Learning Deep Network

$$\mathbf{I} \underset{\mathbf{W}_1}{\rightarrow} \mathbf{H}_1 \underset{\mathbf{W}_2}{\rightarrow} \mathbf{H}_2 \underset{\mathbf{W}_3}{\rightarrow} \mathbf{O}$$

I: Input
O: Output

Example:

$$\mathbf{O} = \mathrm{ReLu}(\mathbf{W}_3\mathbf{H}_2)$$
$$\mathbf{H}_2 = \mathrm{ReLu}(\mathbf{W}_2\mathbf{H}_1)$$
$$\mathbf{H}_1 = \mathrm{ReLu}(\mathbf{W}_1\mathbf{I})$$

More generally,

$$\mathbf{O} = O(\mathbf{W}_3, \mathbf{H}_2)$$
$$\mathbf{H}_2 = H_2(\mathbf{W}_2, \mathbf{H}_1)$$
$$\mathbf{H}_1 = H_1(\mathbf{W}_1, \mathbf{I})$$

Only require that the functions $O(\circ, \circ)$, $H_2(\circ, \circ)$, $H_1(\circ, \circ)$ are differentiable

The functions can be composed together to give an output $\mathbf{O} = O(\mathbf{W}, \mathbf{I})$, $\mathbf{W} = (\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3)$

Loss Function $\mathcal{L}\left(\mathrm{O}(\mathbf{W},\mathbf{I}),\mathbf{T}\right)=\mathbf{L}$

$O(\mathbf{W}, \mathbf{I})$: the output of network

$\mathbf{T}$: the ground truth

Dataset: $\left\{\left(\mathbf{I}_n,\mathbf{T}_n\right): n=1,\ldots,N\right\}$

To train the deep network, we need to compute the derivatives $\frac{d}{d\mathbf{W}}\mathcal{L}\left(\mathrm{O}(\mathbf{W},\mathbf{I}),\mathbf{T}\right)$

Batch Mode → Steepest descent (SD)

$$\mathbf{W}^{t+1}=\mathbf{W}^{t}-\eta_t\frac{1}{N}\sum_{n=1}^{N}\frac{d}{d\mathbf{W}}\mathcal{L}\left(O(\mathbf{W},\mathbf{I}_n),\mathbf{T}_n\right)$$

Online Learning → Stochastic Gradient Descent (SGD)

At time step, select example $n(t)$ at random $\mathbf{W}^{t+1}=\mathbf{W}^{t}-\eta_t\frac{d}{d\mathbf{W}}\mathcal{L}\left(O(\mathbf{W},\mathbf{I}_{n(t)}),\mathbf{T}_{n(t)}\right)$

This is old fashioned SGD. In practice at time t, select a subset $N(t)$ of the data

$$\mathbf{W}^{t+1}=\mathbf{W}^{t}-\eta_t\frac{1}{|N(t)|}\sum_{n\in N(t)}\frac{d}{d\mathbf{W}}\mathcal{L}\left(O(\mathbf{W},\mathbf{I}_n),\mathbf{T}_n\right)$$

Loss Function

Note The loss function $\mathcal{L}(O(\mathbf{W},\mathbf{I}),\mathbf{T})$ is a non-convex function of $\mathbf{W}$

➔ Convergence cannot be guaranteed.
We will discuss later why it is non-convex

SGD has a stochastic property which enables it to avoid some local minima in the loss function.

There are theoretical results, informally called 'Robbins-Monro theory', which can even guarantee convergence to the global minimum of the loss function provided certain conditions apply.

But these results do not apply to deep networks

JOHNS HOPKINS UNIVERSITY

We need to compute $\frac{\partial \mathbf{L}}{\partial \mathbf{W}_1}, \frac{\partial \mathbf{L}}{\partial \mathbf{W}_2}, \frac{\partial \mathbf{L}}{\partial \mathbf{W}_3}$?

To do this, we use the chain rule of differentiation

→ This is called back propagation

To compute $\frac{\partial \mathbf{L}}{\partial \mathbf{W}_3}$ ➡ $\frac{\partial \mathbf{L}}{\partial \mathbf{W}_3} = \frac{\partial}{\partial \mathbf{W}_3} \mathcal{L}\left(O(\mathbf{H}_2, \mathbf{W}_3), \mathbf{T}\right) = \frac{\partial \mathbf{L}}{\partial \mathbf{O}} \cdot \frac{\partial \mathbf{O}}{\partial \mathbf{W}_3}$

To compute $\frac{\partial \mathbf{L}}{\partial \mathbf{W}_2}$, recall that $\mathbf{H}_2 = H_2(\mathbf{H}_1, \mathbf{W}_2)$

First compute $\frac{\partial \mathbf{L}}{\partial \mathbf{H}_2} = \frac{\partial \mathbf{L}}{\partial \mathbf{O}} \cdot \frac{\partial \mathbf{O}}{\partial \mathbf{H}_2}$ ➡ $\frac{\partial \mathbf{L}}{\partial \mathbf{W}_2} = \frac{\partial \mathbf{L}}{\partial \mathbf{H}_2} \cdot \frac{\partial \mathbf{H}_2}{\partial \mathbf{W}_2}$

To compute $\frac{\partial \mathbf{L}}{\partial \mathbf{W}_1}$, recall that $\mathbf{H}_1 = H_1(\mathbf{I}, \mathbf{W}_1)$

First compute $\frac{\partial \mathbf{L}}{\partial \mathbf{H}_1} = \frac{\partial \mathbf{L}}{\partial \mathbf{H}_2} \cdot \frac{\partial \mathbf{H}_2}{\partial \mathbf{H}_1}$ ➡ $\frac{\partial \mathbf{L}}{\partial \mathbf{W}_1} = \frac{\partial \mathbf{L}}{\partial \mathbf{H}_1} \cdot \frac{\partial \mathbf{H}_1}{\partial \mathbf{W}_1}$

Key Idea

Computing the derivatives exploits the compositionality of the function $\mathbf{O} = O(\mathbf{W}, \mathbf{I})$