1: MNIST classification

Solution:
The classic MNIST images dataset comprised 3 partitions(All images were of dimension 28 x 28 provided in the form of a flattened 784 dimensional vector(values normalized)):

1. Train set: Size = 50,000
2. Validation set: Size = 10,000
3. Test set: Size = 10,000

The following classifications were trained:

- SVM: Various Kernels like linear, poly and rbf were tested.
- Single Hidden layer neural network: Variations of hidden units were tried
- 2 Hidden layer neural network Similar to single hidden layer, many hidden units were tried.
- 5 Hidden layer neural network Multiple hidden units were tried
- Convolutional neural network Multiple convolutional layers were stacked.

Experiments & Observations

1. SVM parameter C was tuned using the validation data set. Final model : RBF with C = 1×10^4. The final test accuracy measured was best for rbf kernels, followed by poly and linear. Following is the test accuracy.

	Linear	Poly	Rbf
SVM	77.01%	81.83%	97.40%

2. Neural networks were also tuned using validation data sets. The weight initialization was done using constant weights, randomized weights(Sampled from standard Gaussian dist) and normalized initialization given by Glorot, et al. Glorot initialization was comparable to randomized weights and occasionally performed better in case of CNN. Constant weights performed worst(Measured on max epoch of 10000, or until convergence (tolerance 1×10^5)).

3. Adam optimizer was used for the optimization of loss function for each neural network. This outperformed the vanilla SGD and momentum based SGD.

4. For single hidden layer neural network, the parameters were: Sigmoid activation function, 500 hidden units, randomized weights. Cross entropy loss function

Epochs	Validation Accuracy
1	91.01%
10	97.6%
50	97.12%
200	98.21%

Test accuracy = 97.75%

5. For 2 hidden layer neural network, the parameters were: Sigmoid activation functions on both layers. Hidden layer 1: 500 units, Hidden layer 2: 250 units, randomized weights. Cross entropy loss function

Epochs	Validation Accuracy
1	90.87%
10	98.22%
50	98.11%
200	98.06%

Test accuracy = 98.03%

6. For 5 layers:

	Layer1	Layer2	Layer3	Layer4	Layer5
Hidden Units	500	250	200	300	150

Sigmoid activation function was used in hidden layer 1 and 2, followed by relu in 3 and 4 and sigmoid again in layer 5.

Epochs	Validation Accuracy
1	90.17%
10	96.86%
50	96.64%
200	97.02%

Test accuracy = 96.87%

7. It can be observed that the performance increased when we moved from one layer to two, since we have better functional approximator since the number of learning parameters have increased. However, we observe a decrease in case of 5 hidden layer model. This is perhaps because the model is now over-fitting on the training data and giving worse generalization. Loss was also measured after every 10 epochs and showed steady decrease in the value. It was observed that the loss was more on single layer than 2 hidden layer and very less on 5 hidden layer, which essentially shows that the model is most probably over fitting.

8. CNN: These were trained using GPU Nvidia GTX 1080 Titan.Convolutional neural network architecture:

Layers	Units
Conv Layer 1	32 filters (5×5), Stride: (1,1), Padding: (2,2)
Maxpool1	(2,2) $\rightarrow$ 1
Conv Layer 1	64 filters (5×5), Stride: (1,1), Padding: (2,2)
Maxpool2	(2,2) $\rightarrow$ 1
FC1	3136 $\rightarrow$ 1024
FC2	1024 $\rightarrow$ 10

Test accuracy = 99.54%. After 1 epoch the accuracy was around 92-93% and after 20 epochs, it was around 99%.

9. Dropout regularization was applied in case of CNN and the accuracy improved in such a case. With dropout we could see a 0.2-1% increase in the accuracy on same parameters. Relu activation function was used in the fully connected (FC) layer. Finally the cross entropy loss function was used.
10. On increasing the convolutional layers, there was no distinguishable increase in the validation accuracy.
11. Convolutional neural networks outperformed SVM and feedforward neural networks. This is because images are two dimensional data and nearby pixels are correlated to each other. Such a property is better captured by CNNs. In case of SVM and Feed forward neural network, we feed only a 1 dimensional squashed version of the original 2d/3d image
12. Various combinations of layers, optimizers,etc were tried. The above results are measured on what gave the most accuracy.
13. Cross validation was not performed given the huge size of the data set. Instead the validation was done on the provided validation data set.

References:

1. Code written in python notebook format with the help of pytorch library and scikit-learn libraries. Available upon request
2. Normalized initialization, Glorot, et. al.