Lab report 13

Fall 2021

CSE422L Data Analytics Lab

Submitted by: **Ayaz Mehmood**

Registration No.:**18PWCSE1652**

Section: **A**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: _____

Submitted to:

Engr. Mian Ibad Ali Shah

Last date of Submission:

9 March 2022

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Task 1:
Mnist classification

CODE:
Importing libraries, reading dataset and splitting the dataset:

In [1]:
```
from tensorflow import keras
```

In [2]:
```
(X_train, y_train), (X_test, y_test) = keras.datasets.mnist.load_data()
```

In [3]:
```
from sklearn.model_selection import train_test_split
```

In [4]:
```
X_train, X_val, y_train, y_val = train_test_split(X_train,
                                                  y_train)
```

Plotting the dataset values:

In [5]:
```
import matplotlib.pyplot as plt
```

In [6]:
```
fig, axs = plt.subplots(1, 4, figsize= (10, 10))
axs[0].imshow(X_train[0], cmap='gray')
axs[1].imshow(X_train[5], cmap='gray')
axs[2].imshow(X_train[9], cmap='gray')
axs[3].imshow(X_train[12], cmap='gray')
axs[0].axis(False)
axs[1].axis(False)
axs[2].axis(False)
axs[3].axis(False)
fig.show()
```

```
<ipython-input-6-398868551ec8>:10: UserWarning: Matplotlib is currently using module://ipykernel.pylab.backend_inline, which is
a non-GUI backend, so cannot show the figure.
  fig.show()
```

Creating Model:

In [8]:
```
model = keras.models.Sequential(
    [keras.layers.Flatten(input_shape=(28, 28)),
     keras.layers.Dense(300, activation = 'relu'),
     keras.layers.Dense(100, activation = 'relu'),
     keras.layers.Dense(50, activation = 'relu'),
     keras.layers.Dense(10, activation = 'softmax')]
)
```

Model Summary:

In [9]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 784)               0
_________________________________________________________________
dense (Dense)                (None, 300)               235500
_________________________________________________________________
dense_1 (Dense)              (None, 100)               30100
_________________________________________________________________
dense_2 (Dense)              (None, 50)                5050
_________________________________________________________________
dense_3 (Dense)              (None, 10)                510
=================================================================
Total params: 271,160
Trainable params: 271,160
Non-trainable params: 0
```

Model compiling and callback:

In [11]:
```
model.compile(loss = [keras.losses.sparse_categorical_crossentropy]
              , optimizer = keras.optimizers.Adam(),
              metrics = [keras.metrics.sparse_categorical_accuracy])
```

In [12]:
```
#check_point = keras.callbacks.ModelCheckpoint("mnist_keras_v0.h5")
early_stopping = keras.callbacks.EarlyStopping(patience=10,
                                               restore_best_weights = True)
```

In [13]:
```
X_train.shape
```

Out[13]:
```
(45000, 28, 28)
```

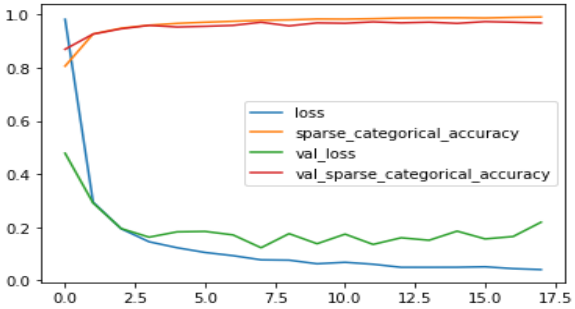
Training Model:

In [14]:
```
history = model.fit(X_train, y_train, epochs = 30,
                    validation_data = (X_val, y_val),
                    callbacks = [early_stopping])
```

```
Epoch 1/30
1407/1407 [==============================] - 7s 4ms/step - loss: 0.9820 - sparse_categorical_accuracy: 0.8059 - val_loss: 0.478
3 - val_sparse_categorical_accuracy: 0.8694
Epoch 2/30
1407/1407 [==============================] - 6s 4ms/step - loss: 0.2947 - sparse_categorical_accuracy: 0.9269 - val_loss: 0.291
5 - val_sparse_categorical_accuracy: 0.9265
Epoch 3/30
1407/1407 [==============================] - 6s 4ms/step - loss: 0.1946 - sparse_categorical_accuracy: 0.9485 - val_loss: 0.195
4 - val_sparse_categorical_accuracy: 0.9463
Epoch 4/30
1407/1407 [==============================] - 6s 4ms/step - loss: 0.1456 - sparse_categorical_accuracy: 0.9598 - val_loss: 0.162
8 - val_sparse_categorical_accuracy: 0.9593
Epoch 5/30
1407/1407 [==============================] - 6s 4ms/step - loss: 0.1231 - sparse_categorical_accuracy: 0.9667 - val_loss: 0.182
9 - val_sparse_categorical_accuracy: 0.9532
Epoch 6/30
```

Loss and accuracy graph:

```
In [16]:
1 pd.DataFrame(history.history).plot()
2 plt.show()
```



Model accuracy:

```
In [17]:
1 model.evaluate(X_test, y_test)

313/313 [==============================] - 1s 3ms/step - loss: 0.1250 - sparse_categorical_accuracy: 0.9710

Out[17]: [0.12498439848423004, 0.9710000157356262]
```

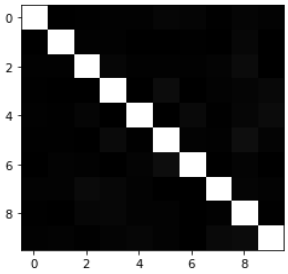
Confusion Matrix:

```
In [19]:
1 from sklearn.metrics import confusion_matrix

In [20]:
1 import numpy as np

In [21]:
1 conf_mat = confusion_matrix(y_true = y_test, y_pred = np.argmax(model.predict(X_test), axis = 1))

In [22]:
1 plt.imshow(conf_mat, cmap = 'gray', vmin = 0, vmax = 255)
2 plt.show()
```



The file uploaded was giving me an error which is below that's why I work on ANN to perform this lab task. I tried to resolve the error but failed to get rid of it that's why I tried this one.

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-41-e5fdfe3b9dbc> in <module>
      1 import numpy as np
      2 import tensorflow as tf
----> 3 from tensorflow.examples.tutorials.mnist import input_data
      4 mnist = tf.keras.datasets.mnist.load_data()

ModuleNotFoundError: No module named 'tensorflow.examples'
```