Classification

Name	Roll No
Noor Ahmed	15-4265

1. Classification Algorithms used

Two types of models were trained

- Multilayer Neural Network (with different layer sizes, activation and optimisers)
- Convolutional Neural Network

2. Preprocessing

Neural Networks are highly dependant on normalization. So normalization was performed which improved the accuracy slightly. Deskewing(from the previous assignment) also helped in the first model.

3. Result

Performance of ANN without scaling			
Parameters	Accuracy	Time to Train	Time to test
Activation="relu", alpha=1e-5, hidden layer=(700, 500, 250, 100), solver="Adam"	0.98174757	46.034562826156616	0.16402220726013184
Activation="tanh", alpha=1e-5, hidden layer=(700, 500, 250, 100), solver="Adam"	0.97553398	62.01548171043396	0.1714768409729004
Activation="logistic", alpha=1e-5, hidden layer=(700, 500, 250, 100), solver="Adam"	0.9766990	56.328927993774414	0.17512297630310059

Note: The Adam solver gave better results for unscaled data but took longer to train. On average Stochastic Gradient descent was faster to train for all activation functions but its results were not that convincing so wasnt added above

Performance of ANN with scaling

Parameters	Time to Train	Time to Test	Accuracy
Activation="relu", alpha=1e-5, hidden layer=(700, 500, 250, 100), solver="Adam"	0.95106796	4.69872093 2006836	0.1640222072 6013184
Activation="tanh", alpha=1e-5, hidden layer=(700, 500, 250, 100), solver="Adam"	0.95728155	4.33602690 6967163	0.1714768409 729004
Activation="logistic", alpha=1e-5, hidden layer=(700, 500, 250, 100), solver="Adam"	0.97475728	4.19516634 9411011	0.1751229763 0310059
Activation="relu", alpha=1e-5, hidden layer=(700, 500, 250, 100), solver="sgd"	0.96233009	3.31863713 26446533	0.1422901153 564453
Activation="tanh", alpha=1e-5, hidden layer=(700, 500, 250, 100), solver="sgd"	0.94368932	3.67483115 19622803	0.1702852249 1455078
Activation="logistic", alpha=1e-5, hidden layer=(700, 500, 250, 100), solver="sgd"	0.26524271	3.59376311 30218506	0.1806697845 4589844

Note: For scaled data the training time was much lower and manageable. Although the accuracies were not great

Performance of the Convolutional Neural Network

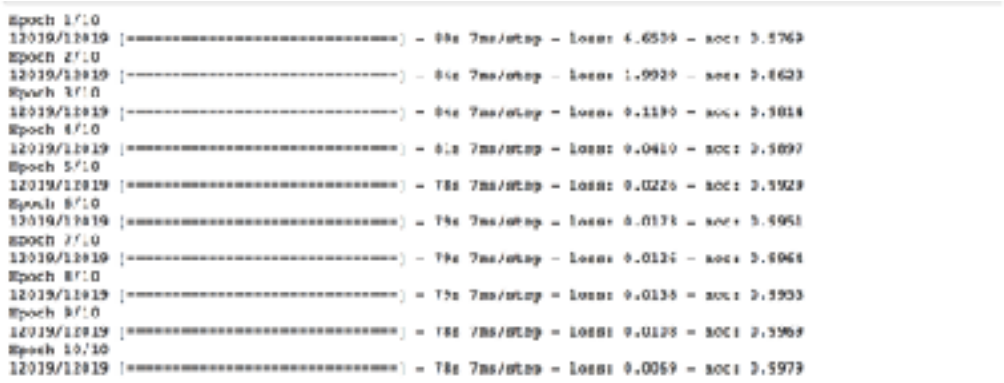
The parameters were selected via trial and error:

2 convolutional layers (100 and 50)

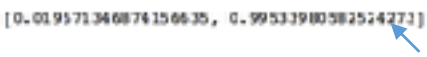
a Max pool layer to reduce the dimension of the input

Two dense layers of size 600 and 400

Above is the training



Above is the training process for the CNN. In total the time taken was around 14 minutes. With training accuracy around 99.79



The accuracy for the test subset dataset was around 99.5 % and the kaggle score was 99.6

4. Discussions and Conclusion

Convolutional Networks work best on handwritten classifications especially paired with faster optimizer (i.e. Adam) functions and RELU activation functions which yielded the best results. This is because unlike classical neural networks, CNN reach a classification based on multiple kernel based convolutional filters. So in a sense the network preprocesses the data for us before hand and finds the features that distinguish one class instance from another.

It is due to this that CNN gave better accuracy although the training time tradeoff is to be considered.