Restricted Boltzmann Machines

(Apr 5, 2017)

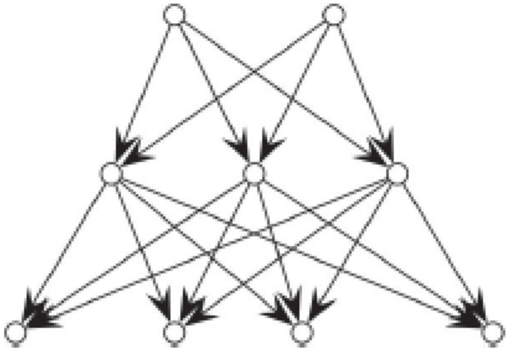
YANG Jiancheng

Outline

- I. Boltzmann Machines
- II. Restricted Boltzmann Machines
- III. Learning: CD and PCD

• I. Boltzmann Machines

- **Neural Network as Graph Models**



(a) (b) (c)

Figure 28.1 Some deep multi-layer graphical models. Observed variables are at the bottom. (a) A directed model. (b) An undirected model (deep Boltzmann machine). (c) A mixed directed-undirected model (deep belief net).

A common practice:

- **Directed models are easy to train, but not easy to inference in parallel**
- **Undirected models are hard to train (need to inference when training), but easy to parallel**

• I. Boltzmann Machines

- **Boltzmann Machines to Restricted Boltzmann Machines**

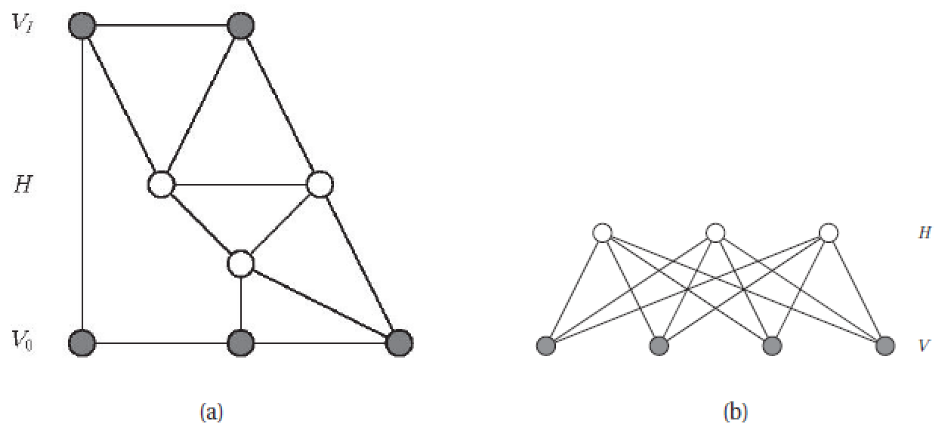


Figure 27.30 (a) A general Boltzmann machine, with an arbitrary graph structure. The shaded (visible) nodes are partitioned into input and output, although the model is actually symmetric and defines a joint density on all the nodes. (b) A restricted Boltzmann machine with a bipartite structure. Note the lack of intra-layer connections.

a) Boltzmann Machines are arbitrary graphs with hidden and visible units, so-called "energy based" model
b) Restricted versions are bipartite, i.e.
   - $h_i \perp h_j | \mathbb{v}$
   - $v_i \perp v_j | \mathbb{h}$

• II. Restricted Boltzmann Machines

- **Binary RBMs**

a) Most common
b) Binary hidden and binary visible units
c) Formula

$$
\begin{aligned}
p(\mathbf{v}, \mathbf{h}|\theta) &= \frac{1}{Z(\theta)} \exp(-E(\mathbf{v}, \mathbf{h}; \theta)) \\
E(\mathbf{v}, \mathbf{h}; \theta) &\triangleq -\sum_{r=1}^{R} \sum_{k=1}^{K} v_r h_k W_{rk} - \sum_{r=1}^{R} v_r b_r - \sum_{k=1}^{K} h_k c_k \\
&= -(\mathbf{v}^T \mathbf{W} \mathbf{h} + \mathbf{v}^T \mathbf{b} + \mathbf{h}^T \mathbf{c}) \\
Z(\theta) &= \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \theta))
\end{aligned}
$$

d) Inference

$$
\begin{aligned}
\mathbb{E}[\mathbf{h}|\mathbf{v}\theta] &= \mathrm{sigm}(\mathbf{W}^T \mathbf{v}) \\
\mathbb{E}[\mathbf{v}|\mathbf{h}, \theta] &= \mathrm{sigm}(\mathbf{W} \mathbf{h})
\end{aligned}
$$

$$
p(h|v, \theta) = \frac{p(h, v|\theta)}{p(v|\theta)} = \frac{e^{-v^T wh}}{\sum_h e^{-v^T wh}} = \frac{1}{1 + e^{-w^T v}}
$$

• II. Restricted Boltzmann Machines

- **Variants of RBMs**

a) Categorical RBMs

$$E(\mathbf{v},\mathbf{h};\theta) \triangleq -\sum_{r=1}^{R}\sum_{k=1}^{K}\sum_{c=1}^{C} v_r^c h_k W_{rk}^c - \sum_{r=1}^{R}\sum_{c=1}^{C} v_r^c b_r^c - \sum_{k=1}^{K} h_k c_k$$

$$p(v_r|\mathbf{h},\theta) = \mathrm{Cat}(\mathcal{S}(\{b_r^c + \sum_k h_k W_{rk}^c\}_{c=1}^{C}))$$

$$p(h_k = 1|\mathbf{c},\theta) = \mathrm{sigm}(c_k + \sum_r \sum_c v_r^c W_{rk}^c)$$

$$S = softmax$$

b) Gaussian RBMs

$$E(\mathbf{v},\mathbf{h}|\theta) = -\sum_{r=1}^{R}\sum_{k=1}^{K} W_{rk} h_k v_r - \frac{1}{2}\sum_{r=1}^{R}(v_r - b_r)^2 - \sum_{k=1}^{K} a_k h_k$$

$$p(v_r|\mathbf{h},\theta) = \mathcal{N}(v_r | b_r + \sum_k w_{rk} h_k, 1)$$

$$p(h_k = 1|\mathbf{v},\theta) = \mathrm{sigm}\left(c_k + \sum_r w_{rk} v_r\right)$$

• III. Learning: CD and PCD

- **Objective: Maxent models (1)**

a) If all units are visible, then it's a Markov Random Field (MRF), in log-linear form:

$$p(v|\theta) = \frac{1}{Z(\theta)} \exp\left(\sum_{c} \theta_c^T \phi_c(v)\right)$$

$$\ell(\theta) = \frac{1}{N} \sum log(p|\theta) = \frac{1}{N} \sum \left(\sum_{c} \theta_c^T \phi_c(v) - logZ(\theta)\right)$$

$$\frac{\partial \ell}{\partial \theta_c} = \frac{1}{N} \sum_{i} [\phi_c(v_i) - \frac{\partial}{\partial \theta_c} \log Z(\theta))]$$

$$= \frac{1}{N} \sum_{i} \phi_c(v_i) - \mathbb{E}_v(\phi_c(v)|\theta)$$

$$= E_{emp}\phi_c(v) - E_{model}\phi_c(v)$$

III. Learning: CD and PCD

- **Objective: Maxent models (2)**

Proof:

$$\frac{\partial}{\partial \theta_c} \log Z(\theta) = \frac{1}{Z(\theta)} \sum_v \frac{\partial}{\partial \theta_c} \exp\left(\sum_c \theta_c^T \phi_c(v)\right)$$

$$= \sum_v \phi_c(v) \frac{\exp\left(\sum_c \theta_c^T \phi_c(v)\right)}{Z(\theta)} = \sum_v \phi_c(v) p(v|\theta)$$

$$= \mathbb{E}_v(\phi_c(v)|\theta)$$

• III. Learning: CD and PCD

- **Objective: Partially observed maxent models**

b) If some units are hidden:

$$p(v,h|\theta) = \frac{1}{Z(\theta)} \exp\left(\sum_{c} \theta_c^T \phi_c(v,h)\right)$$

$$\ell(\theta) = \frac{1}{N} \sum log(p|\theta)$$

Similarly (but with some efforts),

$$\frac{\partial \ell}{\partial \theta_c} = \frac{1}{N} \sum_{i} \mathbb{E}_h \phi_c(v_i, h|\theta) - \mathbb{E}_{h,v} \phi_c(v,h|\theta)$$

III. Learning: CD and PCD

- **Objective: RBMs**

c) RBMs are bipartite with visible and hidden units

$$\frac{\partial \ell}{\partial \theta_c} = \frac{1}{N}\sum_i \mathbb{E}_h \phi_c(v_i, h|\theta) - \mathbb{E}_{h,v}\phi_c(v, h|\theta)$$

$$= \frac{1}{N}\sum_i v_{c,i}\, sigm(v_i, \theta) - \mathbb{E}_{h,v}\phi_c(v, h|\theta)$$

The first part is easy, but what about the second part?

• III. Learning: CD and PCD

- **Block Gibbs Sampling**

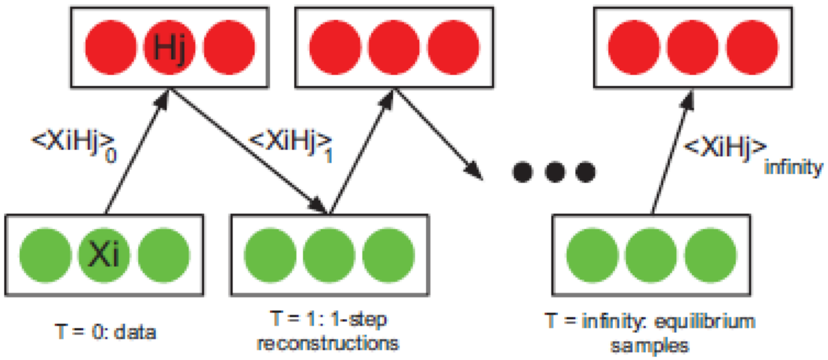


Figure 27.31 Illustration of Gibbs sampling in an RBM. The visible nodes are initialized at a datavector, then we sample a hidden vector, then another visible vector, etc. Eventually (at "infinity") we will be producing samples from the joint distribution $p(\mathbf{v}, \mathbf{h}|\theta)$.

Key ideas of the second part: MCMC

- **Sampling: fantasy data**

III. Learning: CD and PCD

- **Contrastive divergence (CD)**

Algorithm 27.3: CD-1 training for an RBM with binary hidden and visible units

Initialize weights $\mathbf{W} \in \mathbb{R}^{R \times K}$ randomly;
$t := 0$;
for *each epoch* **do**
  $t := t + 1$;
  for *each minibatch of size B* **do**
    Set minibatch gradient to zero, $\mathbf{g} := \mathbf{0}$;
    for *each case $\mathbf{v}_i$ in the minibatch* **do**
      Compute $\boldsymbol{\mu}_i = \mathbb{E}[\mathbf{h}|\mathbf{v}_i, \mathbf{W}]$;
      Sample $\mathbf{h}_i \sim p(\mathbf{h}|\mathbf{v}_i, \mathbf{W})$;
      Sample $\mathbf{v}'_i \sim p(\mathbf{v}|\mathbf{h}_i, \mathbf{W})$;
      Compute $\boldsymbol{\mu}'_i = \mathbb{E}[\mathbf{h}|\mathbf{v}'_i, \mathbf{W}]$;
      Compute gradient $\nabla_{\mathbf{W}} = (\mathbf{v}_i)(\boldsymbol{\mu}_i)^T - (\mathbf{v}'_i)(\boldsymbol{\mu}'_i)^T$;
      Accumulate $\mathbf{g} := \mathbf{g} + \nabla_{\mathbf{W}}$;
    Update parameters $\mathbf{W} := \mathbf{W} + (\alpha_t/B)\mathbf{g}$

• III. Learning: CD and PCD

- **Persistent CD (PCD)**

Algorithm 27.4: Persistent CD for training an RBM with binary hidden and visible units

Initialize weights $\mathbf{W} \in \mathbb{R}^{D \times L}$ randomly;
Initialize chains $(\mathbf{v}_s, \mathbf{h}_s)_{s=1}^{S}$ randomly ;
for $t = 1, 2, \ldots$ **do**
  // Mean field updates ;
  for *each case* $i = 1 : N$ **do**
    $\mu_{ik} = \text{sigm}(\mathbf{v}_i^T \mathbf{w}_{:,k})$
  // MCMC updates ;
  for *each sample* $s = 1 : S$ **do**
    Generate $(\mathbf{v}_s, \mathbf{h}_s)$ by brief Gibbs sampling from old $(\mathbf{v}_s, \mathbf{h}_s)$
  // Parameter updates ;
  $\mathbf{g} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{v}_i (\boldsymbol{\mu}_i)^T - \frac{1}{S} \sum_{s=1}^{S} \mathbf{v}_s (\mathbf{h}_s)^T$;
  $\mathbf{W} := \mathbf{W} + \alpha_t \mathbf{g}$;
  Decrease α_t

Bibliography

- MLaPP: Chapter 27.7 Restricted Boltzman machines (RBMs)
- Neural Networks for Machine Learning by Geoffrey Hinton (Coursera): Week 12,13.14

Thanks for listening!