MID PROJECT REPORT

Effat Jahan, ID: 18-38718-3

October 2021

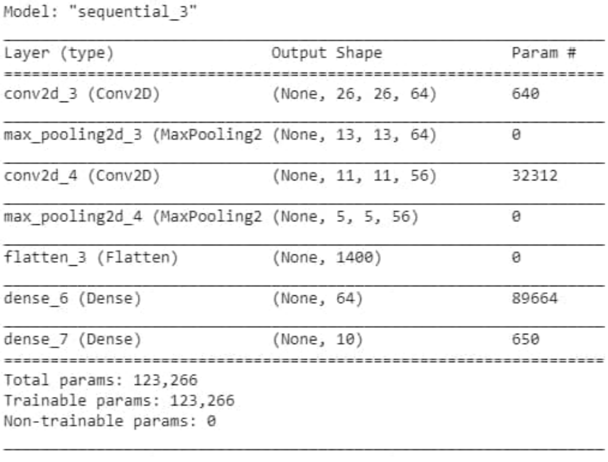
1 Abstract

Handwritten digit recognition is the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screens, etc. As handwriting differ from person to person. The goal of this project is to propose a **CNN** (*Convolutional Neural Network*) model for **MNIST** dataset which will produce a accuracy over 98%. Three different optimizers (**Adam, SGD, RMSprop**) will be used to get the best result possible.

2 Introduction

A Convolutional neural network (CNN) is a neural network that has one or more convolutional layers and are used mainly for image processing, classification, segmentation and also for other auto correlated data. CNNs have been used for understanding in Natural Language Processing (NLP) and speech recognition, although often for NLP Recurrent Neural Nets (RNNs) are used. Each convolutional layer contains a series of filters known as convolutional kernels. The filter is a matrix of integers that are used on a subset of the input pixel values, the same size as the kernel. Each pixel is multiplied by the corresponding value in the kernel, then the result is summed up for a single value for simplicity representing a grid cell, like a pixel, in the output channel/feature map. The **MNIST** (*Modified National Institute of Standards and Technology*)

database of handwritten digits, has a training set of 60,000 examples, and a test set of 10,000 examples. Each image contains 28*28 pixels where each pixel has a value from 0 to 255. I used 4 different layers including Conv2D layers, MaxPooling2D layers, Flatten layer and Dense layers in my model and used different optimizers to get the maximum accuracy. I have used two activation functions which includes *ReLU* and *Softmax* My model Summary is given below-



```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 26, 26, 64)        640
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 13, 13, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 11, 11, 56)        32312
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 5, 5, 56)          0
_________________________________________________________________
flatten_3 (Flatten)          (None, 1400)              0
_________________________________________________________________
dense_6 (Dense)              (None, 64)                89664
_________________________________________________________________
dense_7 (Dense)              (None, 10)                650
=================================================================
Total params: 123,266
Trainable params: 123,266
Non-trainable params: 0
_________________________________________________________________
```

Figure 1: Model Summary

Form this model summary we can see, I have used two conv2D layers and two MaxPooling2D layers. I have also used a flatten layer and two dense layer where one of them is the output layer

3 Results

The result of the model using different optimizers is given below -

Optimizer	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss	Test Accuracy
Adam	99.69%	0.89%	98.46%	6.95%	98.67 %
SGD	100%	0.059%	99.07%	4.64%	99.22%
RMSprop	99.99%	0.026%	99.14%	13.57%	99.17%

The training and validation accuracy, training validation loss after using Adam optimizer is shown in the graph below-

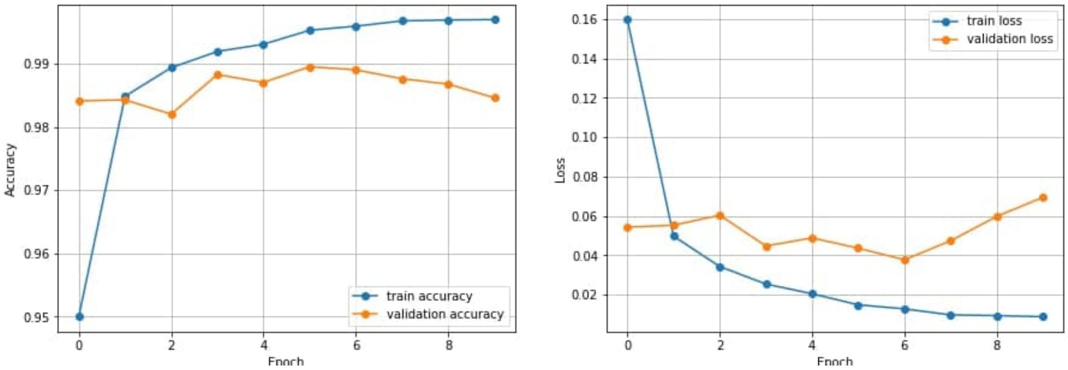


Figure 2: Graph data of Adam Optimizer

The training and validation accuracy, training validation loss after using SGD optimizer is shown in the graph below-

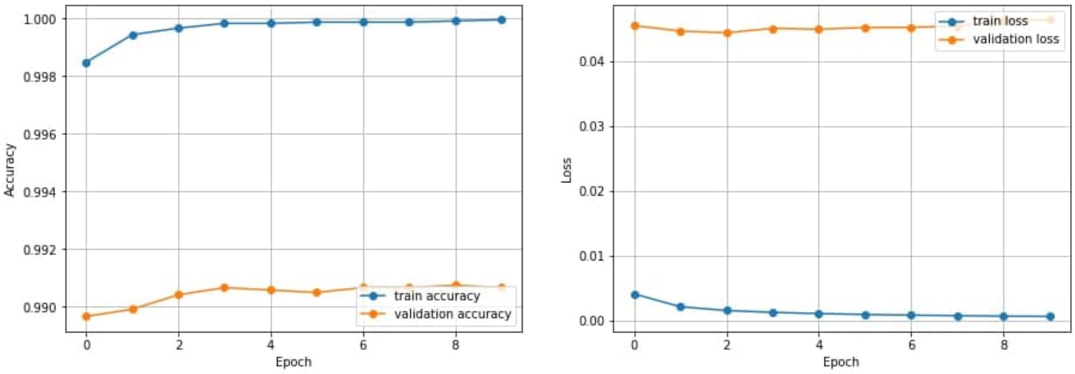


Figure 3: Graph data of SGD Optimizer

The training and validation accuracy, training validation loss after using RMSprop optimizer is shown in the graph below-

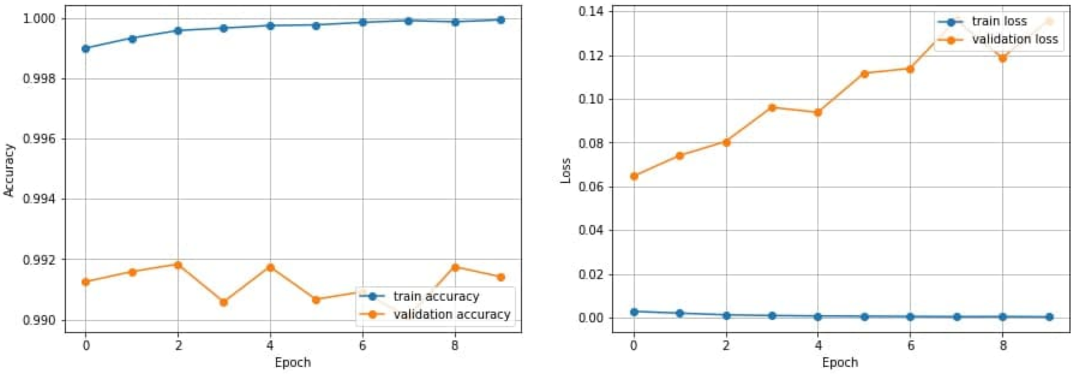


Figure 4: Graph data of RMSprop Optimizer

4 Discussion

From the table data we can see that, on all three optimizer the proposed model accuracy is over 98%.

While using Adam optimizer the training accuracy is 99.69%, testing accuracy is 98.67% and the validation accuracy is 98.46%. In case of train loss and validation loss the data is 0.89% and 6.95% respectively.

While using SGD optimizer the training accuracy is 100%, testing accuracy is 99.22% and the validation accuracy is 99.07%. In case of train loss and validation loss the data is 0.059% and 4.64% respectively.

While using RMSprop optimizer the training accuracy is 99.99%, testing accuracy is 99.17% and the validation accuracy is 99.14%. In case of train loss and validation loss the data is 0.026% and 13.57% respectively.

Although, RMSprop closest value with validation accuracy and testing accuracy and had the lowest training loss It is not the best option as the validation loss is the highest. Considering all the data, it can be concluded that we can get the best output from this model by using SGD optimizer. It gives the highest training and testing accuracy with the lowest validation loss.