CPSC 550: Machine Learning II	2008/9 Term 2

Lecture 14 — March 5th, 2009

Lecturer: Nando de Freitas *Scribe: David Duvenaud*

This lecture derives the learning rules for Gaussian Restricted Boltzmann Machines, along with several variations. It also introduces deterministic autoencoders and derives the learning rule for them.

Gaussian Restricted Boltzmann Machines

In previous lectures, we derived the learning rules for Restricted Boltzmann Machines with binary visible units (inputs). These RBMs took the form:

- Binary visible units $v_i \in \{0, 1\}$
- Binary hidden units $h_j \in \{0, 1\}$
- Parameters $\theta = (c, b, w, \sigma^2)$, where
  - c_i is the bias on visible node i,
  - b_j is the bias on hidden node j,
  - w_{ij} is the weight between visible node j and hidden node i

A simple variant would be one in which the visible units $v_i \in \mathbb{R}$ each had a Gaussian distribution $\mathcal{N}(c_i + \sum_j w_{ij}h_j, \sigma_i^2)$. In this case, the joint probability of $V = v, H = h$ is given by:

$$P_\theta(v, h) = Z(\theta)^{-1} \exp\left\{ (-\frac{1}{2}\sum_i \frac{1}{\sigma_i^2}\left(v_i^2 - 2c_i v_i + c_i^2\right) + \sum_i \sum_j \frac{1}{\sigma_i^2} v_i w_{ij} h_j + \sum_j b_j h_j \right\}$$

We will now show how this joint distribution induces a Normal distribution on the visible nodes given the hidden nodes. Ignoring terms not depending on v, we can get an unnormalized formula for $P(v|h)$:

$$P_\theta(v|h) \propto \exp\left\{ -\frac{1}{2}\sum_i \frac{1}{\sigma_i^2}\left(v_i^2 - 2v_i[c_i + \sum_j w_{ij}h_j] + k_i^2 \right) \right\}$$

Which has a quadratic form (for some constant k_i). Thus we can complete the squares, and again drop terms independent of v to get:

$$P_\theta(v|h) \propto \exp\left\{ -\frac{1}{2}\sum_i \frac{1}{\sigma_i^2}\left(v_i^2 - [c_i + \sum_j w_{ij}h_j] \right)^2 \right\}$$

Which has the form of a normal distribution where each unit is independently given by:

$$\mathcal{N}(c_i + \textstyle\sum_j w_{ij} h_j, \sigma_i^2).$$

By using Bayes' Theorem we can find an expression for $P(h_j|v)$. There are two cases:

$$P(h_j = 1|v) \propto \exp\left\{ b_j + \sum_i \frac{1}{\sigma_i^2} v_i w_{ij} \right\}$$
$$P(h_j = 0|v) \propto 1$$

Thus when we normalize, we get

$$P(h_j = 1|v) = \text{logit}\left(b_j + \sum_i \frac{v_i w_{ij}}{\sigma_i^2} \right) \tag{14.1}$$

14.0.1 Beta RBMs

Another variant of RBMs is one in which the data have range $v_i \in [0, 1]$. Then, we may wish to define our observation model with a Beta distribution:

$$P(v_i) \propto v_i^{\alpha-1} (1 - v_i)^{\beta-1}.$$

In this case, our joint model becomes

$$P_\theta(v, h) = Z(\theta)^{-1} \exp\left\{ (-\frac{1}{2} \sum_i \frac{1}{\sigma_i^2} \left[(\alpha_i - 1) \log(v_i) + (\beta_i - 1) \log(1 - v_i) - 2c_i v_i + c_i^2 \right] \right.$$
$$\left. + \sum_i \sum_j \frac{1}{\sigma_i^2} v_i w_{ij} h_j + \sum_j b_j h_j \right\}$$

14.0.2 Rao-Blackwellization of Contrastive Divergence

The basic contrastive divergence model defines:

$$s = d_i \tilde{h}_j - \tilde{\tilde{v}}_i \tilde{\tilde{h}}_j$$

Where s is the estimated likelihood gradient for W_{ij}, d is the observed data, $\tilde{h}$ are samples of the hidden nodes, and $\tilde{\tilde{v}}$ are samples of the visible nodes given the sampled hidden nodes $\tilde{h}$. This can be called the Monte Carlo estimator of s. We may be able to use our knowledge of the distribution of h to create a better esimate of s. Specifically, we could use the Rao-Blackwellized estimator:

$$s_{RB} = d_i P(h_j = 1|d) - \tilde{\tilde{v}}_i P(h_j = 1|\tilde{\tilde{v}})$$

Where we have replaced the Monte Carlo estimates $d\tilde{h}$ and $d\tilde{\tilde{h}}$ with their expectations given d and $\tilde{\tilde{v}}$, respectively. Recall:

$$\mathbb{E}[dh_j|d] = d\mathbb{E}[h_j|d] = d\sum_{h'_j} I(h_j = h'_j)P(h'_j|d) = dP(h_j = 1|d)$$

The Rao-Blackwell Theorem states that this estimator will have lower variance than the Monte Carlo estimator.

14.0.3 Justification

Specifically, the Rao-Blackwell theorem states that, for an unbiased estimator X,

$$\mathrm{Var}(X) \geq \mathrm{Var}(\mathbb{E}(X|Y))$$

meaning that we can sometimes improve (and never worsen) our estimator by replacing it with its expectation given some relevant statistic Y. In this case, we use the related inequality

$$\mathrm{Var}(dh) \geq \mathrm{Var}(d\mathbb{E}(h|d))$$

(Note that here, d is a variable, not the differentiation operator)

Proof:

$$\begin{aligned}
\mathrm{Var}(dh) &= \mathbb{E}(d^2h^2) - [\mathbb{E}(dh)]^2 \\
&= \mathbb{E}[\mathbb{E}(d^2h^2|d)] - [\mathbb{E}[\mathbb{E}(dh|d)]]^2 \\
&= \mathbb{E}[\mathrm{Var}(dh|d) + [\mathbb{E}(dh|d)]^2] - [\mathbb{E}[\mathbb{E}(dh|d)]]^2 \\
&= \underbrace{\mathbb{E}[\mathrm{Var}(dh|d)}_{\text{always positive}} + \mathrm{Var}[\mathbb{E}(dh|d)]] \\
&\geq \mathrm{Var}[\mathbb{E}(dh|d)]] \\
&= \mathrm{Var}[d\mathbb{E}(h|d)]]
\end{aligned}$$

□

Armed with this knowledge, we can replace the standard contrastive divergence estimator

$$s = d_i \tilde{h}_j - \tilde{\tilde{v}}_i \tilde{\tilde{h}}_j$$

with the Rao-Blackwellized version:

$$d_i P(h_j = 1|d) - P(v_i = 1|\tilde{\tilde{h}}) P(h_j = 1|\tilde{\tilde{v}})$$

which will be gauranteed to give as low or lower variance than our original estimator.

14.0.4 Example: Rao-Blackwellizing a Monte Carlo Estimator

This example will show another example of Rao-Blackwellizing a Monte Carlo estimator in order to get an estimator with lower variance. Consider a switching model, such as a mixture of Gaussians, with parameters (θ, Z) with $\theta \in \mathbb{R}, Z \in 1...k$ and

$$P(\theta, Z) = P(\theta|Z)P(Z)$$

and

$$P(Z = z) = \pi_z$$

If we wish to find $P(\theta \in A)$, we could use a Monte Carlo estimator, which first samples Z, then samples θ, and averages over all samples:

$$\hat{P}(\theta \in A) = \frac{1}{N} \sum_{i=1}^{N} I(\theta^{(i)} \in A).$$

Or, we could replace the estimator $\hat{P}$ with its expectation given Z

$$\hat{P}(\theta \in A) = \frac{1}{N} \sum_{i=1}^{N} I(\theta^{(i)} \in A) \quad (14.2)$$

$$\hat{P}_{RB}(\theta \in A) = \frac{1}{N} \sum_{i=1}^{N} P(\theta \in A|Z^{(i)}) \quad (14.3)$$

and do the same for $P(Z)$:

$$\hat{P}(\theta \in A) = \sum_z P(\theta \in A|z) \frac{1}{N} \sum_{i=1}^{N} I(Z^{(i)} = z) \tag{14.4}$$

$$\hat{P}_{RB}(\theta \in A) = \sum_z P(\theta \in A|z) P(z) \tag{14.5}$$

Thereby recovering the exact expression of $P(\theta \in A)$. Of course, the exact expression has zero variance, which is better than our original estimator. In general, the Rao-Blackwellization method is applicable if an intractable joint distribution can be factored into a conditional distribution times an unconditional distribution.

14.0.5 Autoencoders

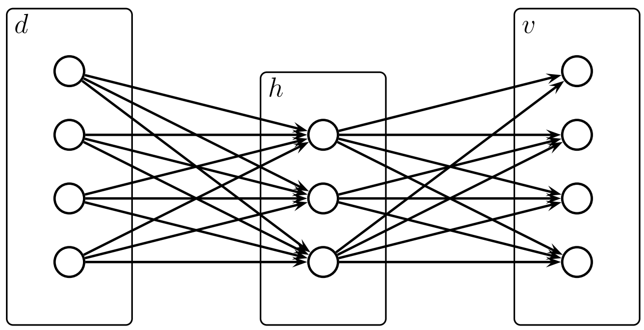

Figure 14.1. An autoencoder network, with 4 observed nodes and 3 hidden nodes.

We define an autoencoder as a transformation from the data d to a set of hidden units h, and from the hidden units back to visible units v in the same domain as the data. Typically, the number of hidden units h is smaller than the number of visible units, meaning that the data is being econded in a lower-dimensional representation.

A sigmoid autoencoder has the following transformations defined: For an individual data point indexed by t, the state of hidden unit h_j is defined as:

$$h_{jt} = \sigma\left(\sum_i w_{ij} d_{it}\right) \qquad\qquad v_{it} = \sigma\left(\sum_j w_{ij} h_{jt}\right)$$

where

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

We can assign an L_2 loss function on the reconstruction of the data:

$$C(w) = \sum_t \sum_i (d_{it} - v_{it})^2$$

or we can assign a cross-entropy loss function:

$$C(w) = \sum_t \sum_i \left[d_{it} \log(v_{it}) + (1 - d_{it}) \log(1 - v_{it})\right].$$

Note: The derivative of $\sigma(x) = \sigma(1 - \sigma(x))$.

Assuming a Bernoulli distribution on d, and using cross-entropy loss, we can now solve for the backpropagation gradient of w:

$$\begin{aligned}
\frac{\delta C(w)}{\delta w_{ij}} &= \sum_t \left[d_{it} \frac{\frac{\delta}{\delta w_{ij}} v_{it}}{v_{it}} + (1 - d_{it}) \frac{\frac{\delta}{\delta w_{ij}}(1 - v_{it})}{1 - v_{it}} \right] \\
&= \sum_t \left[d_{it}(1 - v_{it}) \left(w_{ij} \frac{\delta h_{jt}}{\delta w_{ij}} + h_{jt} \right) - (1 - d_{it}) v_{it} \left(w_{ij} \frac{\delta h_{jt}}{\delta w_{ij}} + h_{jt} \right) \right] \\
&= \sum_t \left[(d_{it} - v_{it})(h_{jt} + w_{ij} h_{jt}(1 - h_{jt}) d_{it}) \right]
\end{aligned}$$

Note that v_{it} is a function of h, and h_{jt} is a function of d.