IE 598 HW2

Q1) Back propogation involves computing of gradients in a multiple layer neural network. This is typically done after the forward propogation step.

F Propogation

$$z^1 = w^1 x + b^1$$

$$h(z) = (\sigma(z_1), \sigma(z_2) \cdots \sigma(z_k)) \; R^2 \to R^2$$

$$a^2 = h(z_1)$$

$$z^2 = w^2 a^2 + b^2$$

$$\vdots$$

$$z^m = w^m a^m + b^m$$

$$a^{m+1} = h(z^m)$$

$$\tilde{a}_j^{m+1} = \underbrace{r_j^{m+1}}_{} \quad a^{m+1}$$

$\longrightarrow$ Bernoulli r.v. $w \to r_j^{m+1}$

$$\therefore z^{m_0+1} = w^{m_0+1} \underline{\tilde{a}^{m_0+1}} + b^{m_0+1} \quad \longrightarrow r_j^{m+1} a^{m+1}$$

$$\therefore z_\theta(x, y) = (g(z^{m_0+1})))_y$$

$$z^{m_0+1} = w^{m_0+1} a^{m_0+1} + b^{m_0+1}$$

$$\Rightarrow z_\theta(x,y) = (g(z^{m_0+1}))_y$$

$\theta \Rightarrow \{w^1, w^{m_0+1}, b^1 \ldots b^{m_0+1}\}$ where $w^1 \in R^{L \times d}$, $w^{m_0+1} \in R^{k \times L}$
$b^m \in R^2$, $b^{m_0+1} \in R^k$ $1 \leq m \leq m_0$ $1 < m < m_2$ $w^m \in R^{L \times L}$

Back propogation

$$z^m = w^m \tilde{a}^m + b^m$$

$$a^{m+1} = h(z^m)$$

Multiply by Bernoulli r.v

$$\tilde{a}_j^{m+1} = r_j^{m+1} a_j^{m+1}$$

where $r_j^{m+1} \sim$ Bernoulli (p) and r_j^m and $r_j^{m'}$ are independent.

Prediction $\Rightarrow z^m = p w^m a^m + b^m$ $\quad a^{m+1} = h(z^m)$ [Outpout of dropout while training]

$$E[\tilde{a}_j^m] = a_j^m p + (1-p)0 = a_j^m p \Rightarrow$$

A good optimizer ⇒ SGD w momentum.

⇒ SGD has learning rate η and initial parameter θ

```
while not finished:-
        sample a minibatch of size m from ha
    set $\{x^{(1)} \ldots x^{(m)}\}$ and $\hat{g} = 0$
    for $i = 1 \ldots m$ do
        Computed Gradient Descent.
            ~~$\hat{g}$ $\hat{g}$ $+ \nabla_\theta L$~~ [illegible]
        Apply update $\theta = \theta_x - \eta\hat{g}$
```

SGD w momentum has the difference when Computing gradient descent ⇒ $\hat{g} \leftarrow g + \nabla_\theta L(f(x^{(i)}, \theta)$

$$r = \alpha r - \epsilon g$$

$$\Rightarrow \theta \leftarrow \theta + r$$

Thus we have derived the back propagation algorithm for the multi layer feedforward neural network. We have also stated the SGD Algorithm for this neural network.

Hence proved. QED.