Deep Belief Net

Xiaogang Wang

xgwang@ee.cuhk.edu.hk

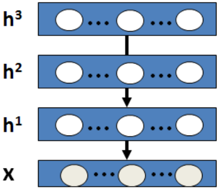
February 9, 2015

Outline

1. Restricted Boltzmann Machine
2. Deep Belief Net
3. Deep Boltzmann Machine

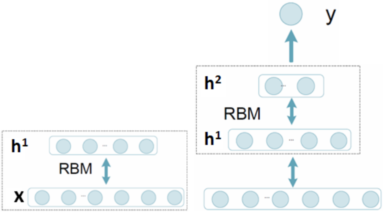
Deep belief net

- Hinton et al. 2006
- DBN is a generative model, modeling the joint distribution of observed data $\mathbf{x}$ and hidden variables $(\{\mathbf{h}^1, \ldots, \mathbf{h}^l\})$, $P(\mathbf{x}, \mathbf{h}^1, \ldots, \mathbf{h}^l; \theta)$
- Hidden variables are structured into l layers and are treated as hierarchical feature representations
  $P(\mathbf{x}, \mathbf{h}^1, \ldots, \mathbf{h}^l) = P(\mathbf{x}|\mathbf{h}^1) P(\mathbf{h}^1|\mathbf{h}^2) \ldots P(\mathbf{h}^{l-1}, \mathbf{h}^l)$
- Learn the network parameters θ by maximizing the data likelihood
  $P(\mathbf{x}; \theta) = \sum_{\mathbf{h}^1, \ldots, \mathbf{h}^l} P(\mathbf{x}, \mathbf{h}^1, \ldots, \mathbf{h}^l; \theta)$
- $x_i, h_i^k \in \{0, 1\}$



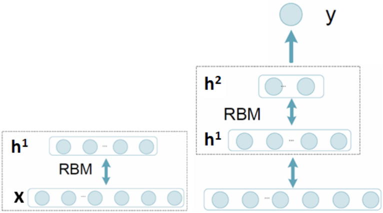
Unsupervised layerwise pre-training

- Each time only consider two layers $\mathbf{h}^{k-1}$ and $\mathbf{h}^k$ ($\mathbf{h}^0 = \mathbf{x}$), assuming $\mathbf{h}^{k-1}$ is known and fixed
- The parameters between the two layers are learned by maximizing the likelihood $P(\mathbf{h}_{k-1})$
- The joint distribution $P(\mathbf{h}^{k-1}, \mathbf{h}^k)$ of the two layers is approximated as Restricted Boltzmann Machine (RBM)
- Parameters of $P(\mathbf{h}^{k-1}, \mathbf{h}^k)$ are learned with Contrastive Divergence (CD) and fixed
- $\mathbf{h}^k$ is sampled from $P(\mathbf{h}^k|\mathbf{h}^{k-1})$ for the training of next layer, or estimated as $\hat{\mathbf{h}}^k = \int_{\mathbf{h}^k} P(\mathbf{h}^k|\mathbf{h}^{k-1})$



Fine tuning

- The top hidden layer is used as feature to predict class label
- After pre-training, the whole network is fine-tuned with backpropagation given supervised information (e.g. class label **y**) provided



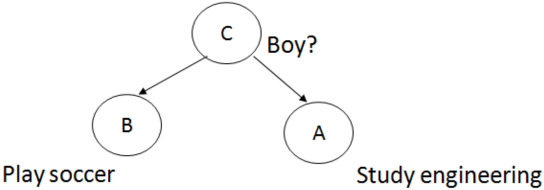
Graphical model

- Graphical models represent conditional independence between random variables
- Given C, A and B are independent:

$$P(A, B|C) = P(A|C)P(B|C)$$

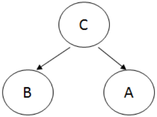
$$P(A, B, C) = P(A, B|C)P(C) = P(A|C)P(B|C)P(C)$$

- Any two nodes are conditionally independent given the values of their parents.

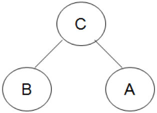


Directed and undirected graphical model

- Directed graphical model
  - $P(A, B, C) = P(A|C)P(B|C)P(C)$
  - Any two nodes are conditionally independent give the values of their parents



- Undirected graphical model
  - $P(A, B, C) = \Phi_1(B, C)\Phi_2(A, C)$
  - A and B are conditionally independent given C, if all the paths connecting A and B are blocked by C



Energy-Based Models (EBM)

- Define a probability distribution through an energy function

$$p(\mathbf{x}) = \frac{e^{-E(\mathbf{x})}}{Z}$$

The normalization factor Z is called the partition function

$$Z = \sum_{\mathbf{x}} e^{-E(\mathbf{x})}$$

- An energy-based model can be learnt by performing (stochastic) gradient descent on the empirical negative log-likelihood of the training data

$$\ell(\theta, \mathcal{D}) = -\frac{1}{N} \sum_{\mathbf{x}^{(i)} \in \mathcal{D}} \log p(\mathbf{x}^{(i)})$$

- Use the stochastic gradient $-\frac{\partial \log p(\mathbf{x}^{(i)})}{\partial \theta}$ to update the parameters θ of the model

EBMs with hidden units

- In some cases, we do not observe the example **x** fully, or we want to introduce some non-observed variables to increase the expressive power of the model. So we consider an observed part (**x**) and a hidden part **h**:

$$P(\mathbf{x}) = \sum_{\mathbf{h}} P(\mathbf{x}, \mathbf{h}) = \sum_{\mathbf{h}} \frac{e^{-E(\mathbf{x},\mathbf{h})}}{Z}$$

- Define the free energy as

$$\mathcal{F}(\mathbf{x}) = -\log \sum_{\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}$$

- $P(\mathbf{x})$ can be written as $P(\mathbf{x}) = \frac{e^{-\mathcal{F}(\mathbf{x})}}{Z}$ with $Z = \sum_{\mathbf{x}} e^{-\mathcal{F}(\mathbf{x})}$
- The data negative log-likelihood gradient has the form

$$-\frac{\partial \log p(\mathbf{x})}{\partial \theta} = \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \theta} - \sum_{\tilde{\mathbf{x}}} p(\tilde{\mathbf{x}}) \frac{\partial \mathcal{F}(\tilde{\mathbf{x}})}{\partial \theta}$$

The first term increases the probability of training data (by reducing the corresponding free energy), while the second term decreases the probability of samples generated by the model.

EBMs with hidden units

- It is usually difficult to determine this gradient analytically, as it involves the computation of $E_P[\frac{\partial \mathcal{F}(\mathbf{x})}{\partial \theta}]$, which is an expectation over all possible configurations of the input $\mathbf{x}$ under the distribution.
- To make the computation tractable, estimate the expectation using a fixed number of samples $\mathcal{N}$, which are generated according to P.

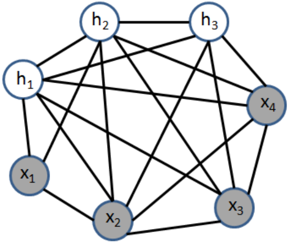
$$-\frac{\partial \log p(\mathbf{x})}{\partial \theta} = \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \theta} - \frac{1}{|\mathcal{N}|} \sum_{\tilde{\mathbf{x}} \in \mathcal{N}} \frac{\partial \mathcal{F}(\tilde{\mathbf{x}})}{\partial \theta}$$

- **The key question is how to generate $\mathcal{N}$ from P.**

Boltzmann Machine (BM)

$$E(\mathbf{x}, \mathbf{h}; \theta) = -(\mathbf{b}'\mathbf{x} + \mathbf{c}'\mathbf{h} + \mathbf{h}'\mathbf{W}\mathbf{x} + \mathbf{x}'\mathbf{U}\mathbf{x} + \mathbf{h}'\mathbf{V}\mathbf{h})$$

$$\theta = \{\mathbf{b}', \mathbf{c}', \mathbf{W}, \mathbf{U}, \mathbf{V}\}$$



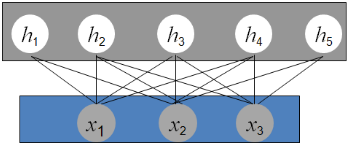
Restricted Boltzmann Machine (RBM)

- RBM does not model the interactions of variables of the same layer
- $E(\mathbf{x}, \mathbf{h})$ is the energy function. $Z = \sum_{\mathbf{x},\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}$ is call partition function and serves as a normalizing factor
- **b**, **c**, and **W** are the parameters to be learned

$$P(\mathbf{x}, \mathbf{h}) = \frac{e^{-E(\mathbf{x},\mathbf{h})}}{\sum_{\mathbf{x},\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}}$$

$$P(\mathbf{x}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}}{\sum_{\mathbf{x},\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}}$$

$$E(\mathbf{x}, \mathbf{h}) = -(\mathbf{b}'\mathbf{x} + \mathbf{c}'\mathbf{h} + \mathbf{h}'\mathbf{W}\mathbf{x})$$



Restricted Boltzmann Machine (RBM)

Let $\mathbf{a} = \mathbf{W}\mathbf{x}$

$$\begin{aligned} P(\mathbf{h}|\mathbf{x}) &\propto e^{(\mathbf{b}'\mathbf{x}+\mathbf{c}'\mathbf{h}+\mathbf{h}'\mathbf{a})} \\ &\propto e^{(\mathbf{c}'\mathbf{h}+\mathbf{h}'\mathbf{a})} \\ &= e^{\sum_i (c_i+a_i)h_i} \\ &= \prod_i e^{(c_i+a_i)h_i} \\ &= \prod_i f(h_i) \end{aligned}$$

Since $\mathbf{x}$ is fixed,

$$P(\mathbf{h}|\mathbf{x}) \propto \prod_i f(h_i)$$

Therefore, $\{h_i\}$ are conditional independent given $\mathbf{x}$.

Restricted Boltzmann Machine (RBM)

- $\{h_i\}$ are conditionally independent given **x**. $\{x_j\}$ are conditionally independent given **h**.

$$P(\mathbf{h}|\mathbf{x}) = \prod_i P(h_i|\mathbf{x})$$

$$P(\mathbf{x}|\mathbf{h}) = \prod_i P(x_i|\mathbf{h})$$

- The conditional distributions have analytical solutions

$$P(x_j = 1|\mathbf{h}) = \sigma(b_j + \mathbf{W}'_{\cdot j} \cdot \mathbf{h})$$

$$P(h_i = 1|\mathbf{x}) = \sigma(c_i + \mathbf{W}_{i\cdot} \cdot \mathbf{x})$$

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

Restricted Boltzmann Machine (RBM)

- Assuming there are N training samples $\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}]$, the parameters θ are estimated by maximizing the log-likelihood

$$L(\mathbf{X}; \theta) = \frac{1}{N} \sum_{n=1}^{N} \log P(\mathbf{x}^{(n)}; \theta)$$

$$\theta_{t+1} = \theta_t + \eta \frac{\partial L(\mathbf{X}; \theta)}{\partial \theta}$$

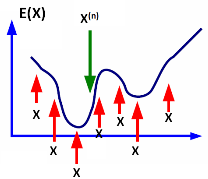
Contrastive Divergence (CD)

$$P(\mathbf{x};\theta) = \frac{\sum_{\mathbf{h}} e^{(\mathbf{b}'\mathbf{x}+\mathbf{c}'\mathbf{h}+\mathbf{h}'\mathbf{W}\mathbf{x})}}{\sum_{\mathbf{x},\mathbf{h}} e^{(\mathbf{b}'\mathbf{x}+\mathbf{c}'\mathbf{h}+\mathbf{h}'\mathbf{W}\mathbf{x})}} = \frac{f(\mathbf{x};\theta)}{Z(\theta)}$$

$$\begin{aligned}
\frac{\partial L(\mathbf{X};\theta)}{\partial w_{ji}} &= \frac{1}{N}\sum_{n=1}^{N}\frac{\partial \log f(\mathbf{x}^{(n)};\theta)}{\partial w_{ij}} - \frac{\partial \log Z(\theta)}{\partial w_{ij}} \\
&= \frac{1}{N}\sum_{n=1}^{N}\frac{\partial \log f(\mathbf{x}^{(n)};\theta)}{\partial w_{ij}} - \frac{1}{Z(\theta)}\frac{\partial Z(\theta)}{\partial \theta} \\
&= \frac{1}{N}\sum_{n=1}^{N}\frac{\partial \log f(\mathbf{x}^{(n)};\theta)}{\partial w_{ij}} - \sum_{\mathbf{x}}\frac{1}{Z(\theta)}\frac{\partial f(\mathbf{x};\theta)}{\partial w_{ij}} \\
&= \frac{1}{N}\sum_{n=1}^{N}\frac{\partial \log f(\mathbf{x}^{(n)};\theta)}{\partial w_{ij}} - \sum_{\mathbf{x}}\frac{f(\mathbf{x};\theta)}{Z(\theta)}\frac{1}{f(\mathbf{x};\theta)}\frac{\partial f(\mathbf{x};\theta)}{\partial w_{ij}} \\
&= \frac{1}{N}\sum_{n=1}^{N}\frac{\partial \log f(\mathbf{x}^{(n)};\theta)}{\partial w_{ij}} - \sum_{\mathbf{x}} P(\mathbf{x};\theta)\frac{\partial \log f(\mathbf{x};\theta)}{\partial w_{ij}}
\end{aligned}$$

Contrastive Divergence (CD)

$$
\begin{aligned}
\frac{\partial L(\mathbf{X};\theta)}{\partial w_{ij}} &= -\sum_{\mathbf{x}} P(\mathbf{x};\theta)\frac{\partial \log f(\mathbf{x};\theta)}{\partial w_{ij}} + \frac{1}{N}\sum_{n=1}^{N}\frac{\partial \log f(\mathbf{x}^{(n)};\theta)}{\partial w_{ij}} \\
&= - < \frac{\partial \log f(\mathbf{x};\theta)}{\partial \theta} >_{model} + < \frac{\partial f(\mathbf{x};\theta)}{\partial \theta} >_{data} \\
&= < \frac{\partial E(\mathbf{x})}{\partial \theta} >_{model} - < \frac{\partial E(\mathbf{x})}{\partial \theta} >_{data} \\
&= - < x_i h_j >_{model} + < x_i^{(n)} h_j >_{data} \\
&= -\sum_{\mathbf{x},\mathbf{h}} P(\mathbf{x},\mathbf{h};\theta) x_i h_j + \frac{1}{N}\sum_{n=1}^{N}\sum_{h_j} P(h_j|\mathbf{x}^{(n)};\theta) x_i^{(n)} h_j
\end{aligned}
$$



E(x) = -log f(x; θ)

Contrastive Divergence

- $< x_i^{(n)} h_j >_{data}$ is the estimation of the average of $x_i h_j$ from the empirical distribution on the training set
- $< x_i h_j >_{model}$ can be esimated from infinite number of samples $\{\tilde{\mathbf{x}}^{(m)}\}_{m=1}^{\infty}$ randomly drawn from $P(\mathbf{x}; \theta)$

$$\frac{\partial L(\mathbf{X}; \theta)}{\partial w_{ji}} \approx -\frac{1}{M} \sum_{\tilde{\mathbf{x}}^{(m)} \sim P(\mathbf{x};\theta)} \sum_{\tilde{h}_j^{(m)}} P(\tilde{h}_j^{(m)} | \tilde{\mathbf{x}}^{(m)}; \theta) \tilde{x}_i^{(m)} \tilde{h}_j^{(m)}$$

$$+ \frac{1}{N} \sum_{n=1}^{N} \sum_{h_j} P(h_j | \mathbf{x}^{(n)}; \theta) x_i^{(n)} h_j$$

Contrastive Divergence

- $\{\tilde{\mathbf{x}}^{(m)}\}$ can be generated from Markov Chain Monte Carlo (MCMC). Gibbs sampling is a commonly used MCMC approach and was used by Hinton et al. (1986).
- Starting from any training sample $\tilde{\mathbf{x}}^{(0)}$, a sequence of samples $\tilde{\mathbf{x}}^{(1)}, \ldots, \tilde{\mathbf{x}}^{(M)}$ are generated in the following way

$$\begin{aligned}
\tilde{\mathbf{h}}^{(0)} &\sim P(\mathbf{h}|\mathbf{x}^{(0)}) \\
\tilde{\mathbf{x}}^{(1)} &\sim P(\mathbf{x}|\tilde{\mathbf{h}}^{(0)}) \\
\tilde{\mathbf{h}}^{(0)} &\sim P(\mathbf{h}|\tilde{\mathbf{x}}^{(1)}) \\
&\ldots \\
\tilde{\mathbf{x}}^{(M)} &\sim P(\mathbf{x}|\tilde{\mathbf{h}}^{(M-1)})
\end{aligned}$$

- $\tilde{\mathbf{x}}^{(1)}, \ldots, \tilde{\mathbf{x}}^{(M)}$ follow the distribution of $P(\mathbf{x})$
- MCMC is slow

Contrastive Divergence

- Stochastic gradient descent: replacing the average over all the training samples with a single training sample
- Starting from the chosen training sample $\mathbf{x}^{(n)}$, only do one step MCMC sampling and use the generated sample $\tilde{x}_i^{(1)}\tilde{h}_j^{(1)}$ to approximate $< x_i h_j >_{model}$ which is supposed to estimated from infinite number of samples generated from MCMC

$$\frac{\partial L(\mathbf{X};\theta)}{\partial w_{ji}} \approx -\tilde{x}_i^{(1)} P(\tilde{h}_j^{(1)} = 1|\tilde{x}_i^{(1)}) + x_i^{(n)} P(h_j^{(n)} = 1|\mathbf{x}^{(n)})$$

$$\frac{\partial L(\mathbf{X};\theta)}{\partial b_i} \approx -\tilde{x}_i^{(1)} + x_i^{(n)}$$

$$\frac{\partial L(\mathbf{X};\theta)}{\partial c_j} \approx -P(\tilde{h}_j^{(1)} = 1|\tilde{x}_i^{(1)}) + P(h_j^{(n)} = 1|\mathbf{x}^{(n)})$$

Contrastive Divergence

$$\frac{\partial L(X;\theta)}{\partial \theta} = -\int p(x,\theta)\frac{\partial \log f(x;\theta)}{\partial \theta}dx + \frac{1}{K}\sum_{k=1}^{K}\frac{\partial \log f(x^{(k)};\theta)}{\partial \theta}$$

Intractable

Easy to compute

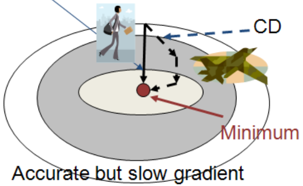
Tractable Gibbs Sampling

Sample $p(z_1, z_2, \ldots, z_M)$

```
1. Initialize $\{z_i : i = 1, \ldots, M\}$
2. For $\tau = 1, \ldots, T$:
    – Sample $z_1^{(\tau+1)} \sim p(z_1 | z_2^{(\tau)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
    – Sample $z_2^{(\tau+1)} \sim p(z_2 | z_1^{(\tau+1)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
      ⋮
    – Sample $z_j^{(\tau+1)} \sim p(z_j | z_1^{(\tau+1)}, \ldots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \ldots, z_M^{(\tau)})$.
      ⋮
    – Sample $z_M^{(\tau+1)} \sim p(z_M | z_1^{(\tau+1)}, z_2^{(\tau+1)}, \ldots, z_{M-1}^{(\tau+1)})$.
```

Fast contrastive divergence
T=1

$$\theta_{t+1} = \theta_t + \lambda \frac{\partial L(X;\theta)}{\partial \theta}$$



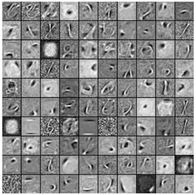
Accurate but slow gradient

Approximate but fast gradient

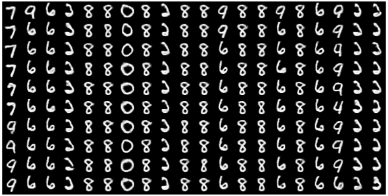
Convergence of Contrastive Divergence

- The stationary points of maximum likelihood (ML) estimation is not the stationary points of CD
- CD is biased, but the bias is typically small
- CD can be used for getting close to ML solution and then ML learning can used for fine-tuning
- It is not clear whether CD learning converges

Filters learned by RBM



Filters learned by RBM



Samples generated by RBM from Gibbs sampling

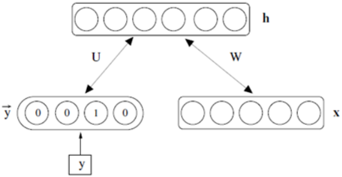
RBM for classification

- Larochelle and Bengio 2008
- **y**: vector of class label

$$p(y, \mathbf{x}, \mathbf{h}) \propto e^{-E(y,\mathbf{x},\mathbf{h})}$$

$$E(y, \mathbf{x}, \mathbf{h}) = -\mathbf{h}'\mathbf{W}\mathbf{x} - \mathbf{b}'\mathbf{x} - \mathbf{c}'\mathbf{h} - \mathbf{d}'\mathbf{y} - \mathbf{h}'\mathbf{U}\mathbf{y}$$

$$p(y|\mathbf{x}) = \frac{e^{d_y} \prod_j (1 + e^{c_j + U_{jy} + \Sigma_i W_{ji} x_i})}{\sum_{y^*} e^{d_{y^*}} \prod_j (1 + e^{c_j + U_{jy^*} + \Sigma_i W_{ji} x_i})}$$



RBM for classification

- Update

$$\frac{\partial \log p(y^{(n)}|\mathbf{x}^{(n)})}{\partial\theta} = \sum_j \sigma(o_{y^{(n)}j}(\mathbf{x}^{(n)}))\frac{\partial o_{y^{(n)}j}(\mathbf{x}^{(n)})}{\partial\theta}$$

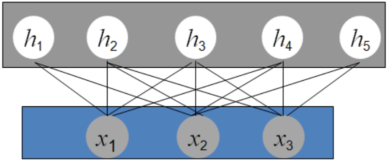
$$- \sum_{j,y^*} \sigma(o_{y^*j}(\mathbf{x}^{(n)}))p(y^*|\mathbf{x}^{(n)})\frac{\partial o_{y^*j}(\mathbf{x}^{(n)})}{\partial\theta}$$

$$o_{yj}(\mathbf{x}) = c_j + \sum_i w_{ji}x_i + U_{jy}$$

RBM leads to distributed representation

- Each hidden unit can be treated as an attribute and creates a 2-region partition of the input space. The binary setting of the N hidden units identifies one region in input space among all the 2^N regions associated with configurations of the hidden units
- If the distribution $P(x)$ represented by RMB is transformed to a mixture model, it is a sum over an exponential number of configurations

$$P(\mathbf{x}) = \sum_{\mathbf{h}} P(\mathbf{x}|\mathbf{h})P(\mathbf{h})$$



RBM leads to distributed representation

- RBM can be represented as a product of experts

$$P(\mathbf{x}) \propto e^{\mathbf{b}'\mathbf{x}+\sum_j \log \sum_{h_j} e^{h_j \mathbf{W}_{j\cdot}\mathbf{x}}} \propto \prod_j e^{f_j(x)}$$

  where $f_j = \log \sum_{h_j} e^{h_j \mathbf{W}_{j\cdot}\mathbf{x}}$

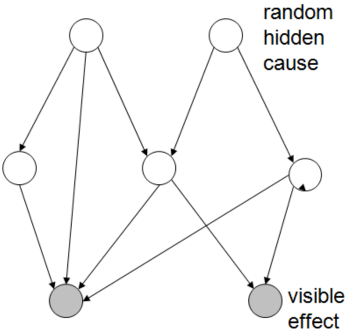
- $f_j(\mathbf{x})$ is an attribute indicator
- Hinton (1999) explains the advantages of a product of experts by opposition to a mixture of experts
  - In a mixture model, the constraint associated with an expert is an indication of belonging to a region which excludes the other regions
  - In a product of experts, the set of $f_j(\mathbf{x})$ form a distributed representation: partition the space according to all the possible configurations (where each expert can have its constraint violated or not)

Product of expert models vs mixture of expert models

- A mixture distribution can have high probability for event **x** when only a **single** expert assigns high probability to that event; while a product can only have high probability for an event **x** when **no expert** assigns an especially low probability to that event
- A single expert in a mixture has the power to pass a bill while a single expert in a product has the power to veto it
- Each component in a product prepresents a soft **constraint**. For an event to be likely under a product model, **all constraints** must be (approximately) satisfied
- Each expert in a mixture represents a soft **template or prototype**. An event is likely under a mixture model if it (approximately) matches with **any single template**

Belief nets

- A belief net is a directed acyclic graph composed of random variables
- Also called Bayesian network



Deep belief nets

- Belief net that is deep
- A generative model

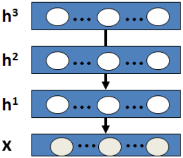
$$P(\mathbf{x},\mathbf{h}_1,\ldots,{}_l) = P(\mathbf{x}|\mathbf{h}_1)P(\mathbf{h}_1|\mathbf{h}_2)\ldots P(\mathbf{h}_{l-2}|\mathbf{h}_{l-1})P(\mathbf{h}_{l-1},\mathbf{h}_l)$$

- $P(\mathbf{h}^{k-1}|\mathbf{h}^k)$ $(\mathbf{x} = \mathbf{h}^0)$ is conditional distribution for the visible units conditional on the hidden units of the RBM at level k

$$P(\mathbf{h}^{k-1}|\mathbf{h}^k) = \prod_i P(h_i^{k-1}|\mathbf{h}^k)$$

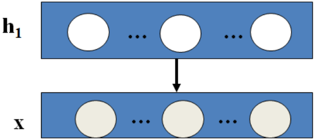
$$P(h_j^{k-1} = 1|\mathbf{h}^k) = \sigma(b_j^k + \mathbf{W}_{\cdot j}^{k\prime}\mathbf{h})$$

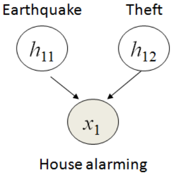
- $P(\mathbf{h}^{l-1},\mathbf{h}^l)$ is model as RBM



The inference of DBN is problematic

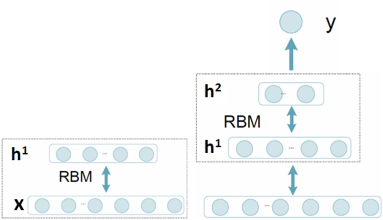
- Explaining away: $P(h_{11}, h_{12}|x_1) \neq P(h_{11}|x_1)P(h_{12}|x_1)$
- Different from RBM, DNB is a directed graphical model. Given $\mathbf{x}$, $h_1, \ldots, h_n$ are not independent any more. Only when $P(\mathbf{h})$ adopts certain prior (called complementary prior), DBN is equivalent to RBM, $P(\mathbf{x}, \mathbf{h}) = P(\mathbf{x}|\mathbf{h})P(\mathbf{h})$. The details of the complementary prior can be found in Hinton et al. 2006.
- Therefore, only feedforward approximate inference is used for DBN: no bottom-up and top-down





Pre-training

- Each time only consider two layers $\mathbf{h}^{k-1}$ and $\mathbf{h}^k$ ($\mathbf{h}^0 = \mathbf{x}$), assuming $\mathbf{h}^{k-1}$ is known and fixed
- The parameters between the two layers are learned by maximizing the likelihood $P(\mathbf{h}_{k-1})$
- The joint distribution $P(\mathbf{h}^{k-1}, \mathbf{h}^k)$ of the two layers is approximated as Restricted Boltzmann Machine (RBM)
- Parameters of $P(\mathbf{h}^{k-1}, \mathbf{h}^k)$ are learned with Contrastive Divergence (CD) and fixed
- $\mathbf{h}^k$ is sampled from $P(\mathbf{h}^k|\mathbf{h}^{k-1})$ for the training of next layer, or estimated as $\hat{\mathbf{h}}^k = \int_{\mathbf{h}^k} P(\mathbf{h}^k|\mathbf{h}^{k-1})$
- **Because of the problem on inference, no joint optimization over all layers**



Fine tuning deep belief net

- Convert DBN to a normal multilayer neural network
- The outputs of hidden units are calculated from the inputs of the lower layer in a deterministic way

$$h_j^k = P(h_j^k = 1|\mathbf{h}^{k-1}) = \sigma(c_j + \mathbf{W}_{j\cdot} \cdot \mathbf{h}^{k-1})$$

$$\mathbf{h}^0 = \mathbf{x}$$

- Fine tune the network with backpropagation

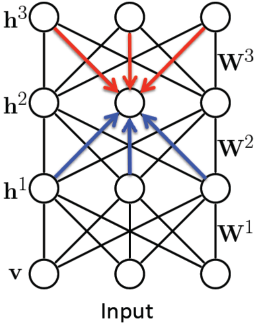
Deep Boltzmann Machine

- Unsupervised feature learning
- The following slides are borrowed from Salakhutdinov's tutorial at CVPR 2012

Model

$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + \underline{\mathbf{h}^{1\top} W^2 \mathbf{h}^2} + \underline{\mathbf{h}^{2\top} W^3 \mathbf{h}^3}\right]$$

Deep Boltzmann Machine



$\theta = \{W^1, W^2, W^3\}$ model parameters

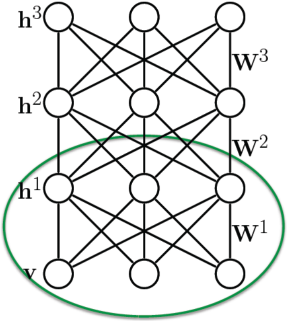
- Dependencies between hidden variables.
- All connections are undirected.
- Bottom-up and Top-down:

$$P(h_j^2 = 1 | \mathbf{h}^1, \mathbf{h}^3) = \sigma\left(\underbrace{\sum_k W_{kj}^3 h_k^3}_{\text{Top-down}} + \underbrace{\sum_m W_{mj}^2 h_m^1}_{\text{Bottom-up}} \right)$$

Learning

$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + \mathbf{h}^{1\top} W^2 \mathbf{h}^2 + \mathbf{h}^{2\top} W^3 \mathbf{h}^3\right]$$

Deep Boltzmann Machine



$\theta = \{W^1, W^2, W^3\}$ model parameters

- Dependencies between hidden variables.

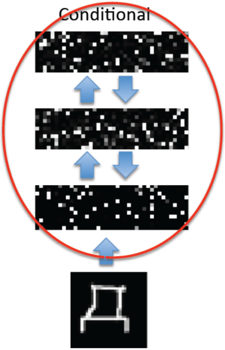
Maximum likelihood learning:

$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathrm{E}_{P_{data}}[\mathbf{v}\mathbf{h}^{1\top}] - \mathrm{E}_{P_\theta}[\mathbf{v}\mathbf{h}^{1\top}]$$

Problem: Both expectations are intractable!

Learning

Posterior Inference



Conditional

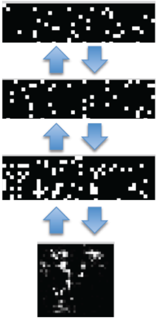
Approximate conditional

$P_{data}(\mathbf{h}|\mathbf{v})$

Simulate from the Model

Approximate the joint distribution

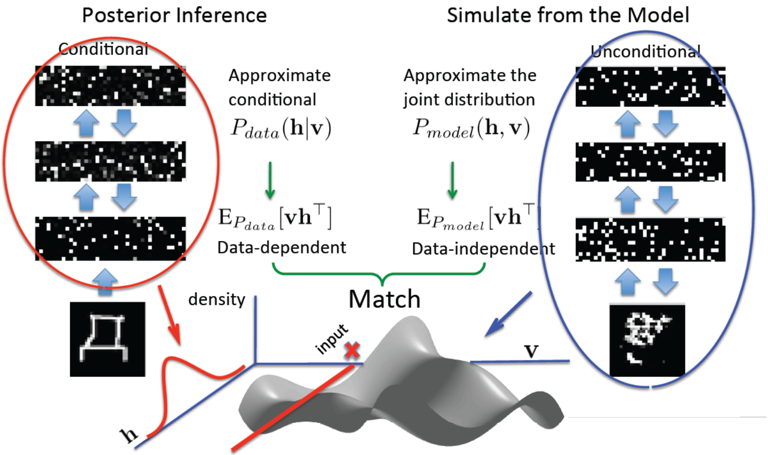
$P_{model}(\mathbf{h}, \mathbf{v})$



Unconditional

(Salakhutdinov, 2008; NIPS 2009)

Learning

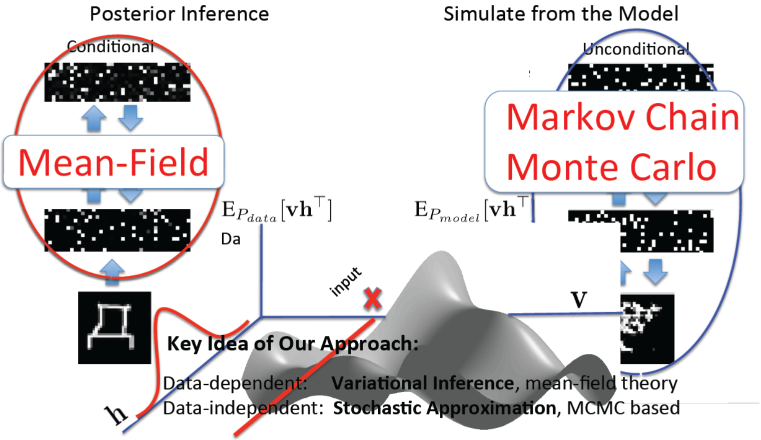


Posterior Inference

Simulate from the Model

Conditional
Unconditional
Approximate conditional
$P_{data}(\mathbf{h}|\mathbf{v})$
Approximate the joint distribution
$P_{model}(\mathbf{h},\mathbf{v})$
$\mathrm{E}_{P_{data}}[\mathbf{v}\mathbf{h}^{\top}]$
Data-dependent
$\mathrm{E}_{P_{model}}[\mathbf{v}\mathbf{h}^{\top}]$
Data-independent
Match
density
input
$\mathbf{v}$
$\mathbf{h}$

Learning



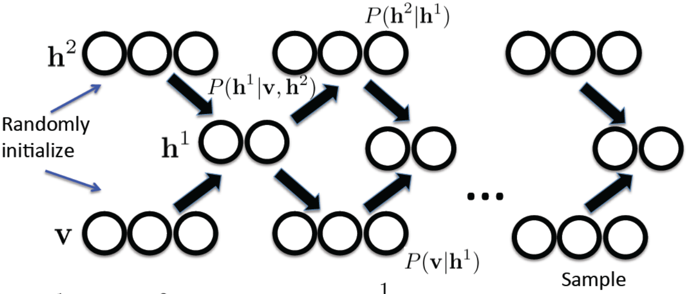
Key Idea of Our Approach:

Data-dependent: **Variational Inference**, mean-field theory

Data-independent: **Stochastic Approximation**, MCMC based

Sampling from DBMs

Sampling from two-hidden layer DBM: by running Markov chain:


$P(\mathbf{h}^2|\mathbf{h}^1)$
$\mathbf{h}^2$
$P(\mathbf{h}^1|\mathbf{v},\mathbf{h}^2)$
Randomly initialize
$\mathbf{h}^1$
...
$\mathbf{v}$
$P(\mathbf{v}|\mathbf{h}^1)$
Sample

$$P(h^1_m = 1|\mathbf{v},\mathbf{h}^2) = \frac{1}{1+\exp(-\sum_i W^1_{im} v_i - \sum_j W^2_{mj} h^2_j)}$$

$$P(h^2_j = 1|\mathbf{h}^1) = \frac{1}{1+\exp(-\sum_m W^2_{mj} h^1_m)}$$

$$P(v_i = 1|\mathbf{h}^1) = \frac{1}{1+\exp(-\sum_m W^1_{im} h^1_m)}$$

Reading materials

- G. E. Hinton, S. Osindero, and Y. Teh, "A Fast Learning Algorithm for Deep Belief Nets," Neural Computation, Vol. 18, pp. 1527-1544, 2006.
- H. Larochelle and Y. Bengio, "Classification using Discriminative Restricted Boltzmann Machines", ICML 2008.
- G. E. Hinton, "Products of Experts," Proc. Int'l Conf. Artificial Neural Networks, 1999.
- R. Salakhutdinov and G. Hinton, "Deep Boltzmann Machines," AISTATS 2009.