Backpropagation Algorithm

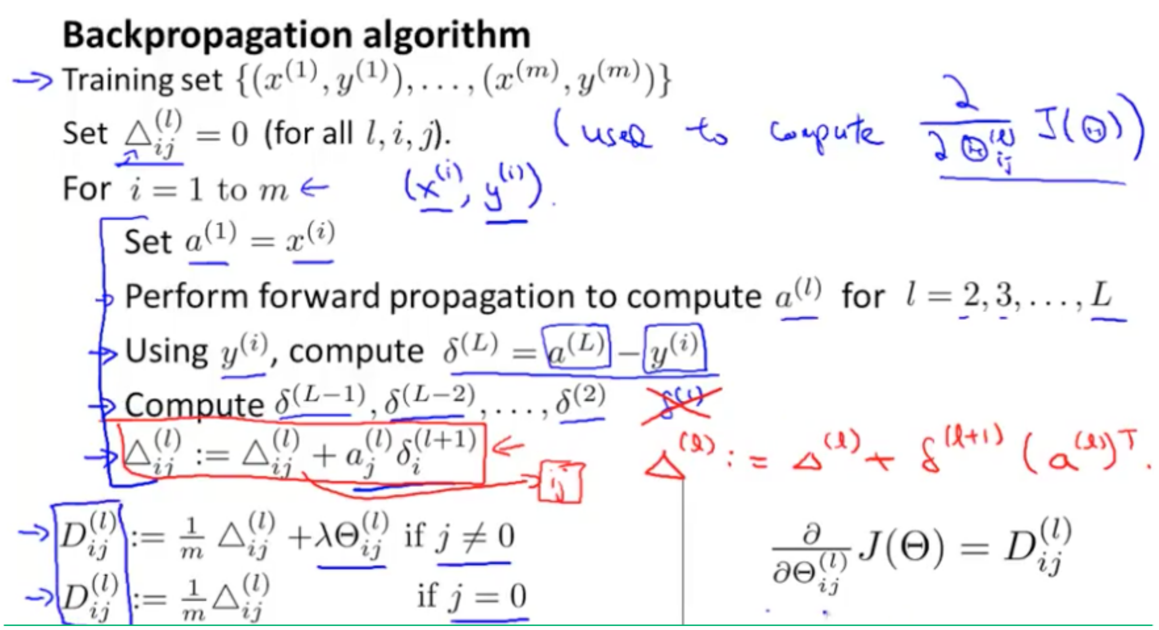
"Backpropagation" is neural-network terminology for minimizing our cost function, just like what we were doing with gradient descent in logistic and linear regression. Our goal is to compute:

$$\min_\Theta J(\Theta)$$

That is, we want to minimize our cost function J using an optimal set of parameters in theta. In this section we'll look at the equations we use to compute the partial derivative of J(Θ):

$$\dfrac{\partial}{\partial \Theta_{i,j}^{(l)}}J(\Theta)$$

To do so, we use the following algorithm:



Back propagation Algorithm

Given training set $\lbrace (x^{(1)}, y^{(1)}) \cdots (x^{(m)}, y^{(m)})\rbrace$

- Set $\Delta^{(l)}_{i,j}$:= 0 for all (l,i,j), (hence you end up having a matrix full of zeros)

For training example t =1 to m: