AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH

Project Report Cover Sheet

Course Name	COMPUTER VISION AND PATTERN RECOGNITION
Report	Implement a CNN architecture to classify the MNIST handwritten dataset.
Submitted To	Dr. Debajyoti Karmaker
Semester	Fall 2021-2022
Section	A

Student Name	Student ID	Department	Section
Md. Sohanur Rahman Sohan	17-35357-3	CoE	A

For faculty use only:	**Total Marks: _____ Marks Obtained: _____**

Faculty comments_____

_____

_____

Abstract

The goal of this research was to propose a simple Convolutional Neural Network (CNN) model to classify MNIST handwriting dataset with an output of more than 98 percent while evaluating various optimizers. In deep learning, a CNN is a sort of artificial neural network that is used to evaluate visual data. Convolutional neural networks (CNNs) are neural networks that evaluate images, classify data, and segment it using one or more convolutional layers. Convolutional neural networks (CNNs) are neural networks containing one or more convolutional layers that are used to analyse images, classify data, and segment it. I used CNN architecture to categorize the MNIST handwritten dataset in this research. To test different levels of accuracy, I employed three types of optimizers: ADAM, SGD, and RMSProp.

Introduction

A CNN, or convolutional neural network, is a deep learning neural network that is meant to analyze organized arrays of data such as photographs. Convolutional neural networks are widely utilized in computer vision and have advanced to the cutting edge of many visual applications such as picture classification. MNIST is a database of labelled images of the handwritten digit image class, with a train image set of 6000 images and a test image set of 10000 images, each of 28*28 pixels. Every pixel has a value between 0 and 255. Optimizers are strategies or approaches for lowering losses by altering the weights and learning rate of neural networks. Adam is an optimization approach for updating network weights based on training data that replaces the traditional stochastic gradient descent process. Adam is a well-known deep learning method because it produces quick and accurate results. SGD is a method for determining an objective function's smoothness qualities. ADAM, on the other hand, is much faster than SGD. RMSprop is a gradient-based optimization method used in neural network training.

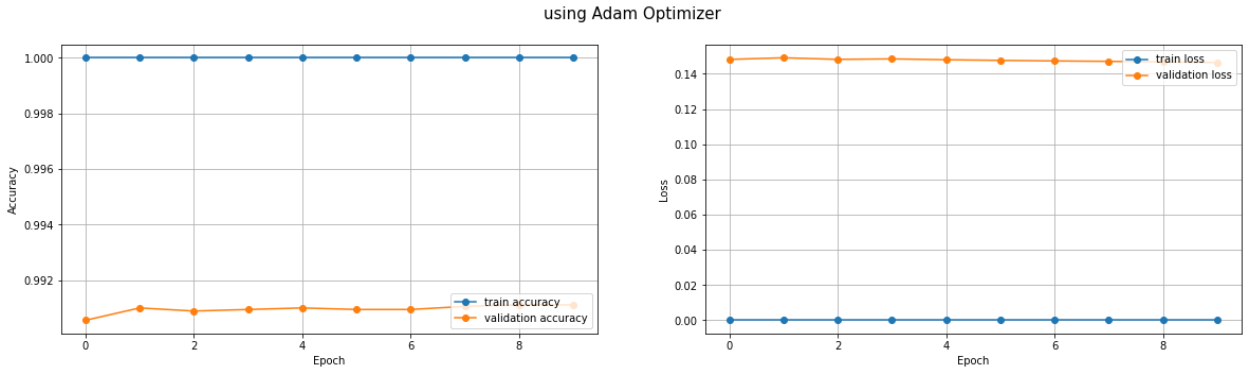
Results

Here ADAM optimizer was used and the test accuracy is 99.29%.

```
[26] test_loss, test_accuracy = model.evaluate(X_test, Y_test)
     print(f'\nTest accuracy: {test_accuracy}')

     313/313 [==============================] - 2s 7ms/step - loss: 0.0968 - accuracy: 0.9929

     Test accuracy: 0.992900013923645
```

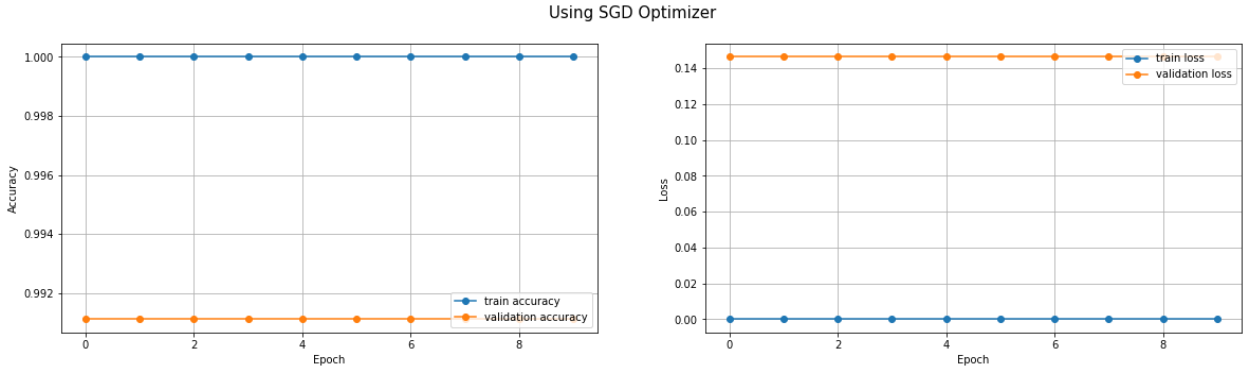


Here SGD optimizer was used and the test accuracy is 99.30%.

```
[36] test_loss, test_accuracy = model.evaluate(X_test, Y_test)
     print(f'\nTest accuracy: {test_accuracy}')
```

```
313/313 [==============================] - 2s 6ms/step - loss: 0.0964 - accuracy: 0.9930

Test accuracy: 0.9929999709129333
```

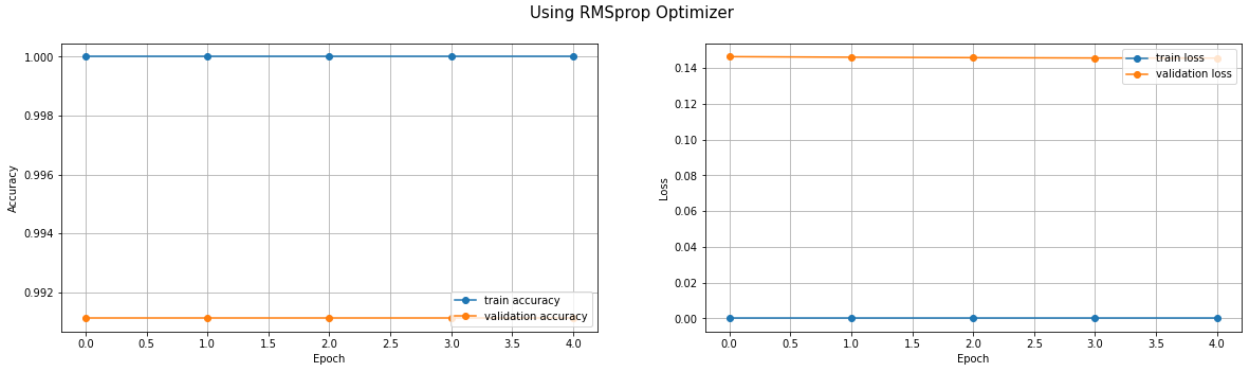


Here RSMprop optimizer was used and the test accuracy is 99.29%.

```
[32] test_loss, test_accuracy = model.evaluate(X_test, Y_test)
     print(f'\nTest accuracy: {test_accuracy}')
```

```
313/313 [==============================] - 2s 6ms/step - loss: 0.0968 - accuracy: 0.9929

Test accuracy: 0.992900013923645
```



Discussion

In this report, I use three different types of optimizers: ADAM, SGD, and RMSProp. As a result, I discovered a minor discrepancy in their accuracy. SGD and RMSProp are much slower than ADAM. Adam optimizer builds on the qualities of earlier models to provide substantially better performance than previously utilized models and outperforms them by a large margin in terms of providing an optimized gradient descent. The model determines the accuracy. It varies depending on the size, filters, and other factors. I attempted to achieve accuracy of greater than 98 percent, and for the Adam, SGD, and RMSprop optimizers, I achieved greater than 99 percent. In terms of computation time, RMSprop was the quickest. However, all of the optimizers performed admirably, with accuracy above 99 percent.