AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh

Assignment Title: Implement a CNN architecture to classify the MNIST handwritten dataset	
Mid Report No**: 1**	Date of Submission**: 30/10/2021**
Course Title: COMPUTER VISION AND PATTERN RECOGNITION	
Course Code:	Section: **B**
Semester: **2021-2022,Fall**	Course Teacher: Dr. DEBAJYOTI KARMAKER

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

No	Name	ID	Program	Signature
1	**ARIFUL ISLAM**	**18-37545-1**	**BSC. CSE**	**ARIF**

Title: Implement a CNN architecture to classify the MNIST handwritten dataset.

Abstract:

In this report I am going to discuss about implementing a CNN architecture to classify the MNIST handwritten dataset. Also I used different optimizer like Adam, SGD, RSMProp to check which one gives best accuracy. Handwritten information varies from person to person, making automatic identification of input data from picture data difficult. The main purpose of this study is to develop a basic convolutional neural network (CNN) model to classify MNIST handwriting datasets that would achieve test accuracy of more than 98 percent and evaluate alternative optimizers.

Introduction:

A Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various objects in the image and be able to differentiate one from the other. The MNIST dataset is an acronym that stands for the Modified National Institute of Standards and Technology dataset. It is a dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9 having each pixels value of 0 to 255.The task is to classify a given image of a handwritten digit into one of 10 classes representing integer values from 0 to 9, inclusively.

Optimizers are techniques or approaches that adjust the characteristics of your neural network, such as weights and learning rate, to decrease losses.

As earlier I said I will use 3 type of optimizer and now I will discuss about them:

Adam: Adam is a deep learning model training technique that replaces stochastic gradient descent. Adam combines the finest features of the AdaGrad and RMSProp methods to provide an optimization technique for noisy issues with sparse gradients.

SGD: SGD is an iterative approach for finding the best smoothness qualities for an objective function. One popular and persuasive argument for optimizers is that SGD generalizes better than Adam.

RMSProp: Root Mean Square Propagation is abbreviated as RMSprop. In neural network training, RMSprop is a gradient-based optimization strategy.

I tested Adam, SGD, and RMSprop optimizers with 1 CNN models.Here,

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 24, 24, 32)        832
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 12, 12, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 10, 10, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 1600)              0
_________________________________________________________________
dense (Dense)                (None, 64)                102464
_________________________________________________________________
dense_1 (Dense)              (None, 10)                650
=================================================================
Total params: 122,442
Trainable params: 122,442
Non-trainable params: 0
_________________________________________________________________
```

Fig:1

Fig 1 is the hyper-parameter picture. It has an input neurons, a two-dimensional convolutional neural network, a maximum pooling layer, a flattening layer, a dense layer, and finally the output layer.

Results:
Result of the different optimizer result are given below,

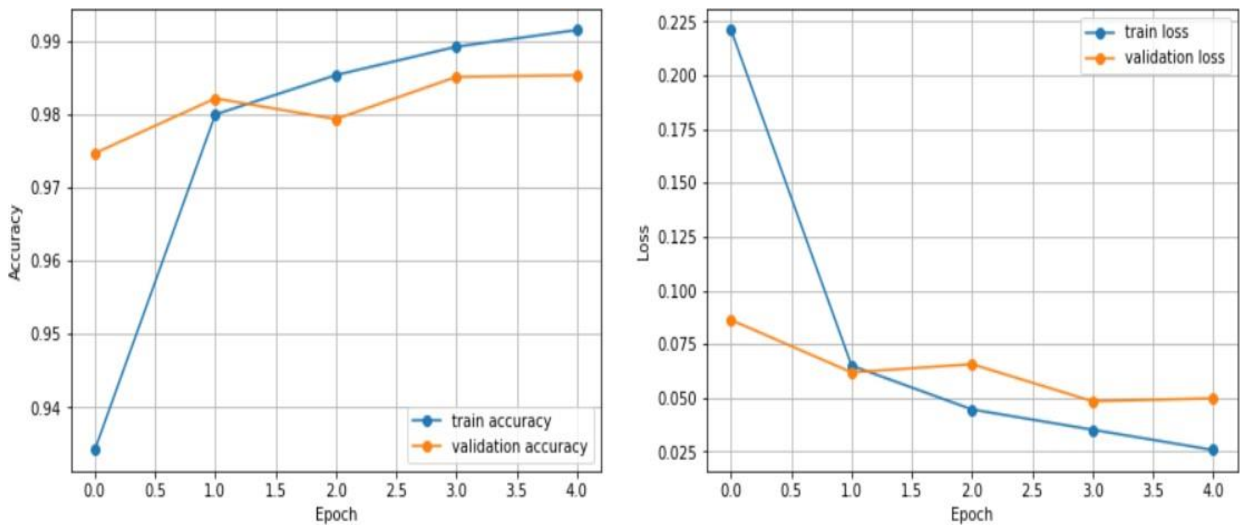

Fig2: Loss of Adam

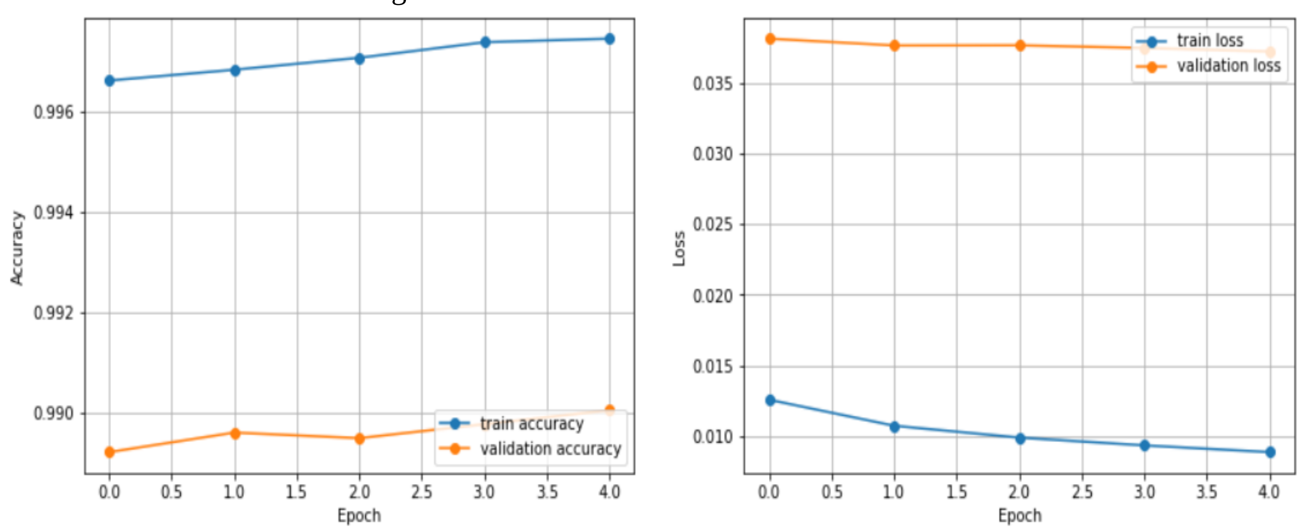

Fig3: Loss of SGD

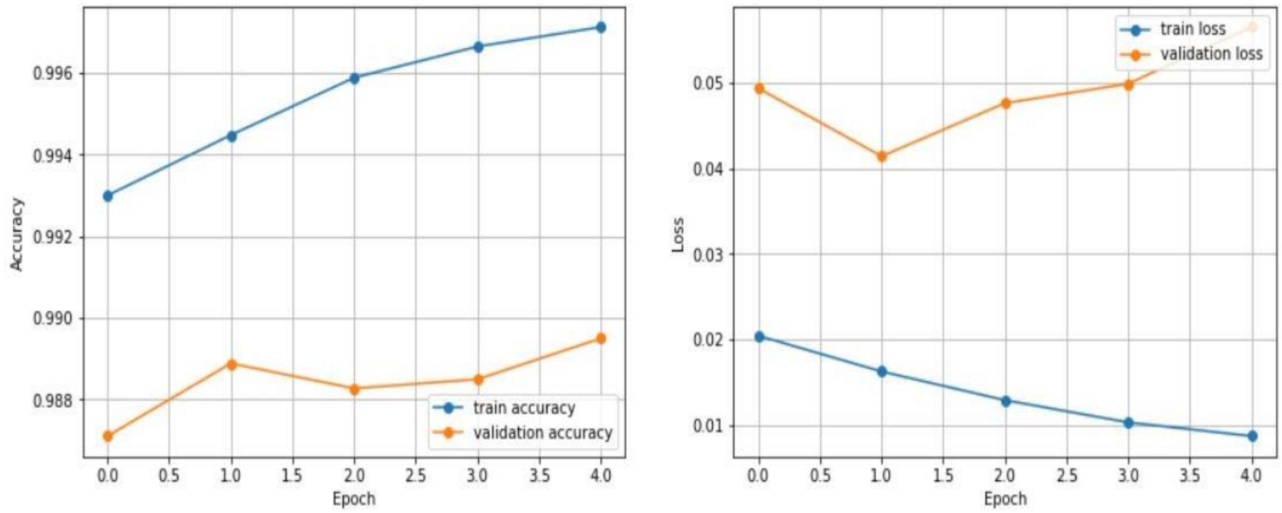

Fig4: Loss of RMSProp

Optimizer	Train Accuracy	Validation Accuracy	Test Accuracy	Test Loss
Adam	99.23%	98.54%	98.82%	3.89%
SGD	99.74%	99.01%	99.11%	2.73%
RMSprop	99.73%	98.95%	99.19%	3.86%

Discussion:

So after using all these, I can say according to my test Adam gives a good result and RMSProp also gives a better result**.** My proposed method has the highest test accuracy of 99.19 percent on RMSprop optimizer and the lowest test accuracy of 98.82 percent on SGD optimizer. The graph analysis reveals a difference in the rates of Train and Validation correctness, indicating that the model will not perform consistently in real-world data. The RMSprop optimizer of the graph shows a somewhat comparable rate of Train and Validation accuracy, indicating that the model will perform better on real-life data.