AAI - 695 Applied Machine Learning.

HW-2

Student := Hinal Patel
CWID = 10473912

Q:1 Cost function of Ridge Regression :-

$$E(w) = MSE(w) + \lambda/2 \sum_{i=1}^{m} w_i^2$$

$$= \frac{1}{m} \sum_{i=1}^{m} (h_w(x_i) - y_i)^2 + \lambda/2 \sum_{i=1}^{m} w_i^2$$

$$= \frac{1}{m} (x_w - y)^T (x_w - y) + \lambda/2 \, w^T w.$$

Neglecting $\frac{1}{m}$ & $\frac{1}{2}$ we get,

$$= ((x_w)^T - y^T)(x_w - y) + \lambda w^T w.$$

$$= (x_w)^T (x_w) - (x_w)^T (y) - (x_w)^T (y) + y^T y + \lambda w^T w$$

$$= w^T x^T x_w - 2 (x_w)^T y + y^T y + \lambda w^T w.$$

Now $\frac{dE}{dw} = 0$

On differentiating we get,

$$(x^T x + \lambda I) w = x^T y$$

multiplying $(x^T x + \lambda I)^{-1}$ on the both side.

$$w = (x^T x + \lambda I)^{-1} x^T y.$$

$$w = (\lambda I + x^T x)^{-1} x^T y \rightarrow \text{Hence Proved.}$$

Q:2

$$\hat{P}_k = \delta(S_k(n))_k = \frac{\exp(S_k(n))}{\sum_{j=1}^{k} \exp(S_j(n))}$$

where, $S_k(n) = \theta_k^T \cdot x$.

Training to minimize the cost function of cross entropy.

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

$$= \frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{k} y(k)^{(i)} \log\left(\frac{\exp(S_k(x)^{(i)})}{\sum_{j=1}^{k} \exp(S_j(n)^{(i)})}\right)$$

$$= -\frac{1}{m} \sum_{i=1}^{m} \left(\sum_{k=1}^{k} y_k^{(i)} \log(\exp(S_k(x^{(i)}))) - \sum_{k=1}^{k} y_{(k)}^{(i)} \log\left(\sum_{j=1}^{k} \exp(S_j(x^{(i)}))\right)\right)$$

$y_k^{(i)} = 1$ if the i^{th} instance belongs to class k.

→ ~~k~~ Softmax Regression gradient for cross-entropy cost funⁿ. –

$$\nabla_{\theta_k} J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} (\hat{P}_k(i) - y_{(k)} x^{(i)}) x(i)$$

$$= -\frac{1}{m} \sum_{i=1}^{m} x(i) - \frac{1}{\sum_{j=1}^{k} \exp(\theta_j^T x(i))} \exp(\theta_k^T x^{(i)}) x_i$$

$$= -\frac{1}{m} \sum_{i=1}^{m} (1 - \hat{P}_k(i)) x(i)$$

$$= \frac{1}{m} \sum_{i=1}^{m} \cdot (\hat{P}_k(i) - 1) x(i)$$

$$= \frac{1}{m} \sum_{i=1}^{m} (\hat{P}_k(i) - y_k(i)) x^{(i)} = \nabla_{\theta_k} J(\theta)$$

Hence proved.