American International University-Bangladesh (AIUB)

Department of Computer Science

COMPUTER VISION AND PATTERN RECOGNITION

Midterm Project Report

Name: Sadia Afrin Swarna

ID: 18-36522-1

Section: A

Abstract: In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of artificial neural network, most commonly applied to analyze visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation equivariant responses known as feature maps. Counter-intuitively, most convolutional neural networks are only equivariant, as opposed to invariant, to translation. They have applications in image and video recognition, recommender systems, image classification, image segmentation. In this report, I have written details about the implementation of CNN architecture to classify the MNIST hand written dataset which has been uploaded before. To classify the MNIST dataset, I used 3 types of optimizer ADAM, SGD, RMSProp.

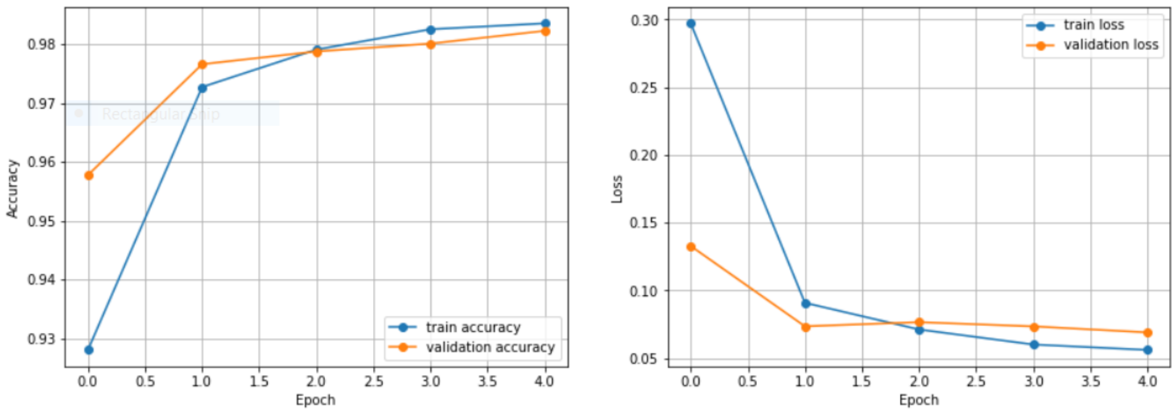
Introduction:

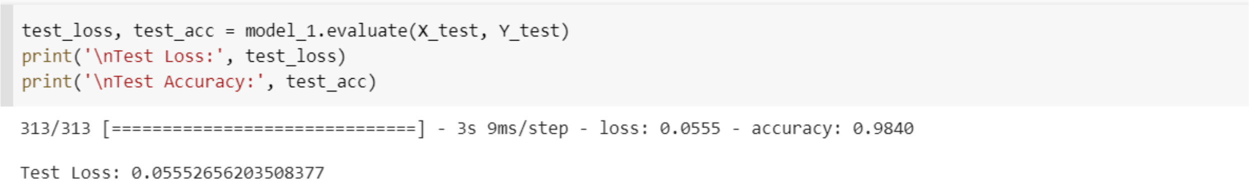
Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models. Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems. Adam is relatively easy to configure where the default configuration parameters do well on most problems.

Stochastic gradient descent (often abbreviated SGD) is an iterative method for optimizing an objective function with suitable smoothness properties (e.g. differentiable or subdifferentiable). It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data). Especially in high-dimensional optimization problems this reduces the computational burden, achieving faster iterations in trade for a lower convergence rate.

Root Mean Squared Propagation, or RMSProp, is an extension of gradient descent and the AdaGrad version of gradient descent that uses a decaying average of partial gradients in the adaptation of the step size for each parameter.

Results: For the results , I used ADAM optimizer and got 98.40% accuracy and loss 5.5%.



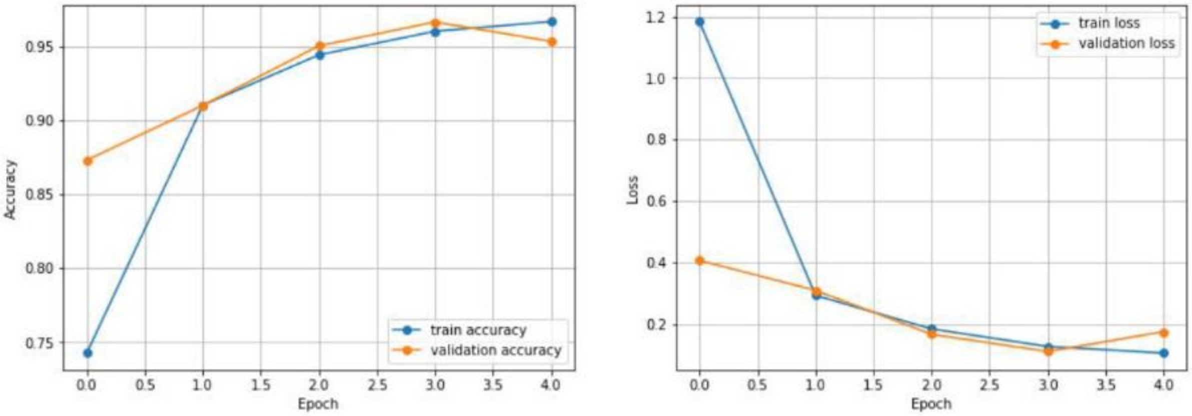


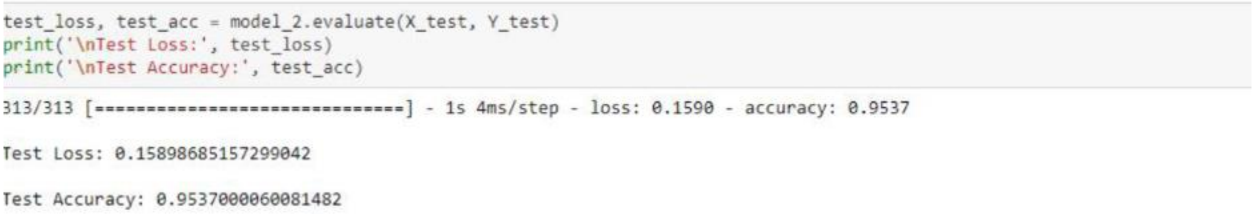
```
test_loss, test_acc = model_1.evaluate(X_test, Y_test)
print('\nTest Loss:', test_loss)
print('\nTest Accuracy:', test_acc)
```

```
313/313 [==============================] - 3s 9ms/step - loss: 0.0555 - accuracy: 0.9840

Test Loss: 0.05552656203508377
```

Then I Used SGD Optimizer with 95.37% accuracy.





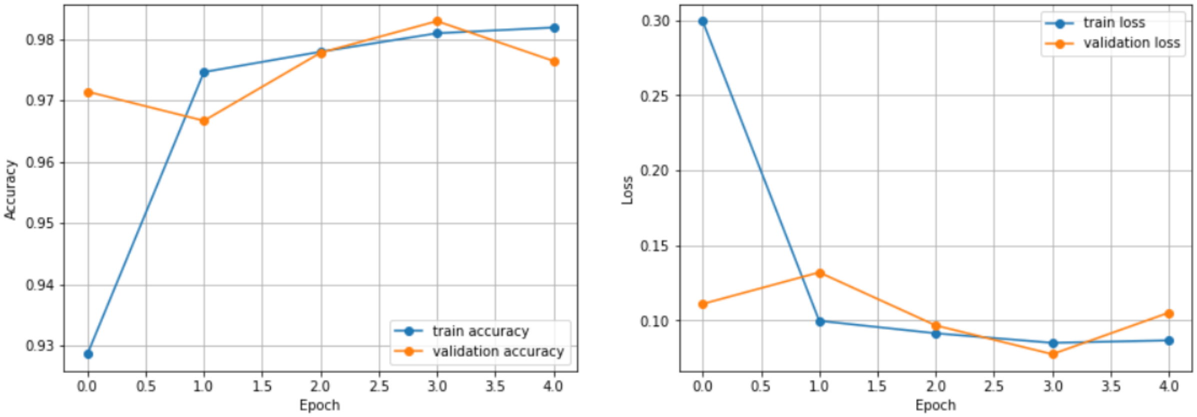
```
test_loss, test_acc = model_2.evaluate(X_test, Y_test)
print('\nTest Loss:', test_loss)
print('\nTest Accuracy:', test_acc)
```

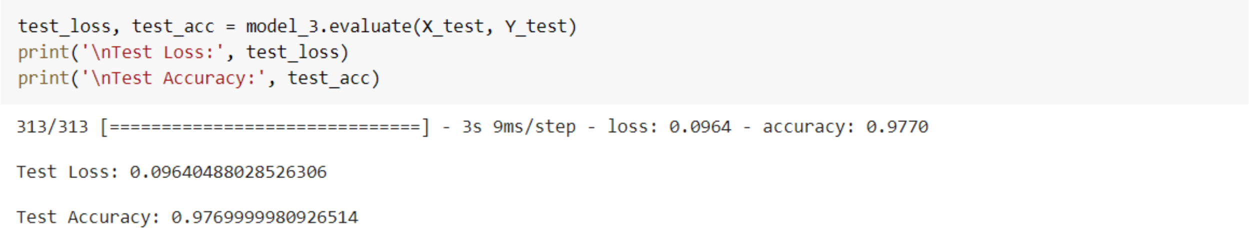
```
313/313 [==============================] - 1s 4ms/step - loss: 0.1590 - accuracy: 0.9537

Test Loss: 0.15898685157299042

Test Accuracy: 0.9537000060081482
```

At last RGBProp and the accuracy is 97.70%.





```
test_loss, test_acc = model_3.evaluate(X_test, Y_test)
print('\nTest Loss:', test_loss)
print('\nTest Accuracy:', test_acc)
```

```
313/313 [==============================] - 3s 9ms/step - loss: 0.0964 - accuracy: 0.9770

Test Loss: 0.09640488028526306

Test Accuracy: 0.9769999980926514
```

Discussion:

In this Project, I have used ADAM, SGD AND RMSProp for optimization. I am performing experiments on the EMNIST validation set using networks with RMSProp, Adam and SGD. I am achieving 95.37% accuracy with SGD. When testing the same exact configuration with RMSProp the accuracy 97.70% and Adam I am achieving accuracy of 98.40%. At first I faced some problem. Then I rerun the project and got the results.

References:

1. https://machinelearningmastery.com/adam-optimizationalgorithm-fordeep-learning/

2. https://ruder.io/optimizing-gradient-descent/

3. https://medium.com/analytics-vidhya/a-complete-guide-to-adamandrmsprop-optimizer-75f4502d83b

4.https://r.search.yahoo.com/_ylt=Awr9DtahLH1h1zAAFBVXNyoA;_ylu=Y29sbwNncTEEcG9zAzEEdnRpZAMEc2VjA3Nj/RV=2/RE=1635622177/RO=10/RU=http%3a%2f%2fen.wikipedia.org%2fwiki%2fConvolutional_neural_network/RK=2/RS=1WThgzdq_jM_g1lmFU1mMA658t8-