

## Model Representation II

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

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

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:

$$\begin{aligned} a_1^{(2)} &= g(z_1^{(2)}) \\ a_2^{(2)} &= g(z_2^{(2)}) \\ a_3^{(2)} &= g(z_3^{(2)}) \end{aligned}$$

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

$$z_k^{(2)} = \theta_{k,0}^{(1)} x_0 + \theta_{k,1}^{(1)} x_1 + \dots + \theta_{k,n}^{(1)} x_n$$

The vector representation of  $x$  and  $z^j$  is:

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} \quad z^{(j)} = \begin{bmatrix} z_1^{(j)} \\ z_2^{(j)} \\ \dots \\ z_n^{(j)} \end{bmatrix}$$

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  $s_j$ . Now we can get a vector of our activation nodes for layer  $j$  as follows:

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

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

We can then add a bias unit (equal to 1) to layer  $j$  after we have computed  $a^{(j)}$ . This will be element  $a_0^{(j)}$  and will be equal to 1. To compute our final hypothesis, let's first compute another  $z$  vector:

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

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

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

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.