Model Representation I

Let's examine how we will represent a hypothesis function using neural networks. At a very simple level, neurons are basically computational units that take inputs (**dendrites**) as electrical inputs (called "spikes") that are channeled to outputs (**axons**). In our model, our dendrites are like the input features $\mathbf{x_1} \dots \mathbf{x_n}$, and the output is the result of our hypothesis function. In this model our $\mathbf{x_0}$ input node is sometimes called the "bias unit." It is always equal to 1. In neural networks, we use the same logistic function as in classification, $\frac{1}{1+e^{-\theta^T x}}$, yet we sometimes call it a sigmoid (logistic) **activation** function. In this situation, our "theta" parameters are sometimes called "weights".

Visually, a simplistic representation looks like:

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

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

We can have intermediate layers of nodes between the input and output layers called the "hidden layers."

In this example, we label these intermediate or "hidden" layer nodes $\mathbf{a_0^2} \dots \mathbf{a_n^2}$ and call them "activation units."

$\mathbf{a_i^j}$**="activation" of unit i in layer j**

Θ^j**=matrix of weights controlling function mapping from layer j to layer j+1**

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

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

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

$$a_1^{(2)} = g(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3)$$

$$a_2^{(2)} = g(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3)$$

$$a_3^{(2)} = g(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3)$$

$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)} + \Theta_{13}^{(2)}a_3^{(2)})$$

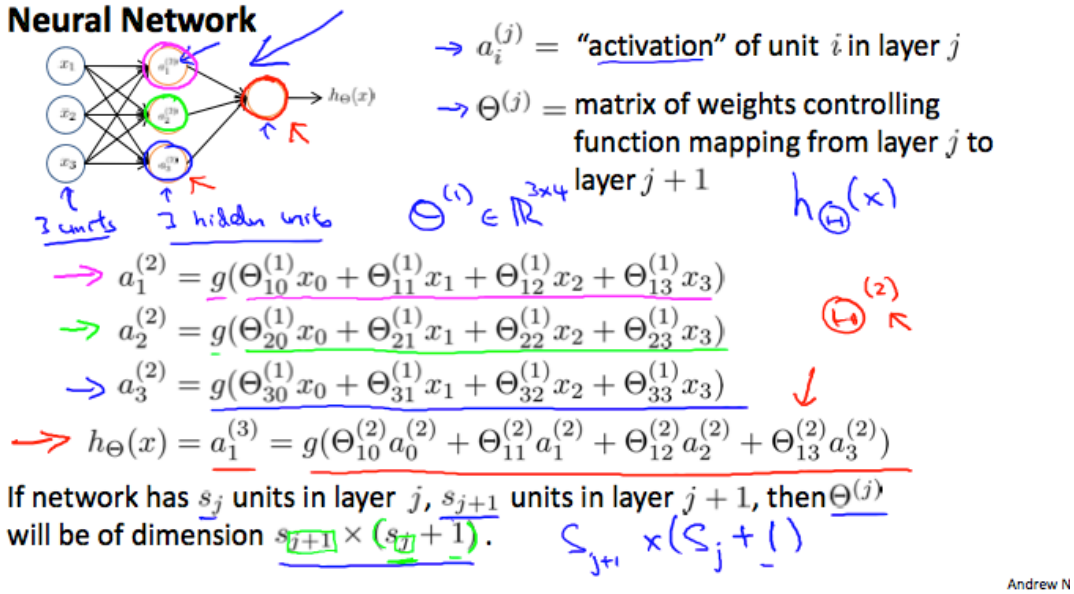
This is saying that we compute our activation nodes by using a 3×4 matrix of parameters. We apply each row of the parameters to our inputs to obtain the value for one activation node. Our hypothesis output is the logistic function applied to the sum of the values of our activation nodes, which have been multiplied by yet another parameter 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:

If network has s_j units in layer j and s_j+1 units in layer j+1, then $\Theta^{(j)}$ will be of dimension $s_j+1 \times (s_j+1)$.

The +1 comes from the addition in $\Theta^{(j)}$ of the "bias nodes," x0 and $\Theta_0^{(j)}$. In other words the output nodes will not include the bias nodes while the inputs will. The following image summarizes our model representation:



Example: If layer 1 has 2 input nodes and layer 2 has 4 activation nodes. Dimension of $\boldsymbol{\Theta^{(1)}}$ is going to be 4×3 where $\boldsymbol{s_j=2}$ and $\boldsymbol{s_j+1=4}$, so $\boldsymbol{s_j+1\times(s_j+1)=4\times3.}$