Computer Vision and Pattern Recognition

Mid Project Report

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 handwritten numbers and corresponding label. A Convolutional Neural Network model is proposed to classify MNIST handwritten dataset that will produce over 98% accuracy.

Introduction: In Recent decades, Deep Learning has proved to be a very powerful tool because of its ability to handle large amounts of data. One of the most popular deep neural networks is Convolutional Neural Networks. Traditionally neural networks are built with perceptron or neurons in a feed forward network manner. Under the hood, the feed forward neural network is just a composite function, that multiplies some matrices and vectors together. On the other hand, Convolutional Neural Networks comes from mathematics where convolution is a mathematical operation on two functions that produces a third 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 60000 images and test image set of 10000 images; the dimension of a particular image is 28*28 pixels and each images has a corresponding label.

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

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_18 (Conv1D)           (None, 24, 32)            4512
_________________________________________________________________
max_pooling1d_16 (MaxPooling (None, 12, 32)            0
_________________________________________________________________
conv1d_19 (Conv1D)           (None, 10, 64)            6208
_________________________________________________________________
max_pooling1d_17 (MaxPooling (None, 5, 64)             0
_________________________________________________________________
conv1d_20 (Conv1D)           (None, 3, 128)            24704
_________________________________________________________________
max_pooling1d_18 (MaxPooling (None, 1, 128)            0
_________________________________________________________________
flatten_7 (Flatten)          (None, 128)               0
_________________________________________________________________
dense_14 (Dense)             (None, 64)                8256
_________________________________________________________________
dense_15 (Dense)             (None, 10)                650
=================================================================
Total params: 44,330
Trainable params: 44,330
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 1.1: Model – 1 Summery

The model consists one input layer and three 1-dimensional convolutional layer and 3 max pooling layer and a flatten layer.

Results:

Result of the CNN Model-1 for different optimizer are given below

Optimizer	Train Accuracy	Validation Accuracy	Test Accuracy
RMSprop	98.50%	97.92%	98.94%

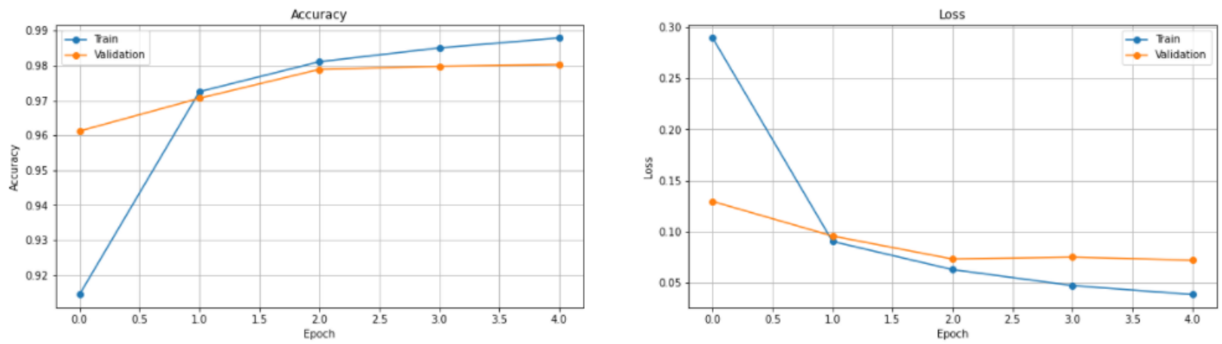

Figure 1.2: Result for RMSprop

Optimizer	Train Accuracy	Validation Accuracy	Test Accuracy
SGD	94.02%	89.45%	87.84%

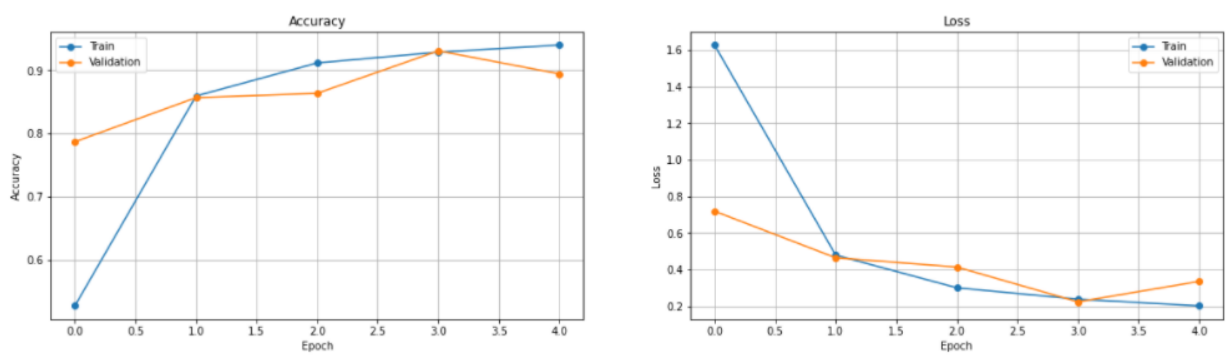

Figure 1.3: Result for SGD

Optimizer	Train Accuracy	Validation Accuracy	Test Accuracy
ADAM	98.66%	97.63%	98.86%

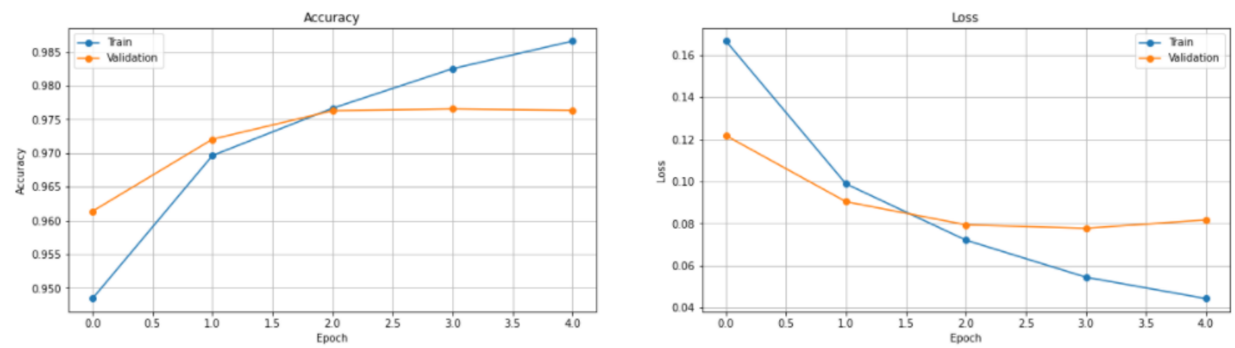

Figure 1.4: Result for ADAM

Discussion: The designed CNN Model performance are very close for ADAM and RMSprop optimizer. The best train accuracy is 98.66 % from Adam optimizer and the lowest train accuracy is 94.2% from SGD optimizer. But the graphs shows that the for the Adam optimizer the 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.