Study Group 10
CSE 144

Max Jacobs
Christopher Mason
Samir Nafez
Ishika Pol
Anya Rajan

Final Project Report - Digit Classification

Abstract - Ishika

According to Banjare and Massey, “digits are universal symbols that find extensive application in various technological fields such as banking, engineering, and postal services.” In this context, the classification of digits holds significant importance. Handwritten digit classification, in particular, has emerged as a practical and essential task in today’s world. With the increasing demand for pattern recognition applications, numerous sectors, including language translation, traffic sign detection, banking, and postal mailing, rely on handwritten character recognition. This necessity arises from the fact that every individual possesses a unique handwriting style, resulting in specific variations in the visual representation of each character. Consequently, accurate digit classification enables the development of efficient systems that can interpret and process diverse handwritten inputs effectively.

Introduction - Anya

This report explores the impact of modifying the learning rate on the performance of a variation of the MNIST dataset. MNIST, a well-known dataset, consists of 60,000 small square images of handwritten single digits between 0 and 9, each image being 28x28 pixels. In our project, we started by testing a baseline model to establish a working foundation. We then examined the effects of adjusting the learning rate on the model's performance. Through this, we wanted to gain a better understanding of the influence of learning rate modifications on the accuracy and efficiency of the model.

The learning rate plays a crucial role in the training process of machine learning models. It determines the step size at which the model adjusts its parameters during optimization. By experimenting with different learning rates, we can gain insights into how this hyperparameter affects the model's ability to learn and generalize patterns from the dataset.

Methodology - Samir

Getting started on this project was initially straightforward as the dataset already has a defined train and test dataset. As a starting point, we created a baseline model using a CNN. The baseline model had two main aspects, one of convolutional and pooling layers and the other a classifier that was able to make a prediction. The baseline model was working fairly quickly however it was taking a lot of time to run the model. The benefit of beginning with the baseline model is that it provided a good foundation to add improvements. To estimate the performance of the model we used k-fold cross validation. The process of k-fold validation helped to eliminate the problem of only working with a single train-test split. We created a line plot showing model performance on the train and test set during each fold of the k-fold

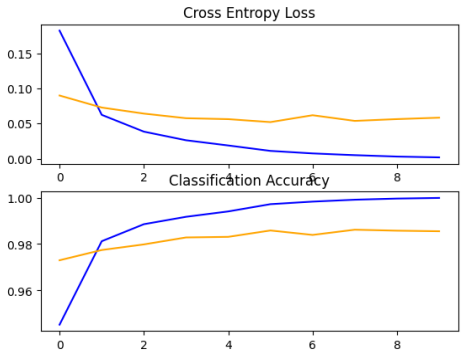
cross validation which helped us see if the model was fitting properly or not. A box and whisker plot was also used to summarize the distribution of accuracy scores.

After establishing the baseline model and ensuring that it was working properly, we were able to implement more features. Batch normalization was one thing that helped standardize the outputs and the normalization helped in stabilizing the learning process and in turn overall improving the learning performance. Modifying the learning rate was also key in order to get optimum performance but toying with this value was time consuming. Initially we were working with learning rates that were far too low which was causing our model to converge very slowly. Once these features were tweaked to their optimal values, we had a model that was already working much better than the baseline model. Something to explore was how changing the capacity of the feature extractor since the feature extractor is what gives the model the ability to recognize and evaluate complex patterns among the data. This in turn slightly reduced underfitting and could help with learning to encode various writing styles and classify them better. One problem that remained even after improving the model was that it was still taking a considerable amount of time to run the model.

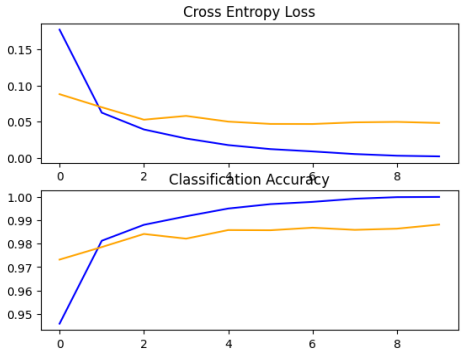
Results - Christopher

For the result metrics, we recorded the cross entropy loss as well as the classification accuracy, and graphed two metrics on the graphs. To define these metrics, the cross entropy loss plot visualizes the training and validation loss for each epoch. The training loss is computed as the average cross-entropy loss over all training examples and the validation loss is similarly computed over all validation examples. The cross-entropy loss is a common loss function for classification problems, and it measures the dissimilarity between the model's predictions and the true labels. A decrease in loss over epochs generally indicates that the model is learning and improving its ability to predict the correct labels. However, if the training loss continues to decrease while the validation loss starts to increase, this may be a sign of overfitting, where the model is learning the training data too well and failing to generalize to unseen data. And then the classification accuracy plot shows the accuracy of the model on the training and validation sets for each epoch. Accuracy is defined as the proportion of correct predictions out of total predictions. An increase in accuracy over epochs generally means the model is improving. However, similar to the loss plot, if the training accuracy continues to increase (approaches 100%) while the validation accuracy stagnates or decreases, it can be a sign of overfitting. There are 4 trials for the first CNN model, and then one for the newer CNN model.

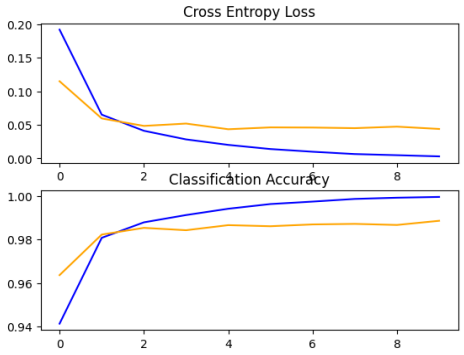
The 4 trails for the first model have the graphs:



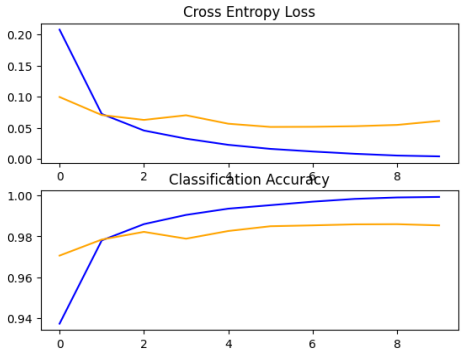
Trial 1



Trial 2

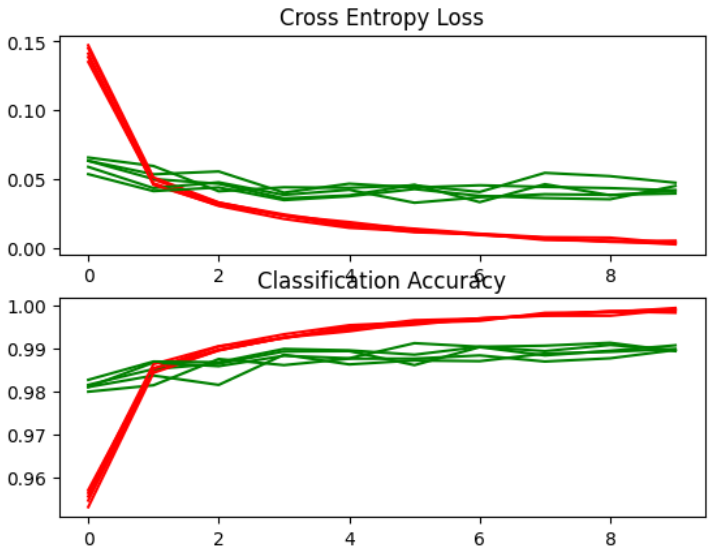


Trial 3



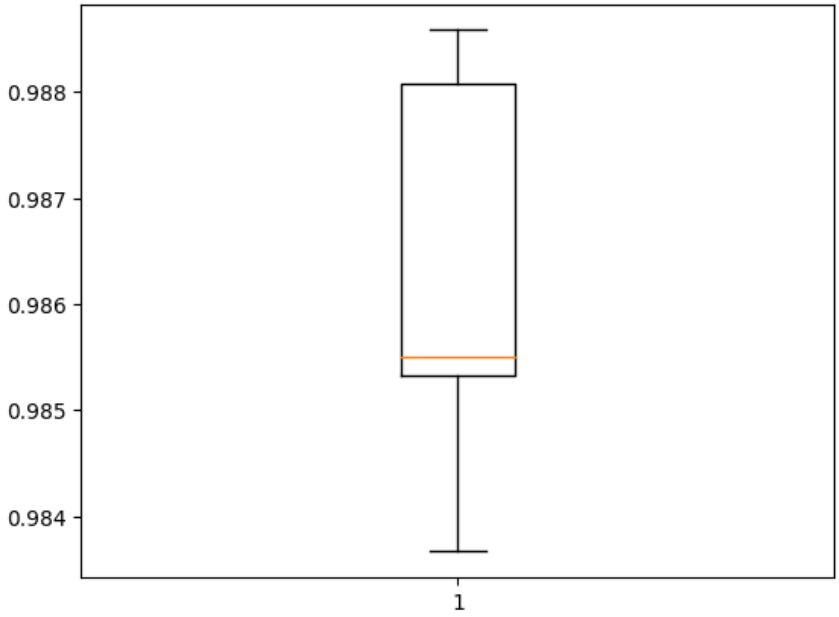
Trial 4

The graph for the new model is as follows:



We also recorded a summary for the first and second CNN model using a box plot. We also recorded the standard deviation, and some other important statistical information.

The summary statical information for the first CNN model is as follows:

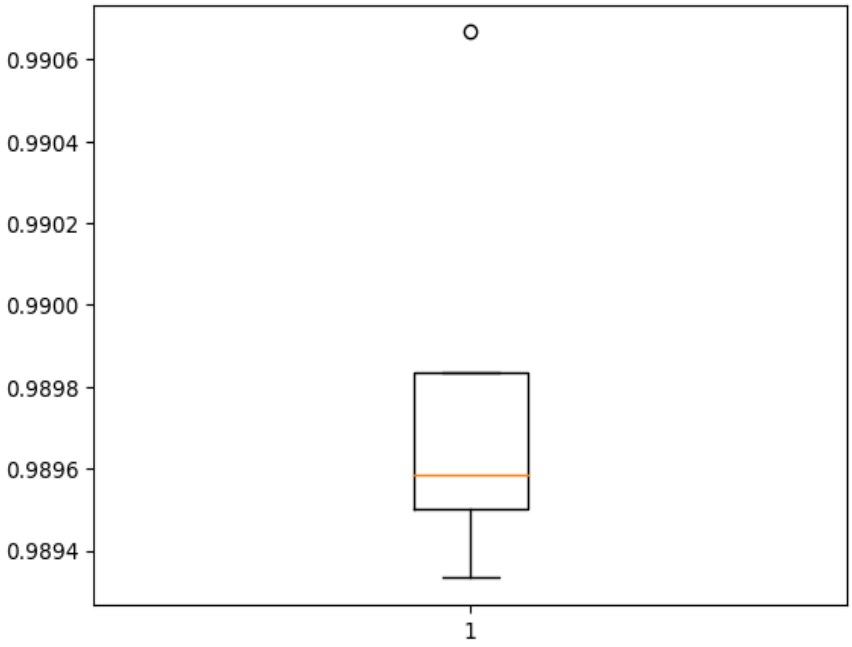


Mean = 98.623

Standard Deviation = 0.184

n = 5

And the summary statistical information for the second model is:



Mean = 98.978
Standard Deviation = 0.047
n = 5

Finally, for the results, we also compared the results of our CNN model to that of a model trained using logistic regression, to see the difference. The final accuracy of the logistic regression model is:

```
Logistic Regression Accuracy: 92.560
```

As we can see, it has an overall much lower accuracy compared to our CNN. In general, this could be because a CNN is particularly well-suited for image classification tasks due to its ability to learn hierarchical feature representations directly from the raw pixel data. This allows it to extract and emphasize local and translation-invariant features, which are crucial in image recognition tasks. On the other hand, traditional machine learning models such as logistic regression operate on flattened features and treat each pixel as an independent feature. They lack the capacity to model spatial hierarchies and local correlations in image data. As a result, they are usually less successful in capturing the complexity of images compared to CNNs.

Limitations of CNNs - Ishika

Convolutional Neural Networks (CNNs) have gained significant popularity in various fields, particularly in image recognition and computer vision tasks. However, they are not without their limitations. One major drawback is the time consumption associated with running the model. According to an article on OpenGenus IQ, each iteration of running the model can be time-consuming, potentially hindering the efficiency of CNN applications. This can be particularly problematic in real-time scenarios where quick results are required.

Additionally, determining the optimal hyperparameters for a CNN can be a challenging and time-consuming task. As mentioned in the same article, finding the specific hyperparameters that yield the best performance often requires a significant amount of experimentation and fine-tuning. This process can be both tedious and resource-intensive, slowing down the development and deployment of CNN models.

Another limitation of CNNs is the dependence on a limited dataset. As stated in an article on Aspiring Youths, CNNs tend to perform best on large and varied datasets. However, compared to other datasets, the commonly used 60,000 training examples might be considered relatively small. Deep architectures like CNNs thrive on having access to a wide range of data to learn and generalize from, making the limited dataset a potential hindrance to their effectiveness.

Furthermore, understanding the decision-making process of a CNN and debugging any issues can be challenging. As the OpenGenus IQ article highlights, comprehending the inner workings of CNNs can be difficult due to their complex structure and numerous layers. This complexity makes it harder to trace and identify errors or biases within the network, making debugging a time-consuming and intricate process.

While CNNs have revolutionized image recognition and computer vision tasks, they do have certain limitations. The time consumption associated with running the model, the challenges in determining optimal hyperparameters, the reliance on limited datasets, and the difficulty in understanding and debugging the decision-making process are all significant drawbacks. Acknowledging these limitations is essential for researchers and practitioners to effectively address the challenges and maximize the potential of CNNs in various applications.

Conclusion - Max

In conclusion, the classification of handwritten digits holds significant importance in various technological fields, given the extensive application of digits in areas such as banking, engineering, and postal services. The need for accurate digit classification arises from the unique handwriting styles of individuals, leading to specific variations in the visual representation of each character. With the increasing demand for pattern recognition applications, accurate digit classification becomes essential for the development of efficient systems that can interpret and process diverse handwritten inputs

effectively. This paper has explored the impact of modifying the learning rate on the performance of a variation of the MNIST dataset, aiming to gain insights into how this hyperparameter affects the accuracy and efficiency of the model. While Convolutional Neural Networks (CNNs) have revolutionized image recognition and computer vision tasks, they do have limitations, including time consumption, challenges in determining optimal hyperparameters, reliance on limited datasets, and difficulties in understanding and debugging the decision-making process. Recognizing and addressing these limitations will allow researchers and practitioners to overcome challenges and maximize the potential of CNNs in various applications.

Works Cited

Banjare, Kiran, and Massey, Sampada. “Handwritten Numeric Digit Classification and Recognition: Recent Advancements.” International Journal of Emerging Technologies in Engineering Research (IJETER), vol. 4, no. 6, June 2016. Accessed 11 June 2023. https://www.ijeter.everscience.org/Manuscripts/Volume-4/Issue-6/Vol-4-issue-6-M-48.pdf.

Bhuiya, Sandeep. “Disadvantages of CNN.” OpenGenus IQ. Accessed 11 June 2023. https://iq.opengenus.org/disadvantages-of-cnn/.

Brownlee, Jason. “How to Develope a CNN for MNIST Handwritten Digit Classification.” 8 May, 2019. https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-from-scratch-for-mnist-handwritten-digit-classification/.

“Advantages and Disadvantages of Convolutional Neural Network (CNN).” Aspiring Youths. Accessed 11 June 2023. https://aspiringyouths.com/advantages-disadvantages/convolutional-neural-network-cnn/.