Problem 3(a).

Given that, the maximum likelihood estimation:

$$\min_{W} F(W) \text{ where } F(W) = \frac{1}{n}\sum_{i=1}^{n} -\log\left[\Pr(y = y_i \mid x = X_i; W)\right] + \frac{\eta}{2}\|W\|_F^2$$

The gradient of $F(W) \rightarrow \frac{\partial F(W)}{\partial w}$:

Each data point is independent. So the probability of all the data is:

$$F(W) = \prod_{i=1}^{n} \Pr(y = y^{(i)} \mid x = X^{(i)})$$

$$= \prod_{i=1}^{n} r(\theta^T x^{(i)})^{y^{(i)}} \left[1 - r(\theta^T x^{(i)})\right]^{(1-y^{(i)})}$$

Here, $r = \dfrac{\exp(z_k - \max_j z_j)}{\sum_{j=1}^{c} \exp(z_j - \max_j z_j)}$ where $z = \theta^T x$

Where, c = number of possible classes.

and $k \in \{1, 2, \ldots\ldots c\}$

Derivative of gradient for one datapoint (X, Y) without regulariz

$$\frac{\partial F(W)}{\partial w} = \frac{\partial}{\partial w} y \log \sigma(\theta^T x) + \frac{\partial}{\partial w}(1-Y)\log[1-\sigma(\theta^T x)]$$

$$= \left[\frac{Y}{\sigma(\theta^T x)} - \frac{1-Y}{1-\sigma(\theta^T x)}\right]\frac{\partial}{\partial w}\sigma(\theta^T x)$$ (derivative of $\log f(x)$)

$$= \left[\frac{Y}{\sigma(\theta^T x)} - \frac{1-Y}{1-\sigma(\theta^T x)}\right]\sigma(\theta^T x)[1-\sigma(\theta^T x)] x_j$$

[chain rule + derivative of sigma]

$$= \left[\frac{Y-\sigma(\theta^T x)}{\sigma(\theta^T x)[1-\sigma(\theta^T x)]}\right]\sigma(\theta^T x)[1-\sigma(\theta^T x)]\, x_j$$

[algebraic manipulation]

$$= [Y - \sigma(\theta^T x)]\, x_j$$ [cancelling term]

The gradient of theta is simply the sum of this term for each training datapoint.

Finally, the gradient of $F(W)$ with regularizer

$$\frac{\partial F(W)}{\partial w} = -\frac{\eta}{2} w_i^{(t)} + \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 | x^j, w^{(t)})]$$

Scanned with CamScanner

The gradient descent rule for w:

- First we have to initialize $w^{(1)} \in R^D$ randomly.

- The we have to iterate through the following convergence:

$$\underbrace{w_i^{(t+1)}}_{\text{new value}} \leftarrow \underbrace{w_i^{(t)}}_{\text{previous value}} + \underbrace{\alpha}_{\text{learning rate}} \underbrace{\left\{-\lambda w_i^{(t)} + \sum_j x_i^j \left[y^j - \hat{P}(Y^j = 1 \mid x^j w^{(t)})\right]\right\}}_{\text{gradient at previous value}}$$

The updates give larger weights to those examples on which the current model makes larger mistakes.

Camparison with least mean square algorithm:

$$w(n+1) = w(n) + \frac{1}{2}\mu[-\nabla \underbrace{J(n)}_{\text{cost function}}]$$

LMS algorithm is based on the idea of gradient descent to search for the minimum error with a cost function equal to the mean squared error, where gradient descent doesn't force the use of any particular cost function, it hunts for the minimum cost solution