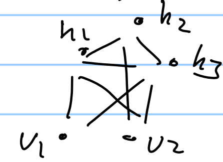
Lecture 14   **Deep Belief Networks & Active Appearance Models**   Note Title   2/28/2010

DBN's : Introduction

Boltzmann Machine:

$$P(\underline{v}, \underline{h}) = \frac{1}{Z} e^{-E(\underline{v}, \underline{h})}$$

observed nodes ($\underline{v}$)   hidden nodes ($\underline{h}$).



$$E(\underline{v}, \underline{h}) = -\sum_{i,j} v_i W_{ij} h_j - \sum_i c_i v_i - \sum_{ij} h_i W^h_{ij} h_j$$

motivated by modeling the brain, but just an MRF.

$v_i \in \{0,1\}$
$h_i \in \{0,1\}$

Observe example. $\{\underline{v}^\mu : \mu = 1 \text{ to } N\}$

$\underline{v} = (v_1 \ldots v_m)$ states of observed nodes.

Task: learn the weights of the model — $\{w_{ij}, c_i, w^h_{ij}\}$.

Can be learnt, in principle, by the EM algorithm or stochastic variant (e.g. data augmentation)

Difficult, because it is hard to do inference on ~~this~~ type of model in general.

Why learn this model? It learns a generative model for data. It learns "receptive" fields — e.g. the w_{ij}, how a hidden unit "h_i" is activated by the inputs.

To build a DBN, you first consider a simplified "Restricted Boltzmann Machine" RBM.

The connection terms w^h_{ij} between the hidden units are removed. ⇝ replace $-\sum_{i,j} h_i w^h_{ij} h_j$ by $-\sum_i b_i h_i$.

This has ~~the property that~~ $P(\underline{h}|\underline{v})$ and $P(\underline{v}|\underline{h})$ have simple forms:

$$P(\underline{h}|\underline{v}) = \prod_i P(h_i|\underline{v})$$

$$P(h_i|\underline{v}) = \frac{e^{-h_i(\sum_j w_{ij} v_j + b_i)}}{e^{\sum_j w_{ij} v_j + b_i} + e^{-(\sum_j w_{ij} v_j + b_i)}}$$

similarly $P(\underline{v}|\underline{h}) = \prod_i P(v_i|\underline{h})$

similar form for $P(v_i|\underline{h})$

(2) <u>Key point RBM's:</u>

<u>Easy</u> to sample from $P(\underline{v}|\underline{h})$ i.e. sample from $p(v_i|\underline{h})$ independ

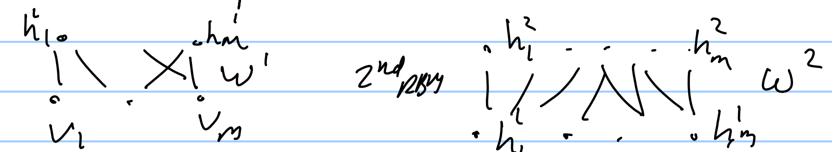
<u>Easy</u> to sample from $P(\underline{h}|\underline{v})$.

<u>This</u> enables us to do inference (stochastically) and makes it practice to learn the parameters $\{W\}$ of an RBM.

<u>But</u> RBM's are too simple → so add another RBM

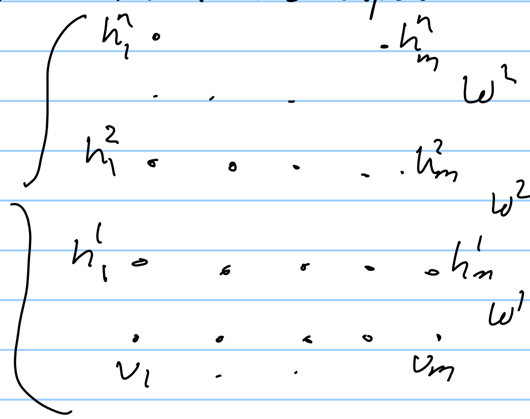
1st RBM

2nd RBM



<u>use</u> the hidden states $(h^1_1 \dots h^1_m)$ of the 1st RBM as input to the 2nd RBM.

then add another RBM and so on



n^{th} order RBM.

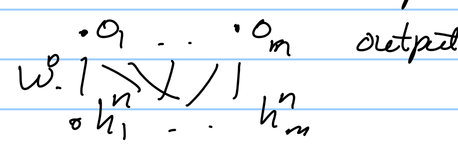
<u>Train</u> the model layer by layer.

<u>Intuition</u>: the "receptive fields"

– the W^1 – gives hidden units (h^1) which capture image features – like a dictionary in the previous two lectures.

→ then the receptive fields $W^2, W^3, \dots$ of the higher levels learn better dictionaries

The hidden states at the top levels can be used as inputs to a classifier level.



This can be trained in supervised mode – i.e. the $\{O_i\}$ are $\{v_i\}$ are provided by the training data.

Effective for learning how to recognize digits – Hinton et al.

For other variants see LeCun, Ng, Bengio.

Hinton's Talk: 4 March (Thurs) 4.15 p.m. CS.

(3)

Active Appearance Models AAM's

Object  Variations in geometry. variations in appearance.

First, suppose objects are aligned (i.e. remove spatial variation.)

Perform PCA principle component analysis.

$\{I^\mu(x) : \mu \in \Lambda\}$, $\bar{I} = \frac{1}{|\Lambda|} \sum_{\mu \in \Lambda} I^\mu(x)$

$$K(\underline{x}, \underline{y}) = \frac{1}{|\Lambda|} \sum_{\mu \in \Lambda} \{I^\mu(x) - \bar{I}(x)\}\{I^\mu(y) - \bar{I}(y)\}.$$

solve $\sum_{\underline{y}} K(\underline{x},\underline{y}) B^\mu(\underline{y}) = \lambda^\mu \underline{B}^\mu(x)$

solve for eigenvalues & eigenvectors.

represent object appearance.

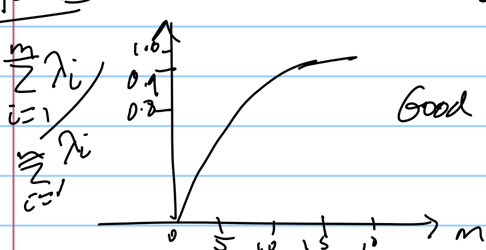
$$I(\underline{x}) = \bar{I}(\underline{x}) + \sum_{a=1}^{m} \underbrace{\alpha_a}_{\text{coefficient}} \underbrace{B_a(x)}_{\text{basis function}} + \epsilon(x)$$

Efficient representation if we can approximate $I(\underline{x})$ using a small no. basis functions i.e. m small

Square Error $\frac{1}{|\Lambda|} \sum_{\mu \in \Lambda} \left| I(x) - \bar{I}(x) - \sum_{a=1}^{m} \alpha_a B_a(x) \right|^2$

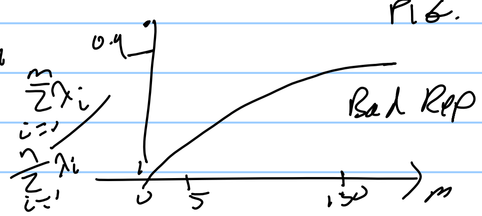
$$= \frac{\lambda^1 + \dots + \lambda^m}{\sum_{n=1}^{n} \lambda_n}$$ all the eigenvalues.

For faces



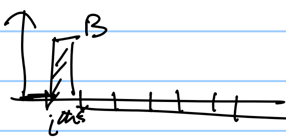
Good Representation

For text — e.g. ABC GOT PIG.

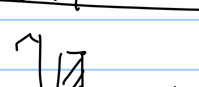
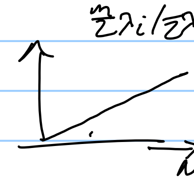


Bad Rep.

Intuition: PCA assumes that the data lies in a linear space. This may be true (approximately) for faces, but certainly is not true for text.

Example: data 

N-states $I(x) = B\delta(x, x_i)$ — ith bin

Suppose the data is  or  or 

i. Then PCA gives $(N-1)$ principal components, all with the same small eigenvalues.

(4)

Second: fix appearance, model geometry

$$I(x) \to I(S(x))$$

where $S(x)$ is a spatial warp

Also represent the spatial warp by PCA also.

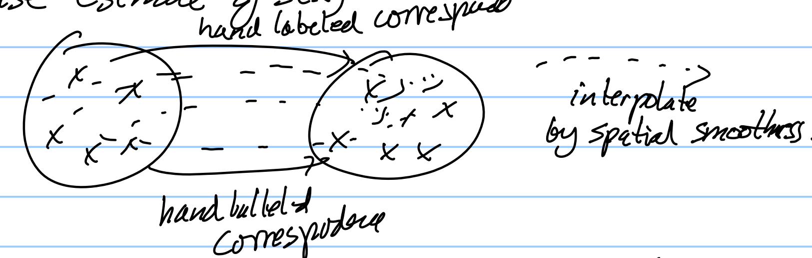
i.e. take training data $\{S^\mu(x) : \mu \in \Lambda\}$

perform PCA on $\{S^\mu\}$

express: $$S(x) = \sum_{b=1}^{q} \beta_b e_b(x) + \epsilon(x)$$

where q is no of coefficients.
$e_b(x)$ are the eigenvectors.

How to get the training data?
Typically hand labelled.
Can label a set of sparse points and interpolate to get dense estimate of $S(x)$



In principle, can learn the spatial and basis functions by EM.

Generative Model $$P(I \mid \{e, B\}, (\alpha, \beta)) = \frac{1}{Z} e^{-E[I : \{e, B\}, \{\alpha, \beta\}]}$$

$$E[I : \{e, B\}, \{\alpha, \beta\}] = \sum_x \left\{ I(x) - \sum \alpha_a B_a\left(\sum \beta_b e_b(x)\right) \right\}^2.$$

set of images $\{I^\mu : \mu \in \Lambda\}$
parameter (α^μ, β^μ) — different coefficients for each image.

$$P(\{I^\mu\} \mid \{e, B\}, \{\alpha^\mu, \beta^\mu\})$$

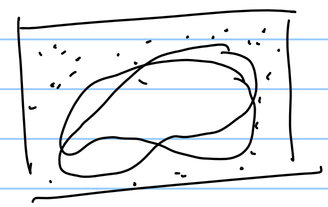
$$= \prod_\mu P(I^\mu \mid \{e, B\}, \alpha^\mu, \beta^\mu)$$

EM
M-step estimate $\{e, B\}$
E-step estimate probabilities $q(\alpha^\mu, \beta^\mu)$ for the coefficients

Problem $\Rightarrow$ many local minima $\to$ —

(5)

Recent work: e.g. I. Kokkinos.



Start with objects only approximately located.
→ e.g. box round object
Background clutter

To simplify the problem and make it tractable:
— filter the image to detect edges and other features like ridges — this removes the appearance variation so need to estimate the B's and α's

Simply need to estimate the spatial warps and their coefficients:

$$I_F^{\mu}(x) = T\left(\sum_{b=1}^{m} \beta_b^{\mu} S_b(x)\right) + \epsilon(x)$$

filtered image   Task: estimate $T(\cdot)$ the edge/ridge image of the object.

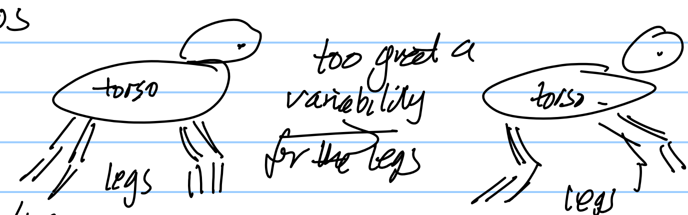
Still require EM,   E-step to estimate $T, \{\beta^{\mu}\}$
M-step to estimate $\{S_b\}$

EM does converge (Kokkinos & Yuille) with reasonable initial conditions.

Complication: what is m? how many basis functions?

Strategy (KEY) → greedy search.
Assume $m=1$, learn $S_1(x)$
then set $m=2$, learn $S_2(x)$ ...

Note: ~~there~~ were limitations using PCA to represent the appearance (i.e. approx. linear assumption).
What are the limitations of PCA to represent spatial warps?
Roughly true for faces, torso/body of a cow.
But bad for representing the spatial variability of the legs of a cows



Better to treat the legs as separate parts connected to the torso/body