# Model Representation II

To re-iterate, the following is an example of a neural network:

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

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

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

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

In this section we'll do a vectorized implementation of the above functions. We're going to define a new variable $z_k^{(j)}$ that encompasses the parameters inside our g function. In our previous example if we replaced by the variable z for all the parameters we would get:

$$a_1^{(2)} = g\left(z_1^{(2)}\right)$$

$$a_2^{(2)} = g\left(z_2^{(2)}\right)$$

$$a_3^{(2)} = g\left(z_3^{(2)}\right)$$

In other words, for layer j=2 and node k, the variable z will be:

$$z_k^{(j)} = \Theta_{k,0}^{(1)}x_0 + \Theta_{k,1}^{(1)}x_1 + \cdots + \Theta_{k,n}^{(1)}x_n$$

The vector representation of x and $z^{(j)}$ is:

$$X = \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \qquad z^{(j)} = \begin{bmatrix} z_1^{(j)} \\ z_2^{(j)} \\ \cdots \\ z_n^{(j)} \end{bmatrix}$$

Setting $x = a^{(1)}$, we can rewrite the equation as:

$$z^{(j)} = \Theta^{(j-1)}a^{(j-1)}$$

We are multiplying our matrix $\Theta^{(j-1)}$ with dimensions $s_j \times (n+1)$ (where $s_j$ is the number of our activation nodes) by our vector $a^{(j-1)}$ with height (n+1). This gives us our vector $z^{(j)}$

with height $\mathbf{s}_j$. Now we can get a vector of our activation nodes for layer j as follows:

$\mathbf{a}^{(j)}$=g ($\mathbf{z}^{(j)}$)

Where our function g can be applied element-wise to our vector $\mathbf{z}^{(j)}$.

We can then add a bias unit (equal to 1) to layer j after we have computed$\mathbf{a}^{(j)}$. This will be element $\mathbf{a}_0^{(j)}$ and will be equal to 1. To compute our final hypothesis, let's first compute another z vector:

$\mathbf{z}^{(j+1)}= \mathbf{\Theta}^{(j)}\mathbf{a}^{(j)}$

We get this final z vector by multiplying the next theta matrix after $\mathbf{\Theta}^{(j-1)}$ with the values of all the activation nodes we just got. This last theta matrix $\mathbf{\Theta}^{(j)}$ will have only **one row** which is multiplied by one column $\mathbf{a}^{(j)}$ so that our result is a single number. We then get our final result with:

$\mathbf{h}_{\mathbf{\Theta}}(x)= \mathbf{a}^{(j+1)}$ = g ($\mathbf{z}^{(j+1)}$)

Notice that in this **last step**, between layer j and layer j+1, we are doing **exactly the same thing** as we did in logistic regression. Adding all these intermediate layers in neural networks allows us to more elegantly produce interesting and more complex non-linear hypotheses.