HW#3

1.

$$L(w) = \frac{1}{2}\left[(Qw - Y)^T(Qw - Y) + \lambda W^T I_{0+M} W\right]$$

W – weight matrix $((M+1)\times 1)$

Y – label vector $(M \times 1)$

I_{0+M} – Identity Matrix $((M+1)\times(M+1))$ $I_{0+M} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 1 & & \\ \vdots & & 1 & \\ 0 & & & 1 \end{bmatrix}$

$$\frac{\partial L(w)}{\partial w} = 2Q^T(Qw - Y) + 2\lambda I_{0+M} W = 0$$

$$\therefore W^* = (Q^TQ + \lambda I_{0+M})^{-1} Q^T Y$$

2. $\sigma \to \infty$ $k(y, z) = 1 \Rightarrow \phi_i(x) = \frac{1}{M}$ & $\phi_0(x) = 1$

$$Q = \begin{bmatrix} 1 & \frac{1}{M} & \frac{1}{M} & \cdots \\ 1 & \frac{1}{M} & & \\ \vdots & & & \\ 1 & \frac{1}{M} & \frac{1}{M} & \frac{1}{M} \end{bmatrix} \qquad Q^T = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \frac{1}{M} & \frac{1}{M} & \frac{1}{M} & & \frac{1}{M} \\ \frac{1}{M} & & & & \\ \vdots & & & & \vdots \\ \frac{1}{M} & & & & \frac{1}{M} \end{bmatrix}$$

$(M+1)\times M$

$$Q^TY = [\Sigma y_i, \bar{y}, \bar{y}, \cdots \bar{y}] \quad ((M+1)\times 1)$$

$$W^* = [\frac{\Sigma y_i}{M}, 0, 0, 0 \cdots] = [\bar{y}, 0, 0, \cdots 0]$$

For $\phi_i(x) = \frac{1}{M}$, we know that all the features are in the same space and expect linear regression. Under this condition, it might be good to only update bias to adjust model and eliminate penalty term by expecting $w_1 - w_M = 0$

$$\bar{y} = \frac{\sum_{i=1}^{M} y_i}{M}$$

3. $L(w) = \frac{1}{2}[(Qw-Y)^T(Qw-Y)]$

It will be possibe to achieve $Qw - Y = 0$

$\sigma \to 0$ $k(y,z) = \begin{cases} 0 & y \neq z \\ 1 & y = z \end{cases}$ $\therefore \phi_i(x) = \begin{cases} 1 & \text{when } i=j \\ 0 & \text{when } i \neq j \end{cases}$

& $\phi_0(x_j) = 1$

$\therefore$ $Q = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 1 & & & \\ \vdots & 0 & 0 & 1 & \ddots & \\ 1 & & & & & 1 \end{bmatrix}$ $M \times (M+1)$

$Q^T = \begin{bmatrix} 1 & 1 & 1 & \cdots & & 1 \\ 1 & 0 & 0 & & & \\ 0 & 1 & 0 & & & \\ 0 & 0 & 1 & & & \\ \vdots & \vdots & \vdots & & & \\ 0 & & & & & 1 \end{bmatrix}$

If there is no penalty, we expect $W = Q^{-1}Y$ & ignore bias term.

$\therefore w^* = [y_1, y_2, y_3, \cdots y_M]$

4. $L(w) = \frac{1}{2}[(Qw-Y)^T(Qw-Y) + \lambda W I_{0+M} W]$

$w^* = (Q^TQ + \lambda I_{0+M})^{-1} Q^T Y$

$Q^TQ = \begin{bmatrix} M & 1 & 1 & \cdots & 1 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ 1 & & & 1 & \\ \vdots & & & & 1 \end{bmatrix}$

$Q^TY = [\Sigma y_i, y_1, y_2, y_3 \cdots y_M]$

$\therefore w^* = [\bar{y}, \frac{y_1}{1+\lambda}, \frac{y_2}{1+\lambda} \cdots \frac{y_M}{1+\lambda}]$

2. programming

I. Binary model

Max_iters (Hidden_size = 32)	Training Accuracy	Validation Accuracy
40	0.921	0.475
60	**0.940**	**0.970**
80	0.941	0.480

Hidden_size (Max_iters = 60)	Training Accuracy	Validation Accuracy
32	0.940	0.970
64	0.950	0.975
128	**0.956**	**0.970**

Findings:

For fixed hidden_size, there are seriously overfitting for max_iter = 40 and max_iter = 80. In the other word, when hidden_size = 32 and max_iters = 60, the model performs better than other two in the first table.

For fixed max_iters, the performance of model will be improved by increasing hidden_size, which is obvious in the second table.

It could conclude that when hidden_size = 128 and max_iters = 60, the binary model performs best.

II. Multi-class model

Max_iters (Hidden_size = 32)	Training Accuracy	Validation Accuracy
300	0.886	0.783
400	0.889	0.799
500	**0.899**	**0.800**

Hidden_size (Max_iters = 400)	Training Accuracy	Validation Accuracy
32	0.899	0.799
64	0.907	0.808
128	**0.928**	**0.829**

Findings:

For fixed hidden_size, the gap between training accuracy and validation accuracy could be decreased by increasing max_iters to some extent. Unfortunately, when max_iters = 500, the gap becomes larger.

For fixed max_iters, it is obvious that training accuracy and validation accuracy are improved for increasing hidden_size. However, the overfitting problem still exists.

It could conclude that to improve accuracy, increasing hidden_size has better effect. And we may need regularization to avoid overfitting.

III. Regularization model

Max_iters (Hidden_size = 32)	Training Accuracy	Validation Accuracy
300	0.872	0.784
400	0.882	0.786
500	**0.887**	**0.791**

Hidden_size (Max_iters = 400)	Training Accuracy	Validation Accuracy
32	0.882	0.786
64	0.904	0.806
128	**0.920**	**0.834**

To avoid overfitting, we could increase the size of training set or add some penalty term. I have tried the dropout for neural network, which randomly sets some neurons to zero. The probability of dropping is set as 0.5.

By comparing with multi-class model and regularization, we are expecting following conclusion. That is, with the same number of neurons and increasing times of iterations, the gap between training accuracy associated with validation accuracy decrease and the accuracy is improved. However, it could not be concluded by observed data. It might be a good idea to collect more training data.