Computer Vision And Pattern Recognition [B]
Mid Project Report
Submitted By: Shohan, Siddikur Rahman
ID: 18-36667-1

Title: Evaluation of proposed CNN model to classify the MNIST handwritten dataset.

Abstract:

The automatic detection of handwritten digit from image data can be tricky as handwritten information varies person to person. My main goal of this work, is to propose a simple convolutional neural network(CNN) model to classify MNIST handwritten dataset which will produce test accuracy of over 98% and evaluate different optimizer.

Introduction:

Convolutional neural network(CNN) is a type of artificial neural network, mostly used in graphical analysis. In order to analyse image data, labeled set of particular image class are fed into a CNN model. The MNIST is a such database of labeled images of handwritten digit image class. The train image set of MNIST is consist of 60000 images and test image set of 10000 images, each image is 28*28 pixels having each pixels value of 0 to 255. Different combination of CNN can produce different results in a single dataset. I have used two different CNN model and evaluated Adam, SGD and RMSprop optimizer. 'Model 1' has a input layer followed by one dimensional convolutional layer and max pooling layer and flatten layer, after that a dense layer and finally the output layer. Below is the hyper-parameter of 'Model 1'

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_4 (Conv1D)            (None, 24, 32)            4512
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 12, 32)            0
_________________________________________________________________
flatten_3 (Flatten)          (None, 384)               0
_________________________________________________________________
dense_6 (Dense)              (None, 128)               49280
_________________________________________________________________
dense_7 (Dense)              (None, 10)                1290
=================================================================
Total params: 55,082
Trainable params: 55,082
Non-trainable params: 0
_________________________________________________________________
```

Fig1: Model 1

'Model 2' has an extra set of layer of one dimensional convolutional layer and max pooling layer. The hyper-parameter of 'Model 2' has been shown below,

```
Model: "sequential_2"
_________________________________________________________________
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_2 (Flatten)          (None, 320)               0
_________________________________________________________________
dense_4 (Dense)              (None, 128)               41088
_________________________________________________________________
dense_5 (Dense)              (None, 10)                1290
=================================================================
Total params: 53,098
Trainable params: 53,098
Non-trainable params: 0
_________________________________________________________________
```

Fig1: Model 2

Results:

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

Optimizer	Train Accuracy	Validation Accuracy	Test Accuracy
SGD	92.08%	92.27%	93.66%
Adam	98.66%	97.96%	98.33%
RMSprop	99.53%	98.45%	98.72%

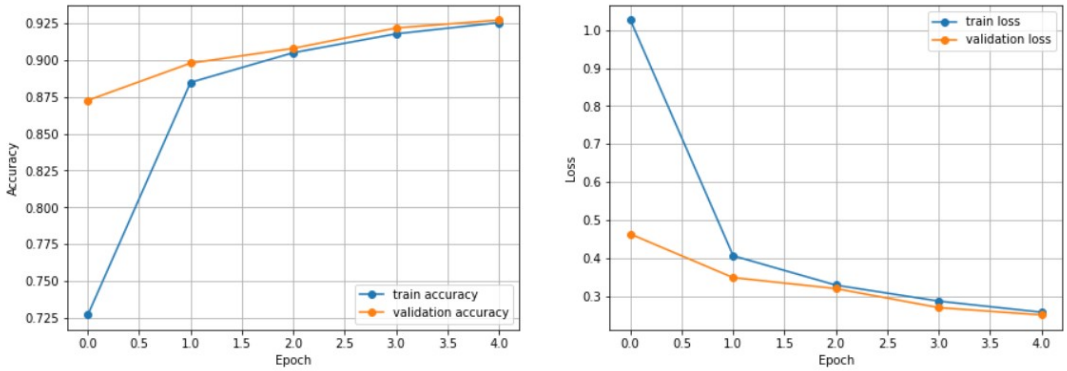

Fig3: 'Model 1' Loss for SGD

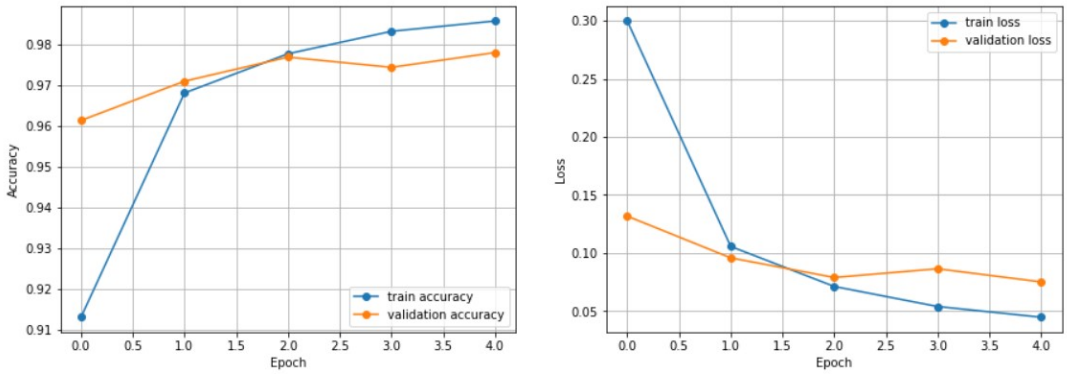

Fig4: ‘Model 1’ Loss for Adam

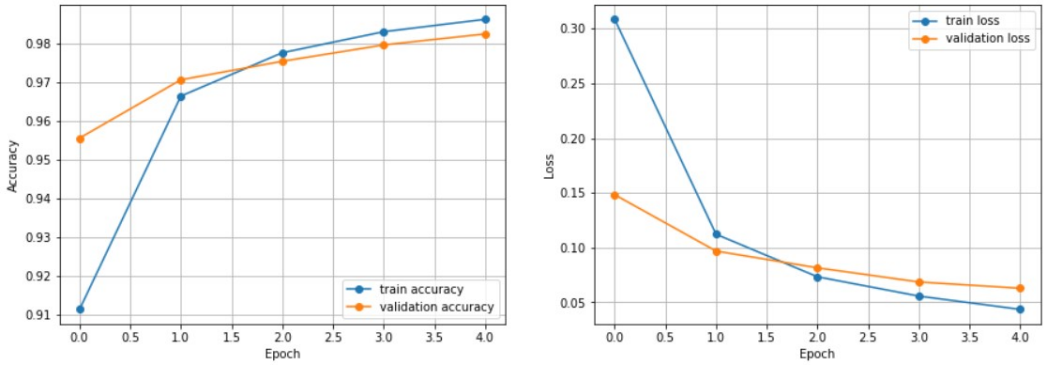

Fig5: ‘Model 1’ Loss for RMSprop

Result of the CNN ‘Model 2’ for different optimizer are given below,

Optimizer	Train Accuracy	Validation Accuracy	Test Accuracy
SGD	93.79%	94.05%	94.49%
Adam	99.39%	98.10%	98.36%
RMSprop	99.39%	98.50%	98.79%

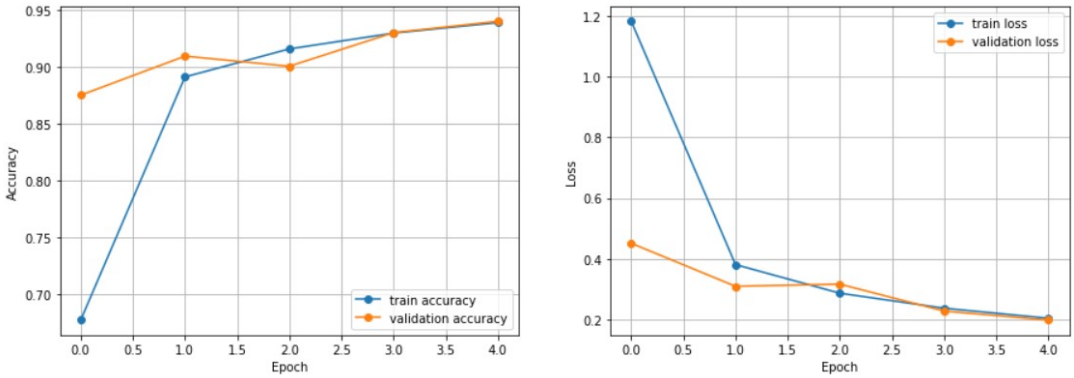

Fig6: ‘Model 2’ Loss for SGD

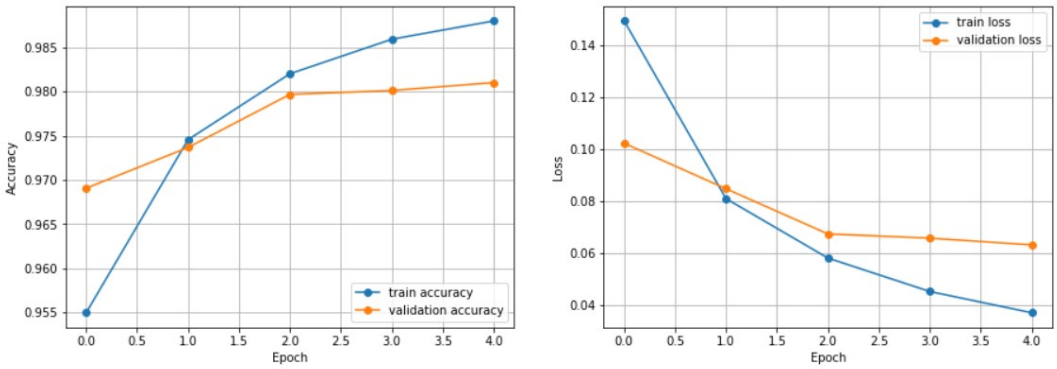

Fig7: ‘Model 1’ Loss for Adam

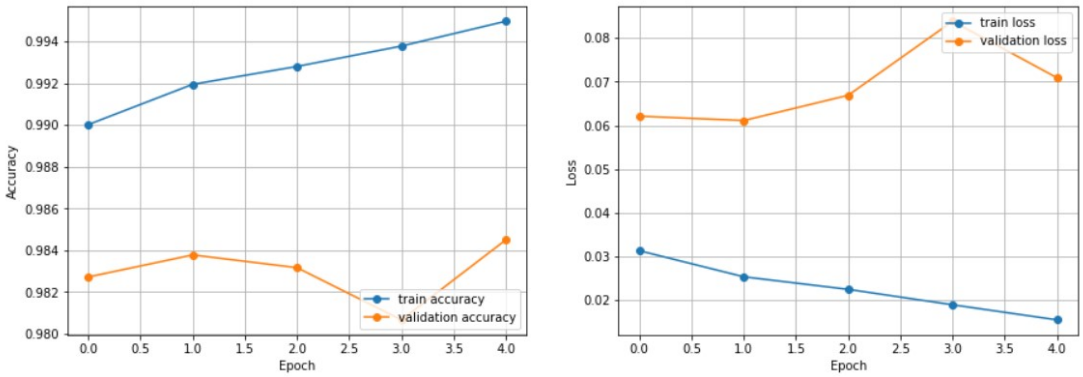

Fig8: ‘Model 1’ Loss for RMSprop

Discussion:

My proposed ‘Model 1’ provides best test accuracy of 98.72% on RMSprop optimizer and lowest test accuracy of 93.66% in SGD optimizer and ‘Model 2’ providers best test accuracy of 98.79% and lowest accuracy of 94.49% on SGD. But the graph analysis shows a different rate of Train and Validation accuracy, thus indicating the model will not perform consistent in real life data. The RMSprop optimizer of ‘Model 1’ graph indicates some what similar rate of Train and Validation accuracy, thus indicates the model will perform better in real life data.