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

1. Linear neural networks

   (a) Suppose we have a three layer linear neural network with one input layer $\mathbf{x}$, one hidden layer $\mathbf{h}$, and one output layer $\mathbf{y}$. Each layer can be expressed as a vector of the values of the nodes in that layer. For example, $\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{pmatrix}$. 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 the a particular output in terms of the hidden layer:

$$y_k = c_k \cdot \mathbf{w_k} \cdot \mathbf{h}$$

   Similarly, we can express the value of each node in the hidden layer in terms of the inputs.

$$h_j = c_j \cdot \mathbf{w_j} \cdot \mathbf{x}$$

   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:

$$\begin{aligned}
y_k =& c_k \cdot \mathbf{w_k} \cdot \mathbf{h} \\
=& c_k \cdot \mathbf{w_k} \cdot \begin{pmatrix} c_{j_1} \cdot \mathbf{w_{j_1}} \cdot \mathbf{x} \\ c_{j_2} \cdot \mathbf{w_{j_2}} \cdot \mathbf{x} \\ ... \\ c_{j_n} \cdot \mathbf{w_{j_n}} \cdot \mathbf{x} \end{pmatrix} \\
=& c_k \cdot \begin{pmatrix} w_{k_1} & w_{k_2} & ... & w_{k_n} \end{pmatrix} \begin{pmatrix} c_{j_1} \cdot \mathbf{w_{j_1}} \cdot \mathbf{x} \\ c_{j_2} \cdot \mathbf{w_{j_2}} \cdot \mathbf{x} \\ ... \\ c_{j_n} \cdot \mathbf{w_{j_n}} \cdot \mathbf{x} \end{pmatrix} \\
=& c_k \sum_{i=1}^{n} w_{k_i} \cdot c_{j_i} \cdot \mathbf{w_{j_i}} \cdot \mathbf{x} \\
=& c_k \left(\sum_{i=1}^{n} w_{k_i} \cdot c_{j_i} \cdot \mathbf{w_{j_i}} \right) \mathbf{x}
\end{aligned}$$

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

$$
\begin{aligned}
y_k =& c_k \cdot \mathbf{w_k} \cdot \mathbf{h_m} \\
=& c_k \cdot \mathbf{w_k} \cdot \begin{pmatrix} c_{j_1} \cdot \mathbf{w_{j_1}} \cdot \mathbf{h_m} \\ c_{j_2} \cdot \mathbf{w_{j_2}} \cdot \mathbf{h_m} \\ \ldots \\ c_{j_n} \cdot \mathbf{w_{j_n}} \cdot \mathbf{h_m} \end{pmatrix} \\
=& c_k \cdot \begin{pmatrix} w_{k_1} & w_{k_2} & \ldots & w_{k_n} \end{pmatrix} \begin{pmatrix} c_{j_1} \cdot \mathbf{w_{j_1}} \cdot \mathbf{h_m} \\ c_{j_2} \cdot \mathbf{w_{j_2}} \cdot \mathbf{h_m} \\ \ldots \\ c_{j_n} \cdot \mathbf{w_{j_n}} \cdot \mathbf{h_m} \end{pmatrix} \\
=& c_k \sum_{i=1}^{n} w_{k_i} \cdot c_{j_i} \cdot \mathbf{w_{j_i}} \cdot \mathbf{h_m} \\
=& c_k \sum_{i=1}^{n} w_{k_i} \cdot c_{j_i} \cdot \begin{pmatrix} w_{j_1} & w_{j_2} & \ldots & w_{j_n} \end{pmatrix} \begin{pmatrix} c_{m_1} \cdot \mathbf{w_{m_1}} \cdot \mathbf{h_n} \\ c_{m_2} \cdot \mathbf{w_{m_2}} \cdot \mathbf{h_n} \\ \ldots \\ c_{m_n} \cdot \mathbf{w_{m_n}} \cdot \mathbf{h_n} \end{pmatrix} \\
=& c_k \sum_{i=1}^{n} w_{k_i} \cdot c_{j_i} \cdot \left(\sum_{a=1}^{nFIXTHIS!!!} w_{j_a} \cdot c_{m_a} \cdot \mathbf{w_{m_a}} \cdot \mathbf{h_n} \right)
\end{aligned}
$$

This thus generalizes to any number of hidden layers: TODO

(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 where as without the hidden layer, as shown in (a), the output is only dependent on $O(n)$, so we save those $h-1$ other computations over the inputs.

2. ML estimation of exponential model

3. ML estimation of noisy-OR model