|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sl. no | Number of hidden layers | Number of units in each hidden layer | Loss function | Activation  Function  (Input) | Optimizer | Learning  Rate | Momentum | Input Scaling | Accuracy | Convergence  speed |
| 1 | 1 | 16 | Classification cross entropy | relu | adam | N/A | N/A | N/A | 98.6656 | 11.404842 |
| 2 | 1 | 16 | Classification cross entropy | relu | sgd | No | No | Yes | 99.0842 | 10.313619 |
| 3 | 1 | 32 | Classification cross entropy | relu | adam | N/A | N/A | N/A | 97.5406 | 8.110255 |
| 4 | 2 | 32,16 | Classification cross entropy | relu | adam | N/A | N/A | Yes | 99.5290 | 12.079434 |
| 5 | 1 | 16 | Classification cross entropy | relu | sgd | 0.01 | No | No | 97.4621 | 10.298023 |
| 6 | 1 | 16 | Classification cross entropy | relu | sgd | 0.01 | 0.5 | No | 93.2496 | 6.422601 |
| 7 | 1 | 16 | Classification cross entropy | tanh | adam | N/A | N/A | N/A | 98.1947 | 16.533055 |
| 8 | 1 | 16 | Classification cross entropy | tanh | sgd | No | No | Yes | 99.0319 | 11.282505 |
| 9 | 1 | 32 | Classification cross entropy | tanh | adam | N/A | N/A | N/A | 99.1627 | 13.892142 |
| 10 | 2 | 32,16 | Classification cross entropy | tanh | adam | N/A | N/A | Yes | 99.3459 | 10.188604 |
| 11 | 1 | 16 | Classification cross entropy | tanh | sgd | 0.01 | No | No | 97.1481 | 20.471196 |
| 12 | 1 | 16 | Classification cross entropy | tanh | sgd | 0.01 | 0.5 | No | 95.0288 | 14.157785 |
| 13 | 1 | 16 | mean squared error | relu | adam | N/A | N/A | N/A | 98.9011 | 17.725555 |
| 14 | 1 | 16 | mean squared error | relu | sgd | No | No | Yes | 99.2151 | 12.235702 |
| 15 | 1 | 32 | mean squared error | relu | adam | N/A | N/A | N/A | 98.6656 | 13.595224 |
| 16 | 2 | 32,16 | mean squared error | relu | adam | N/A | N/A | Yes | 99.2936 | 12.079404 |
| 17 | 1 | 16 | mean squared error | relu | sgd | 0.01 | No | No | 95.3951 | 20.25 |
| 18 | 1 | 16 | mean squared error | relu | sgd | 0.01 | 0.5 | No | 97.6190 | 21.67 |

1. Best Model learned has ReLU as activation function with two layers, classification cross entropy, optimizer as adam with accuracy of 99.5290.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sl. no | Number of hidden layers | Number of units in each hidden layer | Loss function | Activation  Function  (Input) | Optimizer | Learning  Rate | Momentum | Input Scaling | Accuracy | Convergence  Speed |
| 4 | 2 | 32,16 | Classification cross entropy | relu | adam | N/A | N/A | Yes | 99.5290 | 12.079434 |

**Confusion Matrix** and **Classification report** for the above model for **Training data**-

confusion matrix for training data with two layer model

[[375 0 0 0 0 0 0 0 0 0]

[ 0 389 0 0 0 0 0 0 0 0]

[ 0 0 380 0 0 0 0 0 0 0]

[ 0 0 0 388 0 1 0 0 0 0]

[ 0 0 0 0 386 0 1 0 0 0]

[ 0 0 0 1 0 374 0 1 0 0]

[ 0 0 0 0 0 0 377 0 0 0]

[ 0 0 0 1 0 1 0 384 0 1]

[ 0 3 1 1 0 0 0 0 375 0]

[ 0 0 1 2 1 1 0 0 1 376]]

classfication report for training data with two layer model

precision recall f1-score support

0 1.00 1.00 1.00 375

1 0.99 1.00 1.00 389

2 0.99 1.00 1.00 380

3 0.99 1.00 0.99 389

4 1.00 1.00 1.00 387

5 0.99 0.99 0.99 376

6 1.00 1.00 1.00 377

7 1.00 0.99 0.99 387

8 1.00 0.99 0.99 380

9 1.00 0.98 0.99 382

avg / total 1.00 1.00 1.00 3822

**Confusion Matrix** and **Classification report** for the above model for **Test data**-

confusion matrix for test data with two layer model

[[177 0 0 0 0 0 0 0 0 0]

[ 0 178 0 0 0 0 3 0 1 0]

[ 0 2 168 1 0 0 3 1 2 0]

[ 0 1 4 174 0 1 0 0 1 2]

[ 0 1 0 0 176 0 0 0 3 1]

[ 0 1 0 2 0 177 0 0 0 2]

[ 1 0 0 0 2 0 177 0 1 0]

[ 0 0 0 0 0 9 0 165 0 5]

[ 0 8 0 1 0 3 1 0 154 7]

[ 0 0 1 3 3 2 0 0 5 166]]

classfication report for test data with two layer model

precision recall f1-score support

0 0.99 1.00 1.00 177

1 0.93 0.98 0.95 182

2 0.97 0.95 0.96 177

3 0.96 0.95 0.96 183

4 0.97 0.97 0.97 181

5 0.92 0.97 0.95 182

6 0.96 0.98 0.97 181

7 0.99 0.92 0.96 179

8 0.92 0.89 0.90 174

9 0.91 0.92 0.91 180

avg / total 0.95 0.95 0.95 1796

The best learned model for ReLU with mean-squared-error function has the following results-

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sl. no | Number of hidden layers | Number of units in each hidden layer | Loss function | Activation  Function  (Input) | Optimizer | Learning  Rate | Momentum | Input Scaling | Accuracy | Convergence  Speed |
| 16 | 2 | 32,16 | mean squared error | relu | adam | N/A | N/A | Yes | 99.2936 | 12.079404 |

**Different results for this model-**

confusion matrix for training data with two layer model

[[375 0 0 0 0 0 0 0 0 0]

[ 0 379 0 0 3 3 2 2 0 0]

[ 0 0 378 1 0 0 0 0 1 0]

[ 0 2 0 384 0 1 0 0 0 2]

[ 0 0 0 0 385 0 1 1 0 0]

[ 0 0 0 0 0 374 1 1 0 0]

[ 0 0 0 0 0 0 377 0 0 0]

[ 0 0 0 1 0 0 0 386 0 0]

[ 0 0 0 0 0 0 0 0 379 1]

[ 0 0 0 1 1 0 0 1 1 378]]

classfication report for training data with two layer model

precision recall f1-score support

0 1.00 1.00 1.00 375

1 0.99 0.97 0.98 389

2 1.00 0.99 1.00 380

3 0.99 0.99 0.99 389

4 0.99 0.99 0.99 387

5 0.99 0.99 0.99 376

6 0.99 1.00 0.99 377

7 0.99 1.00 0.99 387

8 0.99 1.00 1.00 380

9 0.99 0.99 0.99 382

avg / total 0.99 0.99 0.99 3822

confusion matrix for test data with two layer model

[[174 0 0 0 0 2 1 0 0 0]

[ 0 177 0 0 0 0 3 1 1 0]

[ 1 2 165 2 0 0 3 1 3 0]

[ 0 1 1 169 0 1 0 2 0 9]

[ 0 0 0 0 179 1 0 1 0 0]

[ 0 1 0 0 0 180 0 0 0 1]

[ 0 0 0 0 2 0 179 0 0 0]

[ 0 1 0 0 1 4 0 171 0 2]

[ 0 4 0 3 2 4 1 0 153 7]

[ 1 0 0 2 0 2 0 0 2 173]]

classfication report for test data with two layer model

precision recall f1-score support

0 0.99 0.98 0.99 177

1 0.95 0.97 0.96 182

2 0.99 0.93 0.96 177

3 0.96 0.92 0.94 183

4 0.97 0.99 0.98 181

5 0.93 0.99 0.96 182

6 0.96 0.99 0.97 181

7 0.97 0.96 0.96 179

8 0.96 0.88 0.92 174

9 0.90 0.96 0.93 180

avg / total 0.96 0.96 0.96 1796

**Design Choices-** In the experiments we changed different hyperparameters to see which would give the best results and found that the models which were fed with scaled inputs and a model with two layers with ReLU as activation function provided the highest accuracy among all.

**Discussions-** Neural Network architectures with ReLU as activation function and categorical cross entropy performs better than Neural network architectures with other hyperparameters as we have seen from the table above as it provides an accuracy of 99.5290 which is much higher than other results.

We see that the models with ReLU as activation function have better performance in accuracy than models with the mean squared error as the loss function. The tanh optimizer usually involves exponential functions whereas ReLU uses a threshold to output values and are generally less complex.

1. The best model for with tanh activation function has the following results-

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sl. no | Number of hidden layers | Number of units in each hidden layer | Loss function | Activation  Function  (Input) | Optimizer | Learning  Rate | Momentum | Input Scaling | Accuracy | Convergence  Speed |
| 10 | 2 | 32,16 | Classification cross entropy | tanh | adam | N/A | N/A | Yes | 99.3459 | 10.188604 |

**Different results for this model-**

confusion matrix for training data with two layer model

[[375 0 0 0 0 0 0 0 0 0]

[ 0 387 0 0 0 0 2 0 0 0]

[ 1 0 378 1 0 0 0 0 0 0]

[ 0 1 0 387 0 1 0 0 0 0]

[ 0 0 0 0 386 0 1 0 0 0]

[ 0 0 0 1 0 375 0 0 0 0]

[ 0 0 0 0 1 0 376 0 0 0]

[ 0 0 0 1 1 1 0 384 0 0]

[ 0 3 1 1 0 0 0 0 374 1]

[ 0 0 2 3 1 0 0 0 1 375]]

classfication report for training data with two layer model

precision recall f1-score support

0 1.00 1.00 1.00 375

1 0.99 0.99 0.99 389

2 0.99 0.99 0.99 380

3 0.98 0.99 0.99 389

4 0.99 1.00 0.99 387

5 0.99 1.00 1.00 376

6 0.99 1.00 0.99 377

7 1.00 0.99 1.00 387

8 1.00 0.98 0.99 380

9 1.00 0.98 0.99 382

avg / total 0.99 0.99 0.99 3822

confusion matrix for test data with two layer model

[[172 0 0 0 0 3 1 0 0 1]

[ 0 177 1 0 0 0 3 0 1 0]

[ 0 6 166 4 0 0 1 0 0 0]

[ 0 0 5 172 0 1 1 1 0 3]

[ 0 1 0 0 179 0 0 0 1 0]

[ 0 0 0 1 0 179 0 0 0 2]

[ 0 1 0 0 2 0 178 0 0 0]

[ 0 0 0 0 1 2 0 166 0 10]

[ 0 5 0 4 0 5 0 1 150 9]

[ 0 0 2 4 0 2 0 1 0 171]]

classfication report for test data with two layer model

precision recall f1-score support

0 1.00 0.97 0.99 177

1 0.93 0.97 0.95 182

2 0.95 0.94 0.95 177

3 0.93 0.94 0.93 183

4 0.98 0.99 0.99 181

5 0.93 0.98 0.96 182

6 0.97 0.98 0.98 181

7 0.98 0.93 0.95 179

8 0.99 0.86 0.92 174

9 0.87 0.95 0.91 180

avg / total 0.95 0.95 0.95 1796

**Discussion-** The best learned model for tanh activation function as seen from the above table in comparison to models with same activation and different hyperparameters is a model with 32 hidden units in first layer and 16 in the second with input which is scaled using standardization. This model gives an accuracy of 99.2936 and delivers very high performance.

As such the models with ReLU as activation function have better performance in accuracy than models with tanh as activation function.

2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.no | Number of filters | Filter size | Dropout  Probability | Dropout  probability | Accuracy | Convergence  Speed |
| 1 | 3X3 | 32 | 0.3 | 0.5 | 97.72 | 59.38744783401489 |
| 2 | 3X3 | 32 | 0.3 | 0.6 | 97.94 | 36.36566090583801 |
| 3 | 2X2 | 32 | 0.3 | 0.5 | 97.33 | 57.67174482345581 |
| 4 | 2X2 | 32 | 0.3 | 0.6 | 97.66 | 65.45488595962524 |
| 5 | 4X4 | 32 | 0.3 | 0.8 | 96.33 | 28.2467839717865 |
| 6 | 5X5 | 32 | 0.4 | 0.5 | 97.22 | 35.36261534690857 |
| 7 | 5X5 | 32 | 0.5 | 0.6 | 96.99 | 29.938284158706665 |

The best CNN model has a 3X3 filter with a filter size of 32 and dropout probabilities of 0.3 and 0.6 which gave an accuracy of 97.94 and convergence time of 36.36566090583801.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.no | Number of filters | Filter size | Dropout  Probability | Dropout  probability | Accuracy | Convergence  Speed |
| 2 | 3X3 | 32 | 0.3 | 0.6 | 97.94 | 36.36566090583801 |

For the above model-

Confusion matrix for train data

confusion matrix for training data

[[374 0 0 0 1 0 0 0 0 0]

[ 0 387 1 0 0 0 0 0 0 1]

[ 0 0 379 0 0 0 1 0 0 0]

[ 0 0 0 383 0 5 0 0 0 1]

[ 0 0 0 0 385 0 1 0 0 1]

[ 0 0 0 0 0 375 0 0 0 1]

[ 0 1 0 0 1 0 375 0 0 0]

[ 0 0 0 1 1 0 0 385 0 0]

[ 0 3 0 0 0 1 0 0 376 0]

[ 0 0 0 0 2 1 0 1 0 378]]

Classification report for training data

classification report

precision recall f1-score support

0 1.00 1.00 1.00 375

1 0.99 0.99 0.99 389

2 1.00 1.00 1.00 380

3 1.00 0.98 0.99 389

4 0.99 0.99 0.99 387

5 0.98 1.00 0.99 376

6 0.99 0.99 0.99 377

7 1.00 0.99 1.00 387

8 1.00 0.99 0.99 380

9 0.99 0.99 0.99 382

avg / total 0.99 0.99 0.99 3822

Confusion matrix for test data

confusion matrix for testing data

[[176 0 0 0 1 0 0 0 0 0]

[ 0 182 0 0 0 0 0 0 0 0]

[ 0 1 176 0 0 0 0 0 0 0]

[ 1 0 1 176 0 2 0 2 0 1]

[ 0 0 0 0 181 0 0 0 0 0]

[ 0 0 0 0 1 177 1 0 0 3]

[ 0 1 0 0 1 0 179 0 0 0]

[ 0 0 0 0 0 1 0 175 0 3]

[ 0 5 0 0 1 1 0 1 164 2]

[ 0 1 0 1 4 1 0 0 0 173]]

Classification report for test data

classification report

precision recall f1-score support

0 0.99 0.99 0.99 177

1 0.96 1.00 0.98 182

2 0.99 0.99 0.99 177

3 0.99 0.96 0.98 183

4 0.96 1.00 0.98 181

5 0.97 0.97 0.97 182

6 0.99 0.99 0.99 181

7 0.98 0.98 0.98 179

8 1.00 0.94 0.97 174

9 0.95 0.96 0.96 180

avg / total 0.98 0.98 0.98 1796

**Design choices-** The CNN model was experimented with different sizes of filters and different dropouts were experimented starting from 3X3 filters and moderate dropouts. As seen from the selection of parameters we know that the model with low dropout probabilities gives a better accuracy than the model with more filters and higher dropouts in the model.

**Discussion-** Neural networks are used for dimensionality reduction with an architecture which starts with a dense network and progressively becomes narrower. The first half is used for reducing the dimensionality and the second half used to find the back-projection of low dimensional points to high dimensional points. The best model above gives us an accuracy of 97.94 with a convergence speed of 36.36566090583801.