Notes: A fast learning algorithm for deep belief nets

Jiaxin Shi
Department of Computer Science
Tsinghua University
Beijing, 100084
ishijiaxin@126.com

1 Motivation: Solve explaining away

The motivation of this paper is to solve the difficulties caused by explaining away in learning deep directed belief nets. This is a general problem met in directed model inference. We will illustrate below.

Explaining away makes inference hard in directed models

Suppose we have a v-structure model $p(v|h_1, h_2)$, where explaining away exists when v observed. This means that $p(h_1 = 1|v, h_2 = 1)$ is extremely small. Though this is a natural phenomenon because one reason explains away another given the result, it still makes it hard for Gibbs sampling to jump from $h_2 = 1$ to $h_1 = 1$, causing slow mixing rate, or in other words, Gibbs sampling stuck in the local region. This problem is more serious when it comes to directed model with lots of hidden variables.

2 Construct the ideal model without explaining away

Because inference is hard when explaining away exists, we may ask if there are directed models without this phenomenon. A good could be the posterior over hidden variables are factorized. According to Hammersley-Clifford Theorem, their joint likelihood should have the densities like

$$p(v, h) \propto \exp\left\{\sum_j \Phi_j(v, h_j) + \beta(v) + \sum_j \alpha_j(h_j)\right\}$$

Then

$$p(v|h) \propto p(v, h) = \frac{1}{\Omega(h)} \exp\left\{\sum_j \Phi_j(v, h_j) + \beta(v)\right\}$$

which requires the prior $p(h)$ to be designed as

$$p(h) = \frac{p(v, h)}{p(v|h)} \propto \exp\left\{\log \Omega(h) + \sum_j \alpha_j(h_j)\right\}$$

Above is what they called as complementary prior in the original paper.

The problem is we don't know $\Omega(h)$ when we design. So what the authors do is to avoid defining $p(v|h)p(h)$ by directly defining the two conditionals: the model conditional $p(v|h)$ and the posterior conditional $p(h|v)$. By defining $p(h|v)$ to be factorized, the implicit $p(h)$ is forced to be complementary prior, though we don't get the closed-form of it.

3 Extend to infinite layer model without explaining away

In the last section we build a 1-layer model $p(v, h)$ without explaining away by directly defining its $p(v|h)$ and $p(h|v)$. Now it's kind of easy for us to extend it into a deep layered model (in fact, even an infinite layer model). The intuitive explanation is to alternatively stack conditionals of $p(v|h)$ and $p(h|v)$ into deeper layers. Because all h and v in previous layer are in the marginal distribution of $p(v, h)$, the h or v in the newly add layer is also sampled from the same marginal distribution. We stack the model by bottom-up procedure and defines the bottom-up conditional, then we can prove that the top-down conditionals are also alternating $p(v|h)$ and $p(h|v)$s, which is intuitive in our above explanation. The proof is by mathematical induction as below. Let $x = x^{(0)}, y = y^{(0)}, x^{(1)}, y^{(1)}, x^{(2)}, y^{(2)}...$ be a sequence of variables and $x^{(0)}, y^{(0)}$ are identified as original observed and hidden variables.

We define functions over dummy variables y', x'

$$f(x', y') = \frac{1}{Z} \exp(\sum_{i,j} \Psi_{i,j}(x'_i, y'_j) + \sum_i \gamma_i(x'_i) + \sum_j \alpha_j(y'_j))$$

$$f_x(x') = \sum_{y'} f(x', y')$$

$$f_y(y') = \sum_{x'} f(x', y')$$

$$g_x(x'|y') = \frac{f(x', y')}{f_y(y')}$$

$$g_y(y'|x') = \frac{f(x', y')}{f_x(x')}$$

Define a joint distribution over the sequence of variables

$$P(x^{(0)}, y^{(0)}) = f(x^{(0)}, y^{(0)})$$

$$P(x^{(i)}|y^{(i-1)}) = g_x(x^{(i)}|y^{(i-1)}) \;\; i = 1, 2, ...$$

$$P(y^{(i)}|x^{(i)}) = g_y(y^{(i)}|x^{(i)}) \;\; i = 1, 2, ...$$

By induction we have the following marginal distributions

$$P(x^i) = f_x(x^i)$$

$$P(y^{(i)}) = f_y(y^{(i)})$$

$$P(x^{(i)}) = \sum_{y^{(i-1)}} P(x^{(i)}|y^{(i-1)})P(y^{(i-1)}) = f_x(x^{(i)})$$

Then the following "downward" conditional distributions also hold true:

$$P(x^{(i)}|y^{(i)}) = P(x^{(i)}, y^{(i)})/P(y^{(i)}) = g_x(x^{(i)}|y^{(i)})$$

$$P(y^{(i)}|x^{(i+1)}) = P(y^{(i)}, x^{(i+1)})/P(x^{(i+1)}) = g_y(y^{(i)}|x^{(i+1)})$$

4 The infinite tied weight belief nets is equivalent to RBM

A very insightful point in this paper is that they identify the infinite tied weight belief nets in the above section is equivalent to Restricted Boltzman Machine. The key observation is that the top down generation process of this infinite model can be seen as the Gibbs sampling chain for RBM. That is, alternating updates for h and v.

$$h \sim p(h|v)$$

$$v \sim p(v|h)$$

RBM is an well-known undirected model with an effective learning algorithm called Contrastive Divergence. So the learning of the infinite tied weight belief nets can be transformed into a learning process of RBM, thus solved by Constrastive Divergence. Contrastive Divergence is just an approximate Maximum likelihood Estimation for RBM, by taking gradients on the data likelihood $\log p(v_0)$ and approximate the latter term with k-step Gibbs sampling for estimation of the expectation under model distribution.

$$\frac{\partial \log p(v_0)}{\partial w_{ij}} = \mathbb{E}_{Data}[v_i h_j] - \mathbb{E}_{Model}[v_i h_j] = \mathbb{E}_{p(h_j|v_i^0)}[v_i h_j] - \mathbb{E}_{p(h,v)}[v_i h_j]$$

5 Learning by variational inference: Untied weight layers as variational posterior

The learning process for the deep belief nets with untied weights is by variational inference. Generally, for a directed model $p(h, v) = p(h)p(v|h)$, it's hard to get the gradients on $\log p(v)$ for the parameters because of normalization is intractable. However, there is an workaround for this by using a lower bound of $\log p(v)$.

$$\begin{aligned} \log p(v) &\geq \log p(v) - KL(Q(h|v)||p(h|v)) \\ &= \mathbb{E}_{Q(h|v)}[\log p(h, v) - \log Q(h|v)] \\ &= \mathbb{E}_{Q(h|v)}[\log p(h) + \log p(v|h) - \log Q(h|v)] \end{aligned}$$

$Q(h|v)$ is the bottom layers when training one layer of the deep belief nets from bottom to top. Because they have been untied and fixed, maximize the lower bound above is equal to maximize $\log p(h)$, which is MLE of the upper infinite tied weight model. It can be accomplished with Contrastive Divergence learning of a RBM with the same weights.

Above is all critical points for the learning of deep belief nets.