Author: Jonathan Ibifubara Pollyn

Course: DSC-540

Assignment: Image Classifier

Description: tasked to build an image classifier for the MNIST dataset of handwritten numbers, implementing the k-nearest neighbors (k-NN) algorithm.

In [1]:
```
pip install keras
```

```
Requirement already satisfied: keras in c:\anaconda\lib\site-packages (2.7.0)
Note: you may need to restart the kernel to use updated packages.
```

In [11]:
```
import sys

!$sys.executable -m pip install tensorflow
```

```
Collecting tensorflow
  Downloading tensorflow-2.7.0-cp38-cp38-win_amd64.whl (430.8 MB)
Collecting libclang>=9.0.1
  Downloading libclang-12.0.0-py2.py3-none-win_amd64.whl (13.1 MB)
Collecting protobuf>=3.9.2
  Downloading protobuf-3.19.1-cp38-cp38-win_amd64.whl (895 kB)
Collecting grpcio<2.0,>=1.24.3
  Downloading grpcio-1.41.1-cp38-cp38-win_amd64.whl (3.2 MB)
Collecting tensorboard~=2.6
  Downloading tensorboard-2.7.0-py3-none-any.whl (5.8 MB)
Collecting absl-py>=0.4.0
  Downloading absl_py-0.15.0-py3-none-any.whl (132 kB)
Requirement already satisfied: h5py>=2.9.0 in c:\anaconda\lib\site-packages (from tensorflow) (2.10.0)
Collecting opt-einsum>=2.3.2
  Downloading opt_einsum-3.3.0-py3-none-any.whl (65 kB)
Collecting termcolor>=1.1.0
  Downloading termcolor-1.1.0.tar.gz (3.9 kB)
Requirement already satisfied: wheel<1.0,>=0.32.0 in c:\anaconda\lib\site-packages (from tensorflow) (0.36.2)
Collecting google-pasta>=0.1.1
  Downloading google_pasta-0.2.0-py3-none-any.whl (57 kB)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\jonathan pollyn\appdata\roaming\python\python38\site-p
ackages (from tensorflow) (1.12.1)
Requirement already satisfied: numpy>=1.14.5 in c:\anaconda\lib\site-packages (from tensorflow) (1.20.1)
Collecting tensorflow-io-gcs-filesystem>=0.21.0
  Downloading tensorflow_io_gcs_filesystem-0.21.0-cp38-cp38-win_amd64.whl (1.5 MB)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\anaconda\lib\site-packages (from tensorflow) (3.
7.4.3)
Collecting gast<0.5.0,>=0.2.1
  Downloading gast-0.4.0-py3-none-any.whl (9.8 kB)
Collecting tensorflow-estimator<2.8,~=2.7.0rc0
  Downloading tensorflow_estimator-2.7.0-py2.py3-none-any.whl (463 kB)
Collecting keras-preprocessing>=1.1.1
  Downloading Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
Collecting flatbuffers<3.0,>=1.12
  Downloading flatbuffers-2.0-py2.py3-none-any.whl (26 kB)
Requirement already satisfied: six>=1.12.0 in c:\users\jonathan pollyn\appdata\roaming\python\python38\site-pac
kages (from tensorflow) (1.14.0)
Collecting astunparse>=1.6.0
  Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: keras<2.8,>=2.7.0rc0 in c:\anaconda\lib\site-packages (from tensorflow) (2.7.0)
Collecting google-auth-oauthlib<0.5,>=0.4.1
  Downloading google_auth_oauthlib-0.4.6-py2.py3-none-any.whl (18 kB)
Requirement already satisfied: werkzeug>=0.11.15 in c:\anaconda\lib\site-packages (from tensorboard~=2.6->tenso
rflow) (1.0.1)
Requirement already satisfied: requests<3,>=2.21.0 in c:\anaconda\lib\site-packages (from tensorboard~=2.6->ten
sorflow) (2.25.1)
Requirement already satisfied: setuptools>=41.0.0 in c:\anaconda\lib\site-packages (from tensorboard~=2.6->tens
orflow) (52.0.0.post20210125)
Collecting tensorboard-data-server<0.7.0,>=0.6.0
  Downloading tensorboard_data_server-0.6.1-py3-none-any.whl (2.4 kB)
Collecting google-auth<3,>=1.6.3
  Downloading google_auth-2.3.3-py2.py3-none-any.whl (155 kB)
Collecting markdown>=2.6.8
  Downloading Markdown-3.3.4-py3-none-any.whl (97 kB)
Collecting tensorboard-plugin-wit>=1.6.0
  Downloading tensorboard_plugin_wit-1.8.0-py3-none-any.whl (781 kB)
Collecting rsa<5,>=3.1.4
  Downloading rsa-4.7.2-py3-none-any.whl (34 kB)
Collecting pyasn1-modules>=0.2.1
  Downloading pyasn1_modules-0.2.8-py2.py3-none-any.whl (155 kB)
Collecting cachetools<5.0,>=2.0.0
  Downloading cachetools-4.2.4-py3-none-any.whl (10 kB)
Collecting requests-oauthlib>=0.7.0
  Downloading requests_oauthlib-1.3.0-py2.py3-none-any.whl (23 kB)
Collecting pyasn1<0.5.0,>=0.4.6
  Downloading pyasn1-0.4.8-py2.py3-none-any.whl (77 kB)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\anaconda\lib\site-packages (from requests<3,>=2.21.0
->tensorboard~=2.6->tensorflow) (1.26.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\anaconda\lib\site-packages (from requests<3,>=2.21.0->t
ensorboard~=2.6->tensorflow) (2020.12.5)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\anaconda\lib\site-packages (from requests<3,>=2.21.0->te
nsorboard~=2.6->tensorflow) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in c:\anaconda\lib\site-packages (from requests<3,>=2.21.0->tensorb
oard~=2.6->tensorflow) (2.10)
Collecting oauthlib>=3.0.0
  Downloading oauthlib-3.1.1-py2.py3-none-any.whl (146 kB)
Building wheels for collected packages: termcolor
  Building wheel for termcolor (setup.py): started
  Building wheel for termcolor (setup.py): finished with status 'done'
  Created wheel for termcolor: filename=termcolor-1.1.0-py3-none-any.whl size=4829 sha256=75c209ae22a1a6160ca52
93c5d01ca5c40e5985c0b90b5cce5fb567c4dde787
  Stored in directory: c:\users\jonathan pollyn\appdata\local\pip\cache\wheels\a0\16\9c\5473df82468f958445479c5
9e784896fa24f4a5fc024b0f501
Successfully built termcolor
Installing collected packages: pyasn1, rsa, pyasn1-modules, oauthlib, cachetools, requests-oauthlib, google-aut
h, tensorboard-plugin-wit, tensorboard-data-server, protobuf, markdown, grpcio, google-auth-oauthlib, absl-py,
termcolor, tensorflow-io-gcs-filesystem, tensorflow-estimator, tensorboard, opt-einsum, libclang, keras-preproc
essing, google-pasta, gast, flatbuffers, astunparse, tensorflow
Successfully installed absl-py-0.15.0 astunparse-1.6.3 cachetools-4.2.4 flatbuffers-2.0 gast-0.4.0 google-auth-
2.3.3 google-auth-oauthlib-0.4.6 google-pasta-0.2.0 grpcio-1.41.1 keras-preprocessing-1.1.2 libclang-12.0.0 mar
kdown-3.3.4 oauthlib-3.1.1 opt-einsum-3.3.0 protobuf-3.19.1 pyasn1-0.4.8 pyasn1-modules-0.2.8 requests-oauthlib
-1.3.0 rsa-4.7.2 tensorboard-2.7.0 tensorboard-data-server-0.6.1 tensorboard-plugin-wit-1.8.0 tensorflow-2.7.0
tensorflow-estimator-2.7.0 tensorflow-io-gcs-filesystem-0.21.0 termcolor-1.1.0
```

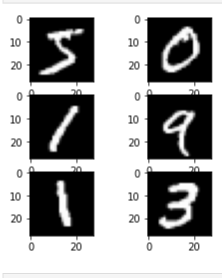
In [3]:
```
#loading the required libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelBinarizer
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense
from sklearn.neighbors import KNeighborsClassifier
from scipy.spatial import distance
import math
```

In [4]:
```
#Loading MNIST Dataset predictor and reponse variable
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

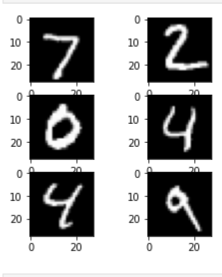
In [5]:
```
#Printing the data shapes
print('x_train: ' + str(x_train.shape))
print('y_train: ' + str(y_train.shape))
print('x_test:  '  + str(x_test.shape))
print('y_test:  '  + str(y_test.shape))
```

```
x_train: (60000, 28, 28)
y_train: (60000,)
x_test:  (10000, 28, 28)
y_test:  (10000,)
```

In [6]:
```
#Plotting the training dataset
for i in range(9):
    plt.subplot(330 + 1 + i)
    plt.imshow(x_train[i], cmap=plt.get_cmap('gray'))
plt.show()
```



In [7]:
```
#Plotting the test dataset
for i in range(9):
    plt.subplot(330 + 1 + i)
    plt.imshow(x_test[i], cmap=plt.get_cmap('gray'))
plt.show()
```



In [8]:
```
#Convert the 2D data into 1D data
x_train = x_train.reshape(60000, 784)
x_test = x_test.reshape(10000, 784)
```

In [9]:
```
#Convert the target variable to a one-hot vector
label_binar = LabelBinarizer()
label_binar.fit(range(10))
y_train = label_binar.transform(y_train)
y_test = label_binar.transform(y_test)
```

In [10]:
```
#Create the model
model = Sequential()
model.add(Dense(units=32, activation='relu', input_dim=784))
model.add(Dense(units=32, activation='relu'))
model.add(Dense(units=10, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics = ['acc'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 32)                25120

 dense_1 (Dense)             (None, 32)                1056

 dense_2 (Dense)             (None, 10)                330

=================================================================
Total params: 26,506
Trainable params: 26,506
Non-trainable params: 0
_________________________________________________________________
```

In [11]:
```
#Train the model to check for the final accuracy
model.fit(x_train, y_train, validation_data = (x_test, y_test), epochs=40,
batch_size=32)
score = model.evaluate(x_test, y_test)
print('Accuracy: {0:.2f}%'.format(score[1]*100))
```

```
Epoch 1/40
1875/1875 [==============================] - 3s 1ms/step - loss: 1.5805 - acc: 0.7386 - val_loss: 0.5162 - val_
acc: 0.8575
Epoch 2/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.3997 - acc: 0.8906 - val_loss: 0.3258 - val_
acc: 0.9086
Epoch 3/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2940 - acc: 0.9184 - val_loss: 0.2522 - val_
acc: 0.9287
Epoch 4/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2461 - acc: 0.9314 - val_loss: 0.2571 - val_
acc: 0.9265
Epoch 5/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2169 - acc: 0.9393 - val_loss: 0.2124 - val_
acc: 0.9426
Epoch 6/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1971 - acc: 0.9450 - val_loss: 0.2095 - val_
acc: 0.9436
Epoch 7/40
1875/1875 [==============================] - 5s 2ms/step - loss: 0.1772 - acc: 0.9504 - val_loss: 0.1919 - val_
acc: 0.9474
Epoch 8/40
1875/1875 [==============================] - 4s 2ms/step - loss: 0.1664 - acc: 0.9529 - val_loss: 0.1835 - val_
acc: 0.9495
Epoch 9/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1550 - acc: 0.9552 - val_loss: 0.1881 - val_
acc: 0.9514
Epoch 10/40
1875/1875 [==============================] - 5s 2ms/step - loss: 0.1450 - acc: 0.9590 - val_loss: 0.1847 - val_
acc: 0.9531
Epoch 11/40
1875/1875 [==============================] - 4s 2ms/step - loss: 0.1364 - acc: 0.9615 - val_loss: 0.1928 - val_
acc: 0.9503
Epoch 12/40
1875/1875 [==============================] - 7s 4ms/step - loss: 0.1309 - acc: 0.9637 - val_loss: 0.1993 - val_
acc: 0.9499
Epoch 13/40
1875/1875 [==============================] - 5s 3ms/step - loss: 0.1249 - acc: 0.9646 - val_loss: 0.2135 - val_
acc: 0.9450
Epoch 14/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1243 - acc: 0.9647 - val_loss: 0.1911 - val_
acc: 0.9524
Epoch 15/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1191 - acc: 0.9665 - val_loss: 0.1765 - val_
acc: 0.9583
Epoch 16/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1121 - acc: 0.9681 - val_loss: 0.1796 - val_
acc: 0.9565
Epoch 17/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1119 - acc: 0.9689 - val_loss: 0.1789 - val_
acc: 0.9564
Epoch 18/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1077 - acc: 0.9690 - val_loss: 0.1812 - val_
acc: 0.9560
Epoch 19/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1040 - acc: 0.9708 - val_loss: 0.1829 - val_
acc: 0.9552
Epoch 20/40
1875/1875 [==============================] - 3s 1ms/step - loss: 0.1044 - acc: 0.9709 - val_loss: 0.1933 - val_
acc: 0.9534
Epoch 21/40
1875/1875 [==============================] - 3s 1ms/step - loss: 0.1011 - acc: 0.9714 - val_loss: 0.1991 - val_
acc: 0.9569
Epoch 22/40
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0993 - acc: 0.9729 - val_loss: 0.2117 - val_
acc: 0.9530
Epoch 23/40
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0979 - acc: 0.9724 - val_loss: 0.1890 - val_
acc: 0.9559
Epoch 24/40
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0903 - acc: 0.9752 - val_loss: 0.2162 - val_
acc: 0.9552
Epoch 25/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0960 - acc: 0.9726 - val_loss: 0.2087 - val_
acc: 0.9522
Epoch 26/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0946 - acc: 0.9741 - val_loss: 0.2066 - val_
acc: 0.9534
Epoch 27/40
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0914 - acc: 0.9743 - val_loss: 0.2145 - val_
acc: 0.9542
Epoch 28/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0939 - acc: 0.9750 - val_loss: 0.2207 - val_
acc: 0.9551
Epoch 29/40
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0897 - acc: 0.9748 - val_loss: 0.2196 - val_
acc: 0.9563
Epoch 30/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0855 - acc: 0.9754 - val_loss: 0.2342 - val_
acc: 0.9533
Epoch 31/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0864 - acc: 0.9758 - val_loss: 0.2105 - val_
acc: 0.9564
Epoch 32/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0872 - acc: 0.9763 - val_loss: 0.2250 - val_
acc: 0.9555
Epoch 33/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0861 - acc: 0.9754 - val_loss: 0.2470 - val_
acc: 0.9542
Epoch 34/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0839 - acc: 0.9761 - val_loss: 0.2253 - val_
acc: 0.9570
Epoch 35/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0805 - acc: 0.9770 - val_loss: 0.2575 - val_
acc: 0.9519
Epoch 36/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0827 - acc: 0.9770 - val_loss: 0.2284 - val_
acc: 0.9595
Epoch 37/40
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0802 - acc: 0.9771 - val_loss: 0.2801 - val_
acc: 0.9530
Epoch 38/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0798 - acc: 0.9779 - val_loss: 0.2467 - val_
acc: 0.9573
Epoch 39/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0871 - acc: 0.9766 - val_loss: 0.2508 - val_
acc: 0.9566
Epoch 40/40
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0743 - acc: 0.9789 - val_loss: 0.2787 - val_
acc: 0.9532
313/313 [==============================] - 0s 1ms/step - loss: 0.2787 - acc: 0.9532
Accuracy: 95.32%
```

In [12]:
```
distance.euclidean(x_test[0], x_train[0])
```

Out[12]: 2204.126357539422

In [13]:
```
distance.euclidean(x_test[1], x_train[1])
```

Out[13]: 2536.5214369289292

In [14]:
```
distance.euclidean(x_test[2], x_train[2])
```

Out[14]: 1909.3464850571256

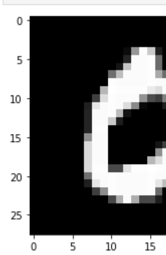
In [15]:
```
distance.euclidean(x_test[3], x_train[3])
```

Out[15]: 2942.6883966876276

In [16]:
```
distance.euclidean(x_test[4], x_train[4])
```

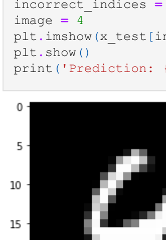
Out[16]: 2154.393882278726

In [17]:
```
#Ploting some test images
image = 3
plt.imshow(x_test[image].reshape(28, 28), cmap=plt.get_cmap('gray'))
plt.show()
y_pred = model.predict(x_test)
print('Prediction: {0}'.format(np.argmax(y_pred[image])))
```



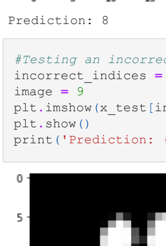
```
Prediction: 0
```

In [18]:
```
#Testing an incorrect prediction
incorrect_indices = np.nonzero(np.argmax(y_pred,axis=1) != np.argmax(y_test,axis=1))[0]
image = 4
plt.imshow(x_test[incorrect_indices[image]].reshape(28,28), cmap=plt.get_cmap('gray'))
plt.show()
print('Prediction: {0}'.format(np.argmax(y_pred[incorrect_indices[image]])))
```



```
Prediction: 8
```

In [19]:
```
#Testing an incorrect prediction
incorrect_indices = np.nonzero(np.argmax(y_pred,axis=1) != np.argmax(y_test,axis=1))[0]
image = 9
plt.imshow(x_test[incorrect_indices[image]].reshape(28,28), cmap=plt.get_cmap('gray'))
plt.show()
print('Prediction: {0}'.format(np.argmax(y_pred[incorrect_indices[image]])))
```



```
Prediction: 9
```

In [20]:
```
#Obtaining the k-Nearest Neighbor
accuracies = []

for k in range(1, 30, 2):
    # train the k-Nearest Neighbor classifier with the current value of `k`
    model = KNeighborsClassifier(n_neighbors=k)
    model.fit(x_train, y_train)

    # evaluate the model and update the accuracies list
    score = model.score(x_test, y_test)
    print("k=%d, accuracy=%.2f%%" % (k, score * 100))
    accuracies.append(score)
```

```
k=1, accuracy=96.91%
k=3, accuracy=96.95%
k=5, accuracy=96.60%
k=7, accuracy=96.55%
k=9, accuracy=96.19%
k=11, accuracy=96.10%
k=13, accuracy=95.85%
k=15, accuracy=95.73%
k=17, accuracy=95.71%
k=19, accuracy=95.64%
k=21, accuracy=95.57%
k=23, accuracy=95.54%
k=25, accuracy=95.33%
k=27, accuracy=95.20%
k=29, accuracy=95.08%
```

In [ ]: