1

Machine Learning Homework-1

Name-SRIHARI TANMAY KARTHIK TADALA
PSU ID-918597835.

Solution 1

1) Given P is invertible if $PP^{-1} = I$

Applying determinant on both sides

$$\det|PP^{-1}| = \det|I|$$

$$\det|P|\det|P^{-1}| = 1$$

$$\det|P| = \frac{1}{\det|P^{-1}|}$$

Now, $P^{-1}AP = B$

since $\det(P^{-1}AP) = \det(B)$

$$\det(P^{-1}) \cdot \det(A) \cdot \det(P) = \det(B)$$

$$\frac{1}{\det(P)} \times \det(A) \times \det(P) = \det(B)$$

Hence $\det(A) = \det(B)$

So therefore $\det(A) = \det(B)$ is true.

ii) Given $P^{-1}AP = B$

To prove A and B have the same eigen values

Eigen values are the solution of charatterstic equation below

$$P(\lambda) = \det(M - \lambda) = 0$$

Now, let consider B

$$B = P^{-1}AP$$

$$P_B(\lambda) = \det(B - \lambda I) = \det(P^{-1}AP - \lambda I)$$

$$P_B(\lambda) = \det(P^{-1}AP - P^{-1}\lambda P)$$

$$= \det(P^{-1}(A - \lambda I)P)$$

$$= \det(P^{-1}) \det(A - \lambda I) \cdot \det(P)$$

$$= \frac{1}{\det(P)} \times \det(A - \lambda I) \cdot \det(P)$$

$$P_B(\lambda) = \det(A - \lambda I)$$

$$P_B(\lambda) = P_A(\lambda)$$

($\because$ charateristic polynomial of matrix A is $P_A(\lambda)$)

This means that A and B have same eigen values.

solution 2

2) (i) $\vec{u} \cdot \vec{u} = \|\vec{u}\| \|\vec{u}\| \cos\theta$

$\Rightarrow \|\vec{u}\| \|\vec{u}\| \cos 0^\circ$

Hence $\vec{u} \cdot \vec{u} \Rightarrow \|\vec{u}\|^2$

(ii) Given Q is a orthogonal matrix and $\vec{x}$ a vector

We need to prove $\|Q\vec{x}\| = \|\vec{x}\|$

We know that Q is orthogonal matrix

So, $QQ^T = I$

(2)

When we multiply both sides with vector $\vec{x}$ then

$$Q \cdot Q^T \bar{x} = \bar{x} I \Rightarrow x \quad \text{ie}$$

$$Q \cdot Q^T x^{-1} = \vec{x} \rightarrow (1)$$

From (1) we know that

$$\|Qx\|_2^2 = \sum_{i=1}^{n} (Qx)_i^2 = \sum_{i=1}^{n} (Qx)_i (Qx)_i$$

$$= (Qx)^T (Qx)$$

$$= Q^T x^T Q x$$

$$= x^T I x \quad (Q^T Q = I)$$

$$= x^T x$$

$$= \|x\|_2^2$$

$$\therefore \|Qx\| = \|x\|$$

solution 3

A3) Considering from the question, we know for 10 Gaussian distribution

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} \times e^{-1/2 \left[\frac{x-u}{\sigma}\right]^2}$$

Now we have to differentiate and double differentiate

$$f'(x) = \frac{1}{\sigma \sqrt{2\pi}} \times e^{-1/2 \left(\frac{x-u}{\sigma}\right)^2} \times \frac{-1}{2} \times 2 \frac{(x-u)}{\sigma^2}$$

substitue for $x = u$ then $f'(x) = 0$

then $f'(x) = 0$

$$f'(x) = -\frac{1(x-u)}{\sigma^3\sqrt{2\pi}} \times e^{-\frac{(x-u)^2}{2\sigma^2}}$$

Now second derivative with respective to x would be

$$f''(x) = \frac{\left(\sqrt{2}(-\sigma^2+(x-u)^2 e^{-\frac{(x-u)^2}{2\sigma^2}} |\sigma|\right)}{2\sqrt{\pi}\sigma^6}$$

substitute for $x = u$ and simplify $f''(x)$, as $(u-u)^2 = 0$

$e^0 = 1$

$$f''(x) = \frac{\sqrt{2}(-\sigma^2)e^0|\sigma|}{2\sqrt{\pi}\sigma^6} = -\frac{\sqrt{2}|\sigma|}{2\sqrt{\pi}\sigma^4}$$

$f''(x) < 0$ as $\sigma^2 > 0$

considering $f'(x) = 0$ and $f''(x)$ is negative the gaussian function has a maximum at $x = u$.

solution 4

A4.) i) Expected number of heads is 3 flips

$$E(x) = \sum_{i=1}^{n} x P x$$

No of heads	ways to achieve	Probability of each outcome	Total probability.
0	1	1/8	1/8
1	3	1/8	3/8
2	3	1/8	3/8
3	1	1/8	1/8.

(3)

Based on above ; expected number of heads $E(X)$ can be found as

$$E(X) = 0 \times (1/8) + 1(3/8) + 2(3/8) + 3(1/8)$$

$$= 3/8 + 6/8 + 3/8 = 12/8 = 1.5$$

Expected number of heads $= 1.5$

4) A) ii) Probability density function of getting first heads is $E(X) = \sum P(E) \times X$

$$= \sum_{i=0} 1/2^i$$

$$E(X) = 1/2^0 + 1/2^1 + 1/2^2 + \ldots$$

$$= 1 + 1/2 + 1/2^2 + \ldots$$

$$= 2.$$

Expected number of tosses required would be 2.

X	P(X)
H	1/2
TH	1/4
TTH	1/8

$$E(X) = \sum P(E) \times X = \sum_{i=0}^{2} 1/2^i$$

$$E(X) = 1/2^0 + 1/2^1 + 1/2^2 + \ldots$$

$$= 1 + 1/2^1 + 1/2^2 + \ldots = 2$$

Expected no of tosses required would be 2.

Solution 5:

5A. $var(X+Y) = E((x+y) - E(x+y))^2$

$= E[((x+y) - E(x) - E(y))^2]$

$= E[((x - E(x)) + (y - E(y)))^2]$

$= E[(x - E(x))^2 + (y - E(y))]^2 + 2(x - E(x))\cdot(y - E(y))]$

since $(a+b)^2 = a^2 + 2ab + b^2$

$= E[(x - E(x))^2] + E[(y - E(y))^2] + 2E[(x - E(x))(y - E(y))]$

As we know

$var[X] = E(X^2) - (E(X))^2 = E[(X - E(X))^2]$

$var[Y] = E(Y^2) - (E(Y))^2 = E[(Y - E(Y))^2]$

$cov(x,y) = E((x - E(x))(y - E(y))]$

Based on the above

$var(X+Y) = var(X) + var(Y) + 2cov(X,Y)$

Hence we proved that $var(X+Y) = var(X) + var(Y) + 2cov(X,Y)$

6 → Solution 6

- A → A person vaguely identified

B → B person correctly identified.

$P(A/B) = 1/2$ (given 1 in 2 of the people recognized)

$P(A) = 1/5$ (given 1 in 5 are vaguely identified)

$$P(A/B) = \frac{P(A \cup B)}{P(B)} \Rightarrow 0.5 = \frac{P(A \cup B)}{0.1}$$

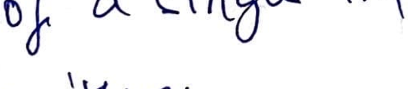
($\because P(B) = 1/10$, given 1 in 10 people were correctly identified)

$P(A \cup B) = 0.05$.

$$P(B/A) = \frac{P(A \cup B)}{P(A)} = \frac{0.05}{0.2} = 0.25$$

7. Solution 7

NOT input output



A	B
0	1
1	0

So the output of NOT gate is the inverse of a single input

perceptron algorithm:

$$y = \begin{cases} 1 & \text{if } wx + b > 0 \\ 0 & \text{if } wx + b \leq 0 \end{cases} \rightarrow (1)$$

For input 0, we initialize $w_1 = 1, b = +1$

$y = x(1) + 1$

$y = 0(1) + 1 \Rightarrow y = 1$

From ① we can conclude $w_1 = 1, b = 1$ as the value of y matches with output of logically NOT gate

For input 1:→ we suppose $w_1 = 1, b = 1$

$y = w_1 x_1 + b \Rightarrow 1(x_1) + 1 \Rightarrow (1)(1) + 1$ $\boxed{y = 2}$

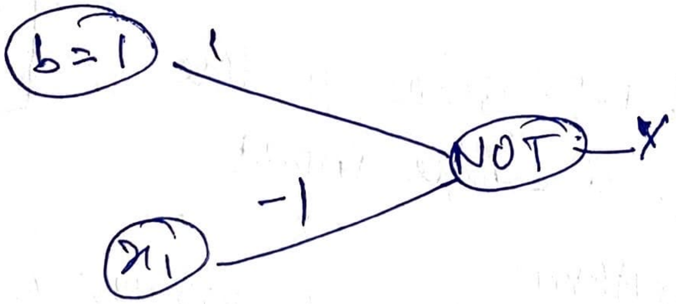
From ① if $wx + b > 0$ then $y = 1$ but it does not match the output as it is 0 for 1.

So we now change $w_1 = -1$

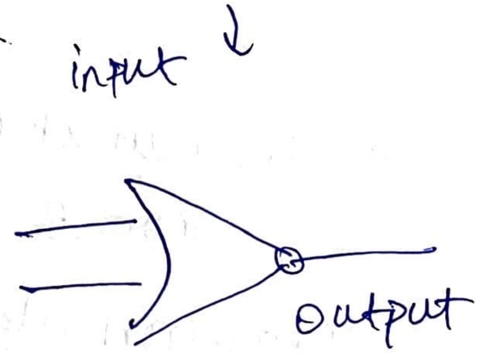
Then $y = w_1 x_1 + b \Rightarrow (-1)(1) + 1 = 0$

From ① then if $wx + b \leq 0$, then $y = 0$ it matches with output of NOT gate

$\therefore w_1 = -1, b = 1$



NOR Gate



A	B	output
0	0	1
0	1	0
1	0	0
1	1	0

⑤ For input (0,0):→

Lets initialize $w_0 = 1, w_1 = -1, w_2 = -1, u = 0.25$

$$y = h[w_0(1) + w_1 x_1 + w_2 x_2]$$
$$= h[(1)(1) + (\cancel{2})(0) + ()(0)]$$

$$y = h(1) > 0$$

So, $y = 1$ which is same as the output in table NOR.

For input (0,1)

$w_0 = 1, w_1 = -1, w_2 = -1, n = 0.25$

$$y = n(w_0(1) + w_1 x_1 + w_2 x_2)$$
$$= n(1 + w_1(0) + w_2(1)) =$$
$$= 0.25(1 + (-1)(\cancel{0}) + (-1)(1))$$
$$0$$
$$= 0$$

so $y = 0$ then since $wx + b$ is ≤ 0 and it matches with the output the output of NOR table.

For input (1,0):→

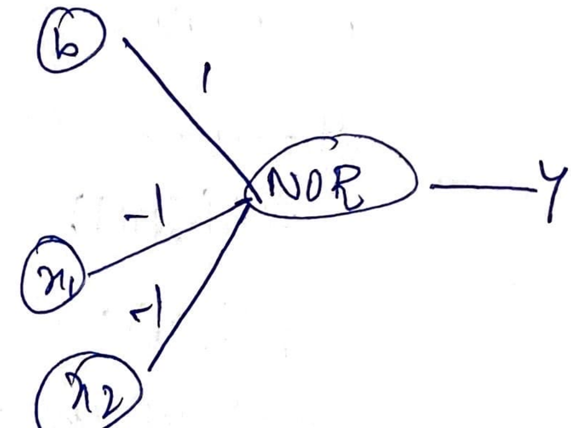
$w_0 = 1, w_1 = -1, w_2 = -1, n = 0.25$

$$y = n(1 + w_1 x_1 + w_2 x_2)$$
$$= n(1 + (-1)(-1) + (-1)(0)] < 0$$

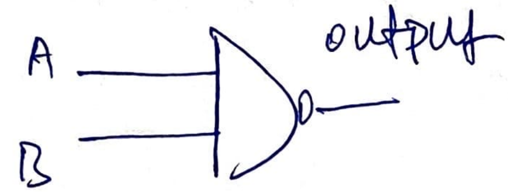
For input (1,1) $w_0 = 1, w_1 = -1, w_2 = -1$

$$y = n(1 + w_1 x_1 + w_2 x_2)$$
$$= n(1 + (-1)(1) + (-1)(0)) = n(1 - 1 - 1)$$
$$y = n(-1) < 0$$

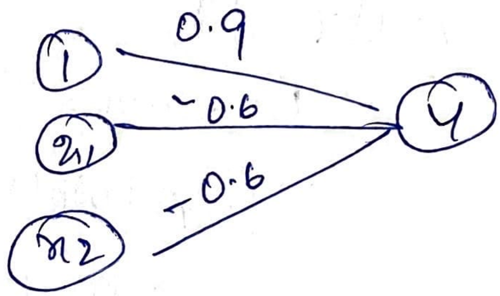
From ① $wx+b \leq 0, y=0$ which equals to output in NOR table



NAND:



The Network Design is



A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

For input (0,0):

$y = a(wx+b)$ - perception algorithm

$y = a(w_1x_1 + w_2x_2 + b)$

Here we initialize $w_1 = -0.6, w_2 = -0.6, w_3 = 0.9$

$y = a[(-0.6)(0) + (-0.6)(0) + 0.9)]$

$y = a(0.9) = 1 \quad y = \begin{cases} 1 & wx+b>0 \\ 0 & wx+b \leq 0 \end{cases}$

which satisfies the NAND output

For input (0,1)

$y = a((0)(-0.6) + (1)(-0.6) + 0.9)) = a(0.3) = 1$

which satisfy the NAND output

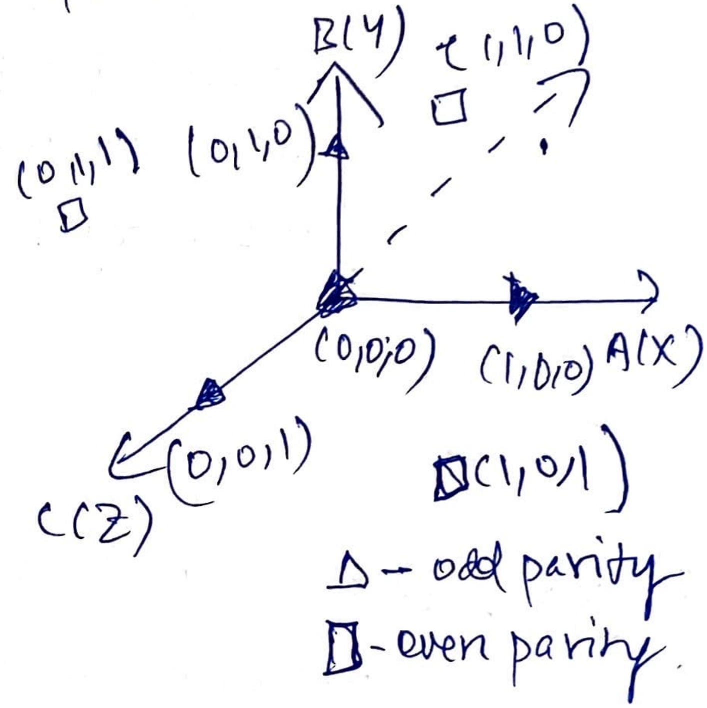
For input (1,1)

$$y = a((1)(-0.6) + (1)(-0.6) + 0.9)) = a(-0.12 + 0.9)$$
$$= a(0.3) = 0$$

A8.) Solutions:

A singular layer perceptron is not capable of learning the parity problem for 3 inputs. This is because the decision boundary that a single layer perceptron is linear & the parity problem for 3 input is a non linear problem.

A	B	C	t
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0
1	0	1	1
1	1	0	1
0	1	1	1
1	1	1	0



I cannot find the linear decision boundary to distinguish even and odd parity.

A9) a)

x_1	x_2	class(y)
0	0	1
0	1	1
1	0	0
1	1	1

$b=1, w_1=-2, w_2=0, w_0=1$

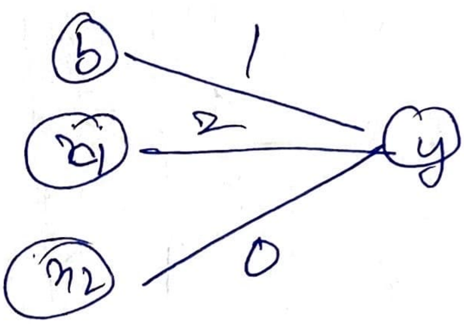
$y=\sum_i wx+b = w_1x_1+w_2x_2+w_0 b$

$(-2)(0)+(0)(0)+(1)(1)=1>0 \quad y=1$

$-2\cdot 0+1\cdot 1+1\cdot 1 = 1+1 = 2>0 \quad y=1$

$-2\cdot 1+1\cdot 0+1\cdot 1 = -2+1 \Rightarrow -1<0 \quad y=0$

$-2\cdot 1+1\cdot 1+1\cdot 1 = -2+1+1 = 0 \quad y=0$



b) $w_0=0.06, w_1=-0.06, w_2=-0.01$

$w_0+w_1x_1+w_2x_2=0$

$0.06-0.06x_1-0.01x_2=0$

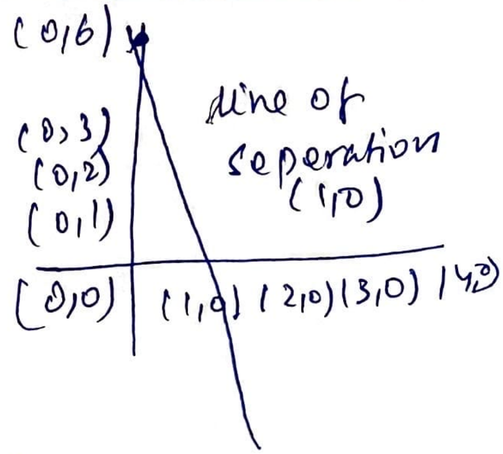
$\boxed{x_2=-6x_1+6} \leftarrow (1)$

Similar to $y=mx+c$ if $x_2=0$

$x_1=6/6=1$ ($\because 0=-6x_1+6$
$6x_1=6$)

(7)

c) Given $\{w_0, w_1, w_2, w_3\} = \{0.3, -0.8, -0.1, 0.2\}$



$x = (-1, 1, 3)$

To find class 0 or 1

$y = a(w \times x)$

$y = a(w_0 x_0 + w_1 x_1 + w_2 x_2 + w_3 x_3)$

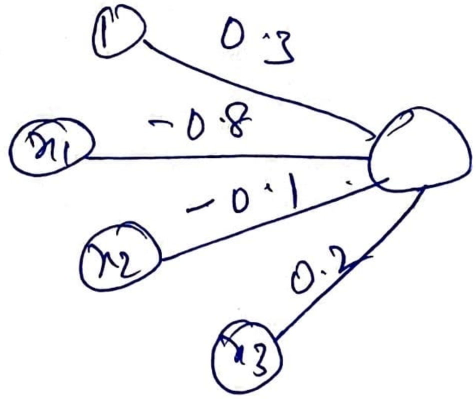
$y = a(0.3 \cdot (1) + (-0.8)(-1) + (-0.1)(1) + 0.2(-3))$

$y = a(0.4)$

Now from perception algorithm $y = \begin{cases} 1 & \text{if } wx > 0 \\ 0 & \text{if } wx \leq 0 \end{cases}$

since $0.4 > 0$ the perceptron will classify the input vector $(-1, 1, -3)$ as class 1.

d) perceptron from c



Ex1: $x = (1, 1, 2)$

$[w_0, w_1, w_2, w_3] = [0.3, -0.8, -0.1, 0.2]$

$y = \bar{a}(w_x \times x)$

$y = a((0.3 \times 1) + (-0.8 \times 1) + (-0.1)(1) + 2(0.2))$

$y = a(-0.2)$

$-0.2 < 0 \quad y = 0$

but the target is 1

So the prediction is wrong.

(8)

Ex 2:

$x = (1, -2, -2)$

$\{w_0, w_1, w_2, w_3\} = \{0.3, -0.8, -0.1, 0.2\}$

$y = a(\bar{w} \cdot x)$

$y = a(0.3 \cdot 1 + -0.8 \cdot 1 + (-0.1)(-2) + 0.2(-2))$

$= a(-0.7) \quad -0.7 < 0;\ y = 0$

Given target $t = 0$

the prediction is correct

e) $\text{Accuracy} = \dfrac{\text{No of prediction that are correct}}{\text{Total no of prediction}} = \dfrac{1}{2} = 0.5$

$= 50\%$

f) For this problem the given inputs are

inputs: $\{1, 1, 2\}$ weights $\{w_0, w_1, w_2, w_3\} = \{0.3, -0.8, -0.1, 0.2\}, n = 0.1$

For (1) the predection is worong so we update the weights wrong

$\Delta w_i = w_i - n(y^k - t^k)x_i^k$

$w_0 = \cancel{(0.3) - 0.1(1}\ 0.3 - 0.1(0-1)(1)) = 0.4$

$w_1 = (-0.8) - (0.1)(1) = -0.7$

$w_2 = (-0.1) - (0.1)(2) = -0.4$

$y = a(\cancel{a}\ w_n x_n)$

$y = a((0.4)(1) + (-0.7)(1) + (0 \times 1) + 0.4(2))$

ee

$y = a(0.4 - 0.7 + 0 + 0.8)$

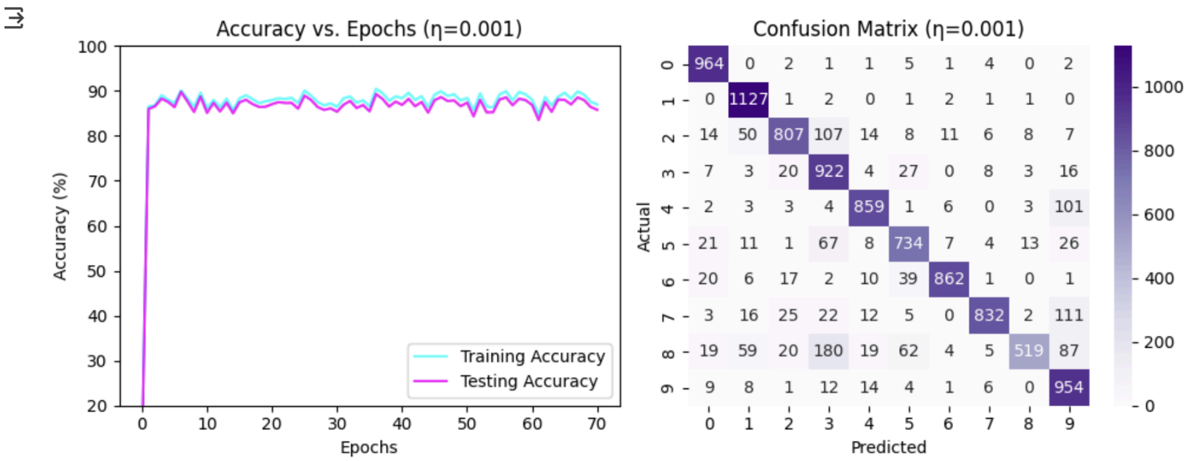
$y = a(0.5)$

From perceptron algorithm we know that

$$y = \begin{cases} 0 & wx \leq 0 \\ 1 & wx > 0 \end{cases}$$

$0.5 < 0$: $y = 1$ which is equal to target

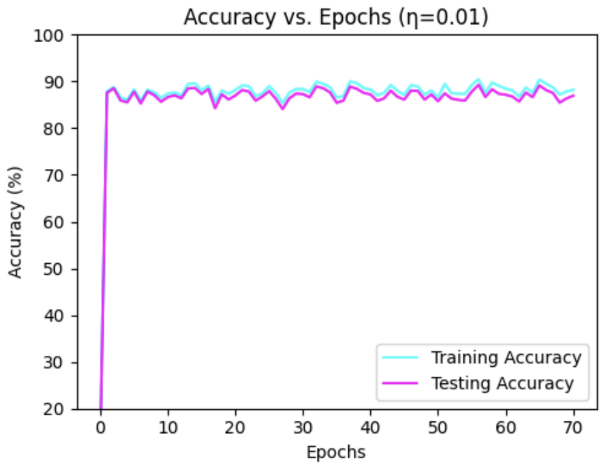
(g) Accuracy = $\dfrac{\text{No of correction predictions}}{\text{Total no of predictions}} = \dfrac{2}{2} = 1$

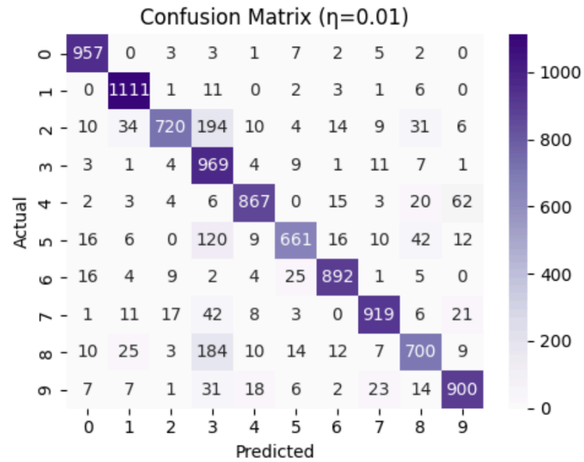


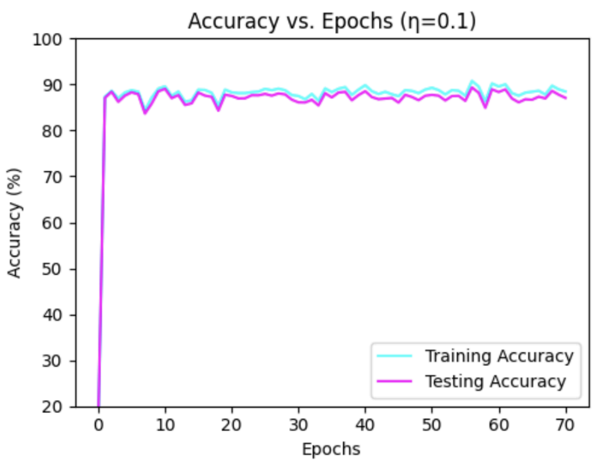
We used the MNIST dataset in this experiment to train a perceptron model using the handwritten digit classification method, where three different learning rates were implemented (η = 0.001, 0.01, and 0.1). For each learning rate, the perceptron was initialized with small random weights and trained for up to 70 epochs. Data normalization scales the pixel values to the range [0, 1]. At each epoch, the model accuracy was evaluated even with an initial evaluation before training (epoch 0).

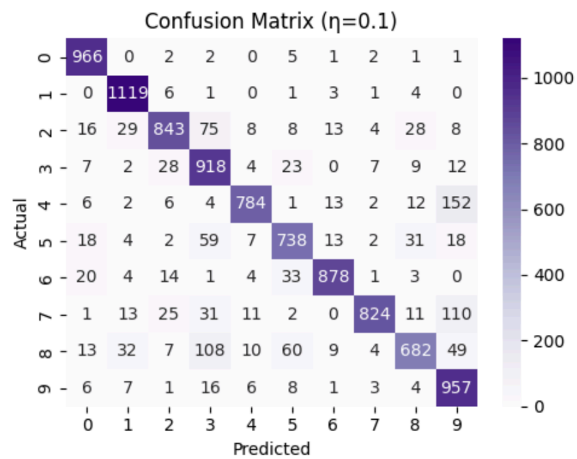
For η = 0.001, this produces accuracy that increases stably with training, indicating a stable learning rate, though relatively slow in convergence. At η = 0.01, the model accuracy increased faster, toward 80%, on the training and test sets without overt signs of overfitting.

However, when η = 0.1, oscillation of the training accuracy was realized, and as such, this learning rate may be unstable at some points, hence very high. The confusion matrix generated at the end of training for each learning rate usually shows the typical classification errors: it confuses, for instance, similar digits (e.g., 5s and 8s). Especially for the lower learning rates, there were very few off-diagonals—that would show classification.


Accuracy vs. Epochs (η=0.01)
Accuracy (%)
Epochs
Training Accuracy
Testing Accuracy


Confusion Matrix (η=0.01)
Actual
Predicted
957 0 3 3 1 7 2 5 2 0
0 1111 1 11 0 2 3 1 6 0
10 34 720 194 10 4 14 9 31 6
3 1 4 969 4 9 1 11 7 1
2 3 4 6 867 0 15 3 20 62
16 6 0 120 9 661 16 10 42 12
16 4 9 2 4 25 892 1 5 0
1 11 17 42 8 3 0 919 6 21
10 25 3 184 10 14 12 7 700 9
7 7 1 31 18 6 2 23 14 900


Accuracy vs. Epochs (η=0.1)
Accuracy (%)
Epochs
Training Accuracy
Testing Accuracy


Confusion Matrix (η=0.1)
Actual
Predicted
966 0 2 2 0 5 1 2 1 1
0 1119 6 1 0 1 3 1 4 0
16 29 843 75 8 8 13 4 28 8
7 2 28 918 4 23 0 7 9 12
6 2 6 4 784 1 13 2 12 152
18 4 2 59 7 738 13 2 31 18
20 4 14 1 4 33 878 1 3 0
1 13 25 31 11 2 0 824 11 110
13 32 7 108 10 60 9 4 682 49
6 7 1 16 6 8 1 3 4 957