

Results and Discussion

September 17, 2018

Over the Assignment, two MLP's, 4-layers and 3-layers deep respectively, a kNN Classifier and a SVM have been trained using the MNIST Dataset. 5-fold cross Validation was used. The trends with respect to convergence and accuracy were studied and have been reported.

Note: This document is intended to be only a discussion of the trends observed. All the data collected is presented with the other documents alongside the training code.

Training: The total dataset of 60000 images were split into 5 folds. 4 folds at a time were taken as the training set and the remaining fold was taken as the test set.

1 Questions 1, 2 and 3:

The given architecture for a 4-layer MLP was implemented with Sigmoid and ReLU activation functions. They were trained using the method described above.

Avg. Sigmoid Accuracy = 91.40 % Avg. ReLU Accuracy = 97.27 %

The test error for ReLU is much lower when compared to Sigmoid. Learning rate decay might help Sigmoid to converge, especially during the later stages of training.

1.1 Observations:

1. The convergence time for Sigmoid is much slower than ReLU - possibly due to the vanishing gradients in the case of Sigmoid.
2. Usage of ReLU can sometimes lead to the exploding gradients problem which leads to numerical overflows. Often, this is the result of un-normalized inputs or parameters with high variances. Gradient Clipping comes handy in such scenarios.
3. Often, ReLU neurons go dead, unlike Sigmoid neurons which can recover in the long-run.
4. Occasionally, it was observed that the loss function kept oscillating during the later stages of training. Learning Rate Decay would help achieve better performance and smooth convergence.

2 Questions 4 and 5:

In the first case, to make the task harder, instead of augmenting noisy data to the original data, the network was trained on noisy data and tested on regular data. L2 Regularisation was employed in another case with $\lambda=1e-3$.

Both cases used ReLU with the same architecture. Slight improvements are already observed over the normal cases in both the situations.

Noisy ReLU Test Accuracy = 97.33 % L2-Regularized ReLU Test Accuracy = 97.18 %

Note that this is on par with the ReLU Accuracy. On also training with regular data, the Noisy ReLU would perform much better. Imposing a heavier regularization might help in the second case.

3 Questions 6 and 7:

HOG descriptors are suitable for this type of problems. HOG yields descriptors that are representative of the overall shape and the orientation of the image which are characteristic for digits. The drawback in this case is that the resolution of the dataset is low implying imprecise feature vectors. Given all that, it is remarkable that models simpler than the ones used above are competitive.

A feature vector of length 81 is the condensed imaged vector used in this portion.

Models Compared: 1. A 3-layer MLP with 500 and 250 hidden units, ReLU activation with Softmax outputs and Cross Entropy Loss. 2. A kNN classifier with k=5. 3. An Linear SVM (Using an RBF Kernel turned out to be computationally intractable)

1. Avg. ReLU MLP Accuracy = 96.80 %
2. Avg. Sigmoid MLP Accuracy = 96.81 %
3. Avg. 5-NN Accuracy = 95.70 %
4. Avg. Linear SVM Accuracy = 90.32 %

3.1 Observations:

1. A smaller MLP compared to the earlier MLP performed competitively well on this task all thanks to the crafted feature vector. Necessitates the significance of representation.
2. Surprisingly, the 5-NN classifier reported accuracies north of 95 percentage.
3. The failure of the SVM could probably be due to similar local structures in the digits. An RBF kernel is obviously expected to perform a better job at classification.
4. Dealing with a smaller model than earlier but the same number of iterations, both Sigmoid and ReLU models have converged.

4 Confusion Matrix and Top 3 Prediction Observations:

Interestingly, the different algorithms used were "confused" differently when given the test set.

4.0.1 HOG-MLP:

Digits that are oriented similarly were often confused in this category. Eg: 4 as 9 and 4 as 7 but not the opposite way.

4.0.2 KNN:

Skeletally similar digits were confused here, naturally. Eg: 5 as 3 and vice-versa; 4 as 7 and vice-versa

4.0.3 SVM:

Often, dissimilar digits were confused. Eg: 3 as 2 and vice-versa. Interestingly, in one case, 1 was confused as 8.

4.0.4 Normal-MLP:

Common error sets like (4,7,9) and (3, 5, 8) were predominant. Some peculiar trends can be found in individual folds which are probably the result of individual bias in each fold setting.