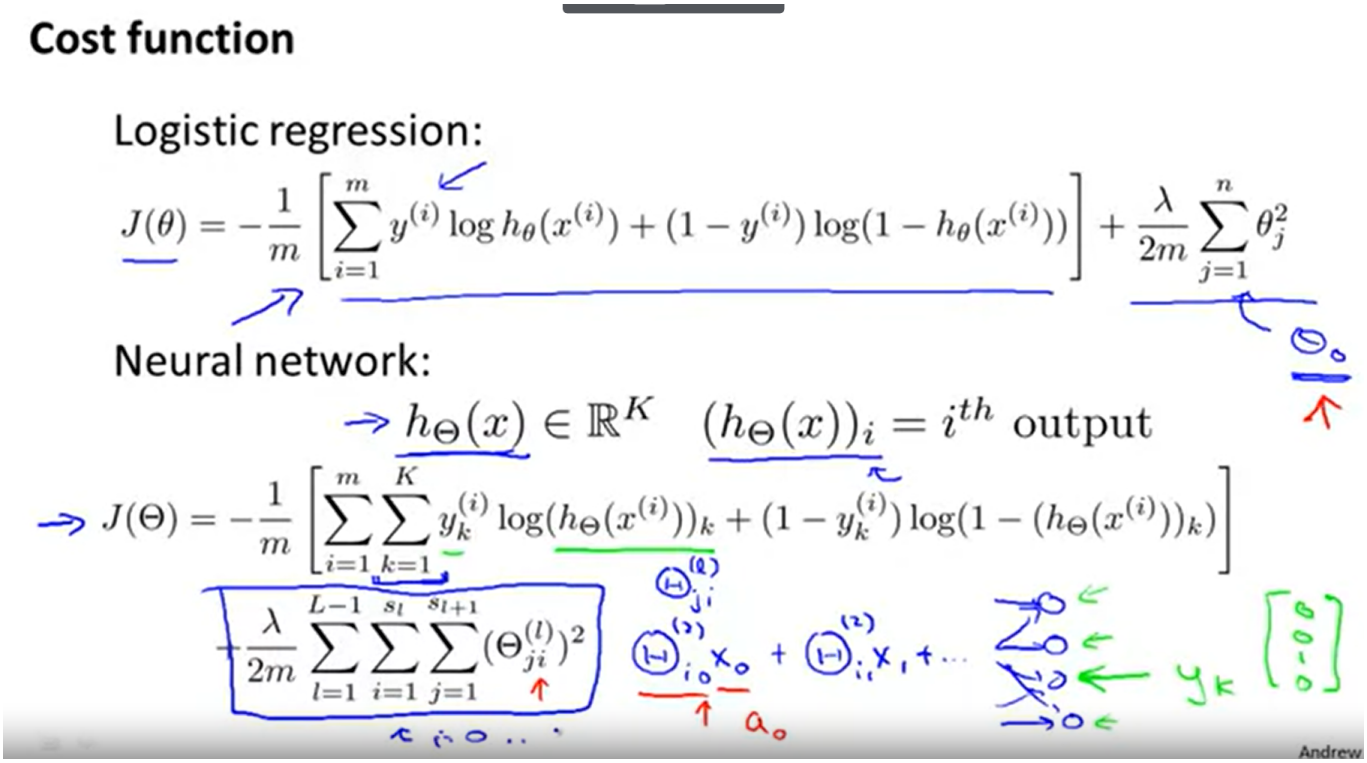
Week 5.1 Neural Networks - Learning

Cost Function of Nerual Networks

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



Backpropagation Algorithm

- $\delta_j^{(l)}$: "error" of node j in layer l
- Initial (for layer $L = 4$):

$$\delta_j^{(4)} = a_j^{(4)} - y_j$$

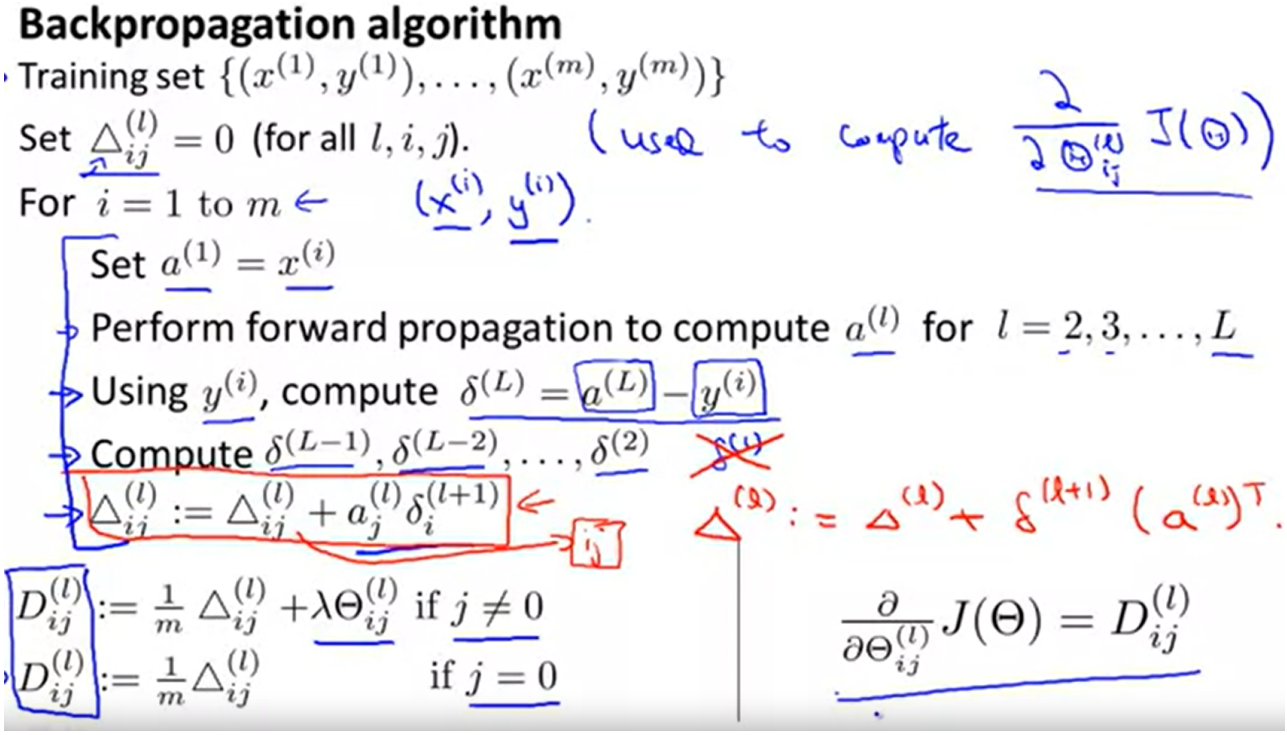
- Propagation:

$$\delta^{(3)} = (\Theta^{(3)})^T \delta^{(4)} .* g'(z^{(3)})$$
$$\delta^{(2)} = (\Theta^{(2)})^T \delta^{(3)} .* g'(z^{(2)})$$
$$g'(z) = a .* (1-a)$$

- if we igore λ

$$\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta) = a^{(l)} \delta^{(l+1)}$$

- **Algorithm:**



Given training set $\{(x^{(1)}, y^{(1)}) \ldots (x^{(m)}, y^{(m)})\}$

- Set $\Delta_{i,j}^{(l)} := 0$ for all (l, i, j)

For training example $t = 1 : m$

1. Set $a^{(1)} := x^{(t)}$
2. Perform forward prpagation to compute a^l for $l = 2, 3, \ldots, L$

$$z^{(l+1)} = \Theta^{(i)} a^{(i)}$$
$$a^{(l+1)} = g(z^{(l+1)})$$

3. Using $y^{(t)}$, compute $\delta^{(L)} = a^{(L)} - y^{(t)}$, L is the total number of layers
4. Compute

$$\delta^{(L-1)}, \delta^{(L-2)}, \ldots, \delta^{(2)}$$

, using

$$\delta^{(l)} = ((\Theta^{(l)})^T \delta^{(l+1)}) .* a^{(l)} .* (1 - a^{(l)})$$

since

$$g'(z^{(l)}) = a^{(l)} .* (1 - a^{(l)})$$

5. $$\Delta_{i,j}^{(l)} := \Delta_{i,j}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$$

or

$$\Delta^{(l)} := \Delta^{(l)} + \delta^{(l+1)}(a^{(l)})^T$$

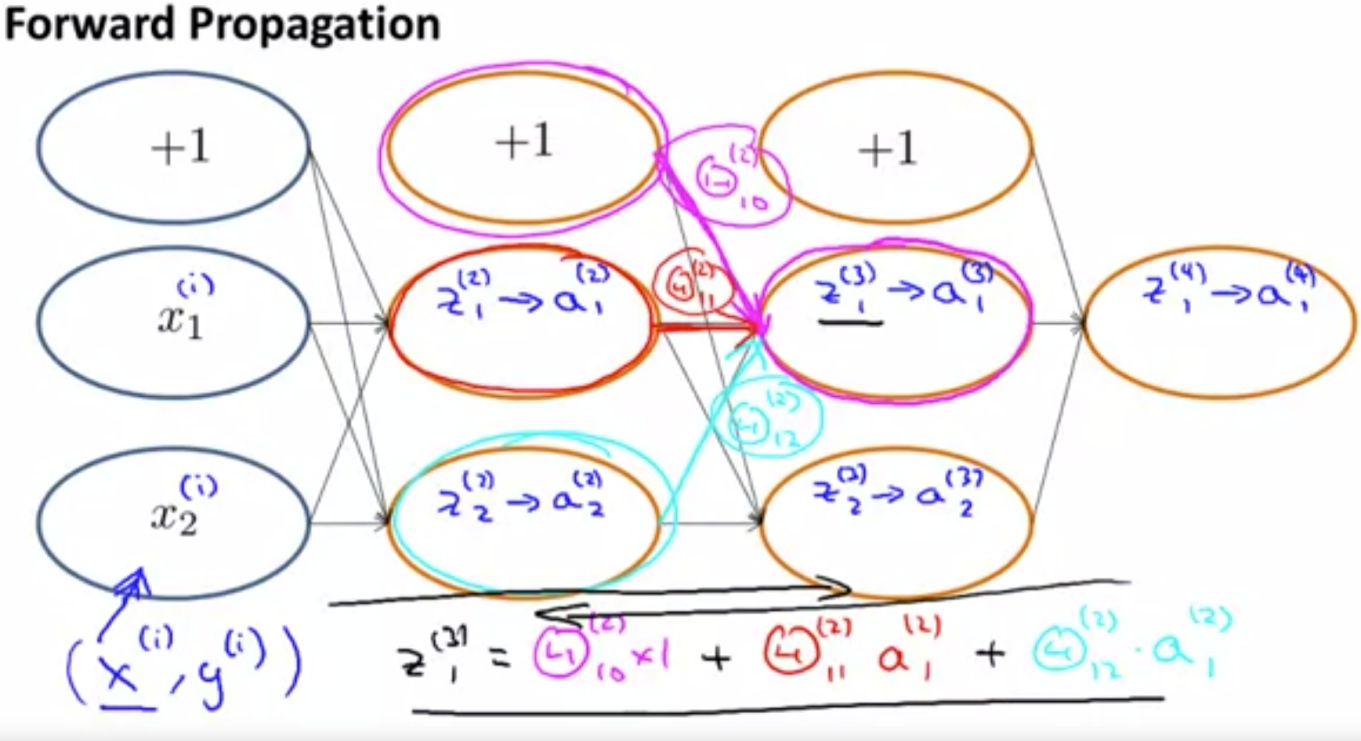
6. Finally, for $j \neq 0$,

$$D_{i,j}^{(l)} := \frac{1}{m}(\Delta_{i,j}^{(l)} + \lambda\Theta_{i,j}^{(l)})$$

for $j = 0$,

$$D_{i,j}^{(l)} := \frac{1}{m}\Delta_{i,j}^{(l)}$$

Intuition



- About Backpropagation,