1

Neural Networks (Week 5)

$$J(\theta) = \frac{1}{m} \sum_{k=1}^{K} \sum_{i=1}^{m} -y_k^{(i)} \log(h_\theta(x^{(i)})_k) - (1-y_k^{(i)}) \log(1-h_\theta(x^{(i)})_k)$$

$$+ \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{S_{L+1}} \sum_{j=1}^{S_l} (\theta_{ij}^{(l)})^2$$

$L \rightarrow$ layers

$K \rightarrow$ classes

$S_l \rightarrow$ no. of features in layer l.

Backpropogat^n algorithm: To reduce $J(\theta)$

$\delta_j^{(l)}$ = "error" of node j in layer l.

$\delta^{(L)} = a^{(L)} - y$; where $L \rightarrow$ no. of layers in the network.

Suppose $L = 4$,

$\delta^{(4)} = a^{(4)} - y$

$\delta^{(3)} = (\theta^{(3)})^T \delta^{(4)} .* a^{(3)}(1 - a^{(3)})$

$\delta^{(2)} = (\theta^{(2)})^T . \delta^{(3)} .* a^{(2)}(1 - a^{(2)})$

$\cancel{\delta_j^{(l)} = \frac{\partial}{\partial \theta^{(l)}} cost(i)}$

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

$$\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta) = a_j^{(l)} \delta_i^{(l+1)}$$

$$\boxed{\frac{\partial}{\partial \theta^{(l)}} J(\theta) = \delta^{(l+1)} \times (a^{(l)})^T}$$

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

Adding Regulariz^n to it,

$$D_{ij}^{(l)} = \frac{1}{m} \Delta_{ij}^{(l)} \quad , \quad j=0$$

$$D_{ij}^{(l)} = \frac{1}{m} \left(\Delta_{ij}^{(l)} + \lambda \Theta_{ij}^{(l)} \right)$$

Checking the obtained Gradients

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{J(\theta_0+\varepsilon, \theta_1, \ldots, \theta_n) - J(\theta_0-\varepsilon, \theta_1, \ldots, \theta_n)}{2\varepsilon}$$

$$\vdots$$

$$\frac{\partial J(\theta)}{\partial \theta_n} = \frac{J(\theta_0, \theta_1, \ldots, \theta_n+\varepsilon) - J(\theta_0, \theta_1, \theta_2, \ldots, \theta_n-\varepsilon)}{2\varepsilon}$$

Calc. these approx gradients and see if they are equal to our calculated gradients found using Backprop.

Random initializ^n of Theta

* Theta can't be initialised to zeroes as it introduce a symmetry into the problem thus grad. descent wouldn't give the desired answer.

∴ Theta is chosen randomly betn values $[-\varepsilon, \varepsilon]$.

Theta = rand(n, 1) * (2 * INIT_EPSILON) - INIT_EPSILON;

Scanned by CamScanner