{m}\epsilon} \\
\implies T &\leq \frac{4(1+G_1\gamma\alpha\kappa\sqrt{m}\epsilon)}{\eta_{\text{tr}}(\gamma-\nu)^2\alpha^2\kappa^2 m\epsilon^2}
\end{aligned}$$

Therefore, we have that $-\mathbb{E}_{\texttt{AdvTr}}\ell'(y_t f_{\delta_t}(\mathrm{W}_t)) = -\mathbb{E}_{\texttt{AdvTr}}\ell'(y_t f_{\mathrm{W}_t}(\delta_{\texttt{att}}(\mathrm{x}_t))) \leq \epsilon$. Moreover,

$$\begin{aligned}
\mathbb{E}_{\texttt{AdvTr}}[-\ell'(y_t f_{\mathrm{W}_t}(\delta_{\texttt{att}}(\mathrm{x}_t)))] &= \mathbb{E}_{\mathcal{S}_t\sim\mathcal{D}^t}[-\ell'(y_t f_{\mathrm{W}_t}(\delta_{\texttt{att}}(\mathrm{x}_t)))] \\
&\qquad \text{(independence from future samples)} \\
&= \mathbb{E}_{\mathcal{S}_{t-1}\sim\mathcal{D}^{t-1}}\mathbb{E}_{(\mathrm{x}_t,y_t)\sim\mathcal{D}}\left[-\ell'(y_t f_{\mathrm{W}_t}(\delta_{\texttt{att}}(\mathrm{x}_t)))|\mathcal{S}_{t-1}\right] \\
&\quad \text{(Smoothing property of the conditional expectation)} \\
&= \mathbb{E}_{\mathcal{S}}\mathbb{E}_{(\mathrm{x},y)\sim\mathcal{D}}[-\ell'(y f_{\mathrm{W}_t}(\delta_{\texttt{att}}(\mathrm{x})))] \\
&\qquad (\mathrm{W}_t \text{ independent of } (\mathrm{x}_t, y_t) \text{ given } \mathcal{S}_{t-1})
\end{aligned}$$

which completes the proof. □

We are now ready to present the proof of the main result.

Proof of Theorem 16. Recall, that β-robust misclassification error is defined as:

$$\begin{aligned}
L_\beta(\mathrm{W}) &= \mathbb{P}\left\{\min_{\delta\in\Delta(\mathrm{x})} y f_{\bar{\mathrm{W}}}(\delta) < -\beta\right\} \\
&= \mathbb{P}\left\{\min_{\delta\in\Delta(\mathrm{x})} y f_{\mathrm{W}}(\delta) < -\beta\|\mathrm{W}\|_F\right\} \qquad \text{(Homogeneity of } f\text{)}
\end{aligned}$$

A key step in the proof is to give an upper bound on ϵ_β in terms of the attack returned by PGD, i.e., $\delta_{\texttt{atk}(\mathrm{x})}$, rather than the optimal attack $\min_{\delta\in\Delta(\mathrm{x})} y f_{\mathrm{W}}(\delta)$. Lemma 18 does provide us with such an upper bound; however, (1) it only holds in expectation, and 2) it is conditioned on existence of an effective attack at the given example (x, y) and the weights W. Naturally, we can use Markov's inequality to bound

the probability above. In order to address the conditional nature of the result in Lemma 18, we introduce a truncated version of the negative loss derivative. In particular, for any c, let $\ell'_c(z) = \ell'(z)\mathbb{I}[z \leq c]$ be the loss derivative thresholded at c. Note that $z \leq c$ implies that $-\ell'_c(z) \geq -\ell'_c(c)$ – therefore, $\mathbb{P}\{z \leq c\} \leq \mathbb{P}\{-\ell'_c(z) \geq -\ell'_c(c)\}$. Let $\beta_\tau := \beta\|\mathrm{W}_\tau\|_F$, where W_τ is the iterate guaranteed by Lemma 19. We have:

$$\begin{aligned}
L_\beta(\mathrm{W}_\tau) &= \mathbb{P}\left\{\min_{\delta\in\Delta(\mathrm{x})} y f_{\mathrm{W}_\tau}(\delta) \leq -\beta_\tau\right\} \\
&\leq \mathbb{P}\left\{-\ell'_{-\beta_\tau}(\min_{\delta\in\Delta(\mathrm{x})} y f_{\mathrm{W}_\tau}(\delta)) \geq -\ell'_{-\beta_\tau}(-\beta_\tau)\right\} \\
&\leq \frac{\mathbb{E}_{\mathcal{D}}\left[-\ell'_{-\beta_\tau}(\min_{\delta\in\Delta(\mathrm{x})} y f_{\mathrm{W}_\tau}(\delta))\right]}{-\ell'_{-\beta_\tau}(-\beta_\tau)} && \text{(Markov's inequality)} \\
&\leq 2\mathbb{E}_{\mathcal{D}}\left[-\ell'_{-\beta_\tau}(\min_{\delta\in\Delta(\mathrm{x})} y f_{\mathrm{W}_\tau}(\delta))\right] && (-\ell'_{-\beta_\tau}(z) \geq 1/2 \text{ for } z \leq 0)
\end{aligned}$$

Given W_τ, for any $(\mathrm{x}, y) \sim \mathcal{D}$, one of the two following cases can happen:

1. *There exists a β-effective attack.* In this case, by Definition 7, it holds that $\min_{\delta\in\Delta(\mathrm{x})} y f_{\mathrm{W}_\tau}(\delta) \leq -\beta\|\mathrm{W}_\tau\|_F = -\beta_\tau$. Therefore, by definition of the truncated negative loss derivative, it also holds that $-\ell'_{\beta_\tau}(\min_{\delta\in\Delta(\mathrm{x})} y f_{\mathrm{W}_\tau}(\delta)) = -\ell'(\min_{\delta\in\Delta(\mathrm{x})} y f_{\mathrm{W}_\tau}(\delta))$. Now, using Lemma 18, we get that

$$-\ell'_{-\beta_\tau}(\min_{\delta\in\Delta(\mathrm{x})} y f_{\mathrm{W}_\tau}(\delta)) \leq -2\ell'(y f_{\mathrm{W}_\tau}(\delta_{\mathtt{atk}}(\mathrm{x}))) + \frac{4\nu^2}{\eta_{\mathtt{atk}} T_{\mathtt{atk}}} \tag{6.8}$$

2. *There does not exist a β-effective attack.* In this case, by Definition 7, it holds that $\min_{\delta\in\Delta(\mathrm{x})} y f_{\mathrm{W}_\tau}(\delta) > -\beta\|\mathrm{W}_\tau\|_F = -\beta_\tau$. Therefore, by definition of the truncated negative loss derivative, it also holds that $-\ell'_{\beta_\tau}(\min_{\delta\in\Delta(\mathrm{x})} y f_{\mathrm{W}_\tau}(\delta)) = 0$, which is trivially bounded by the upper bound in the first case above, given by Equation (6.8).

Putting back the above cases in the upper bound on the β-robust error, we arrive

at:

$$
\begin{aligned}
L_\beta(\mathrm{W}_\tau) &\leq 2\left(2\mathbb{E}_{\mathcal{D}}[-\ell'(yf_{\mathrm{W}_\tau}(\delta_{\mathtt{atk}}(\mathrm{x})))] + \frac{4\nu^2}{\eta_{\mathtt{atk}}T_{\mathtt{atk}}}\right) && \text{(Lemma 18)}\\
&\leq 2\left(\frac{\epsilon}{2} + \frac{4\nu^2}{\eta_{\mathtt{atk}}T_{\mathtt{atk}}}\right) && \text{(Lemma 19 with the proper choice of } T_{\mathtt{Tr}})\\
&\leq 2\left(\frac{\epsilon}{2} + \frac{\epsilon}{2}\right) = 2\epsilon && (T_{\mathtt{atk}} \geq \tfrac{8\nu^2}{\eta_{\mathtt{atk}}\epsilon})
\end{aligned}
$$

which completes the proof of the main result. □

6.5 Empirical Results

Adversarial training is widely used in training robust models and has been shown to be fairly effective in practice. The goal of this section is not to attest or reproduce previous empirical findings. Instead, since the focus in this paper is on the theoretical analysis of adversarial training in non-linear networks, the goal of this section is merely to empirically study the effect of using reflected loss in Algorithm 5.

The experimental results are organized as follows. First, in Section 6.5.1, we compare the optimal attacks found by a grid search on the surrogate loss and its reflected version. In Section 6.5.2, we empirically study adversarial training with reflected loss in the binary classification setting. Finally, in Section 6.5.3, we generalize the reflected loss – which is the key to our theoretical analysis – to the multi-class classification settings. We then report the results on the CIFAR-10 dataset using a deep residual network.

6.5.1 Grid Search Optimization

We look at the following simple 3-dimensional 3-class classification problem. Consider the point (x, y) where $\mathrm{x} = [3, 2, 1]$ and $y = 1$. We focus on the simplest non-trivial function, i.e., the identity mapping, given by $f(\mathrm{x}) = \mathrm{x}$. Obviously, f correctly assigns x to the first class because the first dimension is larger than the others. Also, a

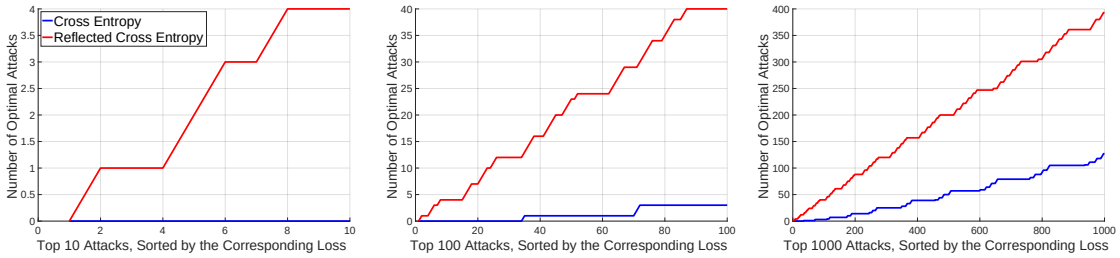

Figure 6-2. Number of the top-k attack vectors that are optimal, i.e., can induce a label flip, for the cross entropy loss (blue) and the reflected version (red), for different values of k: **Left:** $k = 10$, **Middle:** $k = 100$, and **Right:** $k = 1000$.

perturbation of the form $\delta = [-0.501, 0.5, 0]$ with $\|\delta\| = 0.7078$ can flip the label, because $f(\mathrm{x} + \delta) = [2.499, 2.5, 1]$ incorrectly predicts the second class.

We restrict the attack to the set $\{\delta \in (-0.51, +0.51)^3 | \ \|\delta\| \leq 0.7078\}$. We look at every possible attack vector on a grid of size $800 \times 800 \times 800$. We then sort these vectors in a descending order of the corresponding loss function, i.e., the cross entropy loss and its reflected version, and simply count how many of the top-k attack vectors actually induce a label flip. We take this as a measure of how effective is the corresponding loss maximization problem at finding a good attack vector. As we can see in Figure 6-2, the proposed method of maximizing the reflected cross entropy loss is a far more effective way of generating the attacks than maximizing the cross entropy loss.

6.5.2 Binary Classification

Training Alg. \ Attack	FGSM	R-FGSM	PGD-∞	R-PGD-∞	BIM	R-BIM	PGD-2	R-PGD-2
Standard	0.236	0.236	0.033	0.286	0.286	0.286	0.003	0.256
PGD-∞	0.004	0.004	0.005	0.005	0.005	0.005	0.003	0.05
R-PGD-∞	0.003	0.003	0.004	0.004	0.004	0.004	0.002	0.042
PGD-2	0.013	0.013	0.022	0.024	0.024	0.024	0.002	0.034
R-PGD-2	0.004	0.004	0.005	0.006	0.006	0.006	0.0	0.008

Table 6-I. Robust test error of several adversarially trained models with and without reflecting the loss (Standard training, PGD-∞, R-PGD-∞, PGD-2, R-PGD-2), for different attack benchmarks (FGSM, R-FGSM, PGD-∞, R-PGD-∞, BIM, R-BIM, PGD-2, and R-PGD-2).

Experimental Setup. We extract digits 0 and 1 from the MNIST dataset [LBBH98], which provides a (almost) separable distribution, consistent with our theoretical setup. The dataset contains 12665 training samples and 2115 test samples. We evaluate the generalization error as well as the robust generalization error of fully-connected two-layer neural networks which are adversarially trained with and without reflecting the loss. The network has 100 hidden nodes with ReLU activation function.

The outer loop consists of 20 epochs over the training data with batch size equal to 64, randomly shuffled at the beginning of each epoch. The initial learning rate is set to 1, and is decayed by a multiplicative factor of 0.2 every 5 epochs.

We use several benchmark attacks with and without reflecting the loss. The benchmarks include the Fast Gradient Sign Method (FGSM) [GSS15], the Basic Iterative Method (BIM) [KGB17], and the PGD attack with ℓ_2 constraint (PGD-2) and ℓ_∞ constraint (PGD-∞). For each of these attack strategies, we have a corresponding approach that involves reflecting the surrogate loss – we denote the resulting methods as R-FGSM, R-BIM, R-PGD-2, and R-PGD-∞, respectively. The perturbation size for FGSM, PGD-∞, and BIM (and their corresponding reflected version) is set to $\nu = 0.1$. For PGD-2 and R-PGD-2, we let a larger perturbation size of $\nu = 2$ as recommended in Adversarial ML Tutorial.

In the inner-loop, if the attack is iterative, we use a step-decay scheduler with initial step-size of 10, which decreases the step-size every 10 steps by a multiplicative factor of 0.2. In Table 6-I, we report the standard test accuracy as well as the adversarial test accuracy of the trained models over 10 independent random runs of the experiment. Different rows and columns correspond to different training algorithms and different attack models, respectively.

Analysis. We make the following observations in Table 6-I. First, reflecting the loss has a minimal effect on FGSM and BIM attacks, in terms of robust test accuracy of the trained models. In particular, the columns 1 and 2 (similarly columns 5 and 6)

are identical up to the third decimal point.

Second, in PGD-2 attacks, reflecting the loss generally yields a stronger attack – note the striking differences in the last two columns between PGD-2 and R-PGD-2. We observe a milder trend for PGD-∞ attacks, where R-PGD-∞ attacks turns out to be only slightly stronger, except for the standard training setting where reflecting the loss has a huge impact on the robust error.

Third, we would like to remark on the performance of adversarially trained models. We can see that reflecting the loss in general helps robustness. In particular, second and fourth rows (PGD-∞ and PGD-2) are completely dominated by the third and fifth rows (R-PGD-∞ and R-PGD-2), respectively.

Finally, it is notable that even though PGD-2 and PGD-∞ are much weaker than their reflected counterparts, they are still very much competitive in terms of the robustness when used in adversarial training. This, in particular, suggests that finding a "strong" attack is not a necessity for the success of adversarial training.

6.5.3 Extension to multi-label setting

In binary classification using the logistic loss, in essence, adversarial training finds an attack that minimizes the log-likelihood of the correct class. Using the reflected loss, instead, we aim at maximizing the log-likelihood of the wrong class. In a multiclass classification scenario, there are multiple such wrong classes. Therefore, an important design question is which wrong class should be targeted in the attack phase? Here, we focus on the most natural choice: we target the wrong class with the highest log-likelihood. This greedy approach is easy to implement, and has minimal computational overhead over standard adversarial training.

We emphasize that the greedy approach that was described above is sub-optimal, even in a simple linear settings. Intuitively, when the parameters are such that the logits for the true class correlates with the logits for the most likely wrong class,

	Attack Size $\nu = 2/255$							
	Steps $= 2$		Steps $= 4$		Steps $= 16$		Steps $= 32$	
	RA	SA	RA	SA	RA	SA	RA	SA
PGD	14.182	91.254	20.702	90.424	21.014	90.132	20.848	90.09
R-PGD	14.338	91.208	20.726	90.384	20.958	90.06	20.746	89.992
	Attack Size $\nu = 4/255$							
PGD	17.764	90.748	30.344	88.736	37.564	86.65	37.304	86.572
R-PGD	17.162	90.114	30.34	88.826	37.4	86.734	37.374	86.522
	Attack Size $\nu = 8/255$							
PGD	20.064	90.478	34.21	87.746	48.916	78.402	48.936	77.926
R-PGD	20.1	90.564	34.19	87.852	48.792	78.382	48.828	77.982
	Attack Size $\nu = 16/255$							
PGD	16.19	85.908	21.524	86.816	48.722	68.37	45.292	58.526
R-PGD	15.986	89.708	21.362	86.83	48.742	68.456	44.778	58.486

Table 6-II. Robust test accuracy (RA) of adversarially trained models with and without reflecting the loss, for different values of the attack size $\nu \in \{2, 4, 8, 16\}/255$ and number of steps in the attack Steps $\in \{2, 4, 16, 32\}$. The better performance is highlighted in gray, where the intensity corresponds to difference in performance. The clean data standard test accuracy (SA) is also reported for each of the settings.

the greedy approach fails. In particular, consider the following 3-class classification problem on $\mathbb{R}^2$. Let $f_{\mathrm{W}}(\mathrm{x}) = \mathrm{W}\mathrm{x}$, where $\mathrm{W} = [2\mathrm{e}_1, \mathrm{e}_1, 10\mathrm{e}_2] \in \mathbb{R}^{3\times 2}$. Here, e_i denotes the i-th standard basis. Consider the point $\mathrm{x} = [1, 0]$. Clearly, class 1 and 3 have the highest and the smallest likelihoods, respectively. Given a perturbation size $\|\mathrm{x}' - \mathrm{x}\| \leq 0.3$, the likelihood of the second class will never dominate that of the first class:

$$\begin{aligned} \mathrm{w}_1^\top(\mathrm{x}+\delta) = 2\mathrm{e}_1^\top(\mathrm{x}+\delta) &= 2(x_1+\delta_1) \\ &> (x_1+\delta_1) && (x_1 = 1, |\delta_1| \leq 0.3) \\ &= \mathrm{e}_1^\top(\mathrm{x}+\delta) = \mathrm{w}_2^\top(\mathrm{x}+\delta) \end{aligned}$$

Therefore, the greedy approach fails here. Whereas, within the specified perturbation budget, maximizing the likelihood of the third class can indeed find a label-flipping attack. For example, with $\delta = [0, 0.3]$, the point $\mathrm{x}' = [1, 0.3]$ will be assigned to the third class, because $\mathrm{w}_3^\top \mathrm{x}' = 3 > \mathrm{w}_1^\top \mathrm{x} = 2 > \mathrm{w}_2^\top \mathrm{x} = 1$.

Experimental Setup We use adversarial training with and without reflected loss (denoted by R-PGD and PGD, respectively) to train a PreActResNet (PARN) [HZRS16b] on the CIFAR-10 dataset [KH+09]. In the training phase, we conduct experiments for attack size $\nu \in \{2, 4, 8, 16\}/255$. We build on the PyTorch implementation in [ZZK+21], and we follow their experimental setup, which is described next. We use a SGD optimizer with a momentum parameter of 0.9 and weight decay parameter of 5×10^{-4}. We set the batch size to 128 and train each model for 20 epochs. We use a cyclic scheduler which increases the learning rate linearly from 0 to 0.2 within the first 10 epochs and then reduces it back to 0 in the remaining 10 epochs. We report robust test accuracy (RA) of an adversarially-trained model against PGD attacks [MMS+18] (RA-PGD), where we take 50-step PGD with 10 restarts. We report the results for several test-time attack sizes $\nu \in \{2, 4, 16\}/255$.

Analysis. Based on our empirical results, using the (greedy) reflected loss in adversarial training does not significantly impact the standard/robust generalization performance of the learned models.

6.6 Discussion

In this chapter, we studies robust adversarial training of two-layer neural networks as a bi-level optimization problem. We proposed *reflecting* the surrogate loss about the origin in the inner maximization phase when searching for an "optimal" perturbation vector to attack the current model. We gave convergence guarantee for the inner-loop PGD attack and precise iteration complexity results for end-to-end adversarial training, which hold for any width and initialization under a margin assumption. We also provide an empirical study on the effect of reflecting the surrogate loss in real datasets.

Next, we list few natural research directions for future work.

Extension to multiclass setting. In binary classification, which is the focus of

this paper, reflecting the loss about the origin provides a concave lower-bound for the zero one loss (see Figure 6-1). Maximizing the reflected loss then corresponds to maximizing the likelihood of the wrong class. This simple modification enables us to guarantee the convergence of PGD-2 attacks, and yield stronger attacks in our experiments. However, extending this idea to the multiclass setting is not trivial. In particular, the idea of maximizing the likelihood of the wrong class does not trivially generalize to the multiclass setting due to plurality of wrong classes. Nonetheless, as we show in the experimental section, a naive greedy approach to choose a wrong class seems to provide competitive performance in terms of standard/adversarial test error. Is there a simple, principled approach to obtain a lower-bound for the misclassification error in the multiclass setting? It would be interesting to explore theoretical and empirical aspects of such possible extensions.

Beyond β-robustness. The notion of β-robustness is crucial in our analysis. Although we provide robustness guarantees for arbitrarily small positive β (see Corollary 5), our current analysis does not allow for standard robustness guarantees ($\beta = 0$) except for the linear setting ($\alpha = 1$). At a high level, the main challenge here is to guarantee that the attack can always find an adversarial example – if there exists one – regardless of whether the attack is β-effective or not. This is, in particular, challenging to establish for iterative attacks such as PGD, because they can only guarantee getting sufficiently close to an optimal attack in finite time. Therefore, if the optimal attack can just barely flip the sign, the computational time for finding it can grow unboundedly. Therefore, providing robust generalization guarantees ($\beta = 0$) is an interesting research direction that we leave for the future work.

Optimization geometry. In our theoretical results, we focus on PGD-2 attacks, which are based on steepest descent with respect to the ℓ_2 geometry. In our experiments, we also provide empirical results for steepest descent attacks with respect to ℓ_∞ geometry (including FGSM and BIM) on the reflected loss. We leave the theoretical

analysis of such attacks to future work.

Chapter 7

Conclusion

While deep learning continues to advance our technological world, its theoretical underpinnings are not well-understood. In this dissertation, we develop a theory around deep learning, focusing on *regularization* and *robustness* imparted by *algorithmic heuristics*. In this section, we summarize the contributions, and conclude with a discussion on future work.

7.1 Main Contributions

In Chapters 2- 5 of this dissertation, we rigorously argue for *regularization* due to dropout, a popular local-search heuristics in deep learning. We show theoretically and empirically, that in deep regression, dropout induces a nuclear norm penalty, and explicitly biases the learning objective towards low-rank solutions [MAV18, MA19]. We leverage tools from statistical learning theory and prove precise generalization error bounds for dropout training in two important learning problems: matrix sensing, and regression with two-layer ReLU networks [ABMS21]. Finally, we focus on computational and algorithmic aspects of dropout, and give precise iteration complexity bounds for learning two-layer ReLU networks in the lazy regime [MA20].

In Chapter 6, we present a theoretical grounding of *robustness* imparted by local-search heuristics in adversarial training. We provide convergence guarantees for PGD

attacks on two-layer neural networks with leaky ReLU activation, and give global convergence guarantees and establish learning rates for adversarial training.

7.2 Other Contributions

Learning useful representations of data is a major challenges in machine learning. Unsupervised representation learning techniques leverage unlabeled data which is often cheap and plentiful. The goal of these techniques is to learn a representation that captures the intrinsic low dimensional structure in data and disentangles underlying factors of variation. For example, Principal component analysis (PCA) is a ubiquitous representation learning technique in scientific analysis that finds a projection of data that captures as much of the variance in distribution as possible.

In multiview representation learning, multiple "views" of the data measured from different modalities are available. For instance, in web-page classification, the two views can be the text of the page and the hyperlink structure; in automatic speech recognition, one view may be the acoustics features and the other the articulatory measurements [BALHJ12]. In such multiview learning problems, a common representation of the two views is provided by the shared semantic space. In particular, Partial Least Squares (PLS) and Canonical Correlation Analysis (CCA) are two common approaches to extracting this space, which find pairs of maximally covarying and maximally correlated projections of the data in the two views, respectively. Multiview representation learning based on CCA [HSST04] and its non-linear variants [LF00, AABL13, WALB15a, BKG+19] has been shown to be helpful for a variety of tasks [HSST04, WALB15b, AL12, AL13, AL14, BAD16, VCOAA+17, OAVCVB+18, HPM+19, RVDA15].

Instead of formulating representation learning techniques as a problem about a fixed given data set, we argue that these techniques should be studied as a stochastic

optimization problems, especially in a "big data" setting, where the goal is to optimize a population objective based on sample. Such a population-based view of subspace learning has recently been advocated by [ACLS12] and [ACS13] and motivates using Stochastic Approximation (SA) approaches, such as Stochastic Gradient Descent (SGD) and enables a rigorous analysis of their benefits. In a series of previous work, we have developed first-order SA algorithms for ubiquitous representation learning techniques such as PCA [MA18], its robust [MMA18] and non-linear [UMMA18] variants, as well as its multiview cousins PLS [AMM16] and CCA [AMMS17], and provided iteration complexity bounds for the proposed algorithms.

Besides the works above, we have also provided several approximation theoretic results explaining the role of depth in expressivity of deep neural networks [ABMM18], and investigated robustness of SGD against data poisoning attacks [WMA21] as well as adversarial robustness [MA22, WUMA22].

7.3 Future Work

Our theoretical study of regularization and robustness due to local-search heuristics suggests several interesting directions for future research. We detail these research directions in the following.

Trajectory-based analysis. This dissertations contributes to a growing body of literature that leverages trajectory-based analysis of local-search heuristics to explain the success of deep learning algorithms. In particular, In Chapter 5 and Chapter 6, we provide precise iteration complexity results for dropout training and adversarial training, by carefully analyzing the dynamics of the corresponding local-search heuristic at the time of training. However, these results are either limited to the lazy regime, or otherwise require strong distributional assumptions, as is the case for most recent developments in this area. Therefore, extending trajectory-based analysis for local-

search heuristics beyond the lazy regime, and for more general distributional settings, is an important research direction that we leave for future work.

Data-dependent regularization. In Chapter 4 of this thesis, we show that dropout induces a data-dependent regularizer, which directly controls the capacity of the underlying hypothesis class. We gave precise sample complexity results that only depend on the value of the dropout regularizer, without any additional constraints on the norm of the parameters. Our analysis here is tailored to the regularizer due to dropout; given ample empirical evidence arguing for other forms of data-dependent regularizers such as normalization layers and data-augmentation, developing a general theoretical framework for data-dependent capacity control in deep learning is crucial.

Resolving the landscape puzzle. In Chapter 2 of this dissertation, we show that under certain assumptions, all suboptimal critical points in the landscape of dropout objective have negative eigenvalues in their Hessian, which allows dropout to escape saddle points and converge to a global optima. Our results contribute to a large body of literature that attributes the empirical success of local-search heuristics to benign geometric properties of the loss landscape. However, it is still an open question whether poor local minima with suboptimal performance relative to the global optima are common in *practical* deep networks, at least in the space of parameters reachable by local-search heuristics under standard initialization schemes. Therefore, investigating the benign landscape conjecture for practical neural networks is an important direction for future work.

Bibliography

[AABL13] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR, 2013.

[ABB+99] Martin Anthony, Peter L Bartlett, Peter L Bartlett, et al. *Neural network learning: Theoretical foundations*, volume 9. cambridge university press Cambridge, 1999.

[ABMM18] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations (ICLR)*, 2018.

[ABMS21] Raman Arora, Peter Bartlett, Poorya Mianjy, and Nathan Srebro. Dropout: Explicit forms and capacity control. In *International Conference on Machine Learning*, pages 351–361. PMLR, 2021.

[ACGH18] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018.

[ACHL19] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019.

[ACLS12] Raman Arora, Andrew Cotter, Karen Livescu, and Nathan Srebro. Stochastic optimization for PCA and pls. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 861–868. IEEE, 2012.

[ACS13] Raman Arora, Andy Cotter, and Nati Srebro. Stochastic optimization of PCA with capped msg. *Advances in Neural Information Processing Systems*, 26, 2013.

[ACW18] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[ADH+19] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019.

[ADV19] Pranjal Awasthi, Abhratanu Dutta, and Aravindan Vijayaraghavan. On robustness to adversarial examples and polynomial optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[AFM20] Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. Adversarial learning guarantees for linear hypotheses and neural networks. In *International Conference on Machine Learning*, pages 431–441. PMLR, 2020.

[AGCH19] Sanjeev Arora, Noah Golowich, Nadav Cohen, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks.

In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[AH19] Navid Azizan and Babak Hassibi. Stochastic gradient/mirror descent: Minimax optimality and implicit regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[AL12] Raman Arora and Karen Livescu. Kernel cca for multi-view learning of acoustic features using articulatory measurements. In *Symposium on machine learning in speech and language processing*, 2012.

[AL13] Raman Arora and Karen Livescu. Multi-view cca-based acoustic features for phonetic recognition across speakers and domains. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7135–7139. IEEE, 2013.

[AL14] Raman Arora and Karen Livescu. Multi-view learning with supervision for transformed bottleneck features. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2499–2503. IEEE, 2014.

[AM57] Abraham Adrian Albert and Benjamin Muckenhoupt. On matrices of trace zero. *The Michigan Mathematical Journal*, 4(1):1–3, 1957.

[AMM16] Raman Arora, Poorya Mianjy, and Teodor Marinov. Stochastic optimization for multiview representation learning using partial least squares. In *International Conference on Machine Learning*, pages 1786–1794, 2016.

[AMMS17] Raman Arora, Teodor Vanislavov Marinov, Poorya Mianjy, and Nati Srebro. Stochastic approximation for canonical correlation analysis. In

Advances in Neural Information Processing Systems, pages 4775–4784, 2017.

[AZLL19] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems*, pages 6155–6166, 2019.

[AZLS18] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.

[Bac08] Francis R Bach. Consistency of trace norm minimization. *The Journal of Machine Learning Research*, 9:1019–1048, 2008.

[BAD16] Adrian Benton, Raman Arora, and Mark Dredze. Learning multiview embeddings of twitter users. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19, 2016.

[BALHJ12] Sujeeth Bharadwaj, Raman Arora, Karen Livescu, and Mark Hasegawa-Johnson. Multiview acoustic feature learning using articulatory measurements. In *Intl. Workshop on Stat. Machine Learning for Speech Recognition*. Citeseer, 2012.

[Bar98] Peter L Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44(2):525–536, 1998.

[BCM+13] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim vSrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion

attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[Ben09] Yoshua Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[BFT17] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

[BG17] Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. In *International conference on machine learning*, pages 605–614. PMLR, 2017.

[BGMSS18] Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. SGD learns over-parameterized networks that provably generalize on linearly separable data. In *International Conference on Learning Representations*, 2018.

[BH89] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

[BHL18] Peter Bartlett, Dave Helmbold, and Phil Long. Gradient descent with identity initialization efficiently learns positive definite linear transformations. In *ICML*, 2018.

[BKG+19] Adrian Benton, Huda Khayrallah, Biman Gujral, Dee Ann Reisinger, Sheng Zhang, and Raman Arora. Deep generalized canonical correlation analysis. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 1–6, 2019.

[BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[BLL+11] Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 19–26, 2011.

[BLPR19] Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. In *International Conference on Machine Learning*, pages 831–840. PMLR, 2019.

[BM02] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[BNS16] Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3873–3881, 2016.

[BR92] Avrim L Blum and Ronald L Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.

[BRRG18] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018.

[BS13] Pierre Baldi and Peter J Sadowski. Understanding dropout. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2814–2822, 2013.

[CB20] Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020.

[CG19] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pages 10835–10845, 2019.

[CHL+18] Jacopo Cavazza, Benjamin D. Haeffele, Connor Lane, Pietro Morerio, Vittorio Murino, and Rene Vidal. Dropout as a low-rank regularizer for matrix factorization. *Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2018.

[CHM+15] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pages 192–204. PMLR, 2015.

[CM14] Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750, 2014.

[COB18] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. arxiv e-prints, page. *arXiv preprint arXiv:1812.07956*, 2018.

[CRWP19] Zachary Charles, Shashank Rajput, Stephen Wright, and Dimitris Papailiopoulos. Convergence and margin of adversarial training on separable data. *arXiv preprint arXiv:1905.09209*, 2019.

[CT10] Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[CVMG+14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[CW17] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[Cyb89] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[Dan17] Amit Daniely. SGD learns the conjugate kernel class of the network. In *Advances in Neural Information Processing Systems*, pages 2422–2430, 2017.

[DFKL20] Assaf Dauber, Meir Feder, Tomer Koren, and Roi Livni. Can implicit bias explain generalization? stochastic convex optimization as a case study. *arXiv preprint arXiv:2003.06152*, 2020.

[DLL+18] Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018.

[DLSS14] Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. In *Proceedings of*

the forty-sixth annual ACM symposium on Theory of computing, pages 441–448, 2014.

[DSH13] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8609–8613. IEEE, 2013.

[DZPS19] Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

[FC19] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

[FCG21] Spencer Frei, Yuan Cao, and Quanquan Gu. Provable generalization of SGD-trained neural networks of any width in the presence of adversarial label noise. *arXiv preprint arXiv:2101.01152*, 2021.

[FSSS11] Rina Foygel, Ohad Shamir, Nati Srebro, and Ruslan R Salakhutdinov. Learning with the weighted trace-norm under arbitrary sampling distributions. In *Advances in Neural Information Processing Systems*, pages 2133–2141, 2011.

[GBCB16] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[GCL+19] Ruiqi Gao, Tianle Cai, Haochuan Li, Cho-Jui Hsieh, Liwei Wang, and Jason D Lee. Convergence of adversarial training in overparametrized neural networks. *Advances in Neural Information Processing Systems*, 32:13029–13040, 2019.

[GHJY15] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conf. Learning Theory (COLT)*, 2015.

[GJZ17] Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. *arXiv preprint arXiv:1704.00708*, 2017.

[GLM16] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[GLSS18a] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. *arXiv preprint arXiv:1802.08246*, 2018.

[GLSS18b] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Implicit bias of gradient descent on linear convolutional networks. *arXiv preprint arXiv:1806.00468*, 2018.

[GRS18] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pages 297–299, 2018.

[GSS15] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[GWB+17] Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pages 6151–6159, 2017.

[GZ16] Wei Gao and Zhi-Hua Zhou. Dropout rademacher complexity of deep neural networks. *Science China Information Sciences*, 59(7):072104, 2016.

[GZS+19] Aidan N Gomez, Ivan Zhang, Kevin Swersky, Yarin Gal, and Geoffrey E Hinton. Learning sparse networks using targeted dropout. *arXiv preprint arXiv:1905.13678*, 2019.

[HDWF+17] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.

[HK16] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):19, 2016.

[HL15] David P Helmbold and Philip M Long. On the inductive bias of dropout. *Journal of Machine Learning Research (JMLR)*, 16:3403–3454, 2015.

[HL17] David P Helmbold and Philip M Long. Surprising properties of dropout in deep networks. *The Journal of Machine Learning Research*, 18(1):7284–7311, 2017.

[HLL+16] Zhicheng He, Jie Liu, Caihua Liu, Yuan Wang, Airu Yin, and Yalou Huang. Dropout non-negative matrix factorization for independent feature learning. In *Int. Conf. on Computer Proc. of Oriental Languages*. Springer, 2016.

[HLY20] Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization

guarantee. In *International Conference on Learning Representations*, 2020.

[HM16] Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

[Hor91] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[HPM+19] Nils Holzenberger, Shruti Palaskar, Pranava Madhyastha, Florian Metze, and Raman Arora. Learning from multiview correlations in open-domain videos. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8628–8632. IEEE, 2019.

[HSK+12] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[HSST04] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[HV17] Benjamin D Haeffele and Rene Vidal. Structured low-rank matrix factorization: Global optimality, algorithms, and applications. *arXiv preprint arXiv:1708.07850*, 2017.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[HZRS16a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[HZRS16b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

[JGH18] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[JGN+17] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. *arXiv preprint arXiv:1703.00887*, 2017.

[JNM+19] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2019.

[JT18] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018.

[JT19a] Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Conference on Learning Theory*, pages 1772–1798. PMLR, 2019.

[JT19b] Ziwei Ji and Matus Telgarsky. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow ReLU networks. *arXiv preprint arXiv:1909.12292*, 2019.

[Kah99] William Kahan. Only commutators have trace zero, 1999.

[Kaw16] Kenji Kawaguchi. Deep learning without poor local minima. In *Adv in Neural Information Proc. Systems (NIPS)*, 2016.

[KGB14] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[KGB17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world, 2017.

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[KL18] Justin Khim and Po-Ling Loh. Adversarial risk bounds via function transformation. *arXiv preprint arXiv:1810.09519*, 2018.

[KLT11] Vladimir Koltchinskii, Karim Lounici, and Alexandre B Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011.

[Kol01] Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

[KP00] Vladimir Koltchinskii and Dmitriy Panchenko. Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, pages 443–457. Springer, 2000.

[KS09] Adam R Klivans and Alexander A Sherstov. Cryptographic hardness for learning intersections of halfspaces. *Journal of Computer and System Sciences*, 75(1):2–12, 2009.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[KTS+14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[LB18] Thomas Laurent and James Brecht. Deep linear networks with arbitrary loss: All local minima are global. In *International Conference on Machine Learning*, pages 2908–2913, 2018.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LF00] Pei Ling Lai and Colin Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(05):365–377, 2000.

[LL18] Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, pages 8157–8166, 2018.

[LMAPH19] Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda-Pineda, and Radu Horaud. A comprehensive analysis of deep regression. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[LMZ18] Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pages 2–47, 2018.

[LSJR16] Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent converges to minimizers. *arXiv preprint arXiv:1602.04915*, 2016.

[LSO19] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. *arXiv preprint arXiv:1903.11680*, 2019.

[LXS+19] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, pages 8570–8581, 2019.

[LXXZ20] Yan Li, Ethan X.Fang, Huan Xu, and Tuo Zhao. Implicit bias of gradient descent based adversarial training on separable data. In *International Conference on Learning Representations*, 2020.

[MA18] Poorya Mianjy and Raman Arora. Stochastic PCA with ℓ_2 and ℓ_1 regularization. In *International Conference on Machine Learning*, pages 3528–3536, 2018.

[MA19] Poorya Mianjy and Raman Arora. On dropout and nuclear norm regularization. In *International Conference on Machine Learning*, pages 4575–4584, 2019.

[MA20] Poorya Mianjy and Raman Arora. On convergence and generalization of dropout training. In *Advances in Neural Information Processing Systems*, volume 33, pages 21151–21161, 2020.

[MA22] Poorya Mianjy and Raman Arora. Robustness guarantees for adversarially trained neural networks. In *Review*, 2022.

[MAV17] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2498–2507. JMLR. org, 2017.

[MAV18] Poorya Mianjy, Raman Arora, and Rene Vidal. On the implicit bias of dropout. In *International Conference on Machine Learning*, pages 3537–3545, 2018.

[McA13] David McAllester. A PAC-bayesian tutorial with a dropout bound. *arXiv preprint arXiv:1307.2118*, 2013.

[MDFF16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[MKS+15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[MMA18] Teodor Vanislavov Marinov, Poorya Mianjy, and Raman Arora. Streaming principal component analysis in noisy settings. In *International Conference on Machine Learning*, pages 3410–3419, 2018.

[MMS+18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[MZGW18] Wenlong Mou, Yuchen Zhou, Jun Gao, and Liwei Wang. Dropout training, data-dependent regularization, and generalization bounds. In *International Conference on Machine Learning*, pages 3642–3650, 2018.

[Nak19] Preetum Nakkiran. Adversarial robustness may be at odds with simplicity. *arXiv preprint arXiv:1901.00532*, 2019.

[NBS17] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

[NH92] Steven J Nowlan and Geoffrey E Hinton. Simplifying neural networks by soft weight-sharing. *Neural computation*, 4(4):473–493, 1992.

[NK19] Vaishnavh Nagarajan and J. Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[NLB+18] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

[NLG+18] Mor Shpigel Nacson, Jason Lee, Suriya Gunasekar, Nathan Srebro, and Daniel Soudry. Convergence of gradient descent on separable data. *arXiv preprint arXiv:1803.01905*, 2018.

[NS19] Atsushi Nitanda and Taiji Suzuki. Refined generalization analysis of gradient descent for over-parameterized two-layer neural networks with smooth activations on classification problems. *arXiv preprint arXiv:1905.09870*, 2019.

[NSS15] Behnam Neyshabur, Ruslan R Salakhutdinov, and Nati Srebro. Path-SGD: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2422–2430, 2015.

[NTS14] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[NTS15] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

[NTSS17] Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071*, 2017.

[NYC15] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[OAVCVB+18] Juan Rafael Orozco-Arroyave, Juan Camilo Vásquez-Correa, Jesús Francisco Vargas-Bonilla, Raman Arora, Najim Dehak, Phani S Nidadavolu, Heidi Christensen, Frank Rudzicz, Maria Yancheva, H Chinaei, et al. Neurospeech: An open-source software for parkinson's speech analysis. *Digital Signal Processing*, 77:207–221, 2018.

[OBLS14] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[OS20] Samet Oymak and Mahdi Soltanolkotabi. Towards moderate overparameterization: global convergence guarantees for training shallow neural networks. *IEEE Journal on Selected Areas in Information Theory*, 2020.

[PBKL14] Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. Dropout improves recurrent neural networks for handwriting recognition. In *2014 14th international conference on frontiers in handwriting recognition*, pages 285–290. IEEE, 2014.

[PMW+16] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[RFP10] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[RVDA15] Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. Multiview lsa: Representation learning via generalized cca. In *Proceedings of the 2015 conference of the North American chapter of the Association for Computational Linguistics: human language technologies*, pages 556–566, 2015.

[RVM+11] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *In International Conference on Machine Learning*. Citeseer, 2011.

[SCP16] Grzegorz Swirszcz, Wojciech Marian Czarnecki, and Razvan Pascanu. Local minima in training of neural networks. *arXiv preprint arXiv:1611.06310*, 2016.

[SCS20] Albert Senen-Cerda and Jaron Sanders. Almost sure convergence of dropout algorithms for neural networks. *arXiv preprint arXiv:2002.02247*, 2020.

[SHK+14] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1), 2014.

[SHM+16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[SHN+18] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[SK16] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016.

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[SMDH13] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013.

[SMG13] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[SPR18] Arun Suggala, Adarsh Prasad, and Pradeep K Ravikumar. Connecting optimization and regularization paths. *Advances in Neural Information Processing Systems*, 31, 2018.

[SQW16] Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. In *IEEE International Symposium on Information Theory (ISIT)*, pages 2379–2383, 2016.

[SRJ04] Nathan Srebro, Jason Rennie, and Tommi Jaakkola. Maximum-margin matrix factorization. *Advances in neural information processing systems*, 17, 2004.

[SS10] Nathan Srebro and Russ R Salakhutdinov. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. *Advances in neural information processing systems*, 23, 2010.

[SS18] Itay Safran and Ohad Shamir. Spurious local minima are common in two-layer ReLU neural networks. In *International conference on machine learning*, pages 4433–4441. PMLR, 2018.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[SY19] Zhao Song and Xin Yang. Quadratic suffices for over-parametrization via matrix chernoff bound. *arXiv preprint arXiv:1906.03593*, 2019.

[SZS+14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[Tia17] Yuandong Tian. An analytical formula of population gradient for two-layered ReLU network and its applications in convergence and critical point analysis. In *International conference on machine learning*, pages 3404–3413. PMLR, 2017.

[TS14] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[UMMA18] Enayat Ullah, Poorya Mianjy, Teodor V Marinov, and Raman Arora. Streaming kernel PCA with $\tilde{O}(\sqrt{n})$ random features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7322–7332, 2018.

[Vap13] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[VC74] Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition. 1974.

[VCOAA+17] Juan Camilo Vasquez-Correa, Juan Rafael Orozco-Arroyave, Raman Arora, Elmar Nöth, Najim Dehak, Heidi Christensen, Frank Rudzicz, Tobias Bocklet, Milos Cernak, Hamidreza Chinaei, et al. Multi-view representation learning via gcca for multimodal analysis of parkinson's disease. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2966–2970. IEEE, 2017.

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.

[WALB15a] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International conference on machine learning*, pages 1083–1092. PMLR, 2015.

[WALB15b] Weiran Wang, Raman Arora, Karen Livescu, and Jeff A Bilmes. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4590–4594. IEEE, 2015.

[WFWL14] Stefan Wager, William Fithian, Sida Wang, and Percy S Liang. Altitude training: Strong bounds for single-layer dropout. In *Adv. Neural Information Processing Systems*, 2014.

[WK18] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018.

[WLLM19] Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. In *Advances in Neural Information Processing Systems*, pages 9709–9721, 2019.

[WM13] Sida Wang and Christopher Manning. Fast dropout training. In *international conference on machine learning*, pages 118–126, 2013.

[WMA21] Yunjuan Wang, Poorya Mianjy, and Raman Arora. Robust learning for data poisoning attacks. In *International Conference on Machine Learning*, 2021.

[WUMA22] Yunjuan Wang, Enayat Ullah, Poorya Mianjy, and Raman Arora. Adversarial robustness is at odds with lazy training. In *Review*, 2022.

[WWL13] Stefan Wager, Sida Wang, and Percy S Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[WZZ+13] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066, 2013.

[XWZ+18] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.

[YHG+16] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[YKB19] Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *International conference on machine learning*, pages 7085–7094. PMLR, 2019.

[YSJ18] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Small nonlinearities in activation functions create bad local minima in neural networks. In *International Conference on Learning Representations*, 2018.

[YSJ19] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Small ReLU networks are powerful memorizers: a tight analysis of memorization capacity. *Advances in Neural Information Processing Systems*, 32, 2019.

[ZBH+16] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[ZCZG18] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep ReLU networks. *arXiv preprint arXiv:1811.08888*, 2018.

[ZDK+21] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization? In *International Conference on Learning Representations*, 2021.

[ZFG21] Difan Zou, Spencer Frei, and Quanquan Gu. Provable robustness of adversarial training for learning halfspaces with noise. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 13002–13011. PMLR, 18–24 Jul 2021.

[ZL17] Yi Zhou and Yingbin Liang. Critical points of neural networks: Analytical forms and landscape properties. *arXiv preprint arXiv:1710.11205*, 2017.

[ZPD+20] Yi Zhang, Orestis Plevrakis, Simon S Du, Xingguo Li, Zhao Song, and Sanjeev Arora. Over-parameterized adversarial training: An analysis overcoming the curse of dimensionality. In *NeurIPS*, 2020.

[ZSJ+17] Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *International conference on machine learning*, pages 4140–4149. PMLR, 2017.

[ZW18] Ke Zhai and Huan Wang. Adaptive dropout with rademacher complexity regularization. In *International Conference on Learning Representations*, 2018.

[ZZ15] Shuangfei Zhai and Zhongfei Zhang. Dropout training of matrix factorization and autoencoder for link prediction in sparse graphs. In *Proc. of SIAM International Conference on Data Mining (ICDM)*, pages 451–459, 2015.

[ZZK+21] Yihua Zhang, Guanhuan Zhang, Prashant Khanduri, Mingyi Hong, Shiyu Chang, and Sijia Liu. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. *arXiv preprint arXiv:2112.12376*, 2021.

[ZZL15] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

.1 Table of Notations

Numbers, Arrays, and Sets

x	Scalar (integer or real)
x	Vector
X	Matrix
$\mathcal{X}$	Set
$(x)_+$	Scalar $\max\{x, 0\}$
1_d	d-dimensional vector of all ones
I_d	Identity matrix with d rows and d columns
I	Identity matrix with dimensionality implied by context
$[d]$	Set of integers $\{1, \ldots, d\}$
$\binom{\mathcal{S}}{k}$	Set of all k-combinations of a set $\mathcal{S}$

Indexing, Slicing, Flattening, and Vectorizing

x_i	i-th entry of vector x
x_i	i-th column of matrix X
$\mathrm{x}_{:i}$	i-th column of matrix X
$\mathrm{x}_{j:}$	j-th row of matrix X
e_i	i-th standard basis vector
$\mathrm{diag}(\mathrm{X})$	Vector, given by the diagonal entries X
$\mathrm{diag}(\mathrm{x})$	Square, diagonal matrix with diagonal entries given by x

Arrays Operations

$\langle \cdot, \cdot \rangle$	standard inner product, for vectors or matrices
$\|\mathrm{x}\|_p$	p-norm of vector x
Trace(X)	Trace of matrix X
$\|\mathrm{X}\|_2$	Spectral norm of matrix X
$\|\mathrm{X}\|_F$	Frobenius norm of matrix X
$\|\mathrm{X}\|_*$	Nuclear norm of matrix X
$\|\mathrm{X}\|_{p,q}$	q-norm of the p-norm of columns of X
$\lambda_i(\mathrm{X})$	i-th largest eigenvalue of matrix X
$\sigma_i(\mathrm{X})$	i-th largest singular value of matrix X
$(X)_+$	Matrix given by the elementwise application of $(\cdot)_+$ to X $\max\{x, 0\}$
$\|\mathrm{x}\|_{\mathrm{C}}^2$	Mahalonobis norm of x, given by $\mathrm{x}^\top \mathrm{C}\mathrm{x}$, for a positive definite matrix C
$\|\mathrm{X}\|_{\mathrm{C}}^2$	Mahalonobis norm of X, given by $\mathrm{Tr}\left(\mathrm{X}\mathrm{C}\mathrm{X}^\top\right)$, for a positive definite matrix C
$\Pi_{\mathcal{C}}(\mathrm{x})$	Vector $\min_{\mathrm{x}' \in \mathcal{C}} \|\mathrm{x} - \mathrm{x}'\|$, projection of x onto $\mathcal{C}$ with respect to the ℓ_2-norm
$\sqrt{\mathrm{x}}$	elementwise squared root of x
$\mathrm{X}^\dagger$	Moore-Penrose pseudo-inverse of X

Probability and Statistics

$\widehat{\mathbb{E}}[\mathrm{x}]$	Expected value of random variable x
$\widehat{\mathbb{E}}_i[\mathrm{x}_i]$	Empirical average on a sample, given by $\frac{1}{n}\sum_{i \in [n]} \mathrm{x}_i$
C_{x}	Second moment of x, given by $\mathbb{E}[\mathrm{x}\mathrm{x}^\top]$
$\widehat{\mathrm{C}}_{\mathrm{x}}$	Empirical second moment on a sample, given by $\widehat{\mathbb{E}}_i[\mathrm{x}_i \mathrm{x}_i^\top]$

.2 Auxiliary Results

Lemma 20. *For any pair of integers ρ and r, and for any $\lambda \in \mathbb{R}_+$, it holds that*

$$(I_\rho + \frac{\lambda}{r} 1 1^\top)^{-1} = I_\rho - \frac{\lambda}{r + \lambda\rho} 1 1^\top.$$

Lemma 20 is an instance of the Woodbury's matrix identity. Here, we include a proof for completeness.

Proof of Lemma 20. The proof simply follows from the following set of equations.

$$\begin{aligned}(\mathrm{I}_\rho + \frac{\lambda}{r}11^\top)(\mathrm{I}_\rho - \frac{\lambda}{r+\lambda\rho}11^\top) &= \mathrm{I}_\rho + \frac{\lambda}{r}11^\top - \frac{\lambda}{r+\lambda\rho}11^\top - \frac{\lambda^2}{r(r+\lambda\rho)}11^\top 11^\top \\ &= \mathrm{I}_\rho + \left(\frac{\lambda}{r} - \frac{\lambda}{r+\lambda\rho} - \frac{\rho\lambda^2}{r(r+\lambda\rho)}\right)11^\top = \mathrm{I}_\rho\end{aligned}$$

□

Lemma 21. *Let $\lambda > 0$ be a constant. Let $a \in \mathbb{R}^d_+$ such that $a_i \geq a_{i+1}$ for all $i \in [d-1]$. For $r \leq d$, let the function $g : [r] \to \mathbb{R}$ be defined as*

$$g(\rho) := \sum_{i=1}^{\rho}\left(\frac{\lambda\sum_{k=1}^{\rho} a_k}{r+\lambda\rho}\right)^2 + \sum_{i=\rho+1}^{d} a_i^2 + \frac{\lambda}{r}\left(\sum_{i=1}^{\rho}\left(a_i - \frac{\lambda\sum_{k=1}^{\rho} a_k}{r+\lambda\rho}\right)\right)^2.$$

Then $g(\rho)$ is monotonically non-increasing in ρ.

Proof of Lemma 21. Let denote the sum of the top τ elements of a by $\mathrm{h}_\tau = \sum_{i=1}^{\tau} a_i$. Furthermore, let the sum of squared of τ bottom elements of a be denoted by $t_\tau = \sum_{i=\tau+1}^{d} a_i^2$. We can simplify $g(\rho)$ and give it in terms of h_ρ and t_ρ as follows:

$$\begin{aligned}g(\rho) &= \rho\left(\frac{\lambda h_\rho}{r+\lambda\rho}\right)^2 + t_\rho + \frac{\lambda}{r}\left(\left(1 - \frac{\lambda\rho}{r+\lambda\rho}\right)h_\rho\right)^2 \\ &= \frac{\rho\lambda^2 + \lambda r}{(r+\lambda\rho)^2}(h_\rho)^2 + t_\rho = \frac{\lambda h_\rho^2}{r+\lambda\rho} + t_\rho\end{aligned}$$

It suffices to show that $g(\rho+1) \leq g(\rho)$ for all $\rho \in [r-1]$.

$$\begin{aligned}
&g(\rho+1) = \frac{\lambda h_{\rho+1}^2}{r+\lambda\rho+\lambda} + t_{\rho+1} \\
=&\frac{\lambda}{r+\lambda\rho+\lambda}\left(h_\rho^2 + \lambda_{\rho+1}^2(\mathrm{M}) + 2\lambda_{\rho+1}(\mathrm{M})h_\rho\right) \\
&- \lambda_{\rho+1}^2(\mathrm{M}) + t_\rho \\
=&g(\rho) - \frac{\lambda^2 h_\rho^2}{(r+\lambda\rho)(r+\lambda\rho+\lambda)} - \lambda_{\rho+1}^2(\mathrm{M}) \\
&+ \frac{\lambda}{r+\lambda\rho+\lambda}\left(\lambda_{\rho+1}^2(\mathrm{M}) + 2\lambda_{\rho+1}(\mathrm{M})h_\rho\right) \\
=&g(\rho) - \frac{\lambda^2 h_\rho^2}{(r+\lambda\rho)(r+\lambda\rho+\lambda)} - \frac{(r+\lambda\rho)\lambda_{\rho+1}^2(\mathrm{M})}{r+\lambda\rho+\lambda} \\
&+ \frac{\lambda}{r+\lambda\rho+\lambda}\left(2\lambda_{\rho+1}(\mathrm{M})h_\rho\right) \\
=&g(\rho) - \frac{\left(\lambda h_\rho - (r+\lambda\rho)\lambda_{\rho+1}^2(\mathrm{M})\right)^2}{(r+\lambda\rho)(r+\lambda\rho+\lambda)} \leq g(\rho).
\end{aligned}$$

Hence $g(\rho)$ is monotonically non-increasing in ρ. □

Lemma 22 (Khintchine-Kahane inequality)**.** *Let $\{\epsilon_i\}_{i=1}^n$ be i.i.d. Rademacher random variables, and $\{\mathrm{x}\}_{i=1}^n \subset \mathbb{R}^d$. Then there exist a pair of universal constants $c_1, c_2 > 0$ such that*

$$c_1\sqrt{\sum_{i=1}^n \|\mathrm{x}_i\|^2} \leq \mathbb{E}\|\sum_{i=1}^n \epsilon_i \mathrm{x}_i\| \leq c_2\sqrt{\sum_{i=1}^n \|\mathrm{x}_i\|^2}.$$

Theorem 17 (Hoeffding's inequality:[Ver18])**.** *We state Hoeffding's inequality for general Sub-Gaussian random variables, and bounded random variables, as a special case.*

1. **General Sub-Gaussian R.V.:** *Let $X_1, \ldots, X_N$ be independent, mean zero, sub-Gaussian random variables. Then, for every $t \geq 0$, we have*

$$\mathbb{P}\left(\left|\widehat{\mathbb{E}}_i X_i\right| \geq t\right) \leq 2e^{-\frac{ct^2N^2}{\sum_{i=1}^N \|X_i\|_{\psi_2}^2}}$$

2. **Bounded R.V.:** *Let $X_1, \ldots, X_n$ be independent, mean zero random variables.*

Assume that $X_i \in [m_i, M_i]$ for every i. Then, for every $t > 0$, we have

$$\mathbb{P}\{\sum_{i=1}^{n} X_i \geq t\} \leq e^{-\frac{2t^2}{\sum_{i=1}^{n}(m_i - M_i)^2}}$$

Theorem 18 (Theorem 3.1 of [MRT18]). *Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{Z}$ to $[0,1]$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over a sample $\mathcal{S} = \{z_1, \ldots, z_n\}$, the following holds for all $g \in \mathcal{G}$*

$$E[g(z)] \leq \frac{1}{n}\sum_{i=1}^{n} g(z_i) + 2\mathfrak{R}_n(\mathcal{G}) + \sqrt{\frac{\log(1/\delta)}{2n}}$$
$$E[g(z)] \leq \frac{1}{n}\sum_{i=1}^{n} g(z_i) + 2\mathfrak{R}_{\mathcal{S}}(\mathcal{G}) + 3\sqrt{\frac{\log(1/\delta)}{2n}}$$

Theorem 19 (Theorem 10.3 of [MRT18]). *Assume that $\|h - f\|_\infty \leq M$ for all $h \in \mathcal{H}$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over a sample $\mathcal{S} = \{(\mathrm{x}_i, \mathrm{y}_i),\ i \in [n]\}$ of size n, the following inequalities holds uniformly for all $h \in \mathcal{H}$.*

$$\mathbb{E}[|h(\mathrm{x}) - f(\mathrm{x})|^2] \leq \widehat{\mathbb{E}}_i |h(\mathrm{x}_i) - f(\mathrm{x}_i)|^2 + 4M\mathfrak{R}_n(\mathcal{H}) + M^2\sqrt{\frac{\log(2/\delta)}{2n}}$$
$$\mathbb{E}[|h(\mathrm{x}) - f(\mathrm{x})|^2] \leq \widehat{\mathbb{E}}_i |h(\mathrm{x}_i) - f(\mathrm{x}_i)|^2 + 4M\mathfrak{R}_{\mathcal{S}}(\mathcal{H}) + 3M^2\sqrt{\frac{\log(2/\delta)}{2n}}$$

Theorem 20 (Theorem 1 in [FSSS11]). *Assume that $p(i)q(j) \geq \frac{\log(d_2)}{n\sqrt{d_2 d_0}}$ for all $i \in [d_2], j \in [d_0]$. For any $\alpha > 0$, let $\mathcal{M}_\alpha := \{M \in \mathbb{R}^{d_2 \times d_1} :\ \|\operatorname{diag}(\sqrt{p}) M \operatorname{diag}(\sqrt{q})\|_*^2 \leq \alpha\}$ be the class of linear transformations with weighted trace-norm bounded with $\sqrt{\alpha}$. Then the expected Rademacher complexity of $\mathcal{M}_\alpha$ is bounded as follows:*

$$\mathfrak{R}_n(\mathcal{M}_\alpha) \leq O\left(\sqrt{\frac{\alpha d_2 \log(d_2)}{n}}\right)$$

Theorem 21 (Gaussian Concentration [Ver18]). *Consider a random vector $z \sim \mathcal{N}(0, I_d)$ and a ρ-Lipschitz function $f : \mathbb{R}^d \to \mathbb{R}$ (with respect to the Euclidean metric). Then $f(z)$ is ρ-sub-Gaussian and it holds for all $t \geq 0$:*

$$\mathbb{P}\{f(z) - \mathbb{E}[f(z)] \geq t\} \leq e^{\frac{-t^2}{2\rho^2}}$$

Theorem 22 (Theorem 1 of [BLL+11]). *Let $X_1, \ldots, X_T$ be a sequence of real-valued random variables. Let $\mathbb{E}_t[Y] := \mathbb{E}[Y|X_1, \ldots, X_{t-1}]$. Assume, for all t, that $X_t \leq R$ and that $\mathbb{E}_t[X_t] = 0$. Define the random variable $S_t := \sum_{k=1}^{t} X_t$, and $V_t := \sum_{k=1}^{t} \mathbb{E}_k[X_k^2]$. Then for any $\delta > 0$, with probability at least $1 - \delta$, we have the following guarantee:*

$$S_t \leq R \ln \frac{1}{\delta} + (e-2)\frac{V_t}{R}$$