Anurat Bhattacharya 19CS10071 Ques 10

Copy_of_mnist_classifier(2)

October 24, 2021

[37]:
```
import numpy as np
import pandas as pd
```

[56]:
```
from keras.datasets import mnist
```

[57]:
```
#load data
(train_X,train_y),(val_X,val_y) = mnist.load_data()
```

[58]:
```
train_X = np.asarray(train_X)/255.0
val_X = np.asarray(val_X)/255.0
```

[59]:
```
train_X= train_X.reshape(60000,28*28)
val_X= val_X.reshape(val_X.shape[0],28*28)
```

[60]:
```
choices = np.random.choice(train_X.shape[0], 10000, replace=False)
train_X,train_y = train_X[choices, :],train_y[choices]
```

[61]:
```
columns_taken = []
ind = 0
X = []
 #remove the 0 columns to make the matrix with L.I. conlumns
for x in train_X.T:
  if np.sum(x)>0:
    X.append(x)
    columns_taken.append(ind)
  ind+=1
X.append([1]*X[0].shape[0])#add bias
train_X = np.array(X).T
```

[62]:
```
#train_X = [ x  for x in train_X.T if np.sum(x)>0 ]
#train_X.append([1]*train_X[0].shape[0])
#train_X = np.array(train_X).T
```

[63]:
```
train_X.shape
```

[63]:
```
(10000, 678)
```

[63]:

```

```

[64]:

```
#A : train_x
#b : train_y
theta = np.zeros((10,train_X.shape[1]))
for i in range(10):
  #for ith class against others
  y = np.array([1 if y_==i else -1 for y_ in train_y ])
  theta[i] =  np.dot(np.linalg.inv(np.dot(train_X.T,train_X)),np.dot(train_X.
 ↪T,y))
```

[65]:

```
theta.shape
```

[65]:

```
(10, 678)
```

[66]:

```
def predict(x):
  return np.argmax(np.array([np.dot(theta[i].T,x) for i in range(10)]))
```

[70]:

```
#prepare test set
choices = np.random.choice(val_X.shape[0], 1000, replace=False)
val_X,val_y = val_X[choices, :],val_y[choices]
#remove those columns from X which were removed earlier
val_X = val_X.T[columns_taken].T
#also add bias column
bias = np.ones((1000,1))
val_X = np.append(val_X,bias,axis = 1)
#analysis
classes = [i for i in range(10)]
confusion_matrix= {}
for i in classes:
  confusion_matrix[i] = {}
  for j in classes:
    confusion_matrix[i][j] = 0
for i in range(1000):
  x = val_X[i]
  y = val_y[i]
  yc = predict(x)
  #pint(y,yc)
  confusion_matrix[y][yc]+=1
confusion_matrix = pd.DataFrame(confusion_matrix)
confusion_matrix
```

[70]:

```
    0    1   2   3   4  5  6  7  8  9
0  88    0   1   0   0  4  2  0  1  3
1   0  110   4   1   3  1  2  5  3  2
2   0    0  81   2   0  1  0  2  2  0
3   0    0   4  87   0  5  0  1  5  3
```

```
4   0    1   3   2  111   3   4   3   2   7
5   2    0   0   1    1  62   4   0   6   0
6   1    2   2   0    1   1  70   0   2   0
7   0    0   2   2    0   0   0  86   1  12
8   1    3   4   2    1   4   1   0  58   3
9   0    0   0   0    6   2   0   4   2  92
```

[72]:

```
cnt = 1000
correct = 0
for i in range(10): correct+=confusion_matrix[i][i]
accuracy= correct/cnt
print('Accuracy : ',accuracy)
```

```
Accuracy :  0.845
```