CS-454 HW3 Report

Mustafa Alper Sayan

S015674

1. Introduction

In this assignment we are expected to implement a single layer perception and a multi-layer perceptron. We are excepted to train the perceptron algorithms on the MNIST dataset. MNIST dataset consists of handwritten (28*28) pixel images.

2. Implementation Details

For classifying digits, we have 10 classes K = 10. These classes are categorical therefore we need to represent the outputs using encoding. An example is as follows:

If, K = 10 And K_{actual} = 5
Then array representing this class should be
r_classes = [0,0,0,0,0,1,0,0,0,0]

a. Single Layer perception

Initialize the weights randomly using shapes [K=10, Dimensions = 784].
Then the following equations should be used

$$X = \{x^t, r^t\}_t$$

Where x^t = input array shaped with dimensions in the case of MNIST 784
Where r^t = Output class in the case of MNIST a value from 0 – 9

$$o = log \frac{p(x \mid C_i)}{p(x \mid C_k)} = w_i^T x + w_{i0}^o$$

Where W_i = Weight matrix of the class i
Where W_{io} = Bias of the class i
Where x = input

$$y = P(C_i \mid x) = \frac{e^{o_i}}{\sum_{j=1}^{K} e^{o_j}} \ , i = 1, \dots, K$$

$$I(\{w_i, w_{i0}\}_i \,|X) = \prod_t \prod_i y_i^{t\,(r_i^t)}$$

$$E(\{w_i, w_{i0}\}_i \,|X) = -\sum_t r_i^t \log y_i^t$$

$$\Delta w_j = \alpha \sum_t (r_j^t - y_j^t) x^t$$

$$\Delta w_{j0} = \alpha \sum_t (r_j^t - y_j^t)$$

Where α = Learning rate
Where Δw_j, Δw_{j0} = Updated weights

b. Multi-Layer Perceptron

Initialize the weights randomly using shapes for input layer weights
w = [H = (25, 50, 75), Dimensions = 784]
for output layer weights
v = [K=10, H= (25, 50, 75)]

$$o_i^t = \sum_{h=1}^{H} v_{ih}\, w_h^t + v_{i0}$$

Where v = output layer weights
Where w = input layer weights
Where h = hidden layer neurons

$$y_i^t = \frac{e^{o_i^t}}{\sum_k e^{o_k^t}}$$

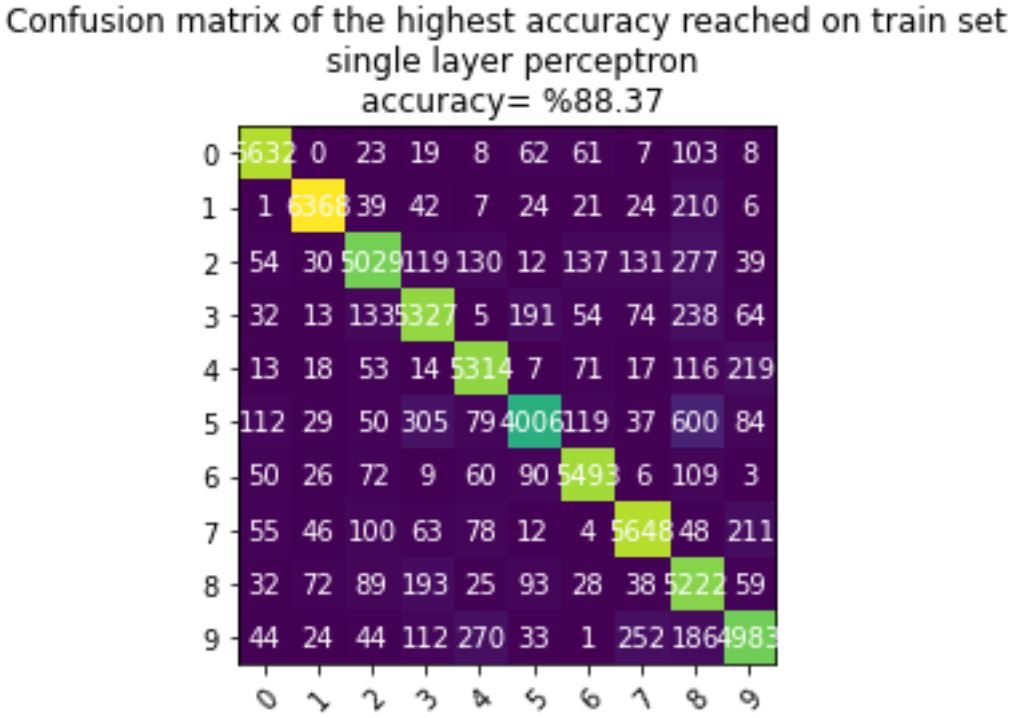
Where k = classes
Where t = sample number

$$E(W, v \,|X) = -\sum_t \sum_i r_i^t \log y_i^t$$

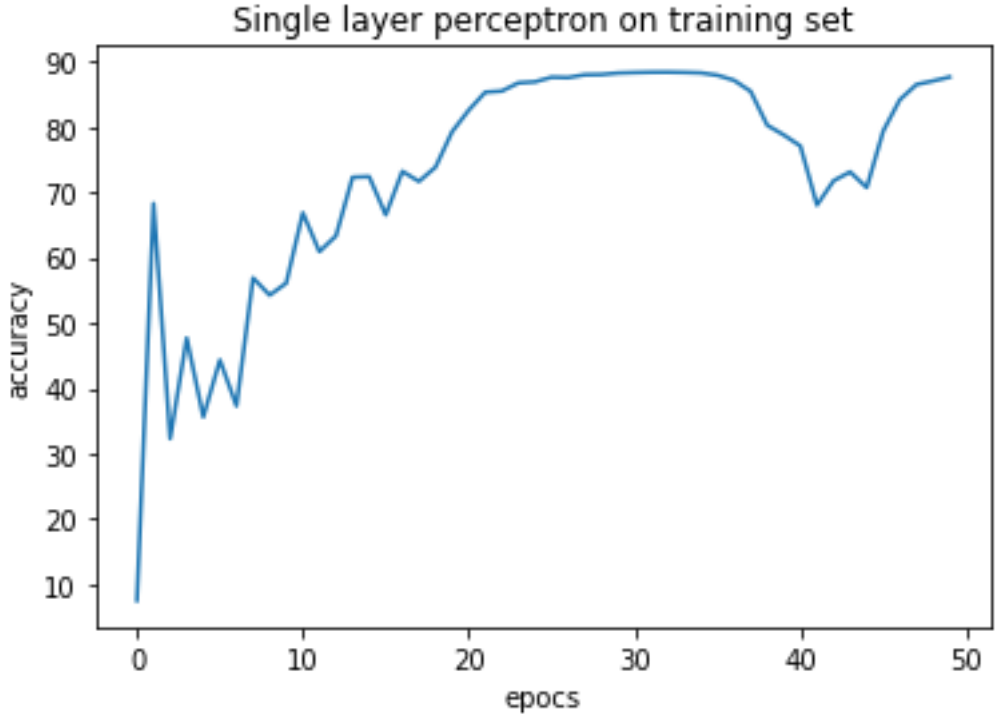
$$\Delta v_{ih} = \alpha \sum_t (r_i^t - y_i^t) z_h^t$$

$$\Delta w_{hj} = \alpha \sum_t \left[\sum_i (r_i^t - y_i^t) v_{ih} \right] z_h^t (1 - z_h^t) x_j^t$$

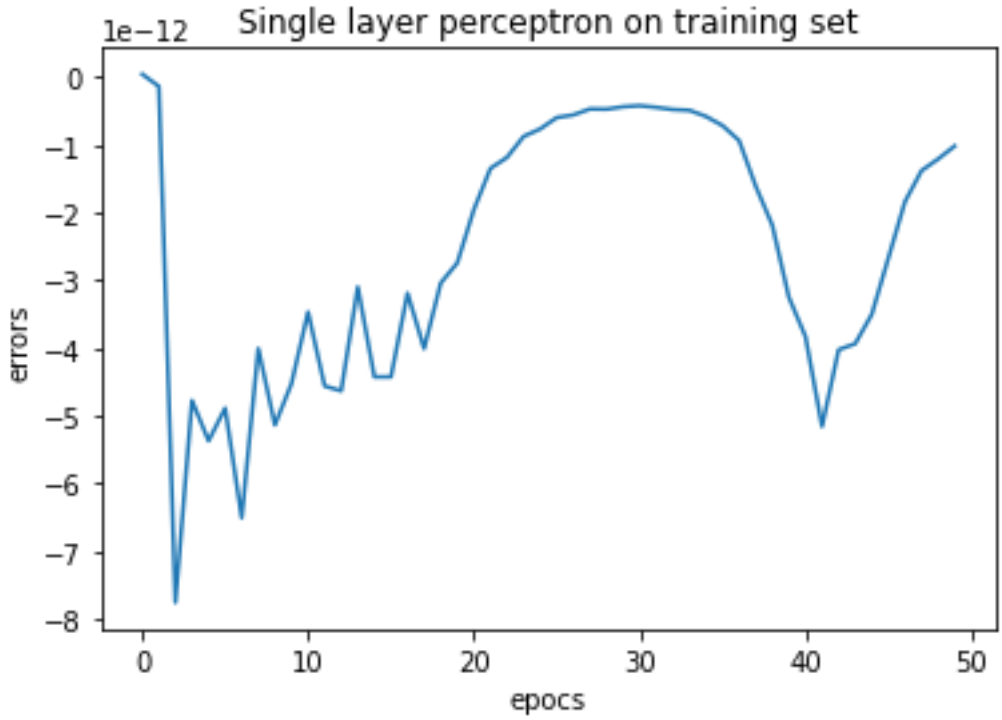
3. Results


Confusion matrix of the highest accuracy reached on train set
single layer perceptron
accuracy= %88.37
0 5632 0 23 19 8 62 61 7 103 8
1 1 6368 39 42 7 24 21 24 210 6
2 54 30 5029 119 130 12 137 131 277 39
3 32 13 133 5327 5 191 54 74 238 64
4 13 18 53 14 5314 7 71 17 116 219
5 112 29 50 305 79 4006 119 37 600 84
6 50 26 72 9 60 90 5493 6 109 3
7 55 46 100 63 78 12 4 5648 48 211
8 32 72 89 193 25 93 28 38 5222 59
9 44 24 44 112 270 33 1 252 186 4983
0 1 2 3 4 5 6 7 8 9

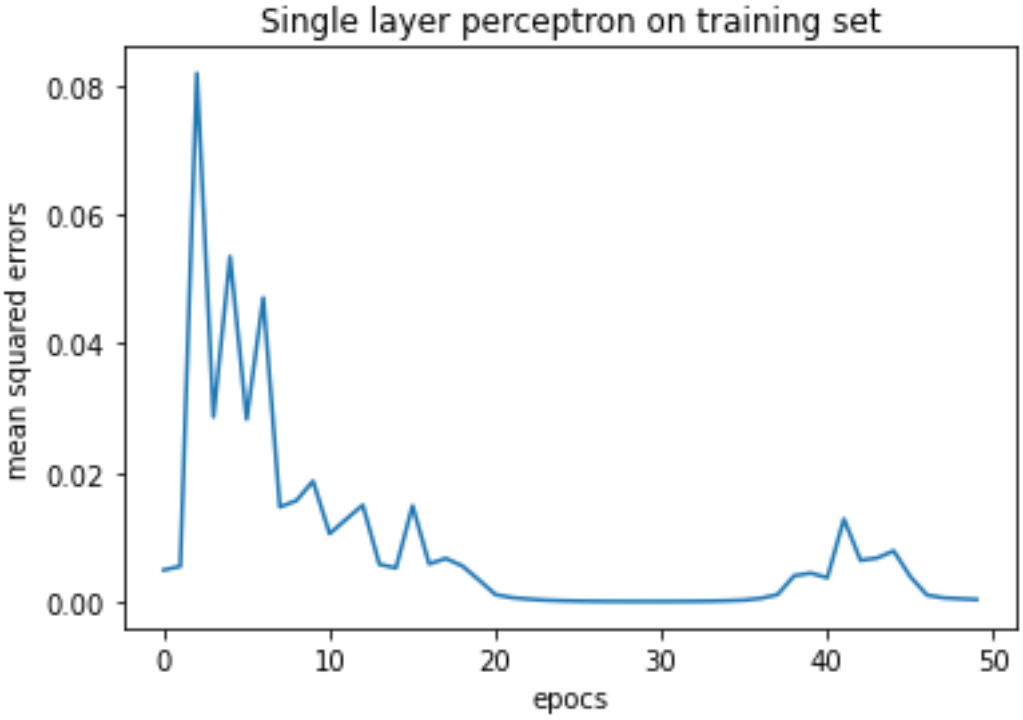
a.


Single layer perceptron on training set
accuracy
epocs
90
80
70
60
50
40
30
20
10
0
10
20
30
40
50

b.

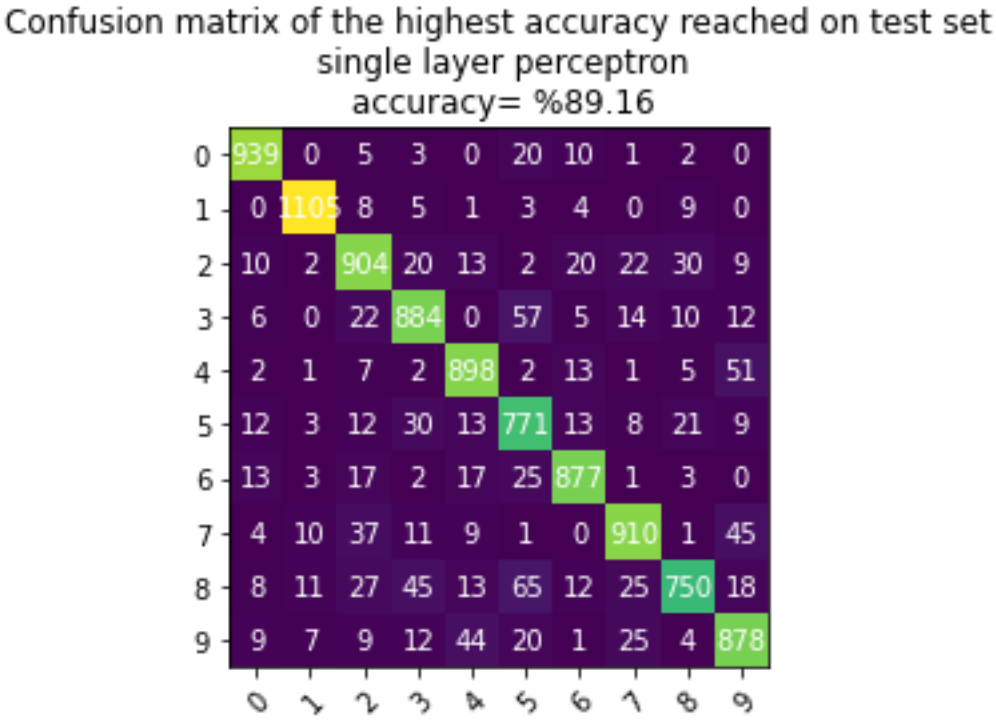

Single layer perceptron on training set
1e−12
0
−1
−2
−3
−4
−5
−6
−7
−8
errors
0
10
20
30
40
50
epocs

c.

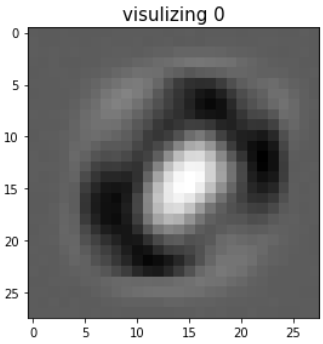

Single layer perceptron on training set
0.08
0.06
0.04
0.02
0.00
mean squared errors
0
10
20
30
40
50
epocs

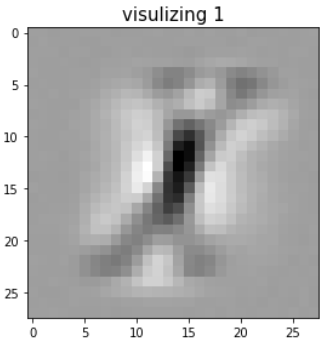
d.

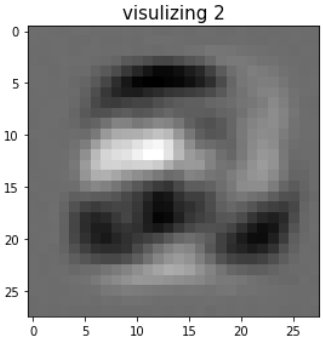
e.

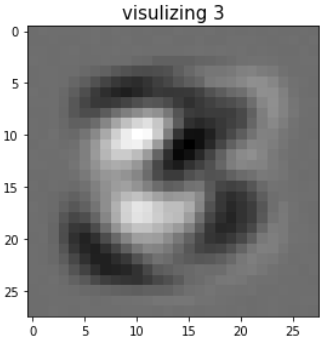

Confusion matrix of the highest accuracy reached on test set
single layer perceptron
accuracy= %89.16
0 939 0 5 3 0 20 10 1 2 0
1 0 1105 8 5 1 3 4 0 9 0
2 10 2 904 20 13 2 20 22 30 9
3 6 0 22 884 0 57 5 14 10 12
4 2 1 7 2 898 2 13 1 5 51
5 12 3 12 30 13 771 13 8 21 9
6 13 3 17 2 17 25 877 1 3 0
7 4 10 37 11 9 1 0 910 1 45
8 8 11 27 45 13 65 12 25 750 18
9 9 7 9 12 44 20 1 25 4 878
0 1 2 3 4 5 6 7 8 9

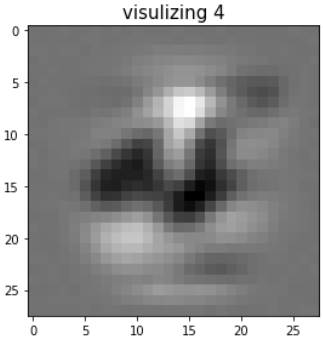
f.

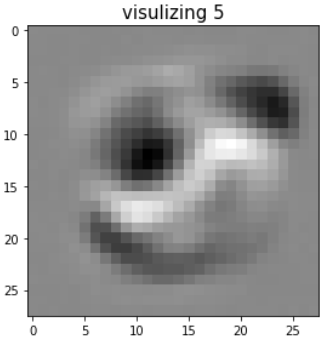

visulizing 0

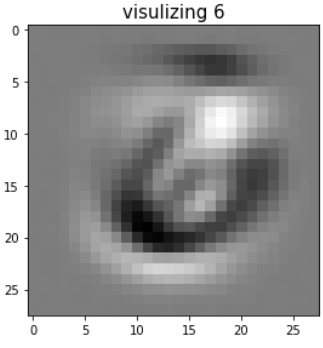

visulizing 1

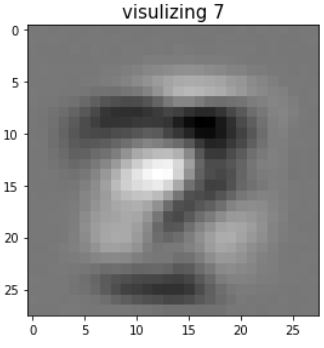

visulizing 2

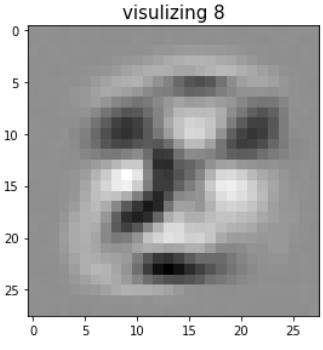

visulizing 3

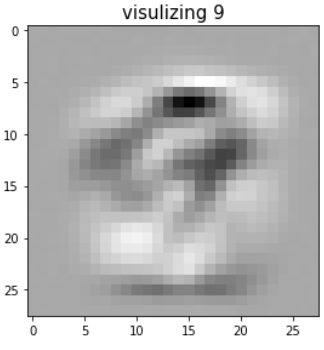

visulizing 4


visulizing 5

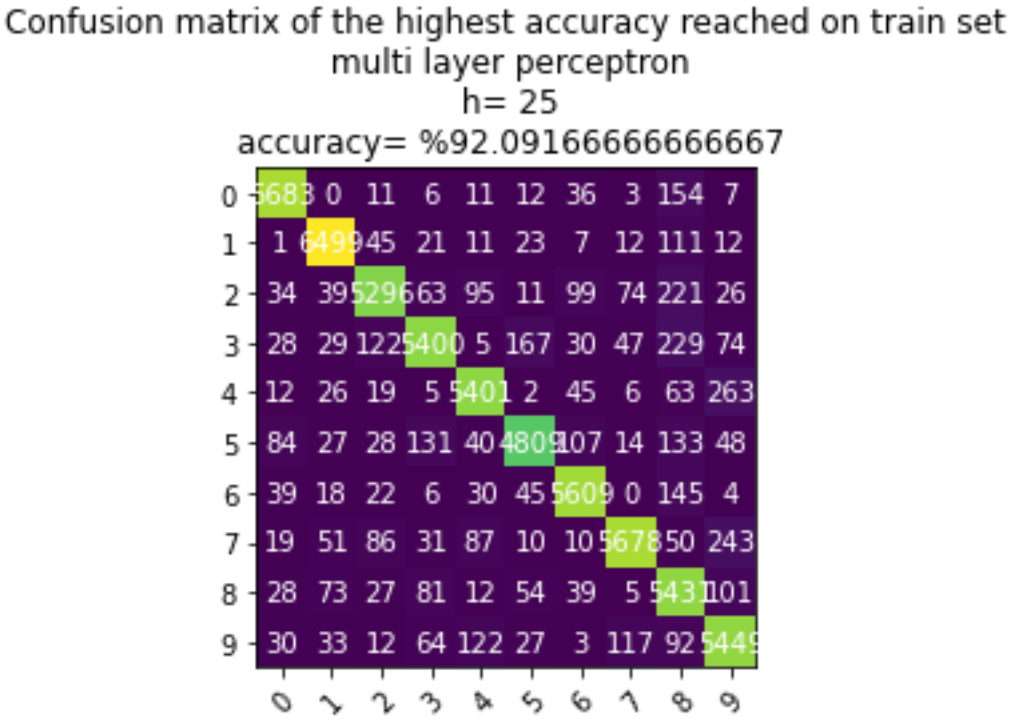

visulizing 6


visulizing 7

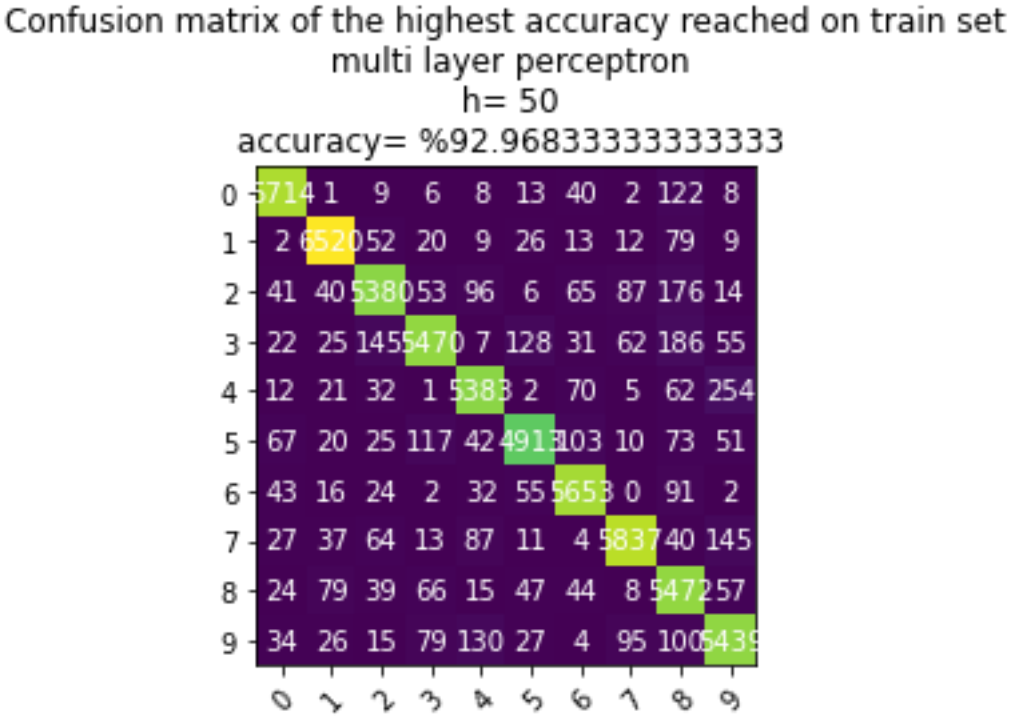

visulizing 8


visulizing 9

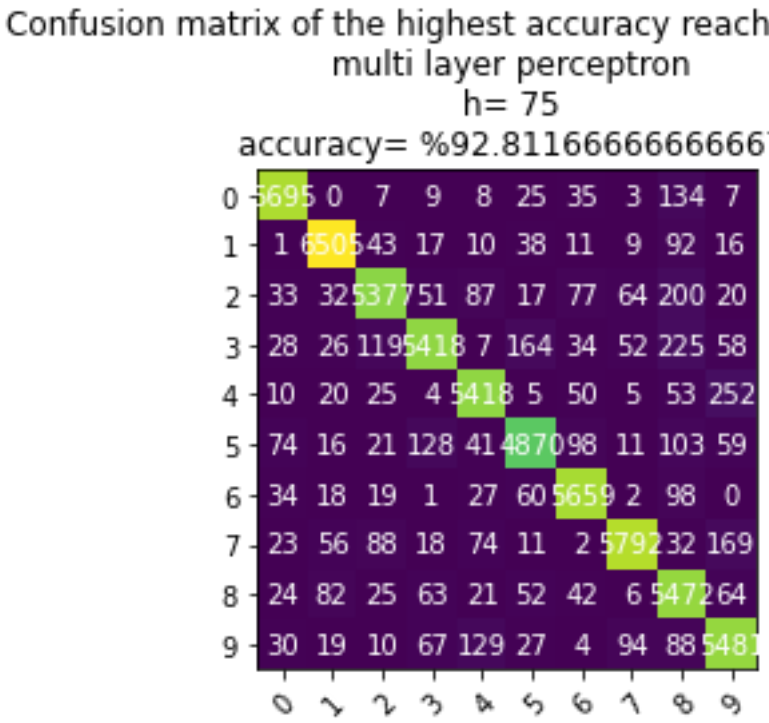
g.


Confusion matrix of the highest accuracy reached on train set
multi layer perceptron
h= 25
accuracy= %92.09166666666667

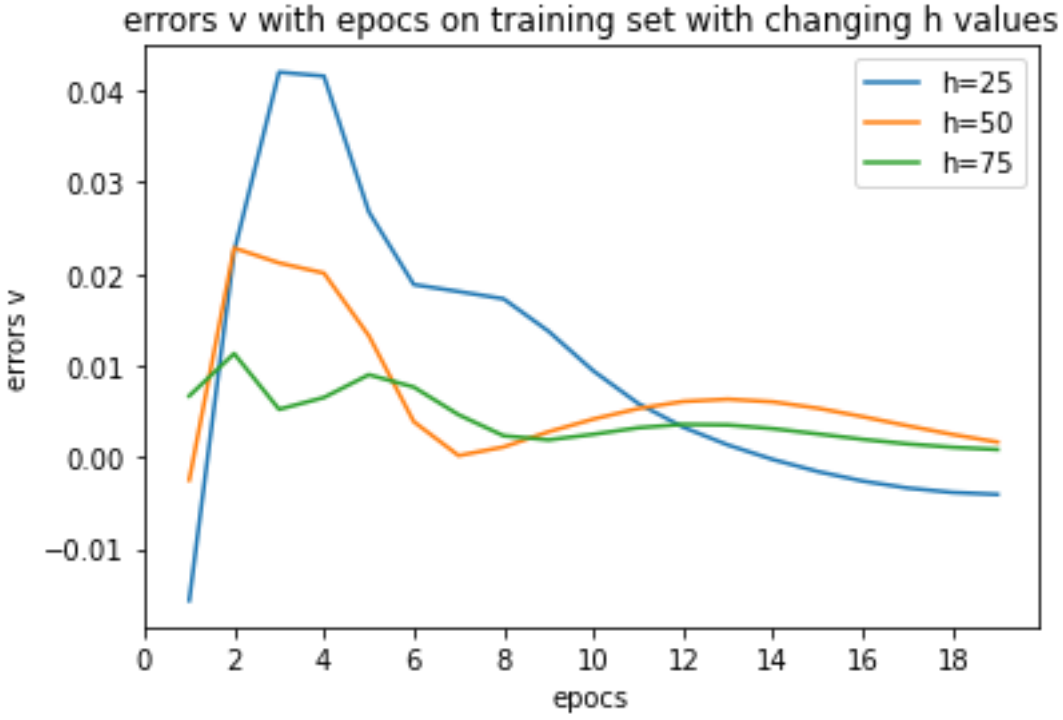
h.

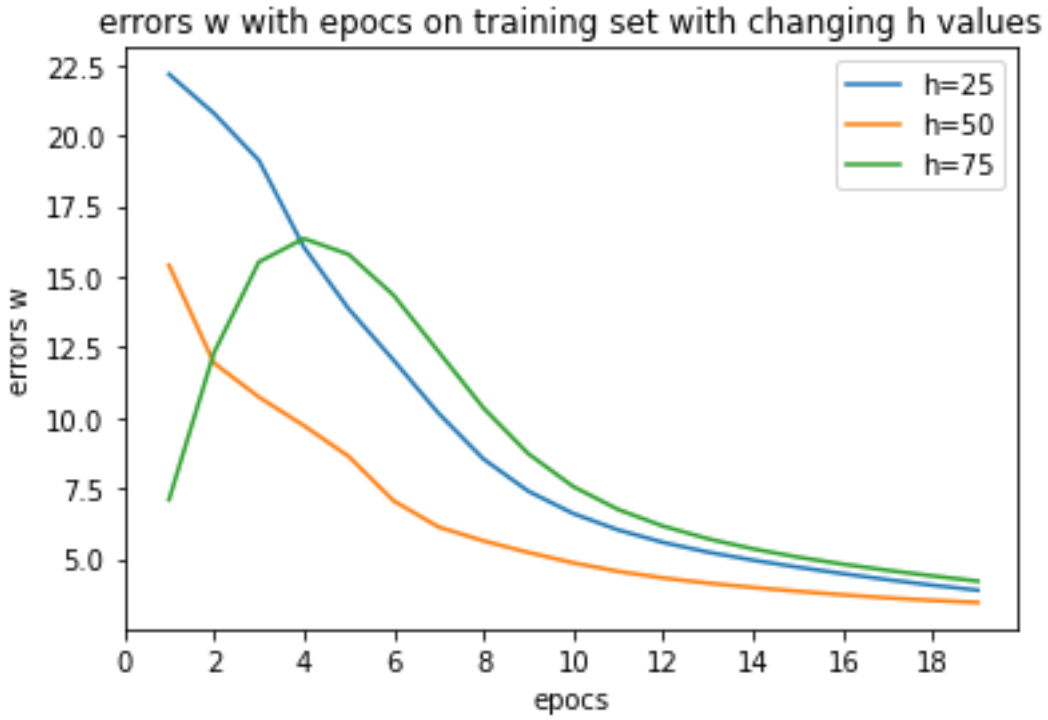

Confusion matrix of the highest accuracy reached on train set
multi layer perceptron
h= 50
accuracy= %92.96833333333333

i.

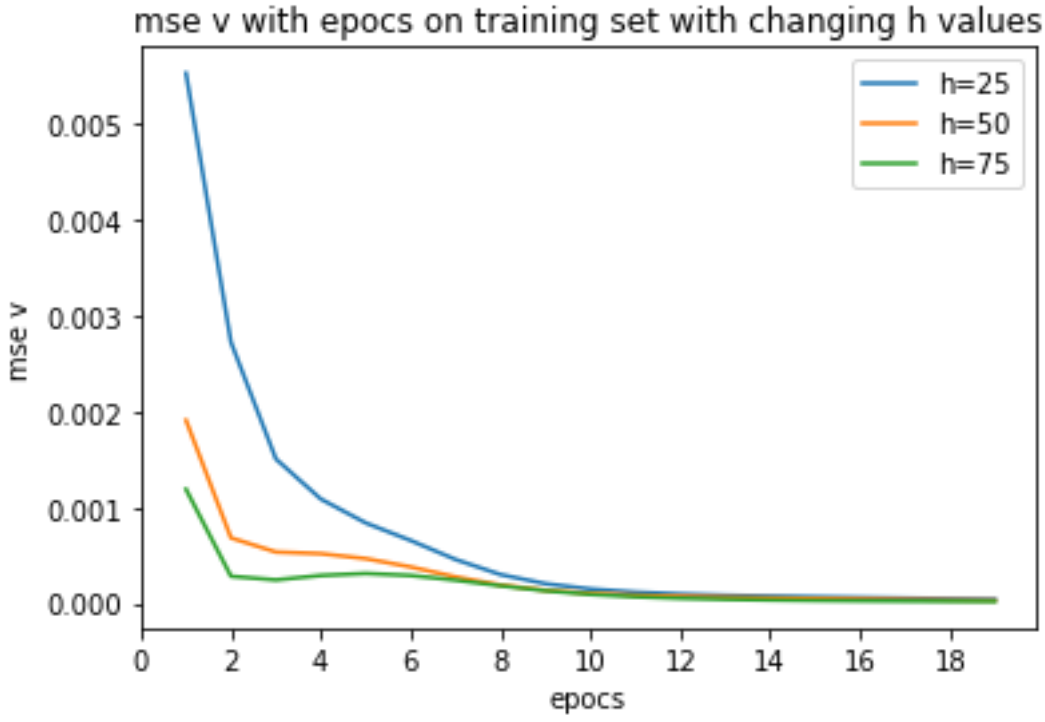

Confusion matrix of the highest accuracy reached on train set
multi layer perceptron
h= 75
accuracy= %92.81166666666667
0 5695 0 7 9 8 25 35 3 134 7
1 1 6506 43 17 10 38 11 9 92 16
2 33 32 5377 51 87 17 77 64 200 20
3 28 26 119 5418 7 164 34 52 225 58
4 10 20 25 4 5418 5 50 5 53 252
5 74 16 21 128 41 4870 98 11 103 59
6 34 18 19 1 27 60 5659 2 98 0
7 23 56 88 18 74 11 2 5792 32 169
8 24 82 25 63 21 52 42 6 5472 64
9 30 19 10 67 129 27 4 94 88 5481
0 1 2 3 4 5 6 7 8 9

j.

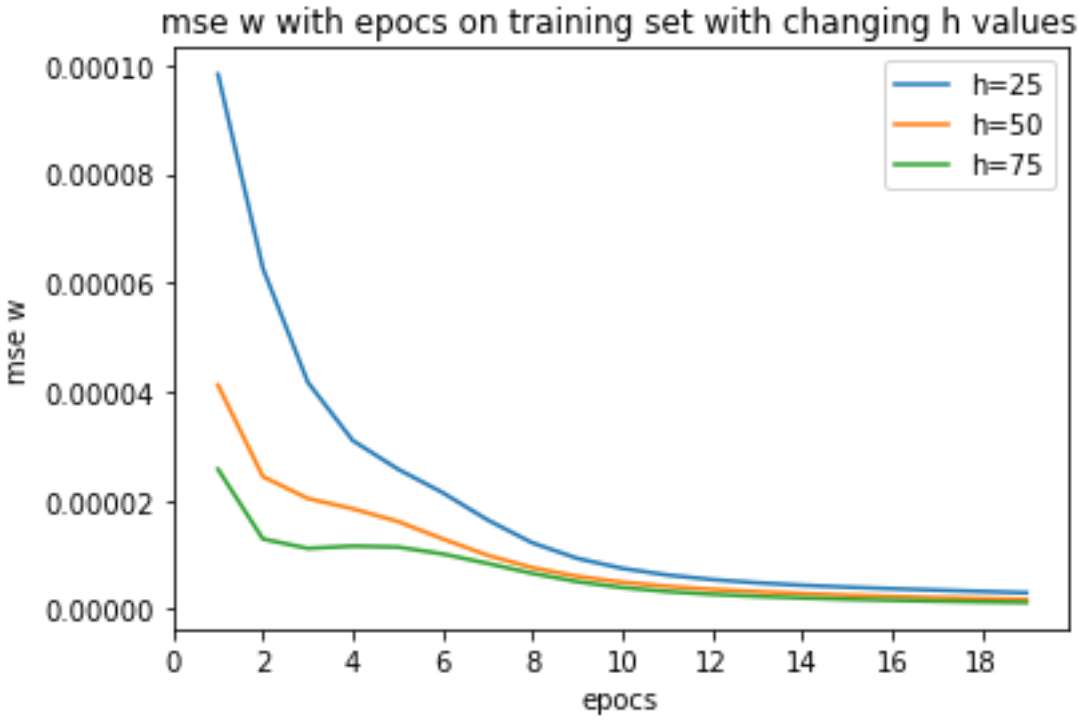

errors v with epocs on training set with changing h values
h=25
h=50
h=75
errors v
epocs
0.04
0.03
0.02
0.01
0.00
−0.01
0
2
4
6
8
10
12
14
16
18


errors w with epocs on training set with changing h values
h=25
h=50
h=75
errors w
epocs
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
0
2
4
6
8
10
12
14
16
18

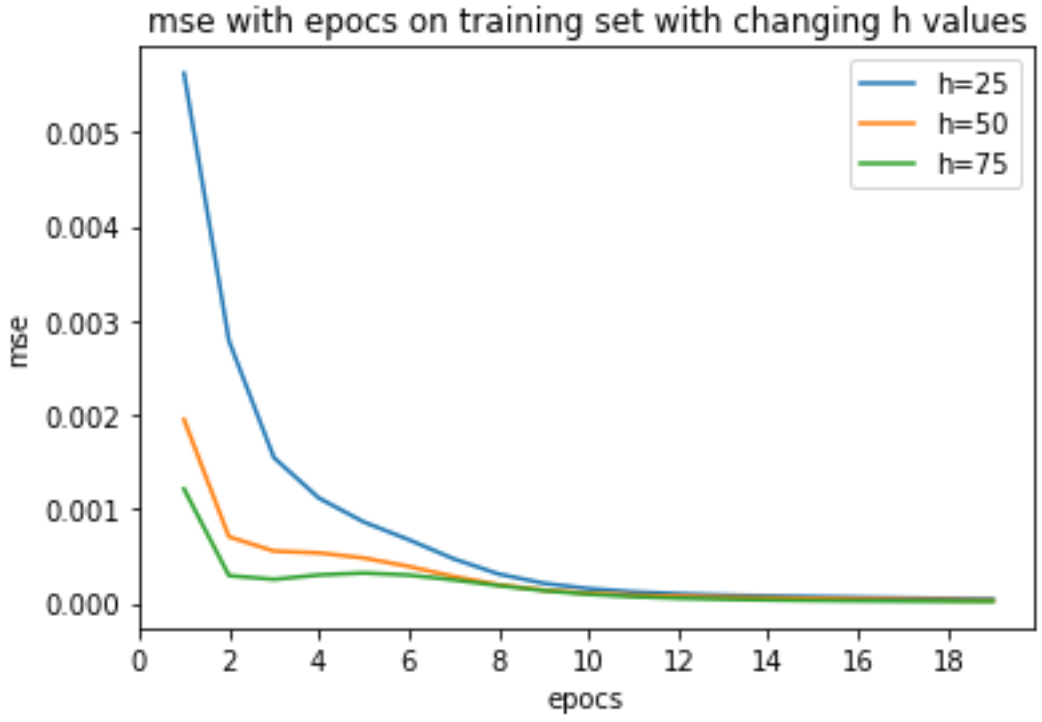
k.


mse v with epocs on training set with changing h values
h=25
h=50
h=75
mse v
epocs
0.005
0.004
0.003
0.002
0.001
0.000
0
2
4
6
8
10
12
14
16
18

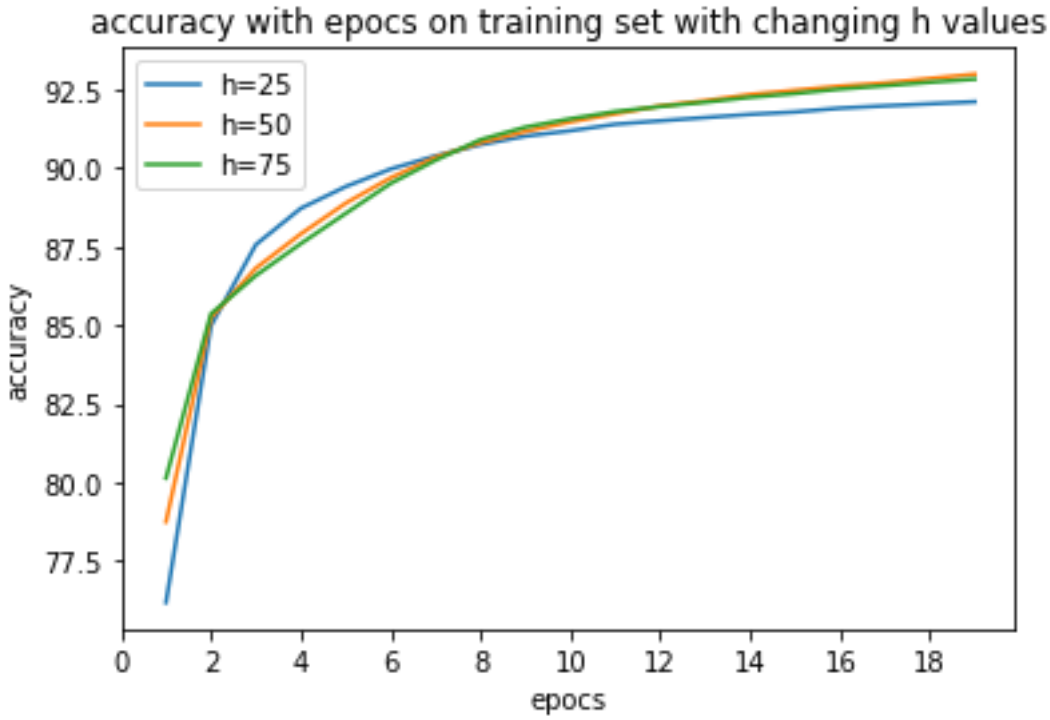
l.


mse w with epocs on training set with changing h values
h=25
h=50
h=75
mse w
epocs
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
0
2
4
6
8
10
12
14
16
18

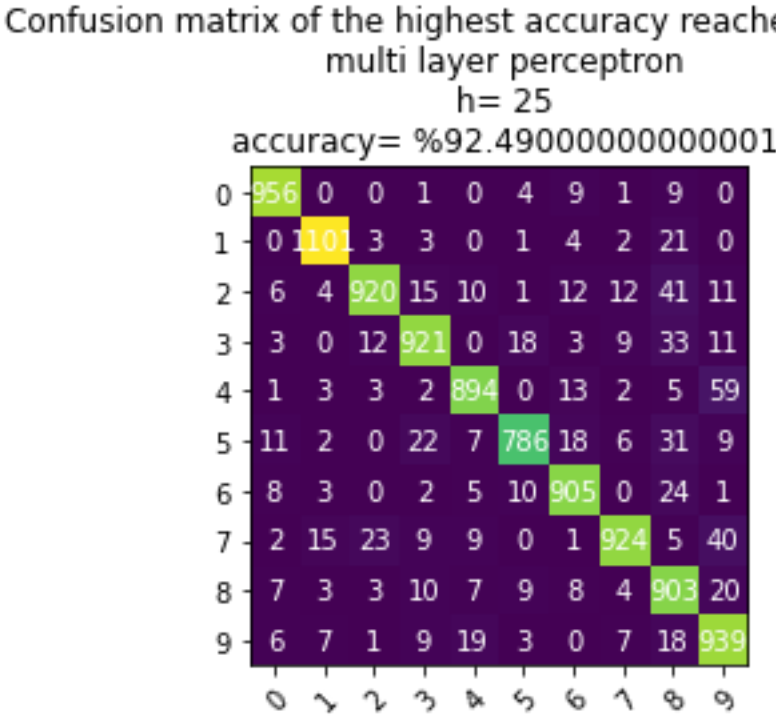
m.


mse with epocs on training set with changing h values
h=25
h=50
h=75
mse
epocs
0.005
0.004
0.003
0.002
0.001
0.000
0
2
4
6
8
10
12
14
16
18

n.

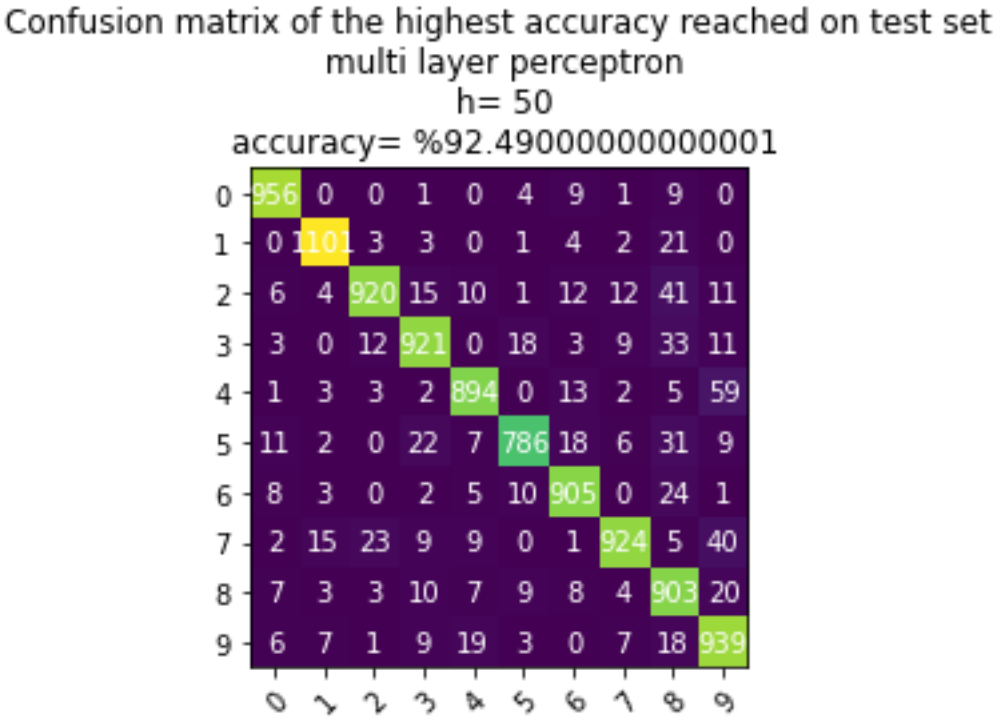

accuracy with epocs on training set with changing h values
h=25
h=50
h=75
accuracy
epocs
92.5
90.0
87.5
85.0
82.5
80.0
77.5
0
2
4
6
8
10
12
14
16
18

o.

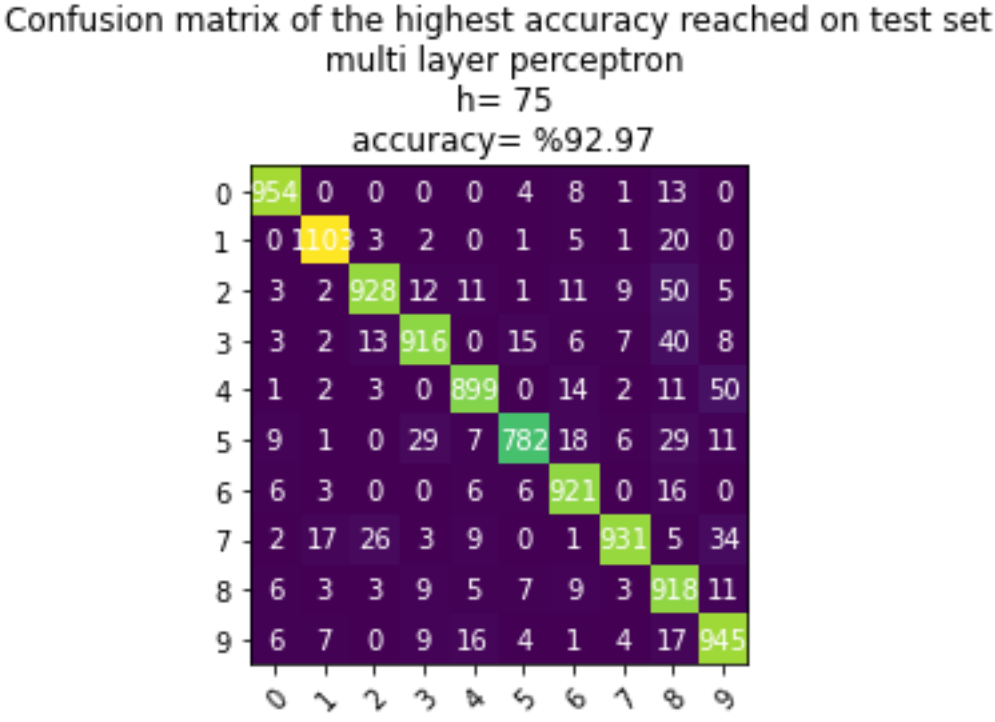

Confusion matrix of the highest accuracy reached on test set
multi layer perceptron
h= 25
accuracy= %92.49000000000001
0 956 0 0 1 0 4 9 1 9 0
1 0 1101 3 3 0 1 4 2 21 0
2 6 4 920 15 10 1 12 12 41 11
3 3 0 12 921 0 18 3 9 33 11
4 1 3 3 2 894 0 13 2 5 59
5 11 2 0 22 7 786 18 6 31 9
6 8 3 0 2 5 10 905 0 24 1
7 2 15 23 9 9 0 1 924 5 40
8 7 3 3 10 7 9 8 4 903 20
9 6 7 1 9 19 3 0 7 18 939
0 1 2 3 4 5 6 7 8 9

p.

q.


Confusion matrix of the highest accuracy reached on test set
multi layer perceptron
h= 50
accuracy= %92.49000000000001
0 956 0 0 1 0 4 9 1 9 0
1 0 1101 3 3 0 1 4 2 21 0
2 6 4 920 15 10 1 12 12 41 11
3 3 0 12 921 0 18 3 9 33 11
4 1 3 3 2 894 0 13 2 5 59
5 11 2 0 22 7 786 18 6 31 9
6 8 3 0 2 5 10 905 0 24 1
7 2 15 23 9 9 0 1 924 5 40
8 7 3 3 10 7 9 8 4 903 20
9 6 7 1 9 19 3 0 7 18 939
0 1 2 3 4 5 6 7 8 9

r.


Confusion matrix of the highest accuracy reached on test set
multi layer perceptron
h= 75
accuracy= %92.97
0 954 0 0 0 0 4 8 1 13 0
1 0 1103 3 2 0 1 5 1 20 0
2 3 2 928 12 11 1 11 9 50 5
3 3 2 13 916 0 15 6 7 40 8
4 1 2 3 0 899 0 14 2 11 50
5 9 1 0 29 7 782 18 6 29 11
6 6 3 0 0 6 6 921 0 16 0
7 2 17 26 3 9 0 1 931 5 34
8 6 3 3 9 5 7 9 3 918 11
9 6 7 0 9 16 4 1 4 17 945
0 1 2 3 4 5 6 7 8 9