Learning with Hidden Variables and RBMs

Ankur Gupta

University of British Columbia

August, 2015

Learning with Hidden Variables

On Monday we looked at learning parameters of a UGM,

- Vancouver rain example, $\mathbf{x}$: rain/no-rain for each month
- Modeled the energy using a log-linear model: $E(\mathbf{x}) = \mathbf{w}^T F(\mathbf{x})$
- NLL: $f(w) = -\frac{1}{N}\sum_t \log(p(\mathbf{x}^{(t)}|\mathbf{w})) = -\mathbf{w}^T F(D) + \log(Z(w))$
- The objective function is convex

Learning with Hidden Variables

On Monday we looked at learning parameters of a UGM,

- Vancouver rain example, $\mathbf{x}$: rain/no-rain for each month
- Modeled the energy using a log-linear model: $E(\mathbf{x}) = \mathbf{w}^T F(\mathbf{x})$
- NLL: $f(w) = -\frac{1}{N} \sum_t \log(p(\mathbf{x}^{(t)}|\mathbf{w})) = -\mathbf{w}^T F(D) + \log(Z(w))$
- The objective function is convex

Today we focus on learning parameters to model $p(\mathbf{x}, \mathbf{h})$

- where only $\mathbf{x}$ is observed and
- $\mathbf{h}$ is not observed (or hidden) in the training examples
- e.g., if the rain entry for a few days each month is missing, how to still use the data for learning

Learning with Hidden Variables

We can obtain $p(\mathbf{x})$ by summing over all values of $\mathbf{h}$

Learning with Hidden Variables

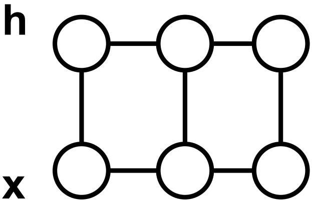
We can obtain $p(\mathbf{x})$ by summing over all values of $\mathbf{h}$

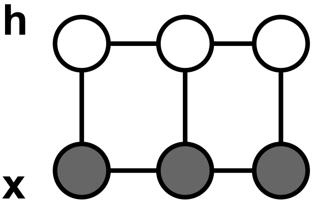
$$
\begin{aligned}
p(\mathbf{x}) &= \sum_{\mathbf{h}} p(\mathbf{x}, \mathbf{h}) \\
&= \sum_{\mathbf{h}} \frac{\exp(-E(\mathbf{x}, \mathbf{h}))}{Z} \\
&= \frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{x}, \mathbf{h}))}{\sum_{\mathbf{x},\mathbf{h}} \exp(-E(\mathbf{x}, \mathbf{h}))} \qquad = \frac{Z_h(\mathbf{x})}{Z}
\end{aligned}
$$

Learning with Hidden Variables

We can obtain $p(\mathbf{x})$ by summing over all values of $\mathbf{h}$

$$
\begin{aligned}
p(\mathbf{x}) &= \sum_{\mathbf{h}} p(\mathbf{x}, \mathbf{h}) \\
&= \sum_{\mathbf{h}} \frac{\exp(-E(\mathbf{x}, \mathbf{h}))}{Z} \\
&= \frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{x}, \mathbf{h}))}{\sum_{\mathbf{x},\mathbf{h}} \exp(-E(\mathbf{x}, \mathbf{h}))} \qquad = \frac{Z_h(\mathbf{x})}{Z}
\end{aligned}
$$





Learning with Hidden Variables

We can obtain $p(\mathbf{x})$ by summing over all values of $\mathbf{h}$

$$\begin{aligned} p(\mathbf{x}) &= \sum_{\mathbf{h}} p(\mathbf{x}, \mathbf{h}) \\ &= \sum_{\mathbf{h}} \frac{\exp(-E(\mathbf{x}, \mathbf{h}))}{Z} \\ &= \frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{x}, \mathbf{h}))}{\sum_{\mathbf{x},\mathbf{h}} \exp(-E(\mathbf{x}, \mathbf{h}))} \qquad = \frac{Z_h(\mathbf{x})}{Z} \end{aligned}$$

NLL:

$$f(w) = -\frac{1}{N} \sum_t \log(p(\mathbf{x}^{(t)} | \mathbf{w})) = \frac{1}{N} \sum_t (-\log(Z_h(\mathbf{x}^{(t)}))) + \log(Z)$$

Learning with Hidden Variables

We can obtain $p(\mathbf{x})$ by summing over all values of $\mathbf{h}$

$$
\begin{aligned}
p(\mathbf{x}) &= \sum_{\mathbf{h}} p(\mathbf{x}, \mathbf{h}) \\
&= \sum_{\mathbf{h}} \frac{\exp(-E(\mathbf{x}, \mathbf{h}))}{Z} \\
&= \frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{x}, \mathbf{h}))}{\sum_{\mathbf{x},\mathbf{h}} \exp(-E(\mathbf{x}, \mathbf{h}))} \qquad = \frac{Z_h(\mathbf{x})}{Z}
\end{aligned}
$$

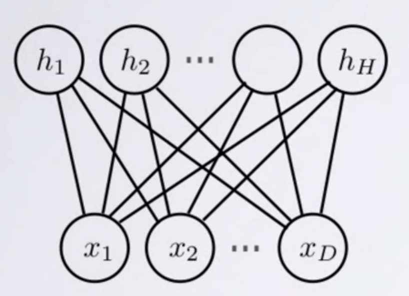
NLL:

$$
f(w) = -\frac{1}{N} \sum_t \log(p(\mathbf{x}^{(t)}|\mathbf{w})) = \frac{1}{N} \sum_t (-\log(Z_h(\mathbf{x}^{(t)}))) + \log(Z)
$$

- Note that the second term is the same as fully observed case
- Now, even for a log-linear model the NLL is no longer convex
- We can use exact or approximate inference (as applicable) to evaluate both the terms

Restricted Boltzmann Machines

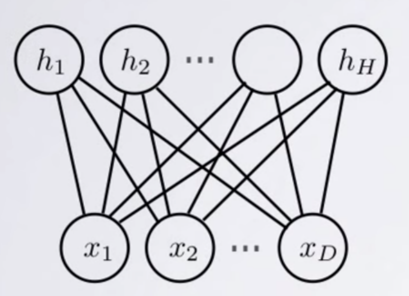
UGM with the following structure:



- No lateral connections
- $\mathbf{x}$ and $\mathbf{h}$ are both binary

Restricted Boltzmann Machines

UGM with the following structure:

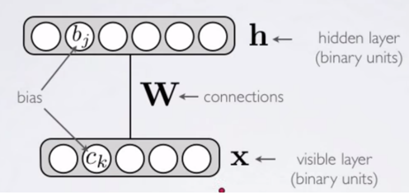


- No lateral connections
- $\mathbf{x}$ and $\mathbf{h}$ are both binary

The figures/slides are from videos by Hugo Larochelle, available at
`https://www.youtube.com/watch?v=p4Vh_zMw-HQ`

Restricted Boltzmann Machines

A compact description:



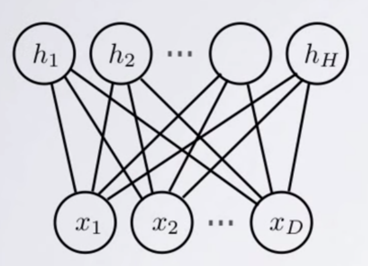
Energy:

$$
\begin{aligned}
E(\mathbf{x}, \mathbf{h}) &= -\mathbf{x}^T W \mathbf{h} - \mathbf{b}^T \mathbf{h} - \mathbf{c}^T \mathbf{x} \\
&= -\sum_{jk} w_{jk} h_j x_k - \sum_j b_j h_j - \sum_k c_k x_k
\end{aligned}
$$

Distribution:

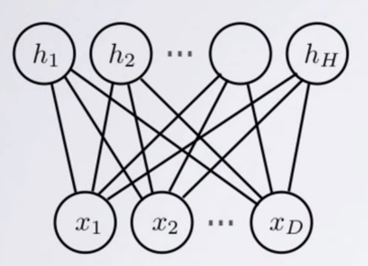
$$
p(\mathbf{x}, \mathbf{h}) = \frac{\exp(-E(\mathbf{x}, \mathbf{h}))}{Z}
$$

RBM: Inference



$$p(\mathbf{x}|\mathbf{h}) = \prod_k p(x_k|\mathbf{h})$$

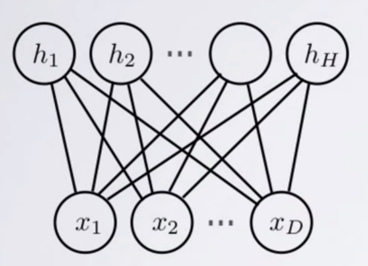
RBM: Inference



$$p(\mathbf{x}|\mathbf{h}) = \prod_k p(x_k|\mathbf{h})$$

$$\begin{aligned} p(x_k = 1|\mathbf{h}) &= \frac{1}{1 + \exp(-(c_k + \mathbf{h}^T \mathbf{W}_{.k}))} \\ &= \text{sigm}(c_k + \mathbf{h}^T \mathbf{W}_{.k}) \end{aligned}$$

RBM: Inference

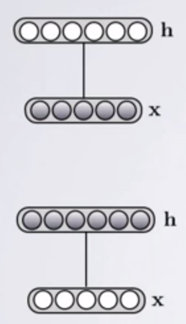


$$p(\mathbf{x}|\mathbf{h}) = \prod_k p(x_k|\mathbf{h})$$

$$p(x_k = 1|\mathbf{h}) = \frac{1}{1 + \exp(-(c_k + \mathbf{h}^T \mathbf{W}_{.k}))}$$
$$= \mathrm{sigm}(c_k + \mathbf{h}^T \mathbf{W}_{.k})$$

$$p(h_j = 1|\mathbf{x}) = \mathrm{sigm}(b_j + \mathbf{W}_{j.} \mathbf{x})$$

RBM: Inference

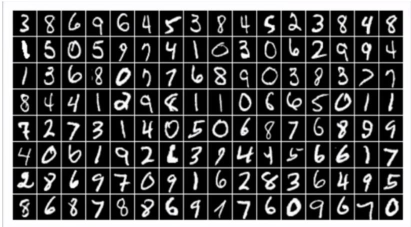


Due to conditional independence:

- conditional distribution $p(\mathbf{x}|\mathbf{h})$ factorizes
- we can calculate it in closed form
- decoding, inference and sampling is easy if $\mathbf{x}$ or $\mathbf{h}$ is given

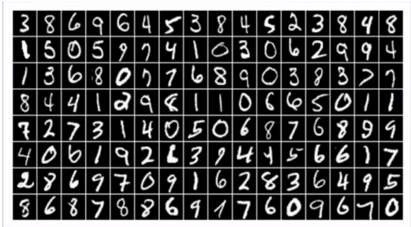
RBM: Learning

- Given a set of examples $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}\}$
- Learn the parameters W, $\mathbf{b}$, and $\mathbf{c}$
- Example: a set of binary images from MNIST dataset



RBM: Learning

- Given a set of examples $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}\}$
- Learn the parameters W, $\mathbf{b}$, and $\mathbf{c}$
- Example: a set of binary images from MNIST dataset



Motivation

- Unsupervised feature discovery
- Compression/non-linear dimensionality reduction
- A generative model of the image

RBM: Learning

To minimize the NLL:

$$\arg\min_{W,b,c} \frac{1}{N} \sum_{t} -\log(p(\mathbf{x}^{(t)}))$$

$$\begin{aligned} -\log(p(\mathbf{x}^{(t)})) &= -\log\left(\frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{x}^{(t)}, \mathbf{h}))}{\sum_{\mathbf{h},\mathbf{x}} \exp(-E(\mathbf{x}, \mathbf{h}))}\right) \\ &= -\log\big(\sum_{\mathbf{h}} \exp(-E(\mathbf{x}^{(t)}, \mathbf{h}))\big) + \log\big(\sum_{\mathbf{h},\mathbf{x}} \exp(-E(\mathbf{x}, \mathbf{h}))\big) \end{aligned}$$

RBM: Learning

To minimize the NLL:

$$\arg\min_{W,b,c} \frac{1}{N}\sum_t -\log(p(\mathbf{x}^{(t)}))$$

$$\begin{aligned}
-\log(p(\mathbf{x}^{(t)})) &= -\log\left(\frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{x}^{(t)},\mathbf{h}))}{\sum_{\mathbf{h},\mathbf{x}} \exp(-E(\mathbf{x},\mathbf{h}))}\right) \\
&= -\log\Big(\sum_{\mathbf{h}} \exp(-E(\mathbf{x}^{(t)},\mathbf{h}))\Big) + \log\Big(\sum_{\mathbf{h},\mathbf{x}} \exp(-E(\mathbf{x},\mathbf{h}))\Big)
\end{aligned}$$

Let's consider,

$$\begin{aligned}
\frac{\partial\big(-\log p(\mathbf{x}^{(t)})\big)}{\partial W_{jk}} &= \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}^{(t)}) \frac{\partial E(\mathbf{x}^{(t)},\mathbf{h})}{\partial W_{jk}} - \sum_{\mathbf{h},\mathbf{x}} p(\mathbf{x},\mathbf{h}) \frac{\partial E(\mathbf{x},\mathbf{h})}{\partial W_{jk}} \\
&= \mathbb{E}_{\mathbf{h}|\mathbf{x}}\left[\frac{\partial E(\mathbf{x}^{(t)},\mathbf{h})}{\partial W_{jk}}\right] - \mathbb{E}_{\mathbf{x},\mathbf{h}}\left[\frac{\partial E(\mathbf{x},\mathbf{h})}{\partial W_{jk}}\right]
\end{aligned}$$

RBM: Learning

To minimize the NLL:

$$\arg\min_{W,b,c} \frac{1}{N} \sum_t -\log(p(\mathbf{x}^{(t)}))$$

$$-\log(p(\mathbf{x}^{(t)})) = -\log\left(\frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{x}^{(t)}, \mathbf{h}))}{\sum_{\mathbf{h},\mathbf{x}} \exp(-E(\mathbf{x}, \mathbf{h}))}\right)$$

$$= -\log\Big(\sum_{\mathbf{h}} \exp(-E(\mathbf{x}^{(t)}, \mathbf{h}))\Big) + \log\Big(\sum_{\mathbf{h},\mathbf{x}} \exp(-E(\mathbf{x}, \mathbf{h}))\Big)$$

Let's consider,

$$\frac{\partial\big(-\log p(\mathbf{x}^{(t)})\big)}{\partial W_{jk}} = \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}^{(t)}) \frac{\partial E(\mathbf{x}^{(t)}, \mathbf{h})}{\partial W_{jk}} - \sum_{\mathbf{h},\mathbf{x}} p(\mathbf{x}, \mathbf{h}) \frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial W_{jk}}$$

$$= \mathbb{E}_{\mathbf{h}|\mathbf{x}}\left[\frac{\partial E(\mathbf{x}^{(t)}, \mathbf{h})}{\partial W_{jk}}\right] - \mathbb{E}_{\mathbf{x},\mathbf{h}}\left[\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial W_{jk}}\right]$$

So far we have not assumed an RBM. This can work for any hidden variable model.

Contrastive Divergence

Recall from previous slides:

$$E(\mathbf{x}, \mathbf{h}) = -\sum_{jk} W_{jk} h_j x_k - \sum_j b_j h_j - \sum_k c_k x_k$$

Contrastive Divergence

Recall from previous slides:

$$E(\mathbf{x}, \mathbf{h}) = -\sum_{jk} W_{jk} h_j x_k - \sum_j b_j h_j - \sum_k c_k x_k$$

Derivative:

$$\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial W_{jk}} = -h_j x_k$$

Contrastive Divergence

Recall from previous slides:

$$E(\mathbf{x}, \mathbf{h}) = -\sum_{jk} W_{jk} h_j x_k - \sum_j b_j h_j - \sum_k c_k x_k$$

Derivative:

$$\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial W_{jk}} = -h_j x_k$$

Plugging in values:

$$\begin{aligned}
\frac{\partial\big(-\log p(\mathbf{x}^{(t)})\big)}{\partial W_{jk}} &= \mathbb{E}_{\mathbf{h}|\mathbf{x}}\left[\frac{\partial E(\mathbf{x}^{(t)}, \mathbf{h})}{\partial W_{jk}}\right] - \mathbb{E}_{\mathbf{x},\mathbf{h}}\left[\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial W_{jk}}\right] \\
&= -\mathbb{E}_{\mathbf{h}|\mathbf{x}}\left[h_j x_k^{(t)}\right] + \mathbb{E}_{\mathbf{x},\mathbf{h}}\left[h_j x_k\right] \\
&\approx -\mathbb{E}_{\mathbf{h}|\mathbf{x}}\left[h_j x_k^{(t)}\right] + \mathbb{E}_{\mathbf{h}|\mathbf{x}}\left[h_j \tilde{x}_k\right] \\
&= -p(h_j = 1|\mathbf{x}^{(\mathbf{t})}) x_k^{(t)} + p(h_j = 1|\tilde{\mathbf{x}}) \tilde{x}_k
\end{aligned}$$

Contrastive Divergence

Update rule:

$$W_{jk} \leftarrow W_{jk} + \alpha\left(p(h_j = 1|\mathbf{x^{(t)}})x_k^{(t)} - p(h_j = 1|\tilde{\mathbf{x}})\tilde{x}_k\right)$$

Contrastive Divergence

Update rule:

$$W_{jk} \leftarrow W_{jk} + \alpha\left(p(h_j = 1|\mathbf{x}^{(\mathbf{t})})x_k^{(t)} - p(h_j = 1|\tilde{\mathbf{x}})\tilde{x}_k\right)$$

- We can obtain similar expressions for b_j and c_k

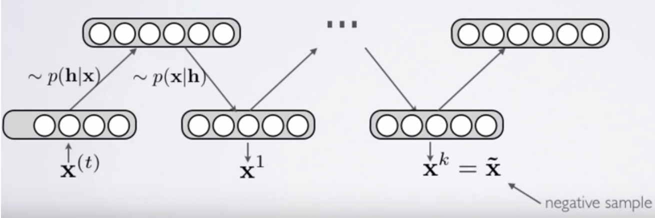
Contrastive Divergence

Update rule:

$$W_{jk} \leftarrow W_{jk} + \alpha\left(p(h_j = 1|\mathbf{x}^{(\mathbf{t})})x_k^{(t)} - p(h_j = 1|\tilde{\mathbf{x}})\tilde{x}_k\right)$$

- We can obtain similar expressions for b_j and c_k

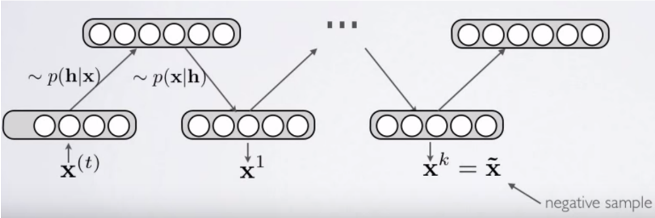
Sampling $\tilde{\mathbf{x}}$: use block Gibb's sampling



Contrastive Divergence

Putting everything together: CD-k algorithm

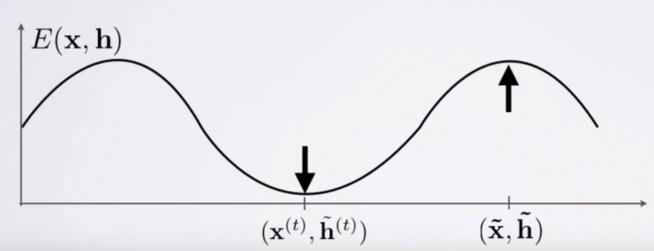
- For each training example $\mathbf{x}^{(t)}$
  - Initialize a Gibb's chain with $\mathbf{x}^{(t)}$
  - Run k rounds to obtain $\tilde{\mathbf{x}}$
  - Update W, b, and c
- Go back to the first step until a stopping criteria



Contrastive Divergence

$$W_{jk} \leftarrow W_{jk} + \alpha\left(p(h_j = 1|\mathbf{x^{(t)}})x_k^{(t)} - p(h_j = 1|\tilde{\mathbf{x}})\tilde{x}_k\right)$$

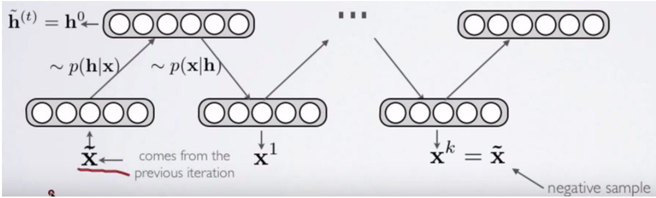
CD intuition:



Persistent CD or Younes' Algorithm

Pseudo code

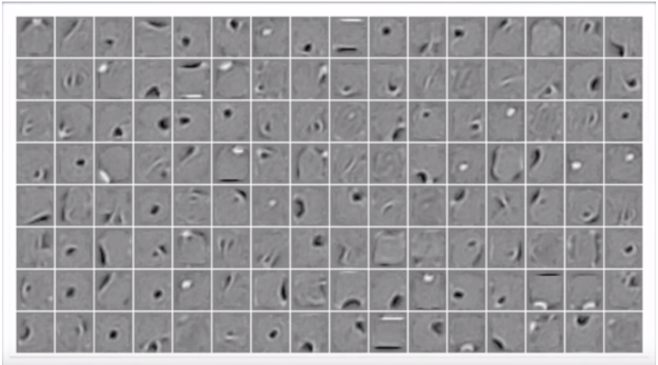
- For each training example $\mathbf{x}^{(t)}$
  - Initialize a Gibb's chain with $\tilde{\mathbf{x}}^{(t-1)}$
  - Run k rounds to obtain $\tilde{\mathbf{x}}$
  - Update W, b, and c
- Go back to the first step until a stopping criteria



Works better in theory as well as in practice.

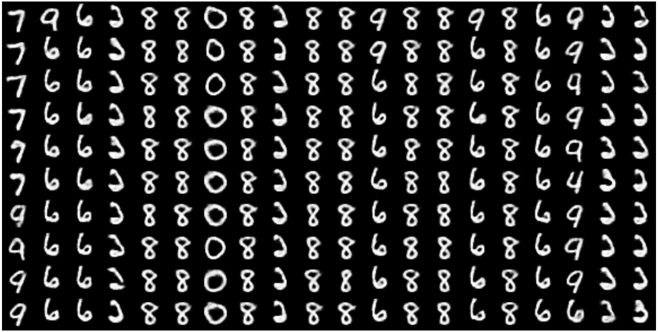
Learned Features

Weights W in an image form:



Sample the generative model

Samples obtained from an RBM trained on MNIST data



Extensions of RBM

Gaussian-Bernoulli RBM

- Input x can be real-valued
- Modified energy function

$$E(\mathbf{x},\mathbf{h}) = -\mathbf{x}^T W\mathbf{h} - \mathbf{b}^T\mathbf{h} - \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{x}$$

- $p(\mathbf{x}|\mathbf{h})$ turns out to be a Gaussian distribution

Extensions of RBM

Gaussian-Bernoulli RBM

- Input x can be real-valued
- Modified energy function

$$E(\mathbf{x},\mathbf{h}) = -\mathbf{x}^T W\mathbf{h} - \mathbf{b}^T\mathbf{h} - \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{x}$$

- $p(\mathbf{x}|\mathbf{h})$ turns out to be a Gaussian distribution

Deep Belief Networks

- Can be trained greedily one layer at a time (same as RBM training)

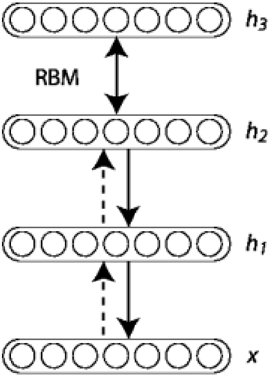


Image: deeplearning.net

Summary

- All the variables may not be observed in the training data. We can still learn the parameters of a UGM.
- Restricted Boltzmann Machines (RBM) are binary UGMs with hidden variables (no lateral connection)
- RBMs are useful for unsupervised feature discovery, non-linear dimensionality reduction etc.
- RBMs can be trained efficiently using Persistent-CD