

SHIVAM KHARE
201505547
ASSIGNMENT 3

AIM:

The aim of this assignment is to experiment with Multilayer Feedforward Neural Network (MLFNN) with Backpropagation.

Question 1:

Implementation of Simple Multilayer Feedforward Neural Network with backpropagation.

Implementation details:

A 3-Layer Neural Network is designed with NN-topology [784,95,10]. The model was trained with data of 60000 images taken from MNIST database. An accuracy of 95.89% was reached while labelling 10000 test dataset taken from the same website.

Features:

The MNIST dataset provides 60000 samples each sample represents an image on 28 X 28 pixels field. This was converted to 784 input parameters to feed into NN.

Confusion Matrix

960	1	0	0	1	0	3	1	3	1
1	1019	2	1	0	0	2	0	10	0
11	4	963	9	2	0	1	3	9	0
1	1	4	979	0	13	0	8	12	2
5	2	0	0	955	0	7	0	6	17
6	2	0	11	0	870	6	2	8	2
11	2	0	1	2	7	940	0	3	0
3	11	13	6	2	0	2	960	2	15
8	3	3	9	4	5	6	2	920	1
10	7	0	14	13	3	1	3	9	920

accuracy: 95.89

sensitivity 0.954944964965

specification 0.994796925048

precision 0.965964964965

5_fold Cross validation:

Fold1

```
[[1178 1 4 6 3 2 14 0 18 0]
 [0 1306 6 5 2 6 6 4 16 1]
 [ 23 19 965 15 24 3 64 28 25 9]
 [17 24 49 1017 6 37 30 14 32 28]
 [1 15 15 3 957 8 30 5 10 82]
 [40 34 6 79 29 834 61 13 30 17]
 [10 8 8 0 7 18 1131 2 8 0]
 [13 26 12 5 27 0 4 1102 7 53]
 [25 26 25 33 8 33 17 4 932 43]
 [20 17 11 23 47 10 3 53 18 935]]
```

Accuracy is 86.6083333333

Fold2:

```
[[1149 0 5 5 5 2 17 1 21 1]
 [ 1 1319 14 5 0 7 1 1 16 2]
 [ 22 17 991 15 36 5 49 27 29 7]
 [ 26 16 35 982 5 33 23 20 31 27]
 [ 6 12 9 4 1046 3 30 3 16 80]
 [ 39 39 9 67 31 785 42 9 23 18]
 [ 8 11 10 1 6 15 1089 2 3 0]
 [ 15 19 13 8 16 2 3 1110 4 49]
 [ 15 45 18 49 10 37 16 4 936 32]
 [ 21 16 13 17 46 15 4 39 14 1030]]
Accuracy is 86.675
```

Fold3:

```
[[1121 0 6 3 4 1 22 1 10 3]
 [ 1 1258 8 2 1 7 9 0 16 1]
 [ 20 21 979 10 26 2 53 28 38 11]
 [ 13 22 43 1017 3 39 27 17 36 31]
 [ 2 12 6 5 1004 6 39 3 14 86]
 [ 52 33 7 72 29 748 53 6 21 26]
 [ 4 6 12 0 4 15 1131 2 6 2]
 [ 3 24 19 9 21 1 5 1120 5 58]
 [ 19 38 24 41 18 29 23 2 990 38]
 [ 19 23 15 25 30 14 2 52 21 996]]
Accuracy is 86.345
```

Fold4:

```
[[1133 0 4 4 1 2 15 1 17 2]
 [ 0 1349 12 1 1 6 7 3 10 4]
 [ 24 23 981 16 21 4 50 30 46 13]
 [ 16 18 45 970 3 43 15 24 31 24]
 [ 2 8 8 4 958 1 30 6 18 86]
 [ 43 52 8 72 35 799 47 12 34 21]
 [ 11 15 13 2 5 18 1128 1 2 0]
 [ 17 25 23 5 38 2 1 1075 14 60]
 [ 20 32 22 51 12 22 21 6 929 29]
 [ 20 16 16 15 34 9 2 42 19 1015]]
Accuracy is 86.123
```

Fold5 :

```
[[1093 1 3 4 4 0 19 0 15 2]
 [ 0 1276 8 1 5 6 8 3 20 1]
 [ 10 30 986 16 25 1 51 28 36 6]
 [ 12 15 43 1035 7 39 9 12 37 33]
 [ 3 8 9 4 1031 6 36 6 17 89]
 [ 40 24 9 75 31 760 56 11 26 14]
 [ 12 8 14 1 7 22 1129 4 7 0]
 [ 12 23 24 6 24 2 4 1120 6 31]
 [ 29 52 33 38 7 36 31 6 910 35]
 [ 13 19 7 18 50 10 1 52 15 1027]]
Accuracy is 86.177788
```

Average accuracy: 86.1366666667

Average precision: 0.863535613531

Average sensitivity: 0.861068073241

Average specificity: 0.984833669018

Question 2:
1NN Classifier.
Accuracy with 1NN Classifier:- 82.08%

Confusion Matrix:

Question 3:
Adding Noise.

Comfusion Matrix obtained:

```
[ 939.  1.  0.  3.  1.  4. 13.  1. 18.  0.]
[  0. 1093. 20.  1.  0.  8.  7.  0.  5.  1.]
[ 18.  23. 909. 15.  5.  1. 38.  8.  8.  7.]
[ 28.  4.  47. 800.  0. 26. 15.  3. 61. 26.]
[  4.  5.  4.  2. 764.  7. 57.  1. 14. 124.]
[ 56.  6.  1. 100.  8. 628. 54.  1. 10. 28.]
[ 30.  4.  5.  1.  3.  9. 902.  0.  4.  0.]
[ 13. 39. 49. 11.  9.  7.  5. 788. 15. 92.]
[ 35.  7. 32. 101.  2. 27. 33.  1. 693. 43.]
[ 18.  8.  8. 12. 34.  1. 21.  8. 21. 878.]
```

sensitivity: 0.943943943944
specification: 0.994855778064
precision: 0.943943943944

Weight Decay:
Weight-decay formula used

Confusion Matrix:

968	0	0	2	1	0	7	0	1	1
0	1118	1	6	1	0	5	1	2	1
13	1	947	43	3	0	8	6	11	0
1	0	3	992	0	0	0	6	5	3
3	0	1	2	906	0	16	1	2	51
11	1	1	36	1	793	20	2	16	11
11	3	1	2	2	1	935	0	3	0
3	11	16	20	4	0	1	926	1	46
6	5	3	44	4	1	9	4	889	9
7	5	1	17	11	0	3	2	1	962