

Week 5.1 Neural Networks - Learning

Cost Function of Neural Networks

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

Cost function

Logistic regression:

$$\underline{J(\theta)} = -\frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

Neural network:

$$\rightarrow h_\Theta(x) \in \mathbb{R}^K \quad (h_\Theta(x))_i = i^{th} \text{ output}$$
$$\rightarrow J(\Theta) = -\frac{1}{m} \left[\sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log(h_\Theta(x^{(i)}))_k + (1 - y_k^{(i)}) \log(1 - (h_\Theta(x^{(i)}))_k) \right] + \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (\Theta_{ji}^{(l)})^2$$

Andrew

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:

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

- if we ignore λ

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

- **Algorithm:**

Backpropagation algorithm

- Training set $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$

Set $\Delta_{ij}^{(l)} = 0$ (for all l, i, j). (use to compute $\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta)$)

For $i = 1$ to m \leftarrow $(x^{(i)}, y^{(i)})$.

Set $a^{(1)} = x^{(i)}$

Perform forward propagation to compute $a^{(l)}$ for $l = 2, 3, \dots, L$

Using $y^{(i)}$, compute $\delta^{(L)} = a^{(L)} - y^{(i)}$

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

$\Delta_{ij}^{(l)} := \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$

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

$D_{ij}^{(l)} := \frac{1}{m} \Delta_{ij}^{(l)} + \lambda \Theta_{ij}^{(l)}$ if $j \neq 0$
 $D_{ij}^{(l)} := \frac{1}{m} \Delta_{ij}^{(l)}$ if $j = 0$

$\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta) = D_{ij}^{(l)}$

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

- 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 propagation to compute a^l for $l = 2, 3, \dots, L$

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

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

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

4. Compute

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

, 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 $j \neq 0$,

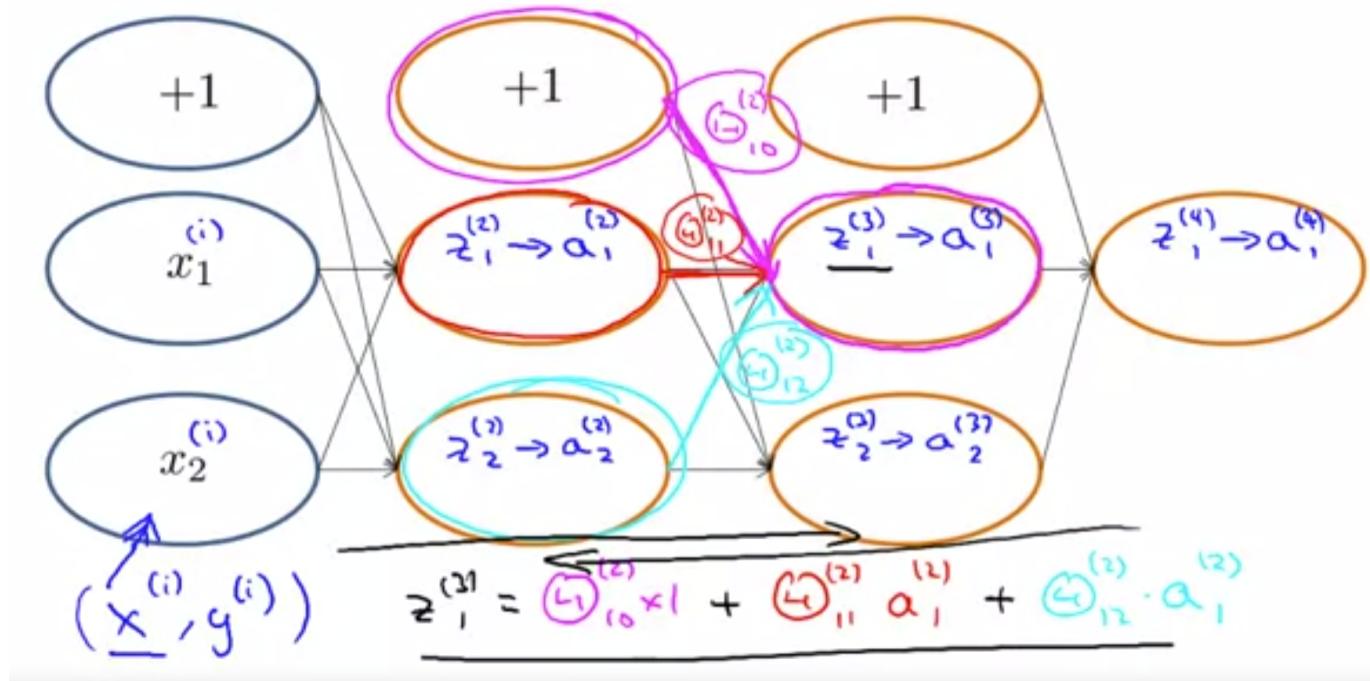
$$D_{i,j}^{(l)} := \frac{1}{m} (\Delta_{i,j}^{(l)} + \lambda \Theta_{i,j}^{(l)})$$

for $j = 0$,

$$D_{i,0}^{(l)} := \frac{1}{m} \Delta_{i,0}^{(l)}$$

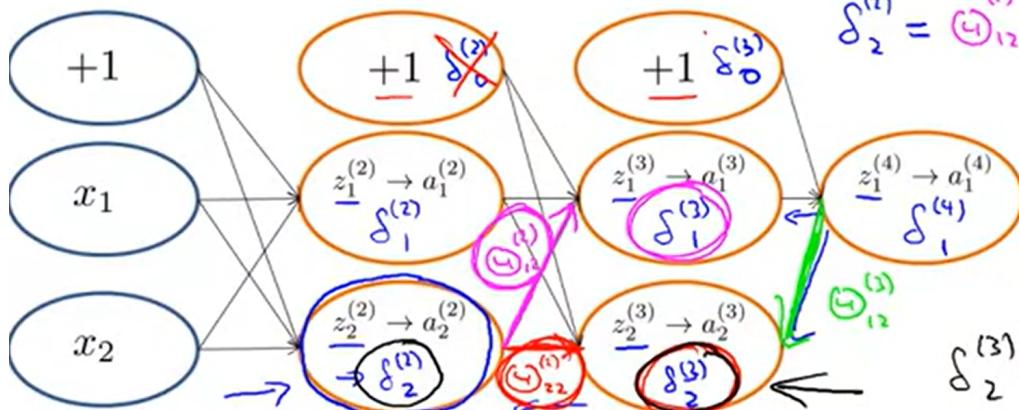
Intuition

Forward Propagation



- About Backpropagation,

Forward Propagation



$\rightarrow \delta_j^{(l)}$ = "error" of cost for $a_j^{(l)}$ (unit j in layer l).

$$\text{Formally, } \delta_j^{(l)} = \frac{\partial \text{cost}(i)}{\partial z_j^{(l)}} \quad (\text{for } j \geq 0), \text{ where}$$

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

$$\delta_1^{(4)} = y^{(1)} - a_1^{(4)}$$

$$\delta_2^{(2)} = \textcolor{blue}{\Theta}_{12}^{(1)} \delta_1^{(3)} + \textcolor{red}{\Theta}_{22}^{(1)} \delta_2^{(3)}$$

$$\delta_2^{(3)} = \textcolor{blue}{\Theta}_{12}^{(2)} \cdot \delta_1^{(4)}.$$