Computer Vision and Pattern Recognition [B]

Submitted by: Akash, Md Mahabubul Hasan

ID: 18-38043-2

Title:

Evaluation of proposed convolutional neural network model to classify the MNIST dataset

Abstract:

The MNIST dataset consists of handwritten numbers and their corresponding label. In this paper, a CNN model is proposed to classify MNIST handwritten datasets that will produce over 98% accuracy by using different activation functions.

Introduction:

In deep learning, CNN is one of the most popular deep neural networks. Traditionally neural networks are built with perceptron /neurons. Perceptron is a single neural network. A feed-forward network is constructed by using these neurons. This neural network is a composite function and it multiplies matrices and vectors. But CNN uses convolution operation on two functions that produce another function that expresses how the shape of one is modified by the other. The MNIST is a large database of handwritten digits that is commonly used for training various image processing systems. The train image set consists of 60000 images and a test image set of 10000 images; the dimension of a particular image is 28*28 pixels and each image has a corresponding label.

A convolutional neural network model is designed for the training dataset and they were evaluated through Adam, SGD, RMSprop optimizers. The model summary is given below

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_2 (Conv1D)            (None, 24, 32)            4512
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 12, 32)            0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 10, 64)            6208
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 5, 64)             0
_________________________________________________________________
flatten_1 (Flatten)          (None, 320)               0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                20544
_________________________________________________________________
dense_4 (Dense)              (None, 64)                4160
_________________________________________________________________
dense_5 (Dense)              (None, 10)                650
=================================================================
Total params: 36,074
Trainable params: 36,074
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 1.1: Model Summary

Results:

Results of the CNN Model for different optimizers.

Optimizer	Train Accuracy	Validation Accuracy	Test Accuracy
Adam	99.27%	98.07%	98.21%

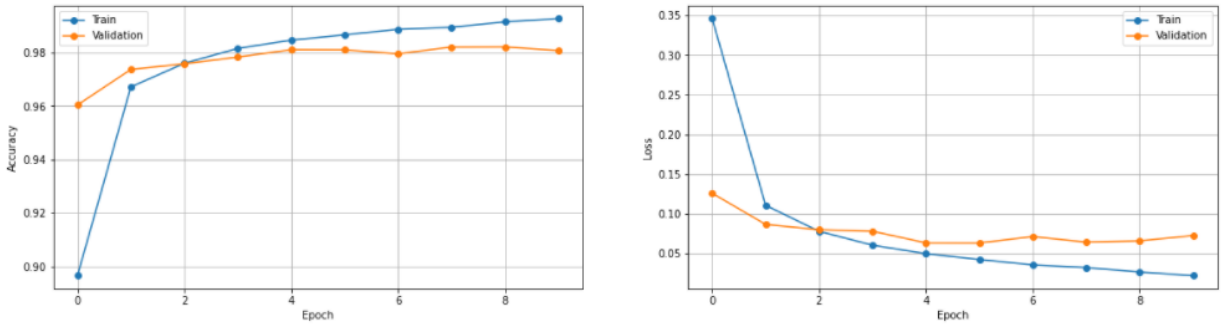

Figure 1.2: Results for Adam

Optimizer	Train Accuracy	Validation Accuracy	Test Accuracy
SGD	96.40%	96.19%	96.42%

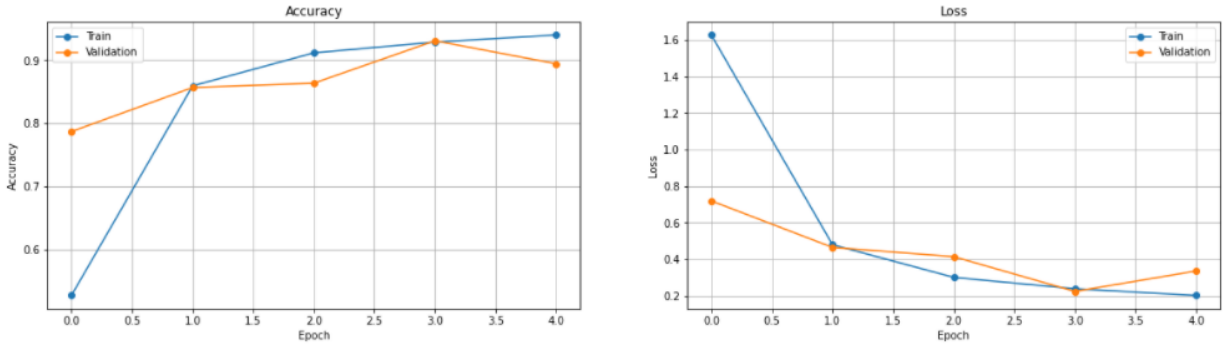

Figure 1.3: Result for SGD

Optimizer	Train Accuracy	Validation Accuracy	Test Accuracy
ADAM	99.38%	98.38%	98.61%

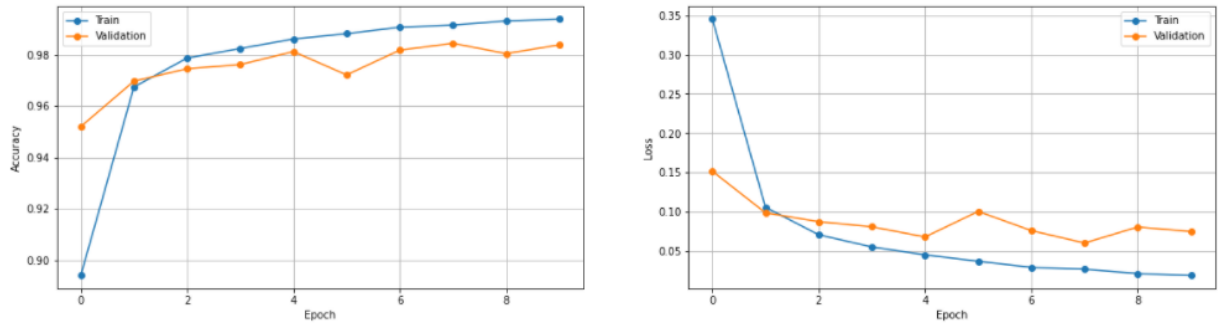

Figure 1.4: Result for ADAM

Discussion:

The designed CNN Model performance is very close for ADAM and RMSprop optimizer. The best train accuracy is 99.38% from the Adam optimizer and the lowest train accuracy is 96.40% from the SGD optimizer. As the graphs show that for the Adam optimizer, validation accuracy differs from train accuracy on the other hand the RMSprop graph is showing the most prominent result. That means in terms of real-life data Adam will not perform well. In that case, RMSprop will be a better option for the model.