Tutorial:
restricted Boltzmann machines

Chris J. Maddison

March 27, 2014

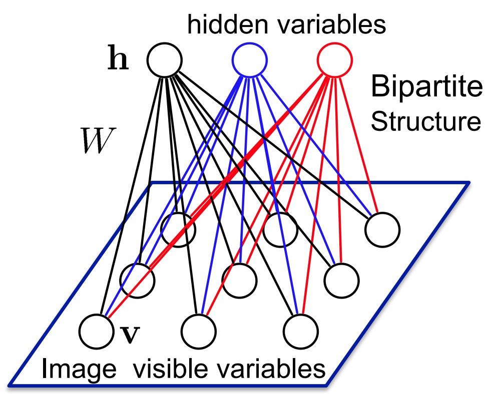
1 restricted Boltzmann machines

A Boltzmann machine is a family of probability distributions over binary vectors $\mathbf{s}$ of length K

$$\mathrm{P}(\mathbf{s}) = \exp\left(\sum_{1\leq i<j\leq K} W_{ij}s_is_j + \sum_{i=1}^{K} b_is_i\right)/Z \equiv \frac{\exp(-E(\mathbf{s}))}{Z} \qquad s_i \in \{0,1\}, W_{ij}, b_i \in \mathbb{R}$$

where $Z = \sum_{\mathbf{s}} \exp(-E(\mathbf{s}))$ is the sum over all possible configurations of $\mathbf{s}$.

A restricted Boltzmann machine (RBM) has a bipartite structure: partition $\mathbf{s}$ into V visible bits $\mathbf{v}$ and H hidden bits $\mathbf{h}$ and set W_{ij} to zero if it connects a hidden bit to a hidden bit or a visible bit to a visible bit.



The energy is a function of the configuration and parameters, but we omit the parameters sometimes if the parameters are implied

$$-E(\mathbf{v}, \mathbf{h}) = \sum_{i=1}^{H}\sum_{j=1}^{V} W_{ij}h_iv_j + \sum_{i=1}^{n} b_ih_i + \sum_{j=1}^{n} c_jv_j$$

2 gradients

Fit an RBM to a data set of bit vectors $(\mathbf{v}_1, \ldots, \mathbf{v}_N)$ by following the average gradient (with respect to the parameters W, b, c)

$$\frac{1}{N}\sum_{n=1}^{N} \nabla \log \mathrm{P}(\mathbf{v}_n)$$

We need the partial derivatives of

$$\begin{aligned}
\log \mathrm{P}(\mathbf{v}_n) &= \log\left(\sum_{\mathbf{h}} \mathrm{P}(\mathbf{v}_n, \mathbf{h})\right) \\
&= \log\left(\sum_{\mathbf{h}} \frac{\exp(-E(\mathbf{v}_n, \mathbf{h}))}{Z}\right) \\
&= \log\left(\sum_{\mathbf{h}} \exp(-E(\mathbf{v}_n, \mathbf{h}))\right) - \log Z
\end{aligned}$$

We show how to derive the derivative of the first term. Recall,

$$\begin{aligned}
\frac{\partial}{\partial\theta}\log f(\theta) &= \frac{\frac{\partial}{\partial\theta} f(\theta)}{f(\theta)} \\
\frac{\partial}{\partial\theta}\exp f(\theta) &= \exp(f(\theta))\frac{\partial}{\partial\theta} f(\theta)
\end{aligned}$$

So for parameter θ

$$\begin{aligned}
\frac{\partial}{\partial\theta}\log\left(\sum_{\mathbf{h}} \exp(-E(\mathbf{v}_n, \mathbf{h}))\right) &= \frac{1}{\sum_{\mathbf{h}} \exp(-E(\mathbf{v}_n, \mathbf{h}))}\frac{\partial}{\partial\theta}\sum_{\mathbf{h}} \exp(-E(\mathbf{v}_n, \mathbf{h})) \\
&= \frac{1}{\sum_{\mathbf{h}} \exp(-E(\mathbf{v}_n, \mathbf{h}))}\sum_{\mathbf{h}} \exp(-E(\mathbf{v}_n, \mathbf{h}))\frac{\partial}{\partial\theta} - E(\mathbf{v}_n, \mathbf{h}) \\
&= \sum_{\mathbf{h}} \mathrm{P}(\mathbf{h}|\mathbf{v} = \mathbf{v}_n)\frac{\partial}{\partial\theta} - E(\mathbf{v}, \mathbf{h}) \\
&= \mathbb{E}\left[\frac{\partial}{\partial\theta} - E(\mathbf{v}, \mathbf{h})\middle|\mathbf{v} = \mathbf{v}_n\right]
\end{aligned}$$

A similar trick works for the second term and we get the partial derivative

$$\frac{\partial}{\partial\theta}\log \mathrm{P}(\mathbf{v}_n) = \overbrace{\mathbb{E}\left[\frac{\partial}{\partial\theta} - E(\mathbf{v}, \mathbf{h})\middle|\mathbf{v} = \mathbf{v}_n\right]}^{\text{positive statistic}} - \underbrace{\mathbb{E}\left[\frac{\partial}{\partial\theta} - E(\mathbf{v}, \mathbf{h})\right]}_{\text{negative statistic}}$$

For the RBM

$$\frac{\partial}{\partial W_{ij}} \log \mathrm{P}(\mathbf{v}_n) = \mathbb{E}\left[h_i v_j | \mathbf{v} = \mathbf{v}_n\right] - \mathbb{E}\left[h_i v_j\right]$$
$$\frac{\partial}{\partial b_i} \log \mathrm{P}(\mathbf{v}_n) = \mathbb{E}\left[h_i | \mathbf{v} = \mathbf{v}_n\right] - \mathbb{E}\left[h_i\right]$$
$$\frac{\partial}{\partial c_j} \log \mathrm{P}(\mathbf{v}_n) = \mathbb{E}\left[v_j | \mathbf{v} = \mathbf{v}_n\right] - \mathbb{E}\left[v_j\right]$$

This is how it corresponds to the notation in the lectures

$$\mathbb{E}[h_i v_j | \mathbf{v} = \mathbf{v}_n] = \langle h_i v_j \rangle_{data}$$

That is the expected value under the model of the product of hidden unit j and visible unit j when $\mathbf{v}$ is clamped to $\mathbf{v}_n$ and

$$\mathbb{E}[h_i v_j] = \langle h_i v_j \rangle_{model}$$

is the expected number of times that h_i and v_j are both on if we sample from the model. We can vectorize everything:

$$-E(\mathbf{v}, \mathbf{h}) = \mathbf{h}^T W \mathbf{v} + \mathbf{h}^T b + \mathbf{v}^T c$$

with gradients

$$\nabla_W \log \mathrm{P}(\mathbf{v}_n) = \mathbb{E}\left[\mathbf{h}\mathbf{v}^T \middle| \mathbf{v} = \mathbf{v}_n\right] - \mathbb{E}\left[\mathbf{h}\mathbf{v}^T\right]$$
$$\nabla_b \log \mathrm{P}(\mathbf{v}_n) = \mathbb{E}\left[\mathbf{h} | \mathbf{v} = \mathbf{v}_n\right] - \mathbb{E}\left[\mathbf{h}\right]$$
$$\nabla_c \log \mathrm{P}(\mathbf{v}_n) = \mathbb{E}\left[\mathbf{v} | \mathbf{v} = \mathbf{v}_n\right] - \mathbb{E}\left[\mathbf{v}\right]$$

Remember to get a gradient on a batch we have to average the individual gradients!

3 computing gradients & contrastive divergence

In this section we talk about how to compute $\mathbb{E}\left[h_i v_j | \mathbf{v} = \mathbf{v}_n\right] - \mathbb{E}\left[h_i v_j\right]$ or approximations to it. For the positive statistic we are conditioning on $\mathbf{v}_n$, so we can take it out of the expected value:

$$\mathbb{E}\left[h_i | \mathbf{v} = \mathbf{v}_n\right] v_{nj}$$

$\mathbb{E}\left[h_i | \mathbf{v} = \mathbf{v}_n\right]$ is just the probability that h_i is on when $\mathbf{v}$ is clamped; this is sometimes called the *activation*:

$$\mathbb{E}\left[h_i | \mathbf{v} = \mathbf{v}_n\right] = \frac{1}{1 + \exp(-\sum_j W_{ij} v_{nj} - b_i)}$$

Two quick notes about this

- $\sigma(x) = 1/(1+\exp(-x))$ is called the *logistic function*
- I will use the convention that $\sigma(\mathbf{x})$ of a vector $\mathbf{x}$ is taken component-wise

So we can see that

$$\mathbb{E}\left[\mathbf{h}|\mathbf{v}=\mathbf{v}_n\right] = \sigma(W\mathbf{v}_n + b)$$

The negative statistic is the real problem. With M true samples $(\mathbf{v}_m, \mathbf{h}_m)$ from the distribution defined by the RBM, we could approximate

$$\mathbb{E}\left[h_i v_j\right] \approx \frac{1}{M}\sum_{m=1}^{M} h_{mi} v_{mj}$$

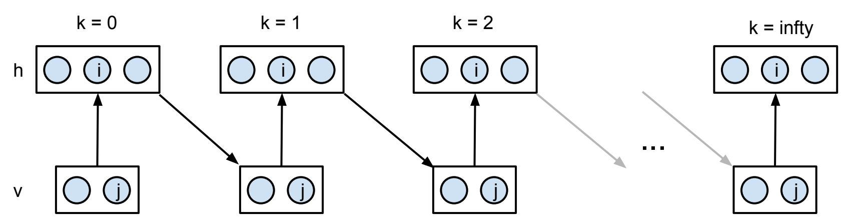
Can get these samples by initializing N independent Markov chain at each data point $\mathbf{v}_n$ and running until convergence $(\mathbf{v}_n^\infty, \mathbf{h}_n^\infty)$. Then,

$$\mathbb{E}\left[h_i v_j\right] \approx \frac{1}{N}\sum_{n=1}^{N} h_{ni}^\infty v_{nj}^\infty$$

The type of Markov transition operator used most often is alternating Gibbs.

$$\begin{aligned}
&\mathbf{v}_n^0 = \mathbf{v}_n \\
&\mathbf{h}_n^k \sim \mathrm{P}(\mathbf{h}|\mathbf{v}=\mathbf{v}_n^k) \text{ for } k \geq 0 \\
&\mathbf{v}_n^k \sim \mathrm{P}(\mathbf{v}|\mathbf{h}=\mathbf{h}_n^{k-1}) \text{ for } k \geq 1
\end{aligned}$$

and in pictures



Sampling from $\mathrm{P}(\mathbf{h}|\mathbf{v}_n^k)$ is easy, compute $\mathbb{E}\left[\mathbf{h}|\mathbf{v}=\mathbf{v}_n\right]$ and sample each bit independently with probability $\mathbb{E}\left[h_i|\mathbf{v}=\mathbf{v}_n\right]$. Similarly for $\mathrm{P}(\mathbf{v}|\mathbf{h}=\mathbf{h}_n^{k-1})$.

The idea behind contrastive divergence is to run the Markov chain for only one step, get samples $(\mathbf{v}_n^1, \mathbf{h}_n^1)$, and hope that

$$\mathbb{E}\left[h_i v_j\right] \approx \frac{1}{N}\sum_{n=1}^{N} h_{ni}^1 v_{nj}^1$$

Because these estimates are often noisy, we use the following smoothed "reconstructions" in their place in gradient calculations

$$\begin{aligned}
\hat{\mathbf{v}}_n^1 &= \mathbb{E}\left[\mathbf{v}\middle|\mathbf{h}=\mathbf{h}_n^0\right] = \sigma(W^T\mathbf{h}_n^0 + c) \\
\hat{\mathbf{h}}_n^1 &= \sigma(W\mathbb{E}\left[\mathbf{v}\middle|\mathbf{h}_n^0\right] + b) = \sigma(W\hat{\mathbf{v}}_n^1 + b)
\end{aligned}$$

In brief we compute the contrastive divergence gradients on data point $\mathbf{v}_n$ as follows:

$$
\begin{aligned}
\mathbf{h}_n^0 &\sim \mathrm{P}(\mathbf{h}|\mathbf{v} = \mathbf{v}_n) \\
\hat{\mathbf{v}}_n^1 &= \sigma(W^T \mathbf{h}_n^0 + c) \\
\hat{\mathbf{h}}_n^1 &= \sigma(W \hat{\mathbf{v}}_n^1 + b) \\
\nabla_W^{CD} \log \mathrm{P}(\mathbf{v}_n) &= \sigma(W \mathbf{v}_n + b)\mathbf{v}_n^T - \hat{\mathbf{h}}_n^1 \hat{\mathbf{v}}_n^{1T} \\
\nabla_b^{CD} \log \mathrm{P}(\mathbf{v}_n) &= \sigma(W \mathbf{v}_n + b) - \hat{\mathbf{h}}_n^1 \\
\nabla_c^{CD} \log \mathrm{P}(\mathbf{v}_n) &= \mathbf{v}_n - \hat{\mathbf{v}}_n^1
\end{aligned}
$$

To get the gradient on a batch, just average these individual gradients.