Report Mid Project

Name : Al Amin Siddique

ID: 18-37870-2

Subject : Computer Vision and Pattern Recognition

Abstract:

In this report I am have implemented a CNN architecture to classify the MNIST handwritten dataset. Here also different optimizer like Adam, SGD, RSMProp has been used to check which one gives best accuracy.

Introduction:

Optimizers are techniques or approaches that adjust the characteristics of neural network, such as weights and learning rate, to decrease losses. Optimization algorithms or methods are in charge of lowering losses and delivering the most accurate outcomes. Optimizers are algorithms or techniques for changing the properties of neural network, such as weights and learning rate, in order to decrease losses. Optimization algorithms or strategies are in charge of decreasing losses and providing the most accurate results feasible. Here some discussion about 3 type of optimizer:

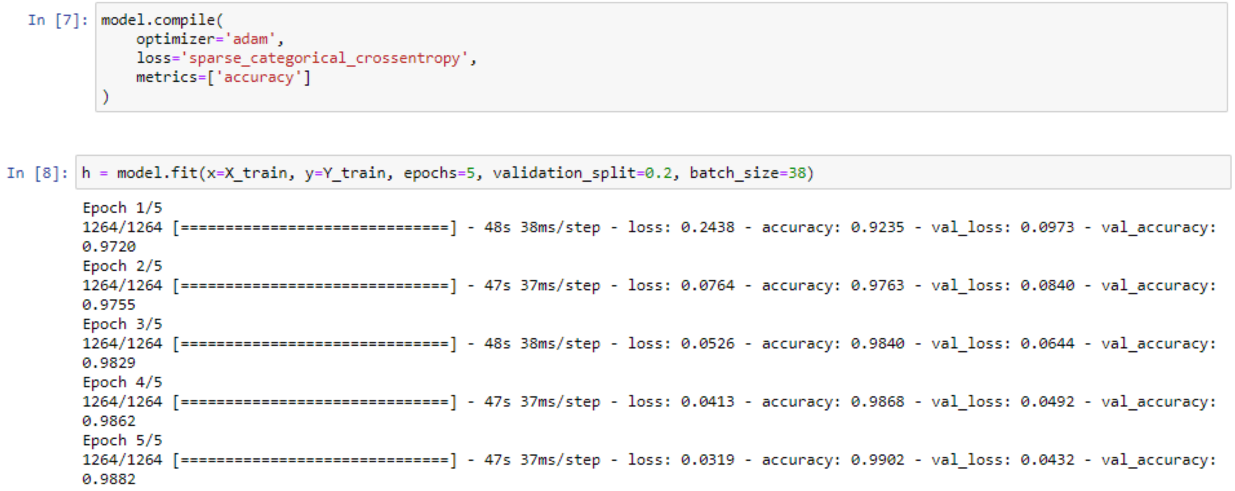
Adam: Adam is a deep learning model training technique that replaces stochastic gradient descent. Adam combines the finest features of the AdaGrad and RMSProp methods to provide an optimization technique for noisy issues with sparse gradients.

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.

Result :

<u>Here is the result :</u>

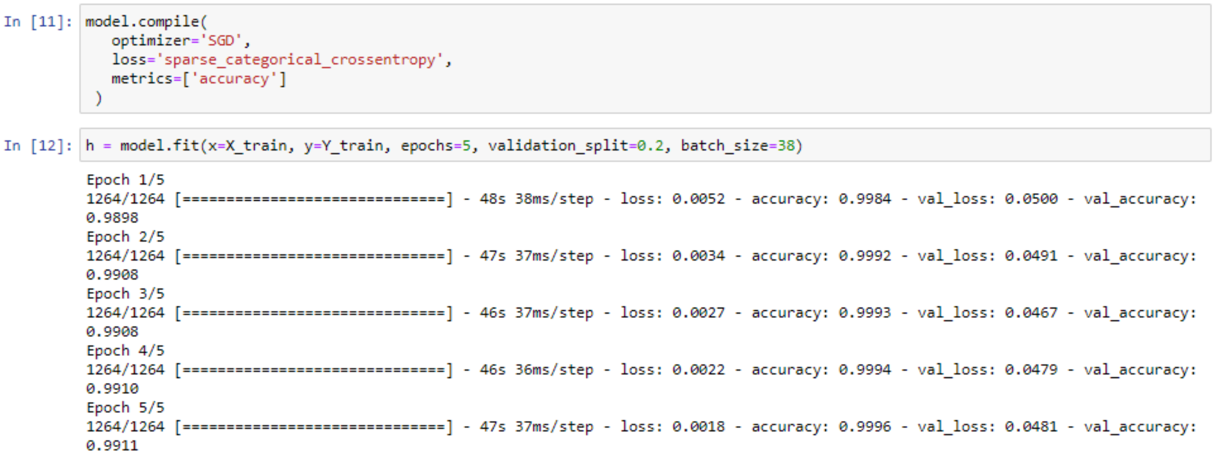


```
In [7]: model.compile(
            optimizer='adam',
            loss='sparse_categorical_crossentropy',
            metrics=['accuracy']
        )
```

```
In [8]: h = model.fit(x=X_train, y=Y_train, epochs=5, validation_split=0.2, batch_size=38)
```

```
Epoch 1/5
1264/1264 [==============================] - 48s 38ms/step - loss: 0.2438 - accuracy: 0.9235 - val_loss: 0.0973 - val_accuracy:
0.9720
Epoch 2/5
1264/1264 [==============================] - 47s 37ms/step - loss: 0.0764 - accuracy: 0.9763 - val_loss: 0.0840 - val_accuracy:
0.9755
Epoch 3/5
1264/1264 [==============================] - 48s 38ms/step - loss: 0.0526 - accuracy: 0.9840 - val_loss: 0.0644 - val_accuracy:
0.9829
Epoch 4/5
1264/1264 [==============================] - 47s 37ms/step - loss: 0.0413 - accuracy: 0.9868 - val_loss: 0.0492 - val_accuracy:
0.9862
Epoch 5/5
1264/1264 [==============================] - 47s 37ms/step - loss: 0.0319 - accuracy: 0.9902 - val_loss: 0.0432 - val_accuracy:
0.9882
```

In this picture we used optimizer as Adam and my accuracy is about 0.98+ and loss is about 0.05+

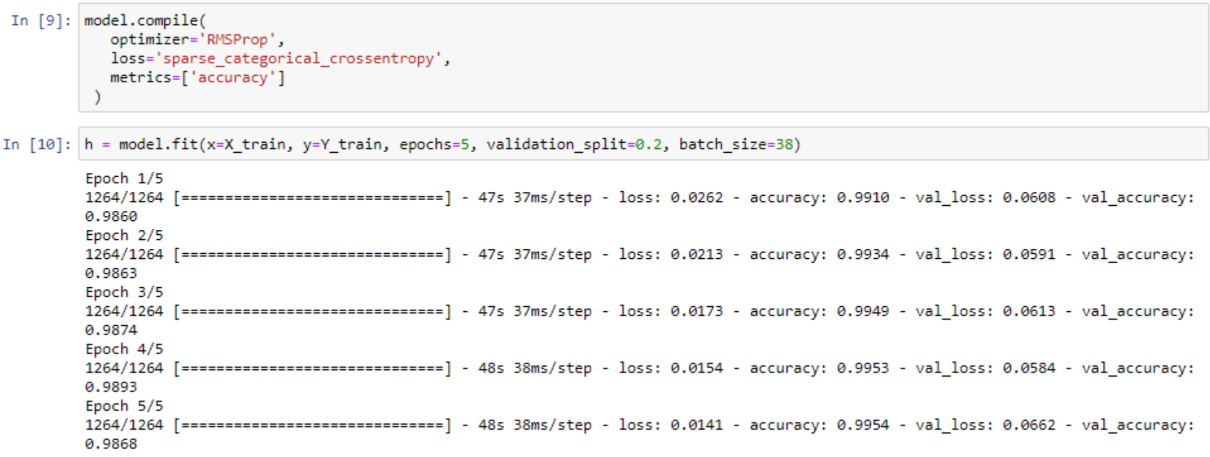


```
In [11]: model.compile(
            optimizer='SGD',
            loss='sparse_categorical_crossentropy',
            metrics=['accuracy']
          )

In [12]: h = model.fit(x=X_train, y=Y_train, epochs=5, validation_split=0.2, batch_size=38)

         Epoch 1/5
         1264/1264 [==============================] - 48s 38ms/step - loss: 0.0052 - accuracy: 0.9984 - val_loss: 0.0500 - val_accuracy:
         0.9898
         Epoch 2/5
         1264/1264 [==============================] - 47s 37ms/step - loss: 0.0034 - accuracy: 0.9992 - val_loss: 0.0491 - val_accuracy:
         0.9908
         Epoch 3/5
         1264/1264 [==============================] - 46s 37ms/step - loss: 0.0027 - accuracy: 0.9993 - val_loss: 0.0467 - val_accuracy:
         0.9908
         Epoch 4/5
         1264/1264 [==============================] - 46s 36ms/step - loss: 0.0022 - accuracy: 0.9994 - val_loss: 0.0479 - val_accuracy:
         0.9910
         Epoch 5/5
         1264/1264 [==============================] - 47s 37ms/step - loss: 0.0018 - accuracy: 0.9996 - val_loss: 0.0481 - val_accuracy:
         0.9911
```

In this output SGD has been used as the optimizer and its gives the best accuracy which h is 0.9996 and loss is 0.0448. This the best accuracy which we can see in below.



```
In [9]: model.compile(
           optimizer='RMSProp',
           loss='sparse_categorical_crossentropy',
           metrics=['accuracy']
         )

In [10]: h = model.fit(x=X_train, y=Y_train, epochs=5, validation_split=0.2, batch_size=38)

         Epoch 1/5
         1264/1264 [==============================] - 47s 37ms/step - loss: 0.0262 - accuracy: 0.9910 - val_loss: 0.0608 - val_accuracy:
         0.9860
         Epoch 2/5
         1264/1264 [==============================] - 47s 37ms/step - loss: 0.0213 - accuracy: 0.9934 - val_loss: 0.0591 - val_accuracy:
         0.9863
         Epoch 3/5
         1264/1264 [==============================] - 47s 37ms/step - loss: 0.0173 - accuracy: 0.9949 - val_loss: 0.0613 - val_accuracy:
         0.9874
         Epoch 4/5
         1264/1264 [==============================] - 48s 38ms/step - loss: 0.0154 - accuracy: 0.9953 - val_loss: 0.0584 - val_accuracy:
         0.9893
         Epoch 5/5
         1264/1264 [==============================] - 48s 38ms/step - loss: 0.0141 - accuracy: 0.9954 - val_loss: 0.0662 - val_accuracy:
         0.9868
```

In this picture you can see I used RMSProp as the optimizer and the accuracy was also like Adam which is 0.98+ but the loss is more than Adam which is (0.04- 0.07)+ .

So after using all the optimizer SGD gives the best output

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 but the loss was more than Adam. But, at the last I used SGD which gives me a better result which have accuracy of 0.99+ and loss about 0.04+. So For me in this project SGD was the best solution. And SGD is faster than Adam and RMSProp.