Backpropagation Intuition

Note: [4:39, the last term for the calculation for z_1^3 (three-color handwritten formula) should be a_2^2 ins of a_1^2. 6:08 - the equation for cost(i) is incorrect. The first term is missing parentheses for the log() funct and the second term should be $(1 - y^{(i)})\log(1 - h_\theta(x^{(i)}))$. 8:50 - $\delta^{(4)} = y - a^{(4)}$ is incorrect an should be $\delta^{(4)} = a^{(4)} - y$.]

Recall that the cost function for a neural network is:

$$J(\Theta) = -\frac{1}{m}\sum_{t=1}^{m}\sum_{k=1}^{K} y_k^{(t)} \log(h_\Theta(x^{(t)}))_k + (1 - y_k^{(t)}) \log(1 - h_\Theta(x^{(t)})_k) + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(\Theta_{j,i}^{(l)})^2$$

If we consider simple non-multiclass classification (k = 1) and disregard regularization, the cost is computed with:

$$cost(t) = y^{(t)}\ \log(h_\Theta(x^{(t)})) + (1 - y^{(t)})\ \log(1 - h_\Theta(x^{(t)}))$$

Intuitively, $\delta_j^{(l)}$ is the "error" for $a_j^{(l)}$ (unit j in layer l). More formally, the delta values are actually the derivative of the cost function:

$$\delta_j^{(l)} = \frac{\partial}{\partial z_j^{(l)}} cost(t)$$

Recall that our derivative is the slope of a line tangent to the cost function, so the steeper the slope the more incorrect we are. Let us consider the following neural network below and see how we could calcu some $\delta_j^{(l)}$: