Model Representation I

Let's examine how we will represent a hypothesis function using neural networks. At a v
are basically computational units that take inputs (**dendrites**) as electrical inputs (calle
channeled to outputs (**axons**). In our model, our dendrites are like the input features x
the result of our hypothesis function. In this model our x_0 input node is sometimes call
always equal to 1. In neural networks, we use the same logistic function as in classificati
sometimes call it a sigmoid (logistic) **activation** function. In this situation, our "theta" p
called "weights".

Visually, a simplistic representation looks like:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \rightarrow [\quad] \rightarrow h_\theta(x)$$

Our input nodes (layer 1), also known as the "input layer", go into another node (layer 2
hypothesis function, known as the "output layer".

We can have intermediate layers of nodes between the input and output layers called t

In this example, we label these intermediate or "hidden" layer nodes $a_0^2 \cdots a_n^2$ and call t

$$\begin{aligned} &a_i^{(j)} = \text{"activation" of unit } i \text{ in layer } j \\ &\Theta^{(j)} = \text{matrix of weights controlling function mapping from layer } j \text{ to layer } j+1 \end{aligned}$$

If we had one hidden layer, it would look like:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \begin{bmatrix} a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix} \rightarrow h_\theta(x)$$

The values for each of the "activation" nodes is obtained as follows:

$$\begin{aligned} a_1^{(2)} &= g(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3) \\ a_2^{(2)} &= g(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3) \\ a_3^{(2)} &= g(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3) \\ h_\Theta(x) = a_1^{(3)} &= g(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)} + \Theta_{13}^{(2)}a_3^{(2)}) \end{aligned}$$

This is saying that we compute our activation nodes by using a 3×4 matrix of parameter
parameters to our inputs to obtain the value for one activation node. Our hypothesis ou
applied to the sum of the values of our activation nodes, which have been multiplied by
matrix $\Theta^{(2)}$ containing the weights for our second layer of nodes.

Each layer gets its own matrix of weights, $\Theta^{(j)}$.

The dimensions of these matrices of weights is determined as follows:

$$\text{If network has } s_j \text{ units in layer } j \text{ and } s_{j+1} \text{ units in layer } j+1, \text{ then } \Theta^{(j)} \text{ will b}$$

The +1 comes from the addition in $\Theta^{(j)}$ of the "bias nodes," x_0 and $\Theta_0^{(j)}$. In other word
include the bias nodes while the inputs will. The following image summarizes our mode