BINGHAMTON UNIVERSITY | The State University of New York

Thomas J. Watson School of Engineering & Applied Science
Department of Electrical & Computer Engineering

Neural Network & Deep Learning
(EECE680C)

Homework_3

Solutions

Submitted By

Alem Haddush Fitwi
afitwi1@binghamton.edu
Graduate Student

Submitted To

Dr. Xiaohua Li
Associate Professor xli@binghamton.edu

Submission Date

05 March 2018

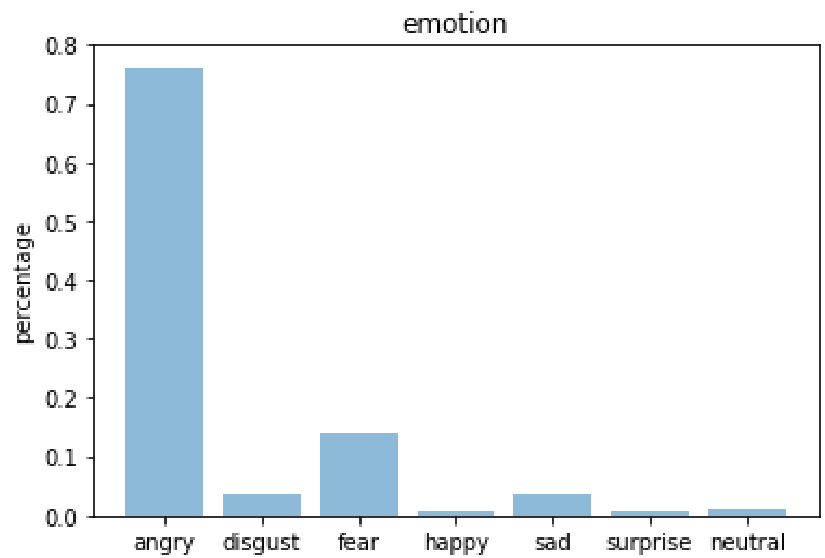
Binghamton, New York

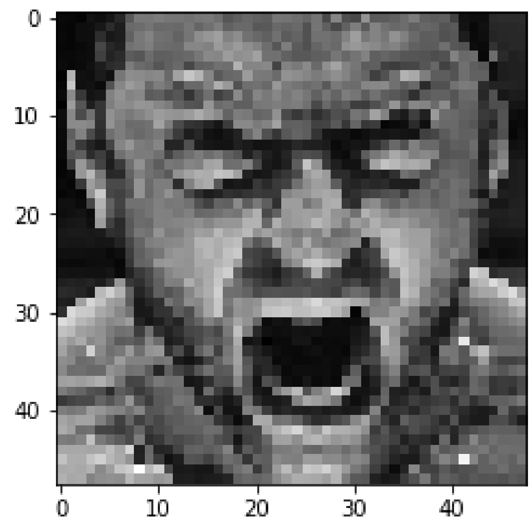
Exercise 12.1: Three flowers (Pink, White, and yellow) classification using CNN model python program. Here is a sample out after the program had been run for 5 epochs.

```
Restarting kernel...

In [1]: runfile('/home/alem/NNDL/Ex_12.1_flowers_Classication.py', wdir='/home/alem/NNDL')
------------------****************------------------------
-----------------Traning is running-----------------------
**********************************************************
Found 210 images belonging to 3 classes.
Found 30 images belonging to 3 classes.
Epoch 1/5
210/210 [==============================] - 199s 949ms/step - loss: -4.8824 - acc: 0.6134 - val_loss: -5.0850 - val_acc: 0.6667
Epoch 2/5
210/210 [==============================] - 199s 950ms/step - loss: -5.3182 - acc: 0.6621 - val_loss: -5.1545 - val_acc: 0.6333
Epoch 3/5
210/210 [==============================] - 216s 1s/step - loss: -5.3444 - acc: 0.6633 - val_loss: -5.2651 - val_acc: 0.6667
Epoch 4/5
210/210 [==============================] - 219s 1s/step - loss: -5.2759 - acc: 0.6641 - val_loss: -5.3065 - val_acc: 0.6667
Epoch 5/5
210/210 [==============================] - 220s 1s/step - loss: -5.2970 - acc: 0.6677 - val_loss: -5.3017 - val_acc: 0.6667
------------------****************------------------------
------------------Traning is over-------------------------
**********************************************************

In [2]:
```

Exercise 14.1: Classification of seven facial expressions (angry, disgust, fear, happy, sad, surprise, and neutral) using CNN model python program. Here under is a sample training result and a bar graph of a prediction test.

```
Python 3.6.4 |Anaconda custom (64-bit)| (default, Jan 16 2018, 18:10:19)
Type "copyright", "credits" or "license" for more information.

IPython 6.1.0 -- An enhanced Interactive Python.

In [1]: runfile('/home/alem/NNDL/Ex_14.1_FacialExpression.py', wdir='/home/alem/NNDL')
Using TensorFlow backend.
Epoch 1/5
256/256 [==============================] - 628s 2s/step - loss: 1.8099 - acc: 0.2508
Epoch 2/5
256/256 [==============================] - 612s 2s/step - loss: 1.6359 - acc: 0.3406
Epoch 3/5
256/256 [==============================] - 614s 2s/step - loss: 1.4049 - acc: 0.4572
Epoch 4/5
256/256 [==============================] - 641s 3s/step - loss: 1.2740 - acc: 0.5132
Epoch 5/5
256/256 [==============================] - 620s 2s/step - loss: 1.1818 - acc: 0.5524
3589/3589 [==============================] - 12s 3ms/step
-------------------************-------------------------
***************Test loss: 1.19384856449 ******************
***************Test accuracy: 54.5277236007 ***************
-------------------************-------------------------
```


emotion
percentage
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
angry
disgust
fear
happy
sad
surprise
neutral


0
10
20
30
40

Solution to Homework #3

#1 Given:

✓ Multilayer perceptron comprising
① An Input layer with 10 neurons
② A Hidden layer with 50 neurons
③ An Output layer with 3 neurons

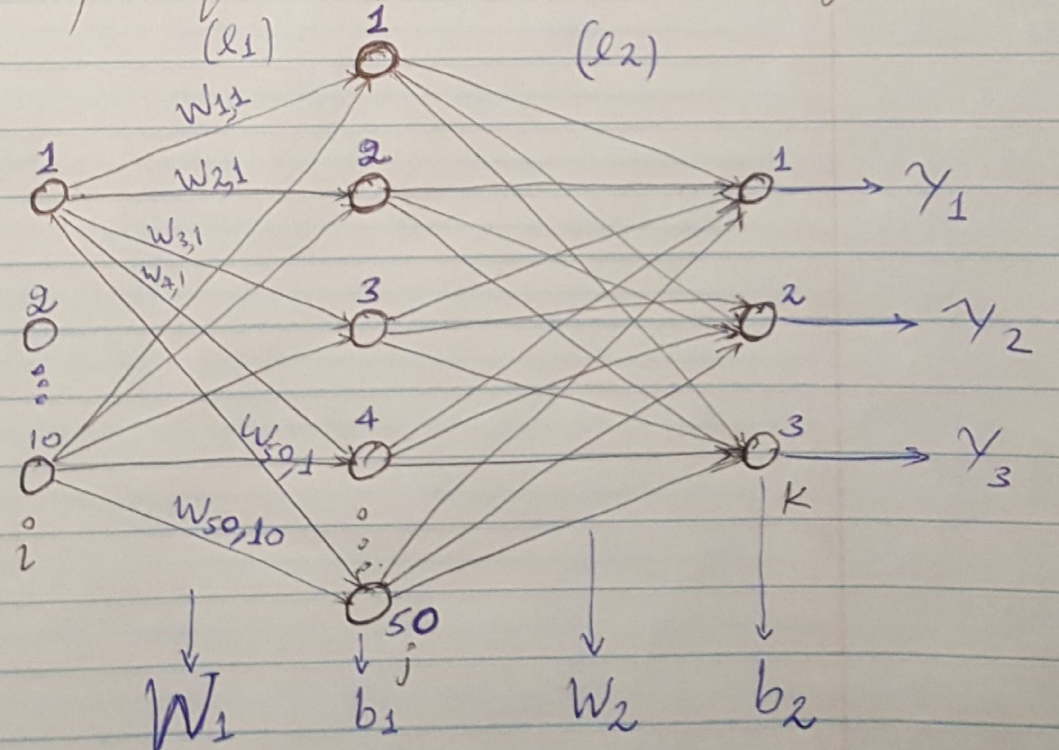
✓ Activation Function = ReLU

$$\phi(V) = \max(V, 0)$$

where $$V = \sum_{i=1}^{m} W_i X_i$$

Sol:

✓ Simplified Network Diagram



✓ Let $i = 1$ to 10 (input counter)
$j = 1$ to 50 (hidden layer output/node counter)

$K = 1$ to 3 (output layer node counter)

(a) Dimensions (or shapes) of X, W, b, & Y

$$X = [x_1, x_2, x_3, \cdots, x_{10}]^T = [1 \times 10]^T$$

$$W_1 = [10 \times 50]^T = \begin{bmatrix} W_{1,1} & W_{1,2} & \cdots & W_{1,50} \\ W_{2,1} & W_{2,2} & \cdots & W_{2,50} \\ \vdots & \vdots & & \vdots \\ W_{10,1} & W_{10,2} & \cdots & W_{10,50} \end{bmatrix}^T$$

$$b_1 = \begin{bmatrix} b_1 \\ \vdots \\ b_{50} \end{bmatrix} = [50 \times 1]$$

$$V_1 = XW_1 = [1 \times 50] \left(\text{or} = [V_1 \ V_2 \ V_3 \ \cdots \ V_{50}]^T \right)$$

$$W_2 = [50 \times 3] = \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ \vdots & \vdots & \vdots \\ W_{50,1} & W_{50,2} & W_{50,3} \end{bmatrix}$$

$$b_2 = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix}$$

(b) Formulate the equation that computes Y as a function of X, W_1, b_1, W_2 & b_2

So $\frac{n}{}$:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} \quad K = 1 \text{ to } 3$$

where

$$Y_1 = \phi\left(\sum_{j=1}^{50} W_{1j} V_j + b_1\right)$$

$$Y_2 = \phi\left(\sum_{j=1}^{50} W_{2j} V_j + b_2\right)$$

$$Y_3 = \phi\left(\sum_{j=1}^{50} W_{3j} V_j + b_3\right)$$

$$Y_K = \phi\left(\sum_{j=1}^{50} W_{Kj} V_j + b_K\right)$$

$K = 1$ to 3

$$V_j = \phi\left(\sum_{i=1}^{10} W_{ji} X_i + b_j\right)$$

- ϕ refers to the activation function ReLu assumed to be applied on both the hidden & output layers!

$\therefore \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix}$, where

$$Y_1 = \phi\left(\sum_{j=1}^{50} W_{1j} V_j + b_1\right)$$

$$= \phi\left(\sum_{j=1}^{50} W_{1j}\left(\phi\left(\sum_{i=1}^{10} W_{ji} X_i + b_j\right)\right) + b_1\right)$$

The same procedure applies to Y_2 & Y_3

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} \phi\left[\sum_{j=1}^{50} W_{1j}\left[\phi\left[\sum_{i=1}^{10} W_{ji} X_i + b_j\right]\right] + b_1\right] \\ \phi\left[\sum_{j=1}^{50} W_{2j}\left[\phi\left[\sum_{i=1}^{10} W_{ji} X_i + b_j\right]\right] + b_2\right] \\ \phi\left[\sum_{j=1}^{50} W_{3j}\left[\phi\left[\sum_{i=1}^{10} W_{ji} X_i + b_j\right]\right] + b_3\right] \end{bmatrix}$$

#2 Given:

- CNN composed of 3 convolutional layers
  - Input image size = $200 \times 300 \times 3$ (RGB)
  - Layer 1 has 100 feature maps ($K_1 = 100$)

- Layer 2 has 200 feature maps ($K_2 = 200$)
- Layer 3 has 400 feature maps ($K_3 = 200$)

Sol^n:

$$200 \times 300 \times 3 \text{ (RGB)} \rightarrow \boxed{\text{conv } 1} \rightarrow \boxed{\text{conv } 2} \rightarrow \boxed{\text{conv } 3} \rightarrow$$

conv 1	?	conv 2	?	conv 3	?
$K = 100$		$K = 200$		$K = 400$	
$S = 2$		$S = 2$		$S = 2$	
$F = 3$		$F = 3$		$F = 3$	

(a) Shape of Weight parameters (w & b)

- At Layer-1, Layer-2, & Layer-3
  - Accept a volume of size $W_1 \times H_1 \times D_1$, where D_1 = depth
  - Requires four hyperparameters
    - K = # of filters (kernel filters)
    - F = spatial extent (or kernel filter size)
    - S = Stride
    - P = the amount of zero-padding
  - produces a volume of size $W_2 \times H_2 \times D_2$

$$W_2 = (W_1 - F + 2P)/S + 1$$

$$H_2 = (W_1 - F + 2P)/S + 1$$

$$D_2 = K$$

- With parameter sharing, it introduces
  - $F * F * D_1$ weights per filter
  - $(F * F * D_1) * K$ Total weights
  - K biases

@ Layer-1 :

- $F=3$, $K=100$, $S=2$, $P=$ SAME
- Input image size $= 200 \times 300 \times 3$

⊛ SAME padding

$$W_2 = \text{Ceil}\left[\text{float}(W_1)/\text{float}(S)\right]$$

$$= \text{ceil}\left[\frac{200.0}{2.0}\right]$$

$$= 100$$ //

$$H_2 = \text{ceil}\left[\text{float}(H_1)/\text{float}(S)\right]$$

$$= \text{ceil}\left[\frac{300.0}{2.0}\right]$$

$$= 150$$ //

$$D_2 = K = 100$$ //

⊛ Output shape $\Rightarrow W_2 \times H_2 \times D_2 = \underline{100 \times 150 \times 100}$

$\Rightarrow W_2 \times H_2 \times D_2 = \underline{100 \times 150 \times 100}$

⊛ Weights (w + b)

- Weights per filter
  $= F * F * D1$
  $= 3 * 3 * 3$ //
- Total weights (for all filters)
  $= (F * F * D_1) * K$
  $= (3 * 3 * 3) * 100$ //

- Bias per filter
  $= 1$
- Total bias (for all filters)
  $= 1 \times K$
  $= 100$ //

@ Layer-2

- $F = 3$, $K = 200$, $S = 2$, $P = \text{SAME}$ //
- Input image size $= 100 \times 150 \times 100$ //
- SAME padding

$$W_3 = \text{ceil}\left[\frac{100.0}{2.0}\right] = 50 \ //$$

$$H_3 = \text{ceil}\left[\frac{150.0}{2.0}\right] = 75 \ //$$

$$D_3 = K = 200$$

- Output shape
  $= W_3 \times H_3 \times D_3 = \underline{\underline{50 \times 75 \times 200}}$

- Weights $(w + b)$
  - Weights per filter
    $= F \times F \times D_2$
    $= (3 \times 3 \times 100)$
  - Weights of all filters (Total)
    $= (F \times F \times D_2) \times K$
    $= (3 \times 3 \times 100) \times 200$ //

- Bias per filter
  $= 1$
- Bias ~~for~~ for all filters
  $= 1 * K$
  $= 1 * 200$
  $= 200$ //

② Layer-3

- $F = 3$, $K = 400$, $S = 2$, $P = SAME$
- Input image size $= 50 \times 75 \times 200$
- SAME padding (float)

$$W_4 = ceil(W_3/S) = ceil\left(\frac{50.0}{2.0}\right)$$
$$= 25 //$$
$$H_4 = ceil(float(H_3)/S) = ceil\left(\frac{75.0}{2.0}\right)$$
$$= ceil(37.5)$$
$$= 38 //$$
$$D_4 = K = 400 //$$

- Output shape $\Rightarrow W_4 \times H_4 \times D_4$
  $= 25 \times 38 \times 400$ //

- Weights ($w + b$)
  - Weights per filter $= F * F * D_3 = 3 \times 3 \times 200$
  - Total weights $= (F \times F \times D_3) * K = (3 \times 3 \times 200) 400$
  - Bias per filter $= 1$
  - Bias for all filters $= 1 \times K = 1 \times 400 = 400$ //

(b) Total number of parameters in the CNN

Sol^n:

ⓐ Layer-1

- Unique set of weights
  $= (3\times3\times3)100$
  $= 900$
- unique set of biases
  $= 1 * 100$
  $= 100$
- Total parameters
  $= w+b$
  $= 1000$ //

ⓐ Layer-2

$W = (3\times3\times100) * 200 = 180,000$

$b = 1\times200 = 200$

Total parameter $= w+b = 180,200$

ⓒ Layer-3

$W = (3\times3\times200)\times400 = 720,000$

$b = 1\times400 = 400$

Total parameters $= w+b = 720,400$

Hence, the total parameters $(w+b)$ of this CNN

$=$ Layer-1 total $+$ Layer-2 total $+$ Layer-3 total

$= 1000 + 180,200 + 720,400$

$= 901,600$ //

#3 Given:

- linearly training samples

Label	x_1	x_2	Given Class
x_3	0	1	C_0
x_2	1	0	C_1
x_1	1	1	C_1

Assume $b = 0$ for simplicity

Sol^n

(a) Assuming

- The activation function is Linear
- The loss function is $\frac{1}{2}\sum_i (d_i - y_i)^2$

⊛ The Activity Rule

(a) $V = \sum_{i=1}^{n} W_i X_i$

(b) $y = \phi(v) = \max(0, V) = \begin{cases} 0, & \text{if } V \le 0 \\ V, & \text{if } V > 0 \end{cases}$

⊛ The Learning Rule

(a) Learning error calculation

$$e(n) = \frac{1}{2}\sum_i (d_i(n) - y_i(n))^2$$

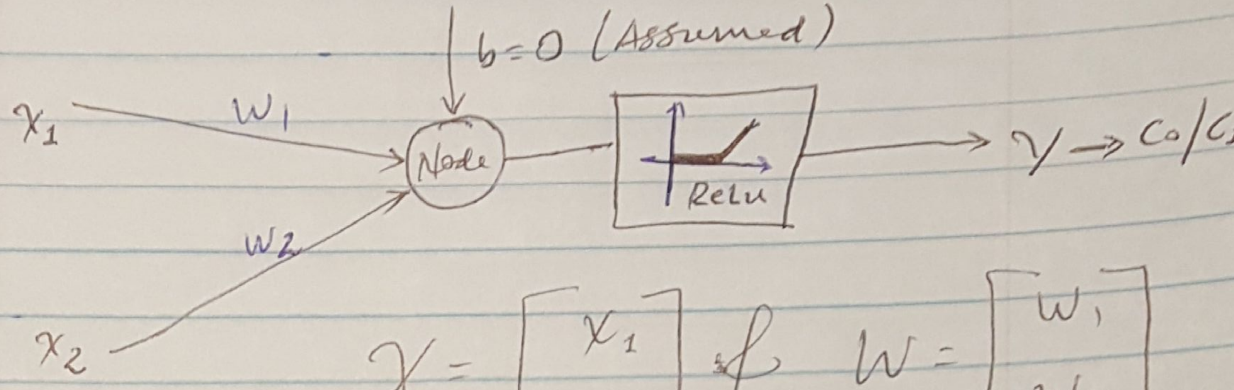
(b) Weight update (or change in weight)

$$\Delta w = \eta e X \rightarrow \Delta w(n) = \eta e(n) X(n)$$

$$W = W_{initial} + \Delta w$$

⊛ Initialize $W = (w_1, w_2) = (0, 0)$

→ b is assumed to be zero!



$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \ \& \ W = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

Let $d = 1$ for class C_0 & $\eta = 0.5$ //
$d = -1$ for class C_1

Iteration - 1: $(x_1(1), x_2(1))^T = (0, 1)^T \rightarrow C_0$

$$V(1) = w_1^0 x_1 + w_2^0 x_2 = 0$$

$$Y(1) = \phi((V(1)) + b=0) = 0 //$$

$$e(1) = \frac{1}{2}(t - y)^2$$

$$\frac{d e(1)}{d(Y(1))} = -(t - y) = -1 //$$

$$\Delta e(1) = -1$$

$$\Delta W(1) = \eta \, \Delta e(1) X(1) = -0.5 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -0.5 \end{bmatrix}$$

$$W(1) = W(0) + \Delta W(1)$$

$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \end{bmatrix} = \begin{bmatrix} 0 \\ -0.5 \end{bmatrix}$$

Iteration - 2: $(x_1(2), x_2(2))^T = (1, 0)^T \rightarrow C_1$

$$V(2) = w_2^1 x_2 + w_2^1 x_2 = 0$$

$$Y(2) = \phi(V(2) + b=0) = 0 //$$

$$\Delta e = \frac{\partial e(2)}{\partial Y(2)} = -(t - Y) = 1 //$$

$$\Delta w(2) = \eta \Delta e(2)\, X(2) = 0.5 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$$

$$W(2) = W(1) + \Delta W(2)$$
$$= \begin{bmatrix} 0 \\ -0.5 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}$$

Iteration-3: $(x_1(3), x_2(3))^T = (1,1)^T \rightarrow C_1$

$V(3) = W_1(2)\, X_1(3) = 0$ //

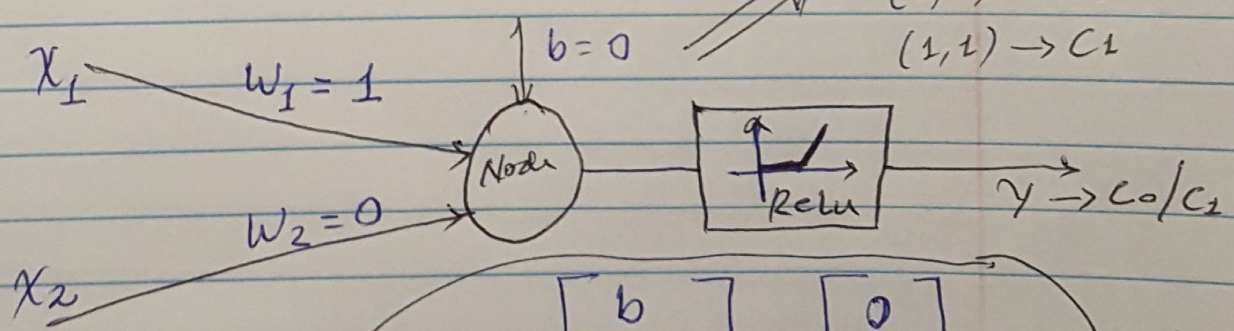
$Y(3) = \phi(V(3) + b = 0) = 0$ //

$\Delta e = 1$

$$\Delta W(3) = \eta \Delta e\, X(3) = 0.5 \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

$$W(3) = W(2) + \Delta W(3)$$
$$= \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$
$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$(0,1) \rightarrow C_0$
$(1,0) \rightarrow C_1$
$(1,1) \rightarrow C_1$



$$W = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$ //

(b) Assuming

- Linear Activation function
- Loss function with Ridge Regression

$$\frac{1}{2}\sum_i (d_i - y_i)^2 + \frac{\lambda}{2}\|w\|^2$$

Sol^n:

Assumption:

$$\|w\|^2 \Rightarrow \text{Normal 2} = \sqrt{w_1^2 + w_2^2}$$

Let $d = 1$ for class C_0

$d = -1$ for class C_1

$\lambda = 0.3$ & Initially $w_1 = 0$ & $w_2 = 0$

Iteration. 1: $(x_1(1), x_2(1))^T = (0, 1)^T \to C_0$

$$V(1) = w_1(0)\,x_1(1) + w_2(0)\,x_2(1) = 0$$

$$y(1) = \varphi(V(1) + b = 0) = 0$$

$$e(1) = \frac{1}{2}(d-y)^2 + \frac{\lambda}{2}\|w\|^2$$

$$= \frac{1}{2}(d-y)^2 + \frac{\lambda}{2}\left(\sqrt{w_1^2 + w_2^2}\right)^2$$

$$\frac{\partial e(1)}{\partial w} = -\frac{2}{2}(d - xw_i) + \lambda(w_1^2 + w_2^2)(2w_1 + 2w_2)$$

$$= -(d - xw_i) + \lambda(w_1^2 + w_2^2)(2w_1 + 2w_2)$$

$$\Delta e(1) = -(1-0) + 0.3(0)(0)$$

$$= -1$$

$$\Delta w(1) = \eta\,\Delta e(1)\,x(1) = -0.5\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\-0.5\end{bmatrix}$$

$$w(1) = w(0) + \Delta w(1)$$

$$= \begin{bmatrix}0\\-0.5\end{bmatrix}$$

Iteration-2: $(x_1(2), x_2(2))^T = (1,0)^T \rightarrow C_1$

$v(1) = w_1(1)x_1(2) + w_2(1)x_2(2) = 0$

$y(1) = \phi(v(1) + b = 0) = 0$

$e(2) = \frac{1}{2}(d-y)^2 + \frac{\lambda}{2}\|w\|^2$

$\Delta e(2) = -(d - x w_i) + \lambda(w_1^2 + w_2^2)(2w_1 + 2w_2)$

$= 1 + 0.3 \times 0.5^2$

$\eta = 0.5$

$= 1.075$

$\Delta w(2) = \eta \, \Delta e(2) \, X(2)$

$= 0.5375 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$= \begin{bmatrix} 0.5375 \\ 0 \end{bmatrix}$

$W(2) = w(1) + \Delta w(2)$

$= \begin{bmatrix} 0.5375 \\ -0.5 \end{bmatrix}$

Iteration-3: $(x_1(3), x_2(3))^T = (1,1)^T \rightarrow C_1$

$v(3) = w_1(2)x_1(3) + w_2(2)x_2(3) = 0.0375$

$y(3) = \phi(v(3) + b = 0) = 1$

$\Delta e(3) = -(d - x w_i) + 0.3(0.0404)$

$= 1.04962$

$\Delta w(3) = \eta \, \Delta e(3) \, X(3) = 0.52481 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$= \begin{bmatrix} 0.5245 \\ 0.5245 \end{bmatrix}$

$W(3) = W(2) + \Delta W(3) = \begin{bmatrix} 1.075 \\ 0.0245 \end{bmatrix}$

Iteration-4: $(X_1(4), X_2(4))^T = (0,1)^T \to C_0$

$v(4) = w_1(3)X_1(4) + w_2(3)X_2(4) = 0.0245$

$y(4) = \varphi(v(4) + 0) = 1$

$\Delta e(4) = -(+1 - 0.0245) + 0.1(1.15623)$

$\Delta = 0.1 \to$ changed.

$= -0.8599$

$= -0.86$

$\Delta W(4) = \eta \Delta e(4) X(4) = -0.43 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$= \begin{bmatrix} 0 \\ -0.43 \end{bmatrix}$

$W(4) = W(3) + \Delta W(4)$

$= \begin{bmatrix} 1.075 \\ 0.0245 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.43 \end{bmatrix}$

$= \begin{bmatrix} 1.075 \\ -0.405 \end{bmatrix}$

Hence, $W = \begin{bmatrix} 1.075 \\ -0.405 \end{bmatrix}$ & $b = 0$