rbm_models

February 15, 2022

1 Modelling Restricted Boltzmann Machines

1.1 Boltzmann Machines

A Boltzmann Machine has an input layer with vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ and an output layer with vector $\mathbf{y} = (y_1, y_2, \ldots, y_M)$. The relationships between input elements x_i and output elements y_j take the form of an undirected graph (see below).

Taking inspiration from random Markov fields and the Boltzmann distribution, we specify an arbitrary energy function

$$E(\mathbf{x}, \mathbf{y}) \quad = \quad -\left[f(\mathbf{x}) + g(\mathbf{y}) + h(\mathbf{x}, \mathbf{y})\right] , \tag{1}$$

such that the joint probability (or density) of $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$p(\mathbf{x}, \mathbf{y}) \quad = \quad \frac{e^{-E(\mathbf{x},\mathbf{y})}}{Z_{X,Y}} = \frac{e^{f(\mathbf{x})+g(\mathbf{y})+h(\mathbf{x},\mathbf{y})}}{Z_{X,Y}} , \tag{2}$$

where $Z_{X,Y}$ is the appropriate partition function obtained by summing (or integrating) the numerator over all possible values of $\mathbf{x}$ and $\mathbf{y}$.

It follows that the conditional distributions are given by

$$p(\mathbf{x} \mid \mathbf{y}) \quad = \quad \frac{e^{f(\mathbf{x})+h(\mathbf{x},\mathbf{y})}}{Z_X(\mathbf{y})} , \tag{3}$$

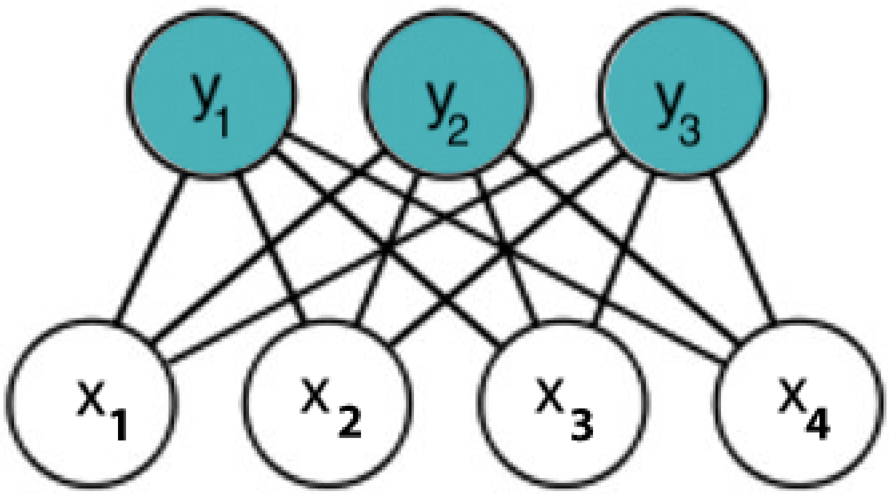
and

$$p(\mathbf{y} \mid \mathbf{x}) \quad = \quad \frac{e^{g(\mathbf{y})+h(\mathbf{x},\mathbf{y})}}{Z_Y(\mathbf{x})} , \tag{4}$$

with respective partition functions.

1.2 Restricted Boltzmann Machines

A *Restricted* Boltzmann Machine (RBM) further restricts the relationships between $\mathbf{x}$ and $\mathbf{y}$ to take the form of an undirected *bipartite* graph, such that the elements of $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ form disconnected nodes in one partition, the elements of $\mathbf{y} = (y_1, y_2, \ldots, y_M)$ form disconnected nodes in the other partition, and edges exist only between nodes in different partitions, i.e. x_i —y_j (see below).



Consequently, by design, the elements x_i of $\mathbf{x}$ are conditionally independent given $\mathbf{y}$, viz.

$$p(\mathbf{x} \mid \mathbf{y}) \quad = \quad \prod_{i=1}^{N} p(x_i \mid \mathbf{y})\,, \tag{5}$$

and the elements y_j of $\mathbf{y}$ are likewise conditionally independent given $\mathbf{x}$, viz.

$$p(\mathbf{y} \mid \mathbf{x}) \quad = \quad \prod_{j=1}^{M} p(y_i \mid \mathbf{x})\,. \tag{6}$$

Therefore, the functions $f(\cdot)$, $g(\cdot)$ and $h(\cdot,\cdot)$ must be linearly separable in their arguments, namely

$$f(\mathbf{x}) = \sum_{i=1}^{N} f_i(x_i)\,, \quad g(\mathbf{y}) = \textstyle\sum_{j=1}^{M} g_j(y_j)\,, \quad h(\mathbf{x},\mathbf{y}) = \displaystyle\sum_{i=1}^{N}\sum_{j=1}^{M} h_{ij}(x_i, y_j)\,, \tag{7}$$

leading to

$$p(x_i \mid \mathbf{y}) = \frac{e^{f_i(x_i)+\sum_{j=1}^{M} h_{ij}(x_i,y_j)}}{Z_{X_i}(\mathbf{y})}\,, \qquad p(y_j \mid \mathbf{x}) = \frac{e^{g_j(y_j)+\sum_{i=1}^{N} h_{ij}(x_i,y_j)}}{Z_{Y_j}(\mathbf{x})}\,, \tag{8}$$

and thus

$$p(\mathbf{x},\mathbf{y}) \quad = \quad \frac{e^{\sum_{i=1}^{N} f_i(x_i)+\sum_{j=1}^{M} g_j(y_j)+\sum_{i=1}^{N}\sum_{j=1}^{M} h_{ij}(x_i,y_j)}}{Z_{X,Y}}\,. \tag{9}$$

1.3 Hidden outputs

Traditionally, the RBM output is considered to be a hidden or latent layer, for which the values of **y** are never observed in practice, and hence must be summed (or integrated) out.

1.3.1 Bernoulli outputs

For tractability, **y** is usually taken to be a vector of binary values, i.e. $\mathbf{y} \in \{0,1\}^M$. Consequently, we obtain

$$\begin{align}
p(y_j = 1 \mid \mathbf{x}) &= \frac{e^{g_j(1)+\sum_{i=1}^N h_{ij}(x_i,1)}}{e^{g_j(0)+\sum_{i=1}^N h_{ij}(x_i,0)} + e^{g_j(1)+\sum_{i=1}^N h_{ij}(x_i,1)}} \tag{10} \\
&= \frac{e^{g_j(1)-g_j(0)+\sum_{i=1}^N [h_{ij}(x_i,1)-h_{ij}(x_i,0)]}}{1 + e^{g_j(1)-g_j(0)+\sum_{i=1}^N [h_{ij}(x_i,1)-h_{ij}(x_i,0)]}} \tag{11} \\
&= \frac{1}{1 + e^{-\left[g_j(1)-g_j(0)+\sum_{i=1}^N [h_{ij}(x_i,1)-h_{ij}(x_i,0)]\right]}} \tag{12} \\
&= \sigma\left(b_j + \sum_{i=1}^N w_{ij}(x_i)\right) \,, \tag{13}
\end{align}$$

where $b_j \doteq g_j(1) - g_j(0)$, $w_{ij}(x_i) \doteq h_{ij}(x_i,1) - h_{ij}(x_i,0)$ and $\sigma(\cdot)$ is the logistic function.

For convenience, note that we may invert these relations and define $g_j(y_j) \doteq b_j y_j$ and $h_{ij}(x_i, y_j) \doteq w_{ij}(x_i)\, y_j$, without loss of generality. The converse conditional distribution then becomes

$$p(x_i \mid \mathbf{y}) \quad = \quad \frac{e^{f_i(x_i)+\sum_{j=1}^M w_{ij}(x_i)\, y_j}}{Z_{X_i}(\mathbf{y})} \,, \tag{14}$$

from above.

The marginal distribution of **x** can also be derived. Recall that

$$p(\mathbf{x}, \mathbf{y}) \quad \propto \quad e^{\sum_{i=1}^N f_i(x_i)+\sum_{j=1}^M g_j(y_j)+\sum_{i=1}^N \sum_{j=1}^M h_{ij}(x_i,y_j)} \,, \tag{15}$$

so that

$$\begin{align}
p(\mathbf{x}) \quad &\propto \quad \sum_{y_1 \in \{0,1\}} \cdots \sum_{y_M \in \{0,1\}} e^{\sum_{i=1}^N f_i(x_i)+\sum_{j=1}^M b_j y_j+\sum_{i=1}^N \sum_{j=1}^M w_{ij}(x_i)\, y_j} \tag{16} \\
&= \quad e^{\sum_{i=1}^N f_i(x_i)} \sum_{y_1 \in \{0,1\}} e^{b_1 y_1+\sum_{i=1}^N w_{ij}(x_i)\, y_1} \cdots \sum_{y_M \in \{0,1\}} e^{b_M y_M+\sum_{i=1}^N w_{ij}(x_i)\, y_M} \,. \tag{17}
\end{align}$$

Therefore, we obtain

$$p(\mathbf{x}) \quad = \quad \frac{e^{\sum_{i=1}^N f_i(x_i)} \prod_{j=1}^M \left(1 + e^{b_j+\sum_{i=1}^N w_{ij}(x_i)}\right)}{Z_X} \,. \tag{18}$$

1.4 Observed inputs

The input vector **x** forms the observed part of the RBM, and hence requires specialised handling to match the assumed input distribution.

1.4.1 Bernoulli inputs

In some analyses, the input $\mathbf{x}$ is a vector of binary values, i.e. $\mathbf{x} \in \{0,1\}^N$. One example is from the field of natural language processing, where each vocabulary word (or token) is either in or not in a given document. Another example is from the field of image processing, where a black-and-white image has pixels that are either on or off. Thus, we obtain

$$\begin{aligned} p(x_i = 1 \mid \mathbf{y}) &= \frac{e^{f_i(1)+\sum_{j=1}^M h_{ij}(1,y_j)}}{e^{f_i(0)+\sum_{j=1}^M h_{ij}(0,y_j)} + e^{f_i(1)+\sum_{j=1}^M h_{ij}(1,y_j)}} & (19) \\ &= \frac{1}{1+e^{-\left[f_i(1)-f_i(0)+\sum_{j=1}^M [h_{ij}(1,y_j)-h_{ij}(0,y_j)]\right]}} & (20) \\ &= \sigma\left(a_i + \sum_{j=1}^M w'_{ij}(y_j)\right)\,, & (21) \end{aligned}$$

where $a_i \doteq f_i(1) - f_i(0)$ and $w'_{ij}(y_j) \doteq h_{ij}(1,y_j) - h_{ij}(0,y_j)$. Hence, we may define $f_i(x_i) \doteq a_i x_i$ and $h_{ij}(x_i,y_j) \doteq x_i w'_{ij}(y_j)$, without loss of generality.

Upon also assuming Bernoulli outputs, we further find that $h_{ij}(x_i,y_j) \doteq x_i w_{ij} y_j$, which gives rise to the standard, bilinear model

$$p(\mathbf{x},\mathbf{y}) = \frac{e^{\mathbf{a}^T\mathbf{x}+\mathbf{b}^T\mathbf{y}+\mathbf{x}^T W\mathbf{y}}}{Z_{X,Y}}\,, \qquad (22)$$

for coefficient vectors $\mathbf{a} = (a_1,\ldots,a_N)$ and $\mathbf{b} = (b_1,\ldots,b_M)$, and coefficient matrix $W = [w_{ij}]$, where we now interpret all vectors column-wise.

The conditional distributions from above therefore take the forms

$$\begin{aligned} p(x_i = 1 \mid \mathbf{y}) &= \sigma\left([\mathbf{a} + W\mathbf{y}]_i\right)\,, & (23) \\ p(y_j = 1 \mid \mathbf{x}) &= \sigma\left(\left[\mathbf{b} + W^T\mathbf{x}\right]_j\right)\,, & (24) \end{aligned}$$

and the marginal distribution becomes

$$p(\mathbf{x}) = \frac{e^{\mathbf{a}^T\mathbf{x}} \prod_{j=1}^M \left(1 + e^{\left[\mathbf{b}+W^T\mathbf{x}\right]_j}\right)}{Z_X}\,. \qquad (25)$$

Note, however, that even for the simplifications inherent in the Bernoulli RBM, the partition functions Z_X and $Z_{X,Y}$ remain intractable.

1.4.2 Free energy

The `free energy` of $\mathbf{x}$, denoted here as $F(\mathbf{x})$, is obtained by reusing the energy distribution formulation, namely:

$$\begin{aligned} p(\mathbf{x}) &= \frac{e^{-F(\mathbf{x})}}{Z_X} & (26) \\ \Rightarrow F(\mathbf{x}) &= -\ln p(\mathbf{x}) - \ln Z_X & (27) \\ &= -\mathbf{a}^T\mathbf{x} - \sum_{j=1}^M \ln\left(1 + e^{[\mathbf{b}+W^T\mathbf{x}]_j}\right)\,. & (28) \end{aligned}$$

The mean free energy of a dataset X is then defined as

$$\bar{F}(X) \;\doteq\; \frac{1}{D}\sum_{d=1}^{D} F(\mathbf{x}_d)\,. \tag{29}$$

Note that since Z_X is unknown in practice, we cannot compute the mean log-likelihood of X, and so we cannot score an individual dataset. However, for fixed RBM parameters, the difference between the scores of two datasets is equal to the difference between their respective mean free energies. Hence, we could, for example, monitor the difference in scores between the training set and a validation set. When this difference starts to grow persistently larger, it is a sign that the RBM might be overfitting the training data.

1.4.3 Gaussian inputs

In other situations, it is more realistic that the $\mathbf{x}$ values are unrestricted, i.e. $\mathbf{x}\in\mathbb{R}^N$. Typically, we take $\mathbf{x}$ to be Gaussian distributed. Recalling that the x_i values (treated as variables) must be conditionally independent in an RBM, we conclude that

$$\begin{aligned} p(x_i\mid\mathbf{y}) &= [2\pi\sigma_i^2(\mathbf{y})]^{-\frac{1}{2}}\,e^{-\,[x_i-\mu_i(\mathbf{y})]^2/2\sigma_i^2(\mathbf{y})} && (30)\\ &= [2\pi\sigma_i^2(\mathbf{y})]^{-\frac{1}{2}}\,e^{-\left[x_i^2-2x_i\mu_i(\mathbf{y})+\mu_i^2(\mathbf{y})\right]/2\sigma_i^2(\mathbf{y})} && (31)\\ &= e^{\alpha_i(\mathbf{y})+\beta_i(\mathbf{y})\,x_i+\gamma_i(\mathbf{y})\,x_i^2}\,, && (32)\end{aligned}$$

for appropriately defined coefficient functions.

Hence, given Bernoulli outputs, we might modify the standard model to include independent squared terms, namely

$$p(\mathbf{x},\mathbf{y}) \;=\; \frac{e^{\mathbf{a}^T\mathbf{x}+\mathbf{x}^T\Gamma(\mathbf{y})\mathbf{x}+\mathbf{b}^T\mathbf{y}+\mathbf{x}^TW\mathbf{y}}}{Z_{X,Y}}\,, \tag{33}$$

where $\Gamma(\mathbf{y})\doteq\mathtt{diag}(\mathbf{c}+D\mathbf{y})$. Letting $\mathbf{x}^2\doteq(x_1^2,\ldots,x_N^2)$ for convenience, this may be rewritten as

$$p(\mathbf{x},\mathbf{y}) \;=\; \frac{e^{(\mathbf{a}\oplus\mathbf{c})^T(\mathbf{x}\oplus\mathbf{x}^2)+\mathbf{b}^T\mathbf{y}+(\mathbf{x}\oplus\mathbf{x}^2)^T(W\oplus D)\mathbf{y}}}{Z_{X,Y}}\,, \tag{34}$$

where the operator $\oplus$ denotes column-wise concatenation. Hence, with some care required when resampling $p(\mathbf{x}\mid\mathbf{y})$, we may notionally augment our feature vector $\mathbf{x}$ with its squared elements and thus reuse the standard bilinear model. In other words, defining $\tilde{\mathbf{x}}\doteq\mathbf{x}\oplus\mathbf{x}^2$, $\tilde{\mathbf{a}}\doteq\mathbf{a}\oplus\mathbf{c}$ and $\tilde{W}\doteq W\oplus D$, we have

$$p(\mathbf{x},\mathbf{y}) \;=\; \frac{e^{\tilde{\mathbf{a}}^T\tilde{\mathbf{x}}+\mathbf{b}^T\mathbf{y}+\tilde{\mathbf{x}}^T\tilde{W}\mathbf{y}}}{Z_{X,Y}}\,. \tag{35}$$

In particular, we immediately obtain the conditional distribution

$$p(y_j=1\mid\mathbf{x}) \;=\; \sigma\left(\left[\mathbf{b}+\tilde{W}^T\tilde{\mathbf{x}}\right]_j\right)\,, \tag{36}$$

and the marginal distribution

$$p(\mathbf{x}) \;=\; \frac{e^{\tilde{\mathbf{a}}^T\tilde{\mathbf{x}}}\prod_{j=1}^{M}\left(1+e^{\left[\mathbf{b}+\tilde{W}^T\tilde{\mathbf{x}}\right]_j}\right)}{Z_X}\,. \tag{37}$$

For the conditional Gaussian, we obtain

$$
\begin{aligned}
p(\mathbf{x} \mid \mathbf{y}) &\propto e^{\tilde{\mathbf{a}}^T\tilde{\mathbf{x}}+\tilde{\mathbf{x}}^T\tilde{W}\mathbf{y}} && (38) \\
&= e^{(\mathbf{a}\oplus\mathbf{c})^T(\mathbf{x}\oplus\mathbf{x}^2)+(\mathbf{x}\oplus\mathbf{x}^2)^T(W\oplus D)\mathbf{y}} && (39) \\
&= \prod_{i=1}^{N} e^{[\mathbf{a}+W\mathbf{y}]_i x_i+[\mathbf{c}+D\mathbf{y}]_i x_i^2} \,. && (40)
\end{aligned}
$$

Equating coefficients with the standard Gaussian form above then gives the individual variances and means as

$$
\sigma_i^2(\mathbf{y}) = -\frac{1}{2[\mathbf{c}+D\mathbf{y}]_i}\,, \qquad \mu_i(\mathbf{y}) = -\frac{[\mathbf{a}+W\mathbf{y}]_i}{2[\mathbf{c}+D\mathbf{y}]_i}\,. \tag{41}
$$

Note that the contstraint $\sigma_i^2(\mathbf{y}) > 0$ requires enforcement, such that $\mathbf{c} + D\mathbf{y} < \mathbf{0}$ $\forall \mathbf{y} \in \{0,1\}^M$. Thus, it is necessary that $\mathbf{c} < \mathbf{0}$, and sufficient that $D < 0$.

1.5 Gradient Approximations

Reconsider the Boltzmann Machine

$$
p(\mathbf{x},\mathbf{y}) \;=\; \frac{e^{-E(\mathbf{x},\mathbf{y})}}{Z_{X,Y}}\,, \tag{42}
$$

where the energy function $E(\mathbf{x},\mathbf{y})$ implicitly has model parameters Θ. Assuming that the output $\mathbf{y}$ is always hidden, then we wish to estimate Θ by maximising the marginal distribution $p(\mathbf{x})$ over all cases of training data. Typically, this is achieved by some gradient ascent procedure.

However, in general the partition function $Z_{X,Y}$ is intractable to compute, and thus the gradient is also intractable. The solution is to approximate the gradient. There are various approaches, including Gibbs sampling and mean field approximations.

For convenience, let us temporarily assume that the input $\mathbf{x}$ and output $\mathbf{y}$ are both discrete valued. However, the derivation below is also valid for continuous variables with summations replaced by integrations.

Thus, let the joint distribution be

$$
p(\mathbf{x},\mathbf{y}) \;=\; \frac{e^{-E(\mathbf{x},\mathbf{y})}}{\sum_{\mathbf{x}'}\sum_{\mathbf{y}'} e^{-E(\mathbf{x}',\mathbf{y}')}}\,, \tag{43}
$$

such that the conditional distribution is

$$
p(\mathbf{y} \mid \mathbf{x}) \;=\; \frac{e^{-E(\mathbf{x},\mathbf{y})}}{\sum_{\mathbf{y}'} e^{-E(\mathbf{x},\mathbf{y}')}}\,, \tag{44}
$$

and the marginal distribution is

$$
\begin{aligned}
p(\mathbf{x}) &= \frac{\sum_{\mathbf{y}} e^{-E(\mathbf{x},\mathbf{y})}}{\sum_{\mathbf{x}'}\sum_{\mathbf{y}'} e^{-E(\mathbf{x}',\mathbf{y}')}} && (45) \\
\Rightarrow \ln p(\mathbf{x}) &= \ln\sum_{\mathbf{y}} e^{-E(\mathbf{x},\mathbf{y})} - \ln\sum_{\mathbf{x}'}\sum_{\mathbf{y}'} e^{-E(\mathbf{x}',\mathbf{y}')}\,. && (46)
\end{aligned}
$$

Then, for each model parameter $\theta \in \Theta$, we have

$$\begin{aligned}
\frac{\partial}{\partial\theta}\ln p(\mathbf{x}) &= -\frac{\sum_{\mathbf{y}} e^{-E(\mathbf{x},\mathbf{y})}\frac{\partial E}{\partial\theta}(\mathbf{x},\mathbf{y})}{\sum_{\mathbf{y}} e^{-E(\mathbf{x},\mathbf{y})}} + \frac{\sum_{\mathbf{x}'}\sum_{\mathbf{y}'} e^{-E(\mathbf{x}',\mathbf{y}')}\frac{\partial E}{\partial\theta}(\mathbf{x}',\mathbf{y}')}{\sum_{\mathbf{x}'}\sum_{\mathbf{y}'} e^{-E(\mathbf{x}',\mathbf{y}')}} && (47)\\
&= -\sum_{\mathbf{y}} p(\mathbf{y}\mid\mathbf{x})\frac{\partial E}{\partial\theta}(\mathbf{x},\mathbf{y}) + \sum_{\mathbf{x}'}\sum_{\mathbf{y}'} p(\mathbf{x}',\mathbf{y}')\frac{\partial E}{\partial\theta}(\mathbf{x}',\mathbf{y}') && (48)\\
&= -\mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\frac{\partial E}{\partial\theta}(\mathbf{x},\mathbf{y})\right] + \mathbb{E}_{\mathbf{x}',\mathbf{y}'}\left[\frac{\partial E}{\partial\theta}(\mathbf{x}',\mathbf{y}')\right] && (49)\\
&= -\mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\frac{\partial E}{\partial\theta}(\mathbf{x},\mathbf{y})\right] + \mathbb{E}_{\mathbf{x}'}\left[\mathbb{E}_{\mathbf{y}'|\mathbf{x}'}\left[\frac{\partial E}{\partial\theta}(\mathbf{x}',\mathbf{y}')\right]\right] && (50)\\
&= +\mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\frac{\partial(-E)}{\partial\theta}(\mathbf{x},\mathbf{y})\right] - \mathbb{E}_{\mathbf{x}'}\left[\mathbb{E}_{\mathbf{y}'|\mathbf{x}'}\left[\frac{\partial(-E)}{\partial\theta}(\mathbf{x}',\mathbf{y}')\right]\right]\,. && (51)
\end{aligned}$$

We assume that $p(\mathbf{y}\mid\mathbf{x})$ is tractable to compute, and thus the conditional expectations are also tractable. However, we have noted above that $p(\mathbf{x})$ and $p(\mathbf{x},\mathbf{y})$ generally are not tractable, so we still have to resort to approximations.

1.5.1 Gibbs sampling

We note that, under suitable conditions, expectations obey

$$\mathbb{E}_X\left[f(X)\right] = \lim_{K\rightarrow\infty}\frac{1}{K}\sum_{k=1}^{K} f(\mathbf{x}_k)\,, \qquad (52)$$

where $\mathbf{x}_k \sim p(X)$. In particular, for the single sample $\mathbf{x}'$, $f(\mathbf{x}')$ is an unbiased estimator of $\mathbb{E}_X\left[f(X)\right]$. Thus, the above gradient could be approximated by the stochastic gradient

$$\frac{\partial}{\partial\theta}\ln p(\mathbf{x}) \approx \mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\frac{\partial(-E)}{\partial\theta}(\mathbf{x},\mathbf{y})\right] - \mathbb{E}_{\mathbf{y}'|\mathbf{x}'}\left[\frac{\partial(-E)}{\partial\theta}(\mathbf{x}',\mathbf{y}')\right]\,. \qquad (53)$$

How are we supposed to sample $\mathbf{x}'$ if computing $p(\mathbf{x}')$ is intractable? This is where the Gibbs sampling comes in. Since we are assuming that the conditional distributions are tractable, then we approximate unconditional distributions by conditional ones. Thus, we let

$$p(\mathbf{x}') = \sum_{\mathbf{y}} p(\mathbf{x}'\mid\mathbf{y})\,p(\mathbf{y}) \quad\Rightarrow\quad \mathbb{E}_{\mathbf{x}'}[\cdot] = \mathbb{E}_{\mathbf{y}}\left[\mathbb{E}_{\mathbf{x}'|\mathbf{y}}[\cdot]\right]\,, \qquad (54)$$

using the other conditional distribution

$$p(\mathbf{x}\mid\mathbf{y}) = \frac{e^{-E(\mathbf{x},\mathbf{y})}}{\sum_{\mathbf{x}'} e^{-E(\mathbf{x}',\mathbf{y})}}\,, \qquad (55)$$

which is also assumed to be tractable to compute.

Now we are faced with the fact that $p(\mathbf{y})$ is also intractable. Hence, we repeat the above step, letting

$$p(\mathbf{y}) = \sum_{\mathbf{x}} p(\mathbf{y}\mid\mathbf{x})\,p(\mathbf{x}) \quad\Rightarrow\quad \mathbb{E}_{\mathbf{y}}[\cdot] = \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathbf{y}|\mathbf{x}}[\cdot]\right]\,. \qquad (56)$$

We could repeat this cycle any number of times, corresponding to multiple steps of sequential Gibbs sampling. However, we are already given $\mathbf{x}$, so we halt with the approximation that

$$\mathbb{E}_{\mathbf{y}}[\cdot] \quad \approx \quad \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\cdot]\,, \tag{57}$$

on the basis that $f(\mathbf{x})$ is an unbiased estimate of $\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})]$, using the same argument as above.

This results in the final approximation

$$\frac{\partial}{\partial\theta}\ln p(\mathbf{x}) \quad \approx \quad \mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\frac{\partial(-E)}{\partial\theta}(\mathbf{x},\mathbf{y})\right] - \mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\mathbb{E}_{\mathbf{x}'|\mathbf{y}}\left[\mathbb{E}_{\mathbf{y}'|\mathbf{x}'}\left[\frac{\partial(-E)}{\partial\theta}\left(\mathbf{x}',\mathbf{y}'\right)\right]\right]\right]\,. \tag{58}$$

In practice, Gibbs sampling of $\mathbf{y}$ and $\mathbf{x}'$ replaces the outer two expectations of the negative term. Hence, the Gibbs sampling algorithm is: 1. For visible input $\mathbf{x}$, compute the distribution $p(\mathbf{y} \mid \mathbf{x})$ of the hidden output, and compute the expectation term $\mathbb{E}_{\mathbf{y}|\mathbf{x}}[\cdot]$ in the gradient. 2. Sample the hidden output $\mathbf{y}$ from the distribution $p(\mathbf{y} \mid \mathbf{x})$. The pair $\mathbf{x},\mathbf{y}$ form the so-called *positive* case. 3. Using $\mathbf{y}$, compute the distribution $p(\mathbf{x}' \mid \mathbf{y})$, and sample $\mathbf{x}'$. 4. Using $\mathbf{x}'$, compute the distribution $p(\mathbf{y}' \mid \mathbf{x}')$, and compute the expectation term $\mathbb{E}_{\mathbf{y}'|\mathbf{x}'}[\cdot]$. The pair $\mathbf{x}',\mathbf{y}'$ form the *negative* case. 5. Compute the approximate stochastic gradient as the difference of expectations.

1.5.2 Mean field approximation

Note that in the special case where $E(\mathbf{x},\mathbf{y})$ is linear in $\mathbf{y}$ for parameter θ, e.g.

$$E(\mathbf{x},\mathbf{y}) \quad = \quad \mathbf{w}(\mathbf{x},\theta)^T\mathbf{y} + \cdots\,, \tag{59}$$

then we have

$$\mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\frac{\partial E}{\partial\theta}(\mathbf{x},\mathbf{y})\right] \quad = \quad \frac{\partial E}{\partial\theta}\left(\mathbf{x},\mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\mathbf{y}\right]\right)\,, \tag{60}$$

exactly. If, however, there are nonlinearities in $\mathbf{y}$ then the above does not hold exactly, but it does still approximately hold true. This is the `mean field` approximation.

To see how the mean field approximation works, we define

$$\bar{\mathbf{y}}_{\mathbf{x}} \quad = \quad \bar{\mathbf{y}}(\mathbf{x}) \doteq \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\mathbf{y}]\,, \tag{61}$$

and consider the first-order Taylor approximation

$$\begin{aligned} E(\mathbf{x},\mathbf{y}) \quad &\approx \quad E(\mathbf{x},\bar{\mathbf{y}}_{\mathbf{x}}) + (\mathbf{y}-\bar{\mathbf{y}}_{\mathbf{x}})^T\frac{\partial E}{\partial\mathbf{y}}(\mathbf{x},\bar{\mathbf{y}}_{\mathbf{x}}) & (62)\\ \Rightarrow \mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[E(\mathbf{x},\mathbf{y})\right] \quad &\approx \quad E(\mathbf{x},\bar{\mathbf{y}}_{\mathbf{x}}) + \left(\mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\mathbf{y}\right]-\bar{\mathbf{y}}_{\mathbf{x}}\right)^T\frac{\partial E}{\partial\mathbf{y}}(\mathbf{x},\bar{\mathbf{y}}_{\mathbf{x}}) & (63)\\ &= \quad E\left(\mathbf{x},\mathbb{E}_{\mathbf{y}|\mathbf{x}}[\mathbf{y}]\right)\,. & (64)\end{aligned}$$

Taking derivatives, we see that

$$\mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\frac{\partial E}{\partial\theta}(\mathbf{x},\mathbf{y})\right] \quad \approx \quad \frac{\partial E}{\partial\theta}\left(\mathbf{x},\mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\mathbf{y}\right]\right) \tag{65}$$

also holds true.

If we now apply the mean field approximation to the Gibbs sampling approximation of the gradient (above), then we obtain

$$\begin{aligned}
\frac{\partial}{\partial\theta}\ln p(\mathbf{x}) &\approx \mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\frac{\partial(-E)}{\partial\theta}(\mathbf{x},\mathbf{y})\right] - \mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\mathbb{E}_{\mathbf{x}'|\mathbf{y}}\left[\mathbb{E}_{\mathbf{y}'|\mathbf{x}'}\left[\frac{\partial(-E)}{\partial\theta}\left(\mathbf{x}',\mathbf{y}'\right)\right]\right]\right] && (66)\\
&\approx \frac{\partial(-E)}{\partial\theta}\left(\mathbf{x},\bar{\mathbf{y}}_{\mathbf{x}}\right) - \mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\mathbb{E}_{\mathbf{x}'|\mathbf{y}}\left[\frac{\partial(-E)}{\partial\theta}\left(\mathbf{x}',\bar{\mathbf{y}}_{\mathbf{x}'}\right)\right]\right] && (67)\\
&\approx \frac{\partial(-E)}{\partial\theta}\left(\mathbf{x},\bar{\mathbf{y}}_{\mathbf{x}}\right) - \mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[\frac{\partial(-E)}{\partial\theta}\left(\bar{\mathbf{x}}_{\mathbf{y}},\bar{\mathbf{y}}(\bar{\mathbf{x}}_{\mathbf{y}})\right)\right] && (68)\\
&\approx \frac{\partial(-E)}{\partial\theta}\left(\mathbf{x},\bar{\mathbf{y}}_{\mathbf{x}}\right) - \frac{\partial(-E)}{\partial\theta}\left(\bar{\mathbf{x}}\left(\bar{\mathbf{y}}_{\mathbf{x}}\right),\bar{\mathbf{y}}\left(\bar{\mathbf{x}}\left(\bar{\mathbf{y}}_{\mathbf{x}}\right)\right)\right)\,. && (69)
\end{aligned}$$

1.5.3 Bernoulli RBM gradient

For the Bernoulli RBM, we have (from above) that

$$-E(\mathbf{x},\mathbf{y}) = \mathbf{a}^T\mathbf{x} + \mathbf{b}^T\mathbf{y} + \mathbf{x}^T W\mathbf{y}\,, \tag{70}$$

and thus

$$\frac{\partial(-E)}{\partial\mathbf{a}} = \mathbf{x}\,,\quad \tfrac{\partial(-E)}{\partial\mathbf{b}} = \mathbf{y}\,,\quad \frac{\partial(-E)}{\partial W} = \mathbf{x}\,\mathbf{y}^T\,. \tag{71}$$

Furthermore, we find that

$$\begin{aligned}
\bar{\mathbf{y}}(\mathbf{x}) &= [p(y_j = 1 \mid \mathbf{x})]_{j=1}^M\,, && (72)\\
\bar{\mathbf{x}}(\mathbf{y}) &= [p(x_i = 1 \mid \mathbf{y})]_{i=1}^N\,. && (73)
\end{aligned}$$

Gibbs sampling Under the Gibbs sampling approximation described above, we first sample $\mathbf{y} \sim \bar{\mathbf{y}}(\mathbf{x})$ and let $\mathbf{y}$ stand in for $\mathbb{E}_{\mathbf{y}|\mathbf{x}}[\cdot]$. Next, we sample $\mathbf{x}' \sim \bar{\mathbf{x}}(\mathbf{y})$, and let $\mathbf{x}'$ stand in for $\mathbb{E}_{\mathbf{x}'|\mathbf{y}}[\cdot]$. This leaves only $\mathbb{E}_{\mathbf{y}|\mathbf{x}}[\cdot]$ and $\mathbb{E}_{\mathbf{y}'|\mathbf{x}'}[\cdot]$ to be evaluated. Hence,

$$\begin{aligned}
\frac{\partial}{\partial\mathbf{a}}\ln p(\mathbf{x}) &\approx \mathbf{x} - \mathbf{x}'\,, && (74)\\
\frac{\partial}{\partial\mathbf{b}}\ln p(\mathbf{x}) &\approx \bar{\mathbf{y}}_{\mathbf{x}} - \bar{\mathbf{y}}_{\mathbf{x}'}\,, && (75)\\
\frac{\partial}{\partial\mathbf{W}}\ln p(\mathbf{x}) &\approx \mathbf{x}\,\bar{\mathbf{y}}_{\mathbf{x}}^T - \mathbf{x}'\bar{\mathbf{y}}_{\mathbf{x}'}^T\,. && (76)
\end{aligned}$$

In practice, this approximation does not work well, with the parameters seemingly wandering about randomly.

Hinton modified gradient Hinton offers a practical guide to training RBMs, although I found the commentary to be rather terse. Briefly, Hinton asserts that the positive (or data) term in the expectations above should couple the binary input with binary (sampled) output. However, the negative (or reconstruction) term can seemingly forego sampling altogether. My interpretation is that this leads to the modified gradient scheme:

$$\begin{aligned}
\frac{\partial}{\partial\mathbf{a}}\ln p(\mathbf{x}) &\approx \mathbf{x} - \bar{\mathbf{x}}_{\mathbf{y}}\,, && (77)\\
\frac{\partial}{\partial\mathbf{b}}\ln p(\mathbf{x}) &\approx \mathbf{y} - \bar{\mathbf{y}}(\bar{\mathbf{x}}_{\mathbf{y}})\,, && (78)\\
\frac{\partial}{\partial\mathbf{W}}\ln p(\mathbf{x}) &\approx \mathbf{x}\,\mathbf{y}^T - \bar{\mathbf{x}}_{\mathbf{y}}\,\bar{\mathbf{y}}(\bar{\mathbf{x}}_{\mathbf{y}})^T\,, && (79)
\end{aligned}$$

where we sample $\mathbf{y} \sim \bar{\mathbf{y}}(\mathbf{x})$ as before.

This scheme seems to work well, after a burn-in training period of random fluctuations. It is interesting to note that in practice this Hinton-modified gradient appears to minmise the RMS error discussed in a later section. This would appear to be due to the fact that both the modified gradient above and the reconstruction probabilities (later) utilise some aspects of mean field approximations (see the next section).

Mean field approximation The mean field approximation, described in detail above, is straightforward to apply. The resulting gradient approximation is

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{a}} \ln p(\mathbf{x}) &\approx \mathbf{x} - \bar{\mathbf{x}}\left(\bar{\mathbf{y}}_{\mathbf{x}}\right) , && (80) \\
\frac{\partial}{\partial \mathbf{b}} \ln p(\mathbf{x}) &\approx \bar{\mathbf{y}}_{\mathbf{x}} - \bar{\mathbf{y}}\left(\bar{\mathbf{x}}\left(\bar{\mathbf{y}}_{\mathbf{x}}\right)\right) , && (81) \\
\frac{\partial}{\partial \mathbf{W}} \ln p(\mathbf{x}) &\approx \mathbf{x}\,\bar{\mathbf{y}}_{\mathbf{x}}^T - \bar{\mathbf{x}}\left(\bar{\mathbf{y}}_{\mathbf{x}}\right)\,\bar{\mathbf{y}}\left(\bar{\mathbf{x}}\left(\bar{\mathbf{y}}_{\mathbf{x}}\right)\right)^T . && (82)
\end{aligned}$$

Clearly, this is closely related to the Hinton-modified gradient above, except that now no Gibbs sampling is required. In practice, this gradient approximation appears to work very well, and seemingly has better convergence than the Hinton-modified gradient (although YMMV).

1.5.4 Gaussian RBM gradient

For the Gaussian RBM (i.e. Gaussian input with Bernoulii output), the negative energy function is

$$-E(\mathbf{x},\mathbf{y}) = \mathbf{a}^T\mathbf{x} + \mathbf{c}^T\mathbf{x}^2 + \mathbf{b}^T\mathbf{y} + \mathbf{x}^T W\mathbf{y} + (\mathbf{x}^2)^T D\mathbf{y} \,, \qquad (83)$$

and thus we obtain the above derivatives in $\mathbf{a}$, $\mathbf{b}$ and W, as well as

$$\frac{\partial(-E)}{\partial \mathbf{c}} = \mathbf{x}^2 \,, \qquad \frac{\partial(-E)}{\partial D} = \mathbf{x}^2\,\mathbf{y}^T . \qquad (84)$$

Note that to preserve the constraints $c_i < 0$ and $d_{ij} < 0$, we might choose $c_i \doteq -e^{c'_i}$ and $d_{ij} \doteq -e^{d'_{ij}}$, such that

$$\frac{\partial(-E)}{\partial c'_i} = \frac{\partial(-E)}{\partial c_i}\frac{\partial c_i}{\partial c'_i} = x_i^2\,c_i \,, \qquad \frac{\partial(-E)}{\partial d'_{ij}} = \frac{\partial(-E)}{\partial d_{ij}}\frac{\partial d_{ij}}{\partial d'_{ij}} = x_i^2\,y_j\,d_{ij} . \qquad (85)$$

Alternatively, we might simply rectify the updated estimates of $\mathbf{c}$ and D to thus obey the constraints.

1.6 Non-standard Estimation

RBMs can be rather difficult to train, since the usual parameter update scheme described above does not really maximise any particular likelihood (Hinton). Additionally, since computing $p(\mathbf{x})$ is intractable, we cannot properly score the updates to test for convergence.

In practice, we need to use some sort of approximation to $p(\mathbf{x})$. One approach is offered by the mean field approximation (described in an earlier section). We note that

$$p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x} \mid \mathbf{y})\,p(\mathbf{y}) = \mathbb{E}_{\mathbf{y}}\left[p(\mathbf{x} \mid \mathbf{y})\right] , \qquad (86)$$

and hence, following previous reasoning, we have

$$p(\mathbf{x}) \approx \mathbb{E}_{\mathbf{y}|\mathbf{x}}\left[p(\mathbf{x} \mid \mathbf{y})\right] \approx p\left(\mathbf{x} \mid \bar{\mathbf{y}}_{\mathbf{x}}\right) . \tag{87}$$

Next, we recall that RBMs obey the conditional independence property

$$p(\mathbf{x} \mid \mathbf{y}) = \prod_{i=1}^{N} p(x_i \mid \mathbf{y}) . \tag{88}$$

Thus, for a Bernoulli RBM we have

$$\begin{aligned} p(\mathbf{x} \mid \mathbf{y}) &= \prod_{i=1}^{N} p\left(x_i = 1 \mid \mathbf{y}\right)^{x_i} \left[1 - p\left(x_i = 1 \mid \mathbf{y}\right)\right]^{1-x_i} && (89) \\ &= \prod_{i=1}^{N} \bar{x}_i\left(\mathbf{y}\right)^{x_i} \left[1 - \bar{x}_i\left(\mathbf{y}\right)\right]^{1-x_i} , && (90) \end{aligned}$$

where

$$\bar{x}_i(\mathbf{y}) \doteq p(x_i = 1 \mid \mathbf{y}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}}[x_i] = \left[\bar{\mathbf{x}}(\mathbf{y})\right]_i . \tag{91}$$

The mean field approximate probability, i.e. the so-called *reconstruction* probability, is therefore

$$p(\mathbf{x}) \approx \prod_{i=1}^{N} \bar{x}_i\left(\bar{\mathbf{y}}_{\mathbf{x}}\right)^{x_i} \left[1 - \bar{x}_i\left(\bar{\mathbf{y}}_{\mathbf{x}}\right)\right]^{1-x_i} . \tag{92}$$

1.6.1 Minimising the reconstruction error

As another approach, suppose we approximate $p(\mathbf{x})$, and score how closely this approximation is to the binary input data. Specifically, we minimise the mean square error

$$R^2 \doteq \frac{1}{D}\sum_{d=1}^{D}\sum_{i=1}^{N}(x_{di} - p_{di})^2 , \tag{93}$$

where

$$\begin{aligned} p_{di} &\doteq \bar{x}_i(\mathbf{q}_d) = \sigma\left(a_i + \sum_{j=1}^{M} w_{ij} q_{dj}\right) , && (94) \\ q_{dj} &\doteq \bar{y}_j(\mathbf{x}_d) = \sigma\left(b_j + \sum_{i=1}^{N} x_{di} w_{ij}\right) . && (95) \end{aligned}$$

We note that, for arbitrary parameter θ, the gradient is

$$\frac{\partial R^2}{\partial\theta} = -\frac{2}{D}\sum_{d=1}^{D}\sum_{i=1}^{N}\delta_i(\theta)\left(x_{di} - p_{di}\right)\frac{\partial p_{di}}{\partial\theta} , \tag{96}$$

where $\delta_i(\theta)$ is a notional 0/1 indicator that causes the summation over i to be dropped if parameter θ is indexed by i. Furthermore, we see that

$$\frac{\partial p_{di}}{\partial\theta} = p_{di}\,(1-p_{di})\left\{\frac{\partial a_i}{\partial\theta}+\sum_{j=1}^{M}\delta_j(\theta)\,\frac{\partial w_{ij}}{\partial\theta}q_{dj}+\sum_{j=1}^{M}\delta_j(\theta)\,w_{ij}\frac{\partial q_{dj}}{\partial\theta}\right\}, \tag{97}$$

since $\sigma'(z)=\sigma(z)\,[1-\sigma(z)]$. Similarly, we have

$$\frac{\partial q_{dj}}{\partial\theta} = q_{dj}\,(1-q_{dj})\left\{\frac{\partial b_j}{\partial\theta}+\sum_{i=1}^{N}\delta_i(\theta)\,x_{di}\frac{\partial w_{ij}}{\partial\theta}\right\}. \tag{98}$$

Consequently, we derive that

$$\begin{aligned}\frac{\partial q_{dj}}{\partial a_i} &= 0\,,\ \frac{\partial p_{di}}{\partial a_i}=p_{di}\,(1-p_{di}) &(99)\\ \Rightarrow\frac{\partial R^2}{\partial\,\mathbf{a}} &= -2\,\mathtt{mean}(A)\,,\ A\doteq(X-P)\otimes P\otimes(1-P)\,, &(100)\end{aligned}$$

for element-wise multiplicative operator $\otimes$, where the function $\mathtt{mean}(\cdot)$ averages over the data.

Similarly, we find that

$$\begin{aligned}\frac{\partial q_{dj}}{\partial b_j} &= q_{dj}\,(1-q_{dj})\,,\ \frac{\partial p_{di}}{\partial b_j}=p_{di}\,(1-p_{di})\,w_{ij}\frac{\partial q_{dj}}{\partial b_j} &(101)\\ \Rightarrow\frac{\partial R^2}{\partial\,\mathbf{b}} &= -2\,\mathtt{mean}([AW]\otimes B)\,,\ B\doteq Q\otimes(1-Q)\,. &(102)\end{aligned}$$

Lastly, we obtain

$$\begin{aligned}\frac{\partial q_{dj}}{\partial w_{ij}} &= x_{di}\,q_{dj}\,(1-q_{dj})\,,\ \frac{\partial p_{di}}{\partial w_{ij}}=p_{di}\,(1-p_{di})\left\{q_{dj}+w_{ij}\frac{\partial q_{dj}}{\partial w_{ij}}\right\} &(103)\\ \Rightarrow\frac{\partial R^2}{\partial W} &= -\frac{2}{D}\left\{A^TQ+W\otimes\left([A\otimes X]^TB\right)\right\}. &(104)\end{aligned}$$

Note that we need to update the parameter estimates in the opposite direction of these gradients in order to minimise the reconstruction error.

1.6.2 Maximising the approximate likelihood

Following similar reasoning to that above, we could instead maximise the approximate likelihood defined earlier. Reusing the definition of the reconstruction probability, we recall that

$$p(\mathbf{x}_d) \approx \prod_{i=1}^{N}p_{di}^{x_{di}}(1-p_{di})^{1-x_{di}}\,. \tag{105}$$

This leads to the average log-likelihood

$$\begin{aligned}
L &= \frac{1}{D}\ln\prod_{d=1}^{D} p(\mathbf{x}_d) && (106)\\
&\approx \frac{1}{D}\sum_{d=1}^{D}\sum_{i=1}^{N}\left[x_{di}\ln p_{di} + (1-x_{di})\ln(1-p_{di})\right] && (107)\\
\Rightarrow \frac{\partial L}{\partial\theta} &\approx \frac{1}{D}\sum_{d=1}^{D}\sum_{i=1}^{N}\delta_i(\theta)\left[\frac{x_{di}}{p_{di}} - \frac{1-x_{di}}{1-p_{di}}\right]\frac{\partial p_{di}}{\partial\theta}\,. && (108)
\end{aligned}$$

Thus, we obtain

$$\begin{aligned}
\frac{\partial L}{\partial a_i} &\approx \frac{1}{D}\sum_{d=1}^{D}\left[x_{di}(1-p_{di}) - (1-x_{di})p_{di}\right] = \frac{1}{D}\sum_{d=1}^{D}(x_{di}-p_{di}) && (109)\\
\Rightarrow \frac{\partial L}{\partial\,\mathbf{a}} &\approx \mathtt{mean}(X-P)\,. && (110)
\end{aligned}$$

Similarly, we have

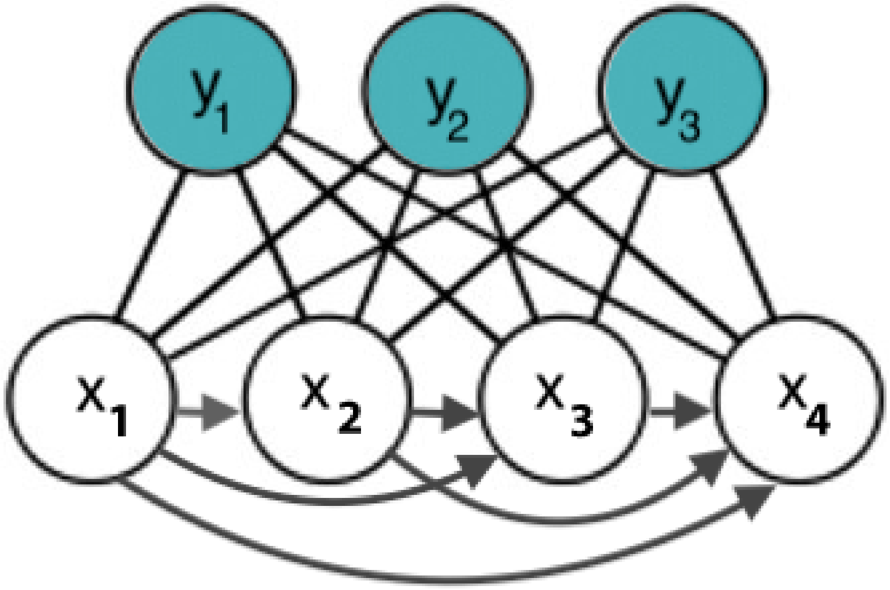
$$\begin{aligned}
\frac{\partial L}{\partial b_j} &\approx \frac{1}{D}\sum_{d=1}^{D}\sum_{i=1}^{N}(x_{di}-p_{di})\,w_{ij}q_{dj}\,(1-q_{dj}) && (111)\\
\Rightarrow \frac{\partial L}{\partial\,\mathbf{b}} &\approx \mathtt{mean}([(X-P)W]\otimes B)\,. && (112)
\end{aligned}$$

Lastly, we obtain

$$\begin{aligned}
\frac{\partial L}{\partial w_{ij}} &\approx \frac{1}{D}\sum_{d=1}^{D}(x_{di}-p_{di})\left[q_{dj} + w_{ij}x_{di}q_{dj}\,(1-q_{dj})\right] && (113)\\
\Rightarrow \frac{\partial L}{\partial W} &\approx \frac{1}{D}\left\{(X-P)^T Q + W\otimes\left([(X-P)\otimes X]^T B\right)\right\}\,. && (114)
\end{aligned}$$

1.7 Sequential Restricted Boltzmann Machines

(Larochelle and Murray) extend the RBM to the case where the input vector $\mathbf{x}$ has internal Bayesian Network dependencies. They call this the Neural Autoregressive Distribution Estimator (NADE). In particular, it is assumed that the elements $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ have been canonically ordered in some fashion, for example either randomly or by reason of causal effects. Thus, the input dependencies form an ordered Markov network (see the figure below).



Borrowing Bayesian Network methodology, and temporarily ignoring the hidden output, we now suppose that the visible input has the joint distribution:

$$
\begin{aligned}
p(\mathbf{x}) &= p(x_1, x_2, \ldots, x_N) && (115)\\
&\doteq p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1, x_2) \cdots p(x_N \mid x_1, x_2, \ldots, x_{N-1}) && (116)\\
&= \prod_{i=1}^{N} p(x_i \mid \mathbf{x}_{1:i-1})\,, && (117)
\end{aligned}
$$

where we define $\mathbf{x}_{j:k} \doteq (x_j, x_{j+1}, \ldots, x_k)$ for $j \leq k \in \{1, 2, \ldots, N\}$. For convenience, whenever $j > k$ we let $\mathbf{x}_{j:k} = (\,)$.

Next, we reintroduce the binary hidden output $\mathbf{y} \in \{0, 1\}^M$, such that

$$
p(x_i \mid \mathbf{x}_{1:i-1}) \;=\; \sum_{\mathbf{y} \in \{0,1\}^M} p(x_i, \mathbf{y} \mid \mathbf{x}_{1:i-1})\,, \tag{118}
$$

where (from the diagram)

$$
p(x_i, \mathbf{y} \mid \mathbf{x}_{1:i-1}) \;=\; p(x_i \mid \mathbf{y}, \mathbf{x}_{1:i-1})\, p(\mathbf{y} \mid \mathbf{x}_{1:i-1})\,. \tag{119}
$$

In order to keep the networked model reasonably simple, we now ignore the input dependencies and define

$$
p(x_i \mid \mathbf{y}, \mathbf{x}_{1:i-1}) \;\doteq\; p(x_i \mid \mathbf{y})\,. \tag{120}
$$

Effectively, we have returned to the standard RBM formulation, but have implicitly retained the dependencies amongst the input units by utilising their effects on the output units.

Thus, for the Bernoulli RBM, we have

$$\bar{x}_i(\mathbf{y}) \;=\; p(x_i = 1 \mid \mathbf{y}) = \sigma\left([\mathbf{a} + W\mathbf{y}]_i\right) \,, \tag{121}$$

as before, but now we use the truncated models

$$\bar{y}_j(\mathbf{x}_{1:k}) \;=\; p(y_j = 1 \mid \mathbf{x}_{1:k}) \doteq \sigma\left(\left[\mathbf{b} + W_{1:k,:}^T \mathbf{x}_{1:k}\right]_j\right) \,, \tag{122}$$

for $k = 1, 2, \ldots, N$, where $W_{1:k,:}$ is the matrix obtained by retaining the first k rows of W. For convenience, with $i = 1$ and $k = i - 1$, we take $\bar{y}_j(\mathbf{x}_{1:0}) = \sigma(b_j)$.

Reusing the derivations above, we similarly assume that

$$p(\mathbf{y} \mid \mathbf{x}_{1:i-1}) \;=\; \prod_{j=1}^{M} \bar{y}_j(\mathbf{x}_{1:i-1})^{y_j} \left[1 - \bar{y}_j(\mathbf{x}_{1:i-1})\right]^{1-y_j} \,, \tag{123}$$

such that

$$p(x_i = 1 \mid \mathbf{x}_{1:i-1}) \;=\; \sum_{\mathbf{y} \in \{0,1\}^M} \bar{x}_i(\mathbf{y})\, p(\mathbf{y} \mid \mathbf{x}_{1:i-1}) = \mathbb{E}_{\mathbf{y}|\mathbf{x}_{1:i-1}}\left[\bar{x}_i(\mathbf{y})\right] \,. \tag{124}$$

However, we note that this summation remains intractable.

In order to obtain a tractable model, (Larochelle and Murray) used a mean field approximation to $p(x_i \mid \mathbf{x}_{1:i-1})$, namely

$$p(x_i = 1 \mid \mathbf{x}_{1:i-1}) \;\approx\; \bar{x}_i(\bar{\mathbf{y}}(\mathbf{x}_{1:i-1}))\,. \tag{125}$$

Hence, the joint probability of input $\mathbf{x}$ is

$$p(\mathbf{x}) \;\approx\; \prod_{i=1}^{N} \bar{x}_i\left(\bar{\mathbf{y}}\left(\mathbf{x}_{1:i-1}\right)\right)^{x_i} \left[1 - \bar{x}_i\left(\bar{\mathbf{y}}\left(\mathbf{x}_{1:i-1}\right)\right)\right]^{1-x_i} \,. \tag{126}$$

This is the essence of the NADE model.

Observe that this is just a modified form of the approximate reconstruction probability derived in an earlier section. It follows that most of the maths we previously derived for the gradient of the log-likelihood still holds. Thus, we compute

$$\bar{y}_j^{(i)}(\mathbf{x}) \;\doteq\; \bar{y}_j(\mathbf{x}_{1:i-1}) = \sigma\left(b_j + \sum_{k=1}^{i-1} x_k w_{kj}\right) \,, \tag{127}$$

$$\bar{x}_i\left(\bar{\mathbf{y}}_{\mathbf{x}}^{(i)}\right) \;=\; \sigma\left(a_i + \sum_{j=1}^{M} w_{ij} \bar{y}_j^{(i)}(\mathbf{x})\right) \,, \tag{128}$$

such that

$$\ln p(\mathbf{x}) \;\approx\; \sum_{i=1}^{N} \left\{ x_i \ln \bar{x}_i\left(\bar{\mathbf{y}}_{\mathbf{x}}^{(i)}\right) + (1 - x_i) \ln\left[1 - \bar{x}_i\left(\bar{\mathbf{y}}_{\mathbf{x}}^{(i)}\right)\right] \right\} \,. \tag{129}$$

Consequently, we obtain the approximate gradients

$$\frac{\partial}{\partial a_i} \ln p(\mathbf{x}) \quad \approx \quad x_i - \bar{x}_i\left(\bar{\mathbf{y}}_{\mathbf{x}}^{(i)}\right) \; , \tag{130}$$

$$\frac{\partial}{\partial b_j} \ln p(\mathbf{x}) \quad \approx \quad \sum_{i=1}^{N} B_{ij} \; , \tag{131}$$

$$\frac{\partial}{\partial w_{ij}} \ln p(\mathbf{x}) \quad \approx \quad \left[x_i - \bar{x}_i\left(\bar{\mathbf{y}}_{\mathbf{x}}^{(i)}\right)\right] \, \bar{y}_j^{(i)}(\mathbf{x}) + x_i \sum_{k=i+1}^{N} B_{kj} \; , \tag{132}$$

where

$$B_{ij} \quad = \quad \left[x_i - \bar{x}_i\left(\bar{\mathbf{y}}_{\mathbf{x}}^{(i)}\right)\right] \, w_{ij} \, \bar{y}_j^{(i)}(\mathbf{x}) \left[1 - \bar{y}_j^{(i)}(\mathbf{x})\right] \; . \tag{133}$$

However, we might recall the various gradient schemes that we have derived so far, namely the Hinton-modified gradient and the mean field approximation, as well the explicit gradients of the reconstruction error and the log reconstruction probability. In application, all of these gradient schemes act to minimise the reconstruction error and maximise the reconstruction probability.

Consequently, it seems reasonable to suppose that we might equally modify one of these existing gradient schemes to allow for dependencies between the input units. Note, however, that whereas previously we computed the component $\bar{x}_i(\bar{\mathbf{y}}_{\mathbf{x}})$ from the vector $\bar{\mathbf{x}}(\bar{\mathbf{y}}_{\mathbf{x}})$, we now need to reverse this procedure and instead compute the vector by stacking the components incrementally.

$$\tag{134}$$