MNIST Data Modeling for Final paper for STA 561

Gongjinghao Cheng, Zhenyu Tian*
Department of Statistical Sciences
Duke University
Durham, NC 27708
gongjinghao.cheng@duke.edu,zhenyu.tian@duke.edu

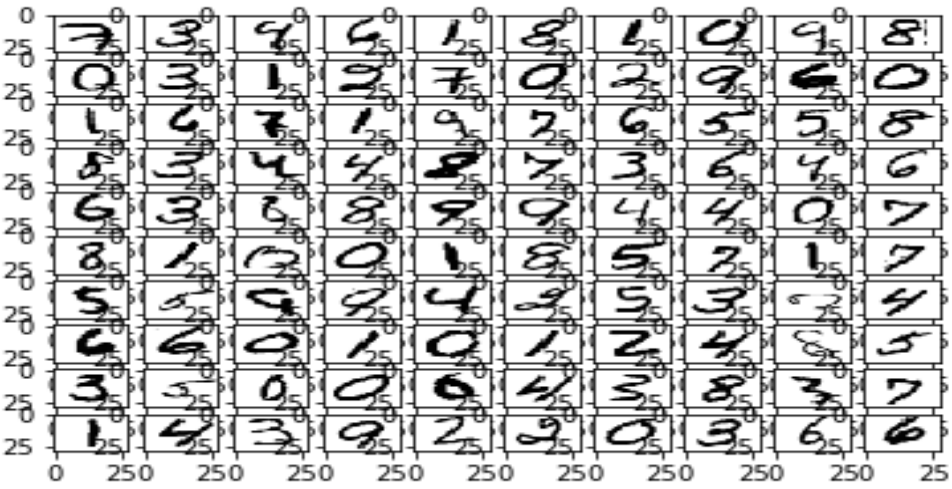
The primary goal of this project is to fit models to the MNIST handwritten digits data set based on the methods we learnt in STA 561 and from external resources. We apply models including Softmax regression, Random Forest, and Convolutional Neural Network with different architectures. Lastly, we compare the test accuracy for each models and discuss the results.

1 Data

The MNIST database of handwritten digits is available on Yann LeCun's website. The data base contains a train set of 60000 samples and a test set of 10000 samples. Both the train and test set contain a set of images and labels. For each observation, the image data has 784 dimension and the label data has 1 dimension. Each image data can be reshaped into a 28×28 matrix and be displayed with 'imshow()' function in Python. The label presents the actual digit of the corresponding image ranges from 0 to 9.

Typical hand written digit plots will look like Figure 1:

Figure 1: Sample Plots of Hand Written Digits in MNIST



*Github Repository: https://github.com/kingcheng12/MNIST_analysis

2 Methodology

Handwritten digits recognition is a classification problem. To study this data set, we implemented Softmax regression, Random forest, Convolutional Neural Network and then evaluated the performance of each method based on their accuracy. In this section, we will briefly introduce each method and discuss the settings of them.

2.1 Softmax Regression

Since our data has 10 classes, the regular logistic regression is not suitable for this problem. Instead, we use Softmax regression to handle this classification. Softmax regression is also known as multinomial logistic regression. It is a generalization of the logistic regression to handle multiple classes. Since this is a rather simple model than the other methods, we expect a relatively inaccurate result. However, this is a good starting point to check the performance of generalized linear model in this modeling problem.

In terms of the packages and settings, we use the `tensorflow` package with `cross entropy loss`, and optimize the parameters with `GradientDescentOptimizer`. We fit the model on the train set and estimate the test accuracy on the test set (Same for the rest of models). The final test accuracy is 85.9%.

2.2 Random Forest

Random forest is another common method for classification problems and it often performs well. Random forest is an ensemble of a large number of individual decision trees whose nodes are decorrelated with a dropout probability. Each tree will generate a prediction, and the class with the most votes will become the model prediction. We expect it to be more accurate than Softmax regression but less accurate than other complex models, such as CNN. Empirically, the test accuracy is 96.9%, which is higher than the result from Softmax regression as expected.

2.3 CNN

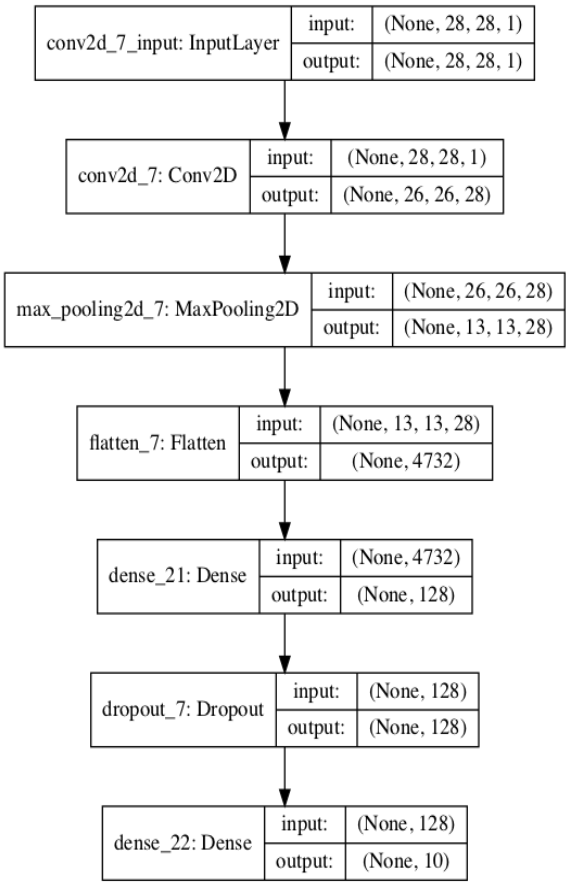
Convolutional Neural Network is one of the most frequently used image classifiers. It is a neural network with convolutional layers designed for image feature recognition; the convolutional layers are usually followed by max pool layers to reduce the dimension; the last few layers will be full connected dense layers followed by a softmax activation node as output layer. Since this is a especially useful image classification method, we expect it to be one of the most accurate algorithm in this case.

In this section, we will first create a naive CNN model with simple architecture and then experiment on different structures and parameters in order to develop the most accurate model.

2.3.1 Naive CNN model

The naive CNN model consists of a single pair of Convolutional layer and Max pooling layer. This first attempt gives us a test accuracy of 98.5%, which is already better than the previous two methods. The architecture we used is shown in Figure 2.

Figure 2: Architecture of Naive CNN



2.3.2 Development of Best CNN model

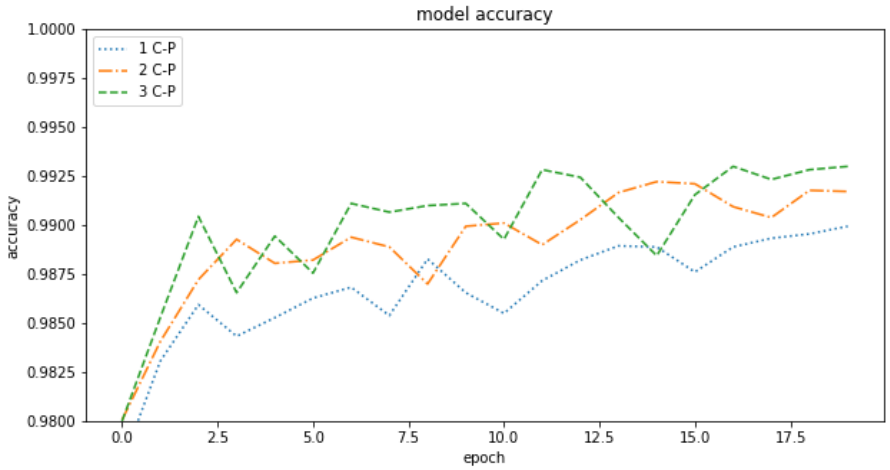
In this section, we are going to explore different features of CNN architecture. We will try to pick out optimal number of C-P (Conv2D and MaxPooling2D) pairs, optimal number of feature maps in each Conv2D layer and the best dropout sequentially.

We first compare the CNN models with number of C-P pairs ranges from $\{1, 2, 3\}$ to determine the ideal number of pairs. The results are shown in Table 1 and Figure 3.

Table 1: Maximum Historical Validation Accuracy for Different Numbers of C-P Pairs

number of C-P pairs	1	2	3
validation accuracy	0.98994	0.99222	0.99300

Figure 3: Accuracy vs. Epoch for Different Numbers of C-P Pairs



Based on the results, it is evident that the model with 3 C-P pairs has highest validation accuracy in this case. However, comparing to that with 2 C-P pairs, the accuracy with 3 C-P pairs only has an increment less than 0.8%, but the corresponding model takes huge amount of extra time to train. Thus, we still choose 2 C-P pairs as optimal.

The next step is to choose the best number of feature maps. We apply 2 C-P pairs for our current CNN and select the optimal map numbers in $\{8,16,24,32,48\}$.

Figure 4: Accuracy vs. Epoch for Different Numbers of Feature Maps

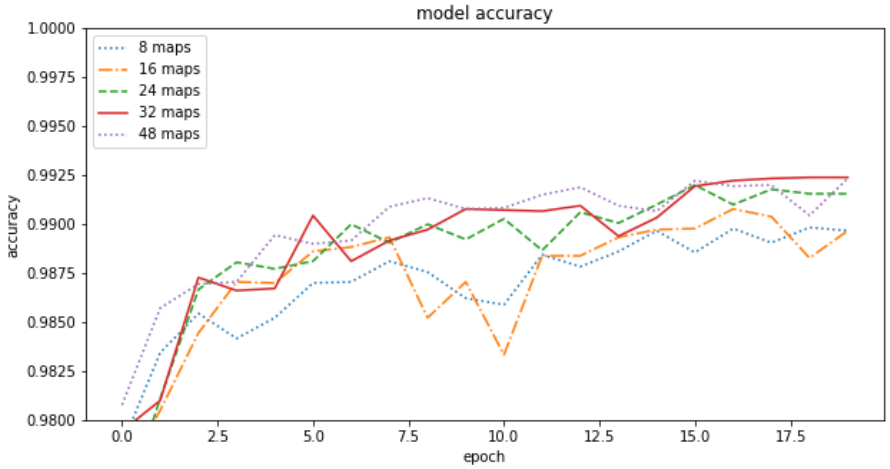


Table 2: Maximum Historical Validation Accuracy for Different Numbers of Feature Maps

number of maps	8	16	24	32	48
validation accuracy	0.98983	0.99078	0.99200	0.99239	0.99233

According to Figure 4 and Table 2, the best choice of map number is 32 in this case. However, since the validation accuracy of models with different map numbers are very close, the best choice might be subject to change in different scenarios.

With 2 C-P pairs and 32 feature maps in first Conv2D layer (64 in 2nd ConvsD layer) as our current optimal choice, we need to find the ideal dropout.

Figure 5: Accuracy vs. Epoch for Different Dropouts

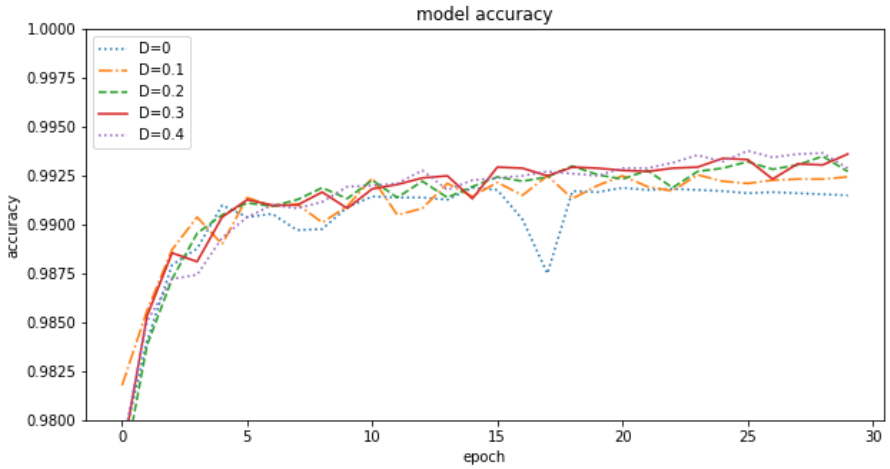


Table 3: Maximum Historical Validation Accuracy for Different Dropouts

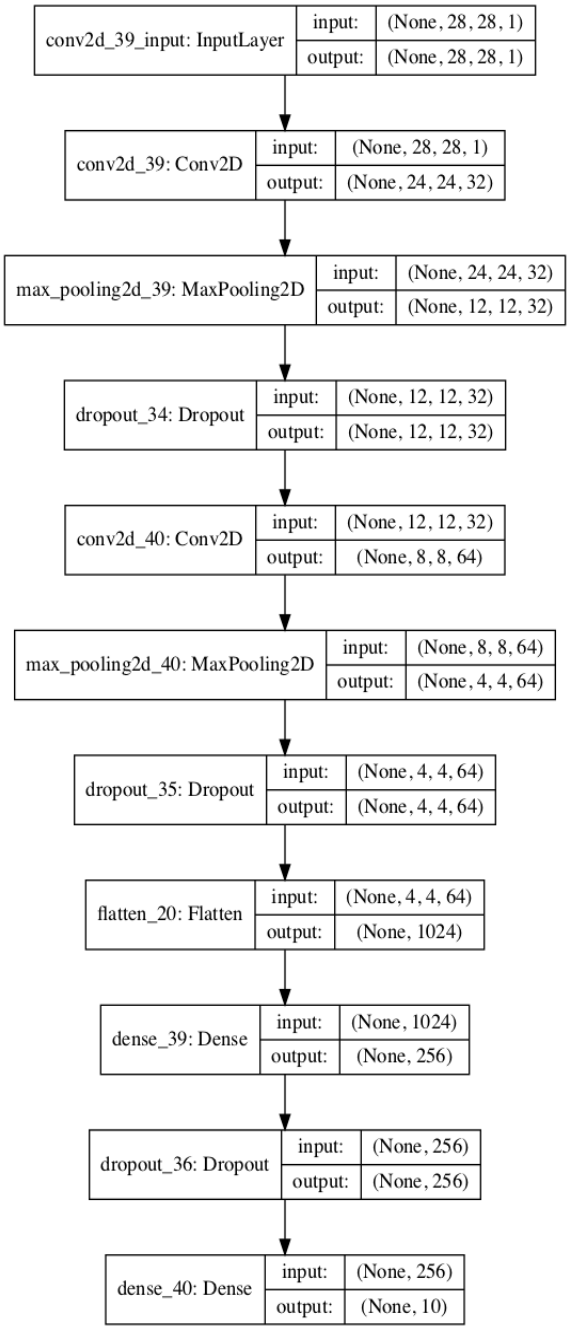
Dropout	0	0.1	0.2	0.3	0.4
validation accuracy	0.99189	0.99256	0.99350	0.99361	0.99378

Based on the results above, the validation accuracy for different dropout is very similar. The optimal dropout is 0.3. Again, it seems that several dropouts have similar effects on the validation accuracy. We should be careful when selecting dropout for different cases.

2.3.3 Optimal CNN model

Based on experiments on **2.3.2**, we have developed a model that hopefully will do better than our naive model. The test result turns out that this final model has a test accuracy of 99.5%, which is the best among all the models. The architecture of the optimal model is shown in Figure 6.

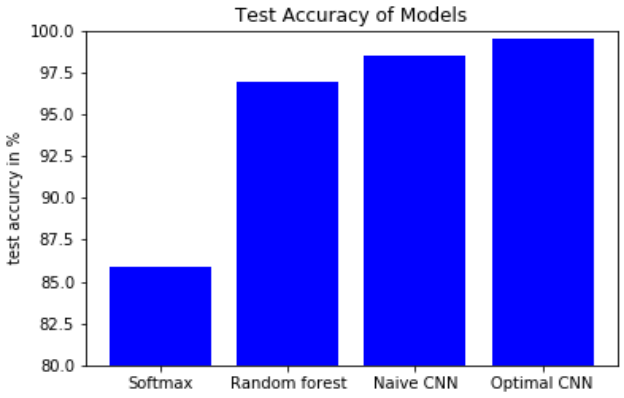
Figure 6: Optimal CNN Model Structure



3 Summary

In this project, we have fitted Softmax Regression, Random Forest and 2 CNN models on the MNIST data set (check out details on our Github). The out of sample errors of all models are displayed in Figure 7. As expected, the Softmax Regression is relatively weak comparing to other models. The Random Forest classification is quite accurate, but not accurate enough since we pursue a nearly perfect prediction in a hand written digit recognition problem. Both CNN models are doing well as they are designed to handle image classification; Nevertheless, the optimal CNN model is exceptional. We suspect that it is approaching Bayes error. Overall, We are satisfied with its result.

Figure 7: Model Comparison



Besides these presented model, we have also tried Gaussian Classifier and SVM. However, due to the limited computational power, we are not able to conduct our analysis. Meanwhile, we believe that these models are less accurate than CNN for image classification problems. While we are not interested in fitting these two models for MNIST data set, they could be an engaging direction for further investigation if we are using a non-image data set.

References

[1] UFLDL Tutorial. Softmax regression. *Stanford University*, n.d.

[2] Tony Yiu. Understanding random forest. *Towards Data Science*, 2019.

[3] S.P. Mishra Pooja Asopa Sakshi Indolia, Anil Kumar Goswami. Conceptual understanding of convolutional neural network- a deep learning approach. *Procedia Computer Science*, 132:679–688, 2018.

[4] Nikhil Kumar. Softmax regression using tensorflow. *GeeksforGeeks*, n.d.

[5] Vinit Neogi. Handwritten-digit-recognition-using-random-forest. *Github*, 2018.