AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Choose an item.

Project Cover Sheet

Assignment Title:	mid_report		
Assignment No:	02	Date of Submission:	Click here to enter a date.
Course Title:	COMPUTER VISION AND PATTERN RECOGNITION		
Course Code:	Click here to enter text.	Section:	A
Semester:	Spring 2019-20	Course Teacher:	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 collaborationhas 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 thatPlagiarism 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.

* *Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.:

No	Name	ID	Program	Signature
1	Rafshan Bin Razzak	18-38310-2	BSc [CSE]	
2			Choose an item.	
3			Choose an item.	
4			Choose an item.	
5			Choose an item.	
6			Choose an item.	
7			Choose an item.	
8			Choose an item.	
9			Choose an item.	
10			Choose an item.	

Faculty use only		
FACULTYCOMMENTS	**Marks Obtained**	
	Total Marks	

Abstract:

Implement a CNN architecture to classify the MNIST handwritten dataset based and test it with different optimizer (Adam, SGD, RSMProp)

Introduction:

Mnist Handwritten Datasets are test datasets used to train and test purpose in machine learning industry. This is a large datasets where all the characters are handwritten. All the images are grey scaled and the resolution is 28*28 pixels.

CNN or convolutional neural network artitecture is a layer based architecture basically used to recognize the written digits. It was created by Mr. Yann LayCun in 1998.

Optimizers determines the way to get the best accuracy by changing the model parameters. In Simple way optimizers works to shape the model in the most accurate way by changing different parameters.

Adam: Gradient descent approaches have a good use in practical life. Adam or adaptive moment estimation is another way of using past gradients to calculate current gradients. This optimizer is practically accepted for use in training neural nets.

SGD: SGD class that implements the stochastic gradient descent optimizer with a learning rate and momentum. First, an instance of the class must be created and configured, then specified to the "optimizer" argument when calling the fit() function on the model.

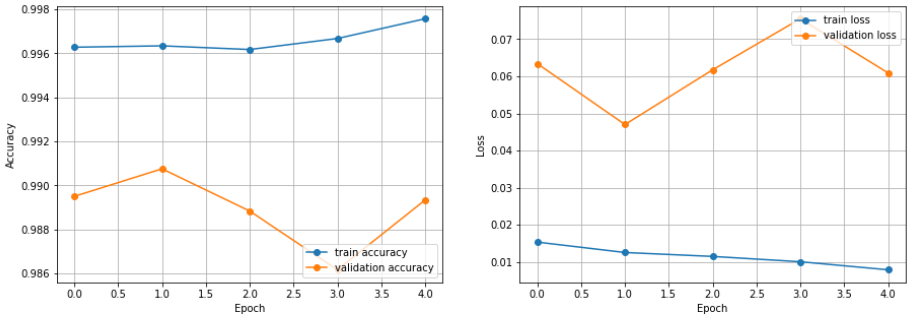
RMSprop :The RMSprop optimizer restricts the oscillations in the vertical direction. Therefore, we can increase our learning rate and our algorithm could take larger steps in the horizontal direction converging faster.

Result:

For Adam Optimizer:

Test Accuracy: 99%

Loss Rate: 0.4%

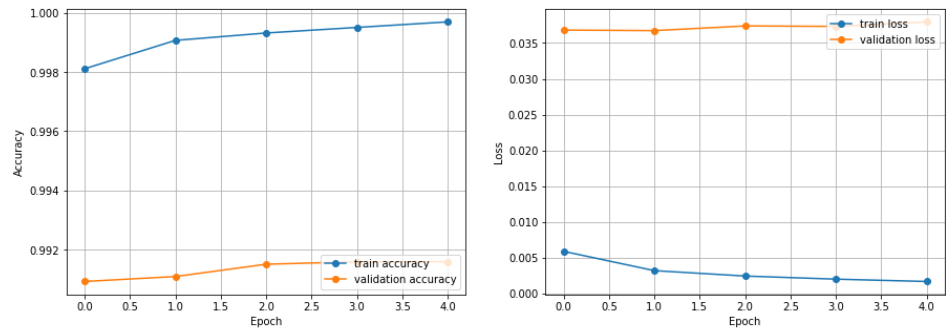



```
In [62]: test_loss, test_acc = model.evaluate(x_test, y_test)
         print('\nTest accuracy:', test_acc)
         print('\nTest loss:', test_loss)

         313/313 [==============================] - 5s 14ms/step - loss: 0.0457 - accuracy: 0.9911

         Test accuracy: 0.991100013256073

         Test loss: 0.04568065330386162
```

For SGD Optimizer:

Test Accuracy: 99%

Loss Rate: 0.3%

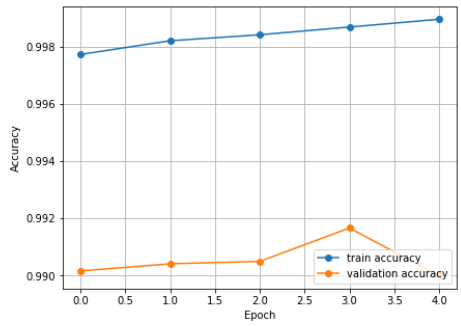


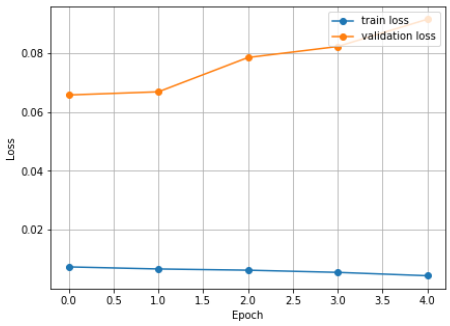

```
49]: test_loss, test_acc = model.evaluate(x_test, y_test)
     print('\nTest accuracy:', test_acc)
     print('\nTest loss:', test_loss)

     313/313 [==============================] - 4s 13ms/step - loss: 0.0311 - accuracy: 0.9920

     Test accuracy: 0.9919999837875366

     Test loss: 0.031087854877114296
```

For RMSprop Optimizer:

Test Accuracy: 99%

Loss Rate: 0.7%






```
: test_loss, test_acc = model.evaluate(x_test, y_test)
  print('\nTest accuracy:', test_acc)
  print('\nTest loss:', test_loss)
```

```
313/313 [==============================] - 4s 13ms/step - loss: 0.0768 - accuracy: 0.9906

Test accuracy: 0.9905999898910522

Test loss: 0.0768323689699173
```

Conclusion: The accuracy depends on the model. It varies based on the size, filters and everything. I tried to get accuracy over 98% and I have got over 99% for the given Adam,SGD and RMSprop optimizer. In case of time RMSprop took the least time to calculate. But overall all the optimizers have worked fine and accuracy is nearly 99%.