

Project Report 15

Handwritten Digit Recognition

Name :DHIVAKAR.R

COURSE: AI and ML(Aug 2020)

Question: Use MNIST dataset to create a classifier for all the 10 digits. First implement the classifier by squeezing the image into a vector and then using a MLP. Now, try the same task using a different machine learning classifier such as an SVM to check the gain in performance by using perceptrons as compared to conventional machine learning techniques.

Prerequisites

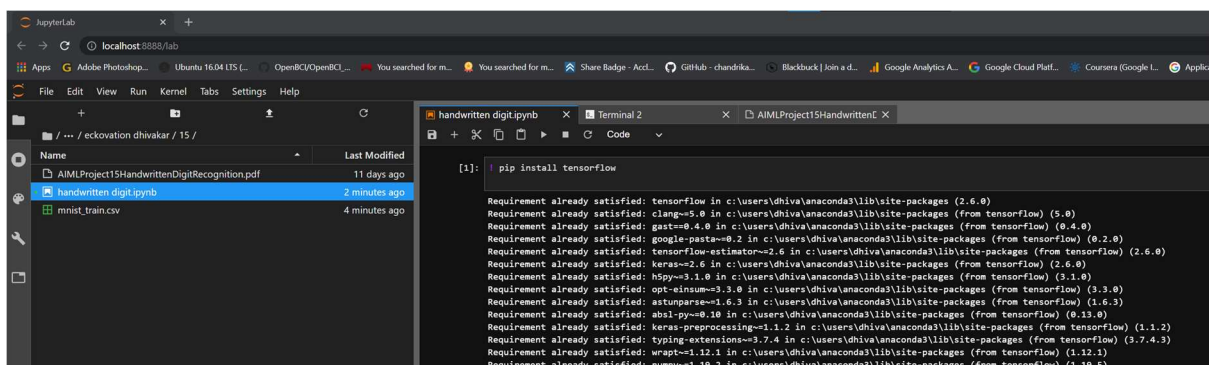
What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has latest version of python. The following url <https://www.python.org/downloads/> can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: <https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-externalcommand/> . Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic.

Second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url <https://www.anaconda.com/download/> You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6 then run below commands in command prompt/terminal to install these packages `pip install -U scikit-learn` `pip install numpy` `pip install scipy` if you have chosen to install anaconda then run below commands in anaconda prompt to install these packages `conda install -c scikit-learn` `conda install -c anaconda numpy` `conda install -c anaconda scipy` . then you have to install tensorflow use `pip install tensorflow` in terminal

Dataset used: The dataset used is MNIST dataset

Importing the required libraires



```
[1]: pip install tensorflow

Requirement already satisfied: tensorflow in c:\users\dhiva\anaconda3\lib\site-packages (2.6.0)
Requirement already satisfied: clang~=5.0 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (5.0)
Requirement already satisfied: gast~=0.4.0 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: google-pasta~=0.2 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: tensorflow-estimator~=2.6 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (2.6.0)
Requirement already satisfied: keras~=2.6 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (2.6.0)
Requirement already satisfied: h5py~=3.1.0 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (3.1.0)
Requirement already satisfied: opt-einsum~=3.3.0 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: astunparse~=1.6.3 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: absl-py~=0.10 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (0.13.0)
Requirement already satisfied: keras-preprocessing~=1.1.2 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (1.1.2)
Requirement already satisfied: typing-extensions~=3.7.4 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (3.7.4.3)
Requirement already satisfied: wrapt~=1.12.1 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (1.12.1)
Requirement already satisfied: numpy~=1.19.2 in c:\users\dhiva\anaconda3\lib\site-packages (from tensorflow) (1.19.5)
```


Accuracy of model :

```
[8]: model.fit(training_images, training_labels, epochs=5)

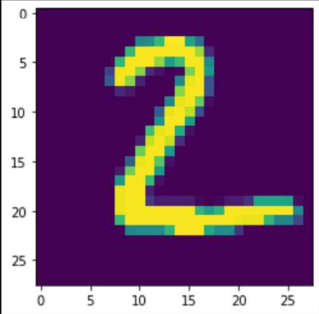
Train on 60000 samples
Epoch 1/5
60000/60000 [=====] - 4s 61us/sample - loss: 0.2292 - accuracy: 0.9325
Epoch 2/5
60000/60000 [=====] - 3s 44us/sample - loss: 0.0947 - accuracy: 0.9715
Epoch 3/5
60000/60000 [=====] - 3s 47us/sample - loss: 0.0662 - accuracy: 0.9796
Epoch 4/5
60000/60000 [=====] - 3s 43us/sample - loss: 0.0507 - accuracy: 0.9837
Epoch 5/5
60000/60000 [=====] - 3s 45us/sample - loss: 0.0420 - accuracy: 0.9862
[8]: <tensorflow.python.keras.callbacks.History at 0x1f3a04d9348>

[9]: print(model.evaluate(test_images, test_labels))

10000/10000 [=====] - 0s 42us/sample - loss: 0.0914 - accuracy: 0.9713
[0.0913783773623989, 0.9713]

[11]: plt.imshow(test_images[1])
prediction = model.predict(test_images)
print(np.argmax(prediction[1]))

2
```



Recognition using Decision Tree Classifier

```
[13]: from sklearn.tree import DecisionTreeClassifier
import pandas as pd

data = pd.read_csv("mnist_train.csv").to_numpy()

[14]: print(data)

[[5 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [4 0 0 ... 0 0 0]
 ...
 [5 0 0 ... 0 0 0]
 [6 0 0 ... 0 0 0]
 [8 0 0 ... 0 0 0]]

[15]: clf = DecisionTreeClassifier()
```

Training the dataset

```
[16]: # Training dataset
x_train = data[0:21000, 1:]
train_label = data[0:21000, 0]

clf.fit(x_train, train_label)
```

```
[16]: DecisionTreeClassifier()
```

```
[17]: # Testing dataset
x_test = data[21000:, 1:]
test_label = data[21000:, 0]
```

Accuracy :

```
[19]: # Accuracy
p = clf.predict(x_test)
count = 0
for i in range(0, 21000):
    count +=1 if p[i] == test_label[i] else 0

print("Accuracy: ", (count/21000)*100)
```

```
Accuracy: 82.8952380952381
```