Franklin Hu, Sunil Pedapudi
CS 194-10 Machine Learning
Fall 2011
Assignment 4

1. Linear neural networks

   (a) Suppose we have a three layer neural network with one input layer x, one hidden layer h, and one output layer y. Each layer can be expressed as a vector of the values of the nodes in that layer. For example,

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Assume that each neural node has its own set of weights $\mathbf{w_i}$ where i is the node index. We can express the value of a particular output in terms of the hidden layer

$$y_k = g(\mathbf{h})$$

Since we are only considering linear activation functions, we can write this equation in terms of a constant multiplied by the weighted sum of inputs

$$y_k = c_{y_k} \cdot \mathbf{w_{y_k}} \cdot \mathbf{h}$$

where c_{y_k} is the constant multiplier of y_k, w_{y_k} is the set of weights for y_k, and h is the vector of hidden nodes.
Similarly, we can express the value of each node in the hidden layer in terms of the inputs.

$$h_j = c_{h_j} \cdot \mathbf{w_{h_j}} \cdot \mathbf{x}$$

Now, we can see that the output layer nodes can simply be written in terms of the inputs without the hidden layer. For a particular output node:

$$\begin{aligned} y_k &= c_{y_k} \cdot \mathbf{w_{y_k}} \cdot \mathbf{h} \\ &= c_{y_k} \cdot \mathbf{w_{y_k}} \cdot \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{pmatrix} \\ &= c_{y_k} \cdot \mathbf{w_{y_k}} \cdot \begin{pmatrix} c_{h1} \cdot \mathbf{w_{h_1}} \cdot \mathbf{x} \\ c_{h2} \cdot \mathbf{w_{h_2}} \cdot \mathbf{x} \\ \vdots \\ c_{hn} \cdot \mathbf{w_{h_n}} \cdot \mathbf{x} \end{pmatrix} \\ &= c_{y_k} \cdot (\mathbf{w_{y_k}} \cdot \mathbf{c_h} \cdot \mathbf{I} \cdot \mathbf{w_h} \cdot \mathbf{x} \end{aligned}$$

where c_h is a vector of the constant weight for each hidden node, I is the identity matrix, and w_h is a matrix of the weight vectors of the hidden nodes. Thus we can define a new weight vector $\mathbf{u_{y_k}}$ for the output node y_k

$$\mathbf{u_{y_k}} = \mathbf{w_{y_k}} \cdot \mathbf{c_h} \cdot \mathbf{I} \cdot \mathbf{w_h}$$

We can thus simply compute the value of y_k in terms of x.

(b) For an arbitrary number of hidden nodes, the same computation can be done. We demonstrate below with two hidden layers: h_m, h_n

(c) For the case when $h \ll n$, a neural net with the hidden layer will do $O(hn)$ computations to find the linear combination of the weighted sum of inputs whereas without the hidden layer, as shown in (a), the output is only dependent on x. This computations is $O(n)$, so we save those $h-1$ other computations over the inputs.

2. ML estimation of exponential model
Knowing

$$P(x) = \frac{1}{b}e^{-\frac{x}{b}}$$

(a) We write the likelihood function given x_i as

$$\begin{aligned}
\mathcal{L}(b) &= \prod_{i=1}^{N} \frac{1}{b}e^{-\frac{x_i}{b}} \\
&= \left(\frac{1}{b}\right)^N \prod_{i=1}^{N} e^{-\frac{x_i}{b}} \\
&= \left(\frac{1}{b}\right)^N e^{\sum_{i=0}^{N} -\frac{x_i}{b}} \\
&= \left(\frac{1}{b}\right)^N exp(-\frac{1}{b}\sum_{i=0}^{N} x_i) \\
\text{Let } \bar{x} &= \frac{1}{N}\sum_{i=1}^{N} x_i, \\
\mathcal{L}(b) &= \left(\frac{1}{b}\right)^N exp(-\frac{1}{b}N\bar{x})
\end{aligned}$$

(b) We first find

$$\begin{aligned}
log(\mathcal{L}) &= log\left(\left(\frac{1}{b}\right)^N exp(-\frac{1}{b}N\bar{x})\right) \\
&= log\left(\frac{1}{b}\right)^N + log\left(e^{-\frac{1}{b}N\bar{x}}\right) \\
&= N(log(1) - log(b)) - \frac{1}{b}N\bar{x} \\
&= -Nlog(b) - \frac{1}{b}N\bar{x}
\end{aligned}$$

Then, let $\theta = \frac{1}{b}$, the parameter variable

$$\begin{aligned}
\frac{\partial log(\mathcal{L}(\theta))}{\partial \theta} &= \frac{\partial N log(\theta)}{\partial \theta} - \frac{\partial \theta N\bar{x}}{\partial \theta} \\
&= \frac{N}{\theta} - \frac{\partial \theta N\bar{x}}{\partial \theta} \\
&= \frac{N}{\theta} - N\bar{x} \\
&= Nb - N\bar{x}
\end{aligned}$$

(c) We aim to maximize $\mathcal{L}$ so,

$$\frac{\partial \mathcal{L}}{\partial \theta} = Nb - N\bar{x} = 0$$

We can solve this to find

$$b = \bar{x} = \frac{1}{N}\sum_{i=0}^{N} x_i$$

3. ML estimation of noisy-OR model