

Neural Networks: Classification

In *binary classification* there is only one K output unit where output $y \in \{0, 1\}$. In *multi-class classification* there are two or more K output units that are K dimensional where $y \in \mathbb{R}^K$. For example if there are two classes A and B then the results of the output units would be:

Two classes where $K = \begin{bmatrix} A \\ B \end{bmatrix}$

Is class A where $K^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Is class B where $K^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Where there are three or more output units for classification then *one-vs-all* will be used.

Cost Function

The cost function for neural networks with regularization is shown below. Instead of having a single output unit we now have K units where $h_{\Theta}(x)_i$ refers to the i^{th} value in the output vector.

$\sum_{k=1}^K$ is summing the normal logistic cost function over each of the K output units and y_k is the i^{th}

output such as $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ (see *Classification*). Including the bias units in the cost is not a big deal but

you generally want to omit them hence below we are not regularizing the bias units so our limits will start at 1.

$$J(\Theta) = -\frac{1}{m} \left[\sum_{i=1}^m \sum_{k=1}^K (y_k^{(i)} \cdot \log(h_{\Theta}(x^{(i)}))_k + (1 - y_k^{(i)}) \cdot \log(1 - h_{\Theta}(x^{(i)}))_k) \right] + \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{s_i} \sum_{j=1}^{s_{l+1}} (\Theta_{ji}^{(l)})^2$$

For the regularization term (also called a *Weight Decay*) it is doing the following:

1. For each layer: $\sum_{l=1}^{L-1}$
2. For each weight in that layer: $\sum_{i=1}^{s_i}$ and $\sum_{j=1}^{s_{l+1}}$ where s_i and $s_i + 1$ are the matrix dimensions at that layer
3. Square the weight at ji : $(\Theta_{ji}^{(l)})^2$