AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

MIDTERM PROJECT

COMPUTER VISION AND PATTERN RECOGNITION

Name	Id	Section
Hasebul Hasan Shihab	18-37031-1	A

Answer To the question no 3

Abstract:-

CNN architecture is for Convolutional Neural Network, and it is a model that has gained a lot of popularity in recent years as a result of its utility. To do computations, CNN employs multilayer perceptrons. In comparison to other image classification methods, CNN employs very little pre-processing. This means that the network picks up information from filters that were previously hand-crafted in traditional methods. As a result, CNN's are the ideal option for image processing tasks.

I use CNN architecture to identify the MNIST handwritten dataset in this report. I apply three types of optimizers to identify the MNIST dataset (ADAM, SGD, RMSProp). In this report analyze the efficiency of different optimizers (ADAM, SGD & RMSProp).

Introduction:-

Optimizers are techniques or approaches that adjust the parameters of a neural network, such as weights and learning rate, to reduce losses. Optimizers acquire faster output. We employ three different types of optimizers in this report. They are ADAM,SGD and RMSProp.

Adam is a deep learning deep learning model training algorithm that replaces stochastic gradient descent. It's a stochastic gradient descent modification that's been increasing popularity in recent years for deep learning applications in computer vision and natural language processing. An iterative method for optimizing an objective function with sufficient smoothness qualities is stochastic gradient descent. It moves at a considerably slower pace than ADAM. Then the Root Mean Square Propagation is abbreviated as RMSprop. It's an unreleased but well-known conjugate gradient optimization technique for neural network mini-batch learning. In neural network training, RMSprop is a gradient-based optimization strategy.

Result:-

```
[18] test_loss_1, test_accuracy_1 = model_1.evaluate(x_test,y_test)
     print(test_loss_1*100, test_accuracy_1*100)

     313/313 [==============================] - 2s 6ms/step - loss: 0.0493 - accuracy: 0.9842
     4.932256042957306 98.42000007629395
```

Here ADAM optimizer used the test accuracy is 0.9842that means 98.42%. The test loss is 0.0493 that means 0.04% loss.

```
[13] test_loss_2, test_accuracy_2 = model_2.evaluate(x_test,y_test)
     print(test_loss_2*100, test_accuracy_2*100)

     313/313 [==============================] - 2s 6ms/step - loss: 0.1754 - accuracy: 0.9429
     17.540858685970306 94.2900002002716
```

Here SGD optimizer used the test accuracy is 0.9429 that means 94.29%. The test loss is 0.175% .

```
test_loss_3, test_accuracy_3 = model_3.evaluate(x_test,y_test)
print(test_loss_3*100, test_accuracy_3*100)
```

```
313/313 [==============================] - 2s 5ms/step - loss: 0.0539 - accuracy: 0.9838
5.391797423362732 98.37999939918518
```

RMSProp optimizer and the test accuracy is 0.98379999939918518 that means 98.38%. The test loss is 0.0539% .

Discussion:-

ADAM, SGD, and RMSProp are the three types of optimizers used in this paper. As a result, I discovered a minor discrepancy in their precision. SGD and RMSProp are a lot slower than ADAM. Adam optimizer builds on the capabilities of prior models to provide significantly better performance and exceeds them by a large margin in terms of providing an optimized gradient descent. My accuracy with ADAM is 98.42%. Then there's SGD, which is a decent second better optimizer, and my SGD accuracy is 94.29% . The last one is RMSProp, and my RMSProp is 98.38%, which is somewhat less accurate than ADAM and SGD. As a result, we can say that the ADAM optimizer is the fastest.