COMP 551 Mini-Project 3

Grace Hu	Xingyu Chen	Jiahui Peng
260776936	260786048	260782511

December 2020

1 Introduction

This mini-project addresses the task of image classification using a modified version of the MNIST database of handwritten digits. The modified MNIST dataset consists of images that contain at most five digits, in the range of 0 to 9, superimposed onto a patterned background. This project involves the application of Convolutional Neural Networks (CNN), a deep learning method, to perform image classification and handwritten digit recognition.

After searching for pre-existing work and image classification libraries, we found several related approaches to create a pipeline to solve our problem. The one approach that proved the most fruitful is based on the end-to-end Deep CNN model proposed by [1]. The authors proposed an end-to-end Tensorflow solution to this problem that does not involve breaking down the problem into the complicated steps of localization, segmentation, and recognition. Inspired by the great results they displayed in their study and their model that could correctly classify multi-digit MNIST images without the assistance of a digit localizer, we adapted their work for this project.

When we first evaluated our model on the synthetic multi-digit MNIST dataset, we obtained an accuracy of 80%. After applying data processing and hyperparameter tuning to our model, we eventually achieved a 98.00% accuracy on the test dataset of the Kaggle competition.

2 Design

2.1 Model Structure

The architecture of our tuned model consists of six convolutional hidden layers and two densely connected hidden layers. All connections are feed-forward and go from one layer to the next without skipping connections. All convolutional layers have the kernel size 5 x 5. The first hidden layer contains maxout units (with three filters per unit), which takes the maximum value of n linear functions [2]. The other hidden layers contain Rectifier Linear Units (ReLU) [3]. The number of units in each layer is [48, 64, 128, 160] for the first four layers and 192 for all other locally connected layers. There are 1024 activation functions on the fully connected layers. Each convolutional layer is followed by a max pooling layer, which reduces the dimensionality of the input (Figure 1). The stride of the model alternates between 2 and 1 at each layer, so that only half of the layers reduce the spatial size of the input (Figure 1). Afterwards, we performed batch normalization before every convolutional layer, which applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. All convolutional layers use zero padding on the input to preserve input size. We trained with dropout applied to all hidden layers except the input layer 2.

Architecture of CNN	
Accuracy	98.00
Parameter	value
Convolutional Layer	6
Maximum Pooling Layers	6
Stride Size	1 and 2
Learning Rate	0.001
Batch Size	128
Epochs	5
Activation Function	ReLu

Figure 1: Description of our CNN model

2.2 Hyper-parameter Optimization

To enhance the model's accuracy, we tried different combinations of hyper-parameters. We first tried changing the type of optimizer used from Adam to Nadam, RMSProp, and GradientDescent in the Tensorflow Keras library. However, we did not see any improvement in the test accuracy and thus stuck with Adam as the optimizer. We also varied the number of epochs we trained for, and saw an increase in accuracy when we increased the number of epochs. We also tried different batch sizes, and found that running time increased as batch size increased. Due to the limits of our computational resources, we compromised and used a batch size of 128. We also used a train-validation split of 80%-20% in order to train our model.

3 Result

In this section, we present the error rate (accuracy rate) obtained performing an image classification task on the modified MNIST dataset. Overall the best accuracy we achieved was **98.00%** using the following hyper-parameters: epochs = 5, learning rate = 0.001, batch size = 128.

4 Conclusion

With the purpose of developing a classification model to recognize the handwritten digits in the a modified MINIST data-set, we developed a multi-layer convolutional neural network. Due to nature of CNNs, no feature extraction was necessary as the first few layers essentially act as a trainable feature extractor. Over the past decades, many researchers have tackled image classification of the original MINIST digits dataset and achieved very promising accuracies of 97 99% using more complex CNN models [4]. In this project, we managed to achieve a fairly good accuracy of 98.00% with a relatively simple CNN, after adjusting the optimizer, learning rate, and batch size of our model.

While we attempted to improve the performance of our model, we learned that after a certain point, the time and space requirements of the training process exceeds what our platform of choice (Deepnote) can support. However, there are still many possible techniques we can employ to improve the accuracy of our model in the future, such as more specific hyper-parameter tuning and image processing (i.e. normalization of pixel intensity).

5 Appendix

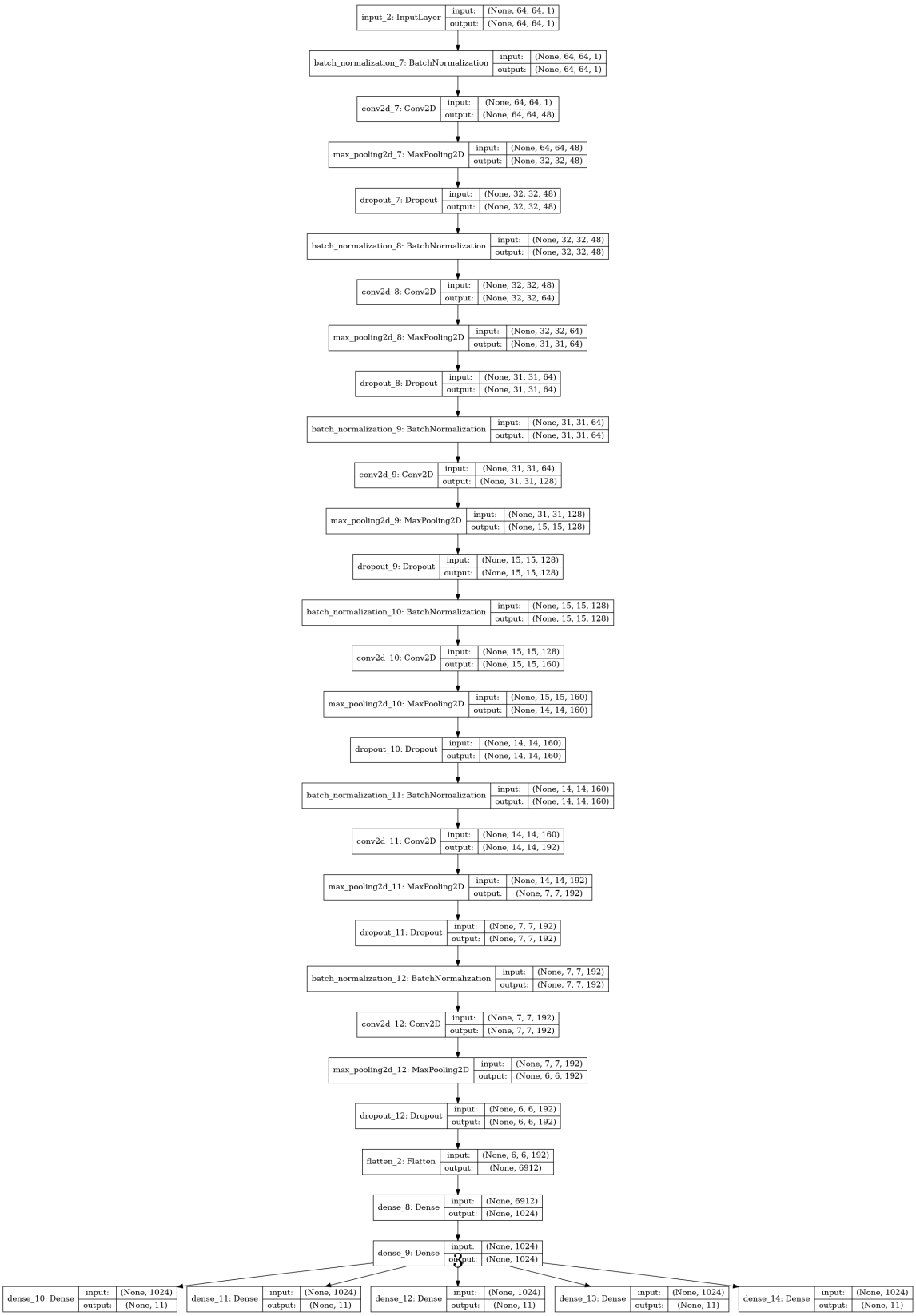

Figure 2: Model Architecture

References

[1] Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, et al. *Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks*. Apr. 2014. URL: https://arxiv.org/abs/1312.6082.

[2] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, et al. "Maxout networks". In: *International conference on machine learning*. PMLR. 2013, pp. 1319–1327.

[3] Kevin Jarrett, Koray Kavukcuoglu, Marc' Aurelio Ranzato, et al. "What is the best multi-stage architecture for object recognition?" In: *2009 IEEE 12th International Conference on Computer Vision* (2009). DOI: 10.1109/iccv.2009.5459469.

[4] Yann LeCun, Lawrence D Jackel, Léon Bottou, et al. "Learning algorithms for classification: A comparison on handwritten digit recognition". In: *Neural networks: the statistical mechanics perspective* 261 (1995), p. 276.