# Machine Learning Week-4

ramay7

February 25, 2017

## Contents



## 1 Neural Network

If network has $s_j$ units in layer $j$, $s_{j+1}$ 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 $\theta^{(j)}$ for the 'bias nodes', $x_0$ and $\theta_0^{(j)}$. The 'bias unit' is always equal to 1. For instance, consider a three layer neural network with two inputs, 4 hidden units, and one output unit, the dimension of $\theta^{(1)}$ is 4 by 3 ($4 \times 3$).

In neural networks, we use the same logistic function as in classification($\frac{1}{1+e^{-\theta^T X}}$), yet we sometimes call it sigmoid(logistic) function. In this situation, our 'theta' parameters are sometimes called 'weights'.

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

Let's set:

$$Z^{(j)} = \theta^{(j-1)} a^{(j-1)}$$

we are multiplying our matrix $\theta^{(j-1)}$ with dimentions $s_j \times (n+1)$ (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:

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

## 2 Equation

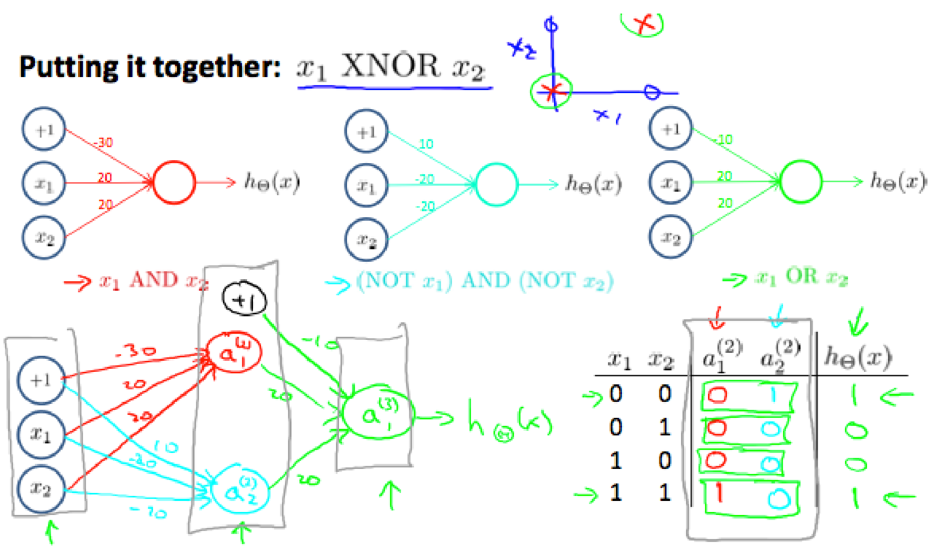
$$\frac{\partial J}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)})$$

$$
\begin{bmatrix}
\frac{\partial J}{\partial \theta_0} \\
\frac{\partial J}{\partial \theta_1} \\
\frac{\partial J}{\partial \theta_2} \\
. \\
. \\
. \\
\frac{\partial J}{\partial \theta_n}
\end{bmatrix}
= \frac{1}{m}
\begin{bmatrix}
\sum_{i=1}^{m}((h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)} \\
\sum_{i=1}^{m}((h_\theta(x^{(i)}) - y^{(i)})x_1^{(i)} \\
\sum_{i=1}^{m}((h_\theta(x^{(i)}) - y^{(i)})x_2^{(i)} \\
. \\
. \\
. \\
\sum_{i=1}^{m}((h_\theta(x^{(i)}) - y^{(i)})x_n^{(i)}
\end{bmatrix}
$$

$$
= \frac{1}{m}\sum_{i=1}^{m}((h_\theta(X^{(i)}) - y^{(i)})X^{(i)}
$$

$$
= \frac{1}{m}X^T(h_\theta(X) - y)
$$

## 3 Simple Neural Network Application: XNOR




## 4 The End

In this week's exercise, I have implemented one-vs-all logistic regression an neural networks to recognize hand-written digits. However, I don't know how to translate a picture into the form of a csv-file. Thanks to the original folder which contains a data set in ex3data1.mat that contains 5000 training examples of handwritten digits. Enjoy myself !