Artificial Neural Network

In [0]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/au
th?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.goog
leusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&r
esponse_type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fa
uth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20
https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20ht
tps%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly

Enter your authorization code:
··········
Mounted at /content/drive
```

In [0]:

```
#Installing important libraries
import numpy as np
import pandas as pd
```

In [0]:

```
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LeakyReLU,PReLU,ELU,Activation
from keras.layers import Dropout
```

```
ERROR: Could not find a version that satisfies the requirement Label
Powerset (from versions: none)
ERROR: No matching distribution found for LabelPowerset
```

In [0]:

```
!pip install --upgrade LabelPowerset
```

```
ERROR: Could not find a version that satisfies the requirement Label
Powerset (from versions: none)
ERROR: No matching distribution found for LabelPowerset
```

In [0]:

```
df3 = pd.read_csv("/content/drive/My Drive/image classification project/df3_final.csv",header=None)
X = df3.iloc[:,7:3079].values
y = df3.iloc[:, 3079].values
#Normalizing the data
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X = sc.fit_transform(X)
np.random.seed(3)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30,random_state = 0)
```

Single-Layer Neural Network

In [0]:

```
# Initialising the Single Layer Neural Network
classifier = Sequential()

# Adding the input layer and the first hidden layer
classifier.add(Dense(100 ,input_dim = X_train.shape[1]))
classifier.add(Activation('relu'))
classifier.add(Dropout(0.3))
classifier.add(Dense(100))
classifier.add(Activation('relu'))
classifier.add(Dropout(0.3))
classifier.add(Dense(7, activation = 'softmax'))
classifier.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics = ['accuracy'])

model_history=classifier.fit(X_train, y_train,validation_split=0.20, batch_size = 128, nb_epoch = 50)
print(model_history.history.keys())
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:13: Use
rWarning: The `nb_epoch` argument in `fit` has been renamed `epoch
s`.
  del sys.path[0]

Train on 5608 samples, validate on 1402 samples
Epoch 1/50
5608/5608 [==============================] - 1s 167us/step - loss:
1.3444 - acc: 0.6004 - val_loss: 0.8975 - val_acc: 0.6726
Epoch 2/50
5608/5608 [==============================] - 0s 45us/step - loss:
1.0582 - acc: 0.6473 - val_loss: 0.8563 - val_acc: 0.6997
Epoch 3/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.9583 - acc: 0.6699 - val_loss: 0.8360 - val_acc: 0.6961
Epoch 4/50
5608/5608 [==============================] - 0s 44us/step - loss:
0.9109 - acc: 0.6764 - val_loss: 0.8190 - val_acc: 0.7097
Epoch 5/50
```

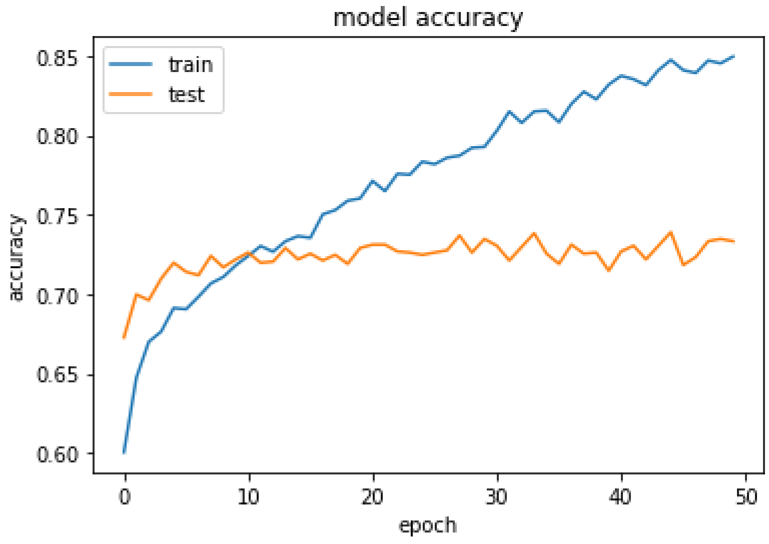
```
5608/5608 [==============================] - 0s 43us/step - loss:
0.8775 - acc: 0.6912 - val_loss: 0.7930 - val_acc: 0.7197
Epoch 6/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.8485 - acc: 0.6904 - val_loss: 0.8105 - val_acc: 0.7140
Epoch 7/50
5608/5608 [==============================] - 0s 44us/step - loss:
0.8221 - acc: 0.6983 - val_loss: 0.7783 - val_acc: 0.7118
Epoch 8/50
5608/5608 [==============================] - 0s 44us/step - loss:
0.8055 - acc: 0.7067 - val_loss: 0.7801 - val_acc: 0.7240
Epoch 9/50
5608/5608 [==============================] - 0s 45us/step - loss:
0.7848 - acc: 0.7108 - val_loss: 0.7943 - val_acc: 0.7168
Epoch 10/50
5608/5608 [==============================] - 0s 44us/step - loss:
0.7844 - acc: 0.7181 - val_loss: 0.7831 - val_acc: 0.7218
Epoch 11/50
5608/5608 [==============================] - 0s 46us/step - loss:
0.7470 - acc: 0.7240 - val_loss: 0.7678 - val_acc: 0.7261
Epoch 12/50
5608/5608 [==============================] - 0s 44us/step - loss:
0.7408 - acc: 0.7302 - val_loss: 0.7675 - val_acc: 0.7197
Epoch 13/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.7431 - acc: 0.7266 - val_loss: 0.7569 - val_acc: 0.7204
Epoch 14/50
5608/5608 [==============================] - 0s 41us/step - loss:
0.7146 - acc: 0.7332 - val_loss: 0.7672 - val_acc: 0.7290
Epoch 15/50
5608/5608 [==============================] - 0s 46us/step - loss:
0.7054 - acc: 0.7364 - val_loss: 0.7506 - val_acc: 0.7218
Epoch 16/50
5608/5608 [==============================] - 0s 46us/step - loss:
0.7056 - acc: 0.7354 - val_loss: 0.7667 - val_acc: 0.7254
Epoch 17/50
5608/5608 [==============================] - 0s 42us/step - loss:
0.6806 - acc: 0.7502 - val_loss: 0.7576 - val_acc: 0.7211
Epoch 18/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.6646 - acc: 0.7529 - val_loss: 0.7375 - val_acc: 0.7247
Epoch 19/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.6542 - acc: 0.7587 - val_loss: 0.7559 - val_acc: 0.7190
Epoch 20/50
5608/5608 [==============================] - 0s 42us/step - loss:
0.6469 - acc: 0.7602 - val_loss: 0.7516 - val_acc: 0.7290
Epoch 21/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.6350 - acc: 0.7712 - val_loss: 0.7577 - val_acc: 0.7311
Epoch 22/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.6280 - acc: 0.7648 - val_loss: 0.7558 - val_acc: 0.7311
Epoch 23/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.6215 - acc: 0.7757 - val_loss: 0.7427 - val_acc: 0.7268
```

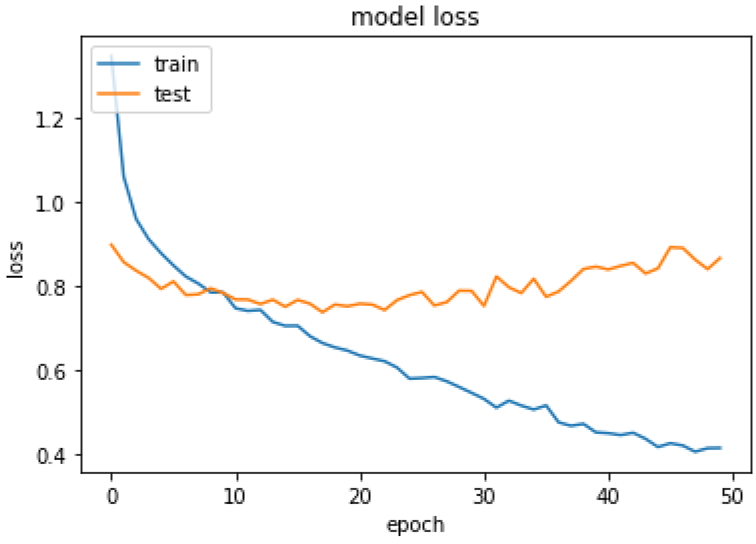
```
Epoch 24/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.6065 - acc: 0.7751 - val_loss: 0.7663 - val_acc: 0.7261
Epoch 25/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.5803 - acc: 0.7833 - val_loss: 0.7780 - val_acc: 0.7247
Epoch 26/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.5817 - acc: 0.7817 - val_loss: 0.7856 - val_acc: 0.7261
Epoch 27/50
5608/5608 [==============================] - 0s 42us/step - loss:
0.5841 - acc: 0.7858 - val_loss: 0.7535 - val_acc: 0.7275
Epoch 28/50
5608/5608 [==============================] - 0s 44us/step - loss:
0.5740 - acc: 0.7871 - val_loss: 0.7615 - val_acc: 0.7368
Epoch 29/50
5608/5608 [==============================] - 0s 42us/step - loss:
0.5605 - acc: 0.7921 - val_loss: 0.7888 - val_acc: 0.7261
Epoch 30/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.5466 - acc: 0.7926 - val_loss: 0.7882 - val_acc: 0.7347
Epoch 31/50
5608/5608 [==============================] - 0s 45us/step - loss:
0.5323 - acc: 0.8028 - val_loss: 0.7524 - val_acc: 0.7304
Epoch 32/50
5608/5608 [==============================] - 0s 46us/step - loss:
0.5115 - acc: 0.8149 - val_loss: 0.8223 - val_acc: 0.7211
Epoch 33/50
5608/5608 [==============================] - 0s 42us/step - loss:
0.5281 - acc: 0.8078 - val_loss: 0.7967 - val_acc: 0.7297
Epoch 34/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.5165 - acc: 0.8149 - val_loss: 0.7830 - val_acc: 0.7382
Epoch 35/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.5071 - acc: 0.8154 - val_loss: 0.8169 - val_acc: 0.7254
Epoch 36/50
5608/5608 [==============================] - 0s 46us/step - loss:
0.5168 - acc: 0.8081 - val_loss: 0.7741 - val_acc: 0.7190
Epoch 37/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.4769 - acc: 0.8197 - val_loss: 0.7867 - val_acc: 0.7311
Epoch 38/50
5608/5608 [==============================] - 0s 42us/step - loss:
0.4685 - acc: 0.8274 - val_loss: 0.8117 - val_acc: 0.7254
Epoch 39/50
5608/5608 [==============================] - 0s 42us/step - loss:
0.4733 - acc: 0.8226 - val_loss: 0.8399 - val_acc: 0.7261
Epoch 40/50
5608/5608 [==============================] - 0s 45us/step - loss:
0.4532 - acc: 0.8318 - val_loss: 0.8455 - val_acc: 0.7147
Epoch 41/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.4510 - acc: 0.8374 - val_loss: 0.8388 - val_acc: 0.7268
Epoch 42/50
5608/5608 [==============================] - 0s 43us/step - loss:
```

```
0.4466 - acc: 0.8352 - val_loss: 0.8472 - val_acc: 0.7304
Epoch 43/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.4521 - acc: 0.8315 - val_loss: 0.8546 - val_acc: 0.7218
Epoch 44/50
5608/5608 [==============================] - 0s 45us/step - loss:
0.4379 - acc: 0.8409 - val_loss: 0.8294 - val_acc: 0.7304
Epoch 45/50
5608/5608 [==============================] - 0s 44us/step - loss:
0.4183 - acc: 0.8474 - val_loss: 0.8414 - val_acc: 0.7389
Epoch 46/50
5608/5608 [==============================] - 0s 42us/step - loss:
0.4272 - acc: 0.8409 - val_loss: 0.8914 - val_acc: 0.7183
Epoch 47/50
5608/5608 [==============================] - 0s 44us/step - loss:
0.4219 - acc: 0.8392 - val_loss: 0.8899 - val_acc: 0.7233
Epoch 48/50
5608/5608 [==============================] - 0s 43us/step - loss:
0.4071 - acc: 0.8470 - val_loss: 0.8621 - val_acc: 0.7332
Epoch 49/50
5608/5608 [==============================] - 0s 42us/step - loss:
0.4156 - acc: 0.8452 - val_loss: 0.8397 - val_acc: 0.7347
Epoch 50/50
5608/5608 [==============================] - 0s 45us/step - loss:
0.4161 - acc: 0.8495 - val_loss: 0.8658 - val_acc: 0.7332
dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
```

In [0]:

```python
import matplotlib.pyplot as plt
# summarize history for accuracy
plt.plot(model_history.history['acc'])
plt.plot(model_history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# summarize history for loss
plt.plot(model_history.history['loss'])
plt.plot(model_history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```





In [0]:

```
# Predicting the Test set results
y_pred = classifier.predict(X_test)
pred = list()
for i in range(len(y_pred)):
    pred.append(np.argmax(y_pred[i]))
#Converting one hot encoded test label to label
test = list()
for i in range(len(y_test)):
    test.append(y_test[i])
```

In [0]:

```
from sklearn.metrics import accuracy_score
a = accuracy_score(pred,test)
print('Accuracy is:', a*100)
```

```
Accuracy is: 72.31281198003327
```

In [0]:

```
# Making the Confusion Matrix
from sklearn.metrics import classification_report, confusion_matrix
cm = confusion_matrix(test, pred)
cm
```

Out[0]:

```
array([[  25,   16,   19,    1,    4,   27,    1],
       [  15,   74,   16,    2,    6,   40,   11],
       [  12,   11,  134,    1,   45,  110,    2],
       [   3,   21,   10,    1,    1,    8,    0],
       [   7,    4,   49,    0,  100,  175,    3],
       [   8,   23,   97,    0,  104, 1777,    3],
       [   0,    5,    2,    0,    3,    9,   20]])
```

In [0]:

```
print(classification_report(test, pred))
```

```
              precision    recall  f1-score   support

           0       0.36      0.27      0.31        93
           1       0.48      0.45      0.47       164
           2       0.41      0.43      0.42       315
           3       0.20      0.02      0.04        44
           4       0.38      0.30      0.33       338
           5       0.83      0.88      0.85      2012
           6       0.50      0.51      0.51        39

    accuracy                           0.71      3005
   macro avg       0.45      0.41      0.42      3005
weighted avg       0.69      0.71      0.70      3005
```

Fully Connected Neural Network

In [0]:

```
classifier1 = Sequential()

# Adding the input layer and the second hidden layer
classifier1.add(Dense(100 ,input_dim = X_train.shape[1]))
classifier1.add(Activation('relu'))
classifier1.add(Dense(100))
classifier1.add(Activation('relu'))
classifier1.add(Dropout(0.3))
classifier1.add(Dense(100))
classifier1.add(Activation('relu'))
classifier1.add(Dropout(0.3))
classifier1.add(Dense(7, activation = 'softmax'))
classifier1.compile(optimizer = 'adam', loss = 'sparse_categorical_cro
ssentropy', metrics = ['accuracy'])
model1_history=classifier1.fit(X_train, y_train,validation_split=0.20,
batch_size = 128, nb_epoch = 50)
print(model1_history.history.keys())
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:14: Use
rWarning: The `nb_epoch` argument in `fit` has been renamed `epoch
s`.


Train on 5608 samples, validate on 1402 samples
Epoch 1/50
5608/5608 [==============================] - 1s 212us/step - loss:
1.1630 - acc: 0.6279 - val_loss: 0.9004 - val_acc: 0.6862
Epoch 2/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.9512 - acc: 0.6730 - val_loss: 0.8552 - val_acc: 0.6983
Epoch 3/50
5608/5608 [==============================] - 0s 49us/step - loss:
0.9025 - acc: 0.6808 - val_loss: 0.8331 - val_acc: 0.7033
Epoch 4/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.8773 - acc: 0.6912 - val_loss: 0.8554 - val_acc: 0.7104
Epoch 5/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.8596 - acc: 0.6945 - val_loss: 0.8139 - val_acc: 0.7147
Epoch 6/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.8096 - acc: 0.7115 - val_loss: 0.8241 - val_acc: 0.7118
Epoch 7/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.7965 - acc: 0.7143 - val_loss: 0.7855 - val_acc: 0.7147
Epoch 8/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.7701 - acc: 0.7238 - val_loss: 0.7971 - val_acc: 0.7175
Epoch 9/50
5608/5608 [==============================] - 0s 49us/step - loss:
0.7568 - acc: 0.7236 - val_loss: 0.7867 - val_acc: 0.7261
Epoch 10/50
5608/5608 [==============================] - 0s 46us/step - loss:
0.7219 - acc: 0.7400 - val_loss: 0.8049 - val_acc: 0.7147
Epoch 11/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.6939 - acc: 0.7495 - val_loss: 0.7993 - val_acc: 0.7247
Epoch 12/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.6773 - acc: 0.7532 - val_loss: 0.7899 - val_acc: 0.7183
Epoch 13/50
5608/5608 [==============================] - 0s 45us/step - loss:
0.6751 - acc: 0.7611 - val_loss: 0.8167 - val_acc: 0.7118
Epoch 14/50
5608/5608 [==============================] - 0s 49us/step - loss:
0.6372 - acc: 0.7636 - val_loss: 0.7902 - val_acc: 0.7268
Epoch 15/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.6407 - acc: 0.7625 - val_loss: 0.8039 - val_acc: 0.7204
Epoch 16/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.6274 - acc: 0.7668 - val_loss: 0.8306 - val_acc: 0.7111
Epoch 17/50
5608/5608 [==============================] - 0s 48us/step - loss:
```

```
0.5858 - acc: 0.7899 - val_loss: 0.8382 - val_acc: 0.7168
Epoch 18/50
5608/5608 [==============================] - 0s 47us/step - loss:

0.5652 - acc: 0.7924 - val_loss: 0.8097 - val_acc: 0.7168
Epoch 19/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.5535 - acc: 0.8001 - val_loss: 0.8275 - val_acc: 0.7240
Epoch 20/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.5512 - acc: 0.7958 - val_loss: 0.8485 - val_acc: 0.7290
Epoch 21/50
5608/5608 [==============================] - 0s 45us/step - loss:
0.5017 - acc: 0.8113 - val_loss: 0.8243 - val_acc: 0.7225
Epoch 22/50
5608/5608 [==============================] - 0s 46us/step - loss:
0.4964 - acc: 0.8222 - val_loss: 0.8731 - val_acc: 0.7268
Epoch 23/50
5608/5608 [==============================] - 0s 50us/step - loss:
0.4744 - acc: 0.8211 - val_loss: 0.9204 - val_acc: 0.7211
Epoch 24/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.4675 - acc: 0.8306 - val_loss: 0.9205 - val_acc: 0.7261
Epoch 25/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.4396 - acc: 0.8395 - val_loss: 0.9316 - val_acc: 0.7161
Epoch 26/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.4135 - acc: 0.8524 - val_loss: 0.8918 - val_acc: 0.7297
Epoch 27/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.4371 - acc: 0.8388 - val_loss: 0.9403 - val_acc: 0.7240
Epoch 28/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.4546 - acc: 0.8340 - val_loss: 0.9244 - val_acc: 0.7211
Epoch 29/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.4093 - acc: 0.8493 - val_loss: 1.0173 - val_acc: 0.7154
Epoch 30/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.3857 - acc: 0.8611 - val_loss: 1.0086 - val_acc: 0.7354
Epoch 31/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.3581 - acc: 0.8693 - val_loss: 1.0383 - val_acc: 0.7168
Epoch 32/50
5608/5608 [==============================] - 0s 46us/step - loss:
0.3595 - acc: 0.8682 - val_loss: 1.0154 - val_acc: 0.7261
Epoch 33/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.3407 - acc: 0.8759 - val_loss: 1.0217 - val_acc: 0.7211
Epoch 34/50
5608/5608 [==============================] - 0s 49us/step - loss:
0.3409 - acc: 0.8762 - val_loss: 1.0240 - val_acc: 0.7147
Epoch 35/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.2923 - acc: 0.8905 - val_loss: 1.0890 - val_acc: 0.7290
Epoch 36/50
```

```
5608/5608 [==============================] - 0s 46us/step - loss:
0.2942 - acc: 0.8907 - val_loss: 1.0665 - val_acc: 0.7389
Epoch 37/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.2921 - acc: 0.8900 - val_loss: 1.2216 - val_acc: 0.7233
Epoch 38/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.2932 - acc: 0.8928 - val_loss: 1.2760 - val_acc: 0.7161
Epoch 39/50
5608/5608 [==============================] - 0s 46us/step - loss:
0.2946 - acc: 0.8976 - val_loss: 1.1550 - val_acc: 0.7325
Epoch 40/50
5608/5608 [==============================] - 0s 45us/step - loss:
0.2634 - acc: 0.9050 - val_loss: 1.1881 - val_acc: 0.7190
Epoch 41/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.2666 - acc: 0.9030 - val_loss: 1.1821 - val_acc: 0.7240
Epoch 42/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.2504 - acc: 0.9105 - val_loss: 1.2085 - val_acc: 0.7389
Epoch 43/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.2702 - acc: 0.9092 - val_loss: 1.2827 - val_acc: 0.7282
Epoch 44/50
5608/5608 [==============================] - 0s 45us/step - loss:
0.2384 - acc: 0.9142 - val_loss: 1.2328 - val_acc: 0.7368
Epoch 45/50
5608/5608 [==============================] - 0s 49us/step - loss:
0.2182 - acc: 0.9237 - val_loss: 1.3069 - val_acc: 0.7354
Epoch 46/50
5608/5608 [==============================] - 0s 49us/step - loss:
0.2110 - acc: 0.9256 - val_loss: 1.4059 - val_acc: 0.7068
Epoch 47/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.2374 - acc: 0.9187 - val_loss: 1.4289 - val_acc: 0.7168
Epoch 48/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.2184 - acc: 0.9206 - val_loss: 1.2973 - val_acc: 0.7061
Epoch 49/50
5608/5608 [==============================] - 0s 47us/step - loss:
0.2198 - acc: 0.9228 - val_loss: 1.3335 - val_acc: 0.7133
Epoch 50/50
5608/5608 [==============================] - 0s 48us/step - loss:
0.2158 - acc: 0.9228 - val_loss: 1.4246 - val_acc: 0.7211
dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
```

In [0]:

```
# Predicting the Test set results
y_pred1 = classifier1.predict(X_test)
pred1 = list()
for i in range(len(y_pred1)):
    pred1.append(np.argmax(y_pred1[i]))
#Converting one hot encoded test label to label
test1 = list()
for i in range(len(y_test)):
    test1.append(y_test[i])
```

In [0]:

```
a1= accuracy_score(pred1,test1)
print('Accuracy is:', a1*100)
```

```
Accuracy is: 71.71381031613977
```

In [0]:

```
# Making the Confusion Matrix
cm1= confusion_matrix(test1, pred1)
cm1
```

Out[0]:

```
array([[  28,   27,    6,    0,    6,   25,    1],
       [   7,   91,    8,    3,    3,   43,    9],
       [   9,   24,  106,    2,   37,  136,    1],
       [   2,   15,    7,    2,    1,   17,    0],
       [  10,   10,   31,    0,  100,  185,    2],
       [   5,   34,   63,    2,   88, 1814,    6],
       [   1,    5,    2,    1,    3,   13,   14]])
```

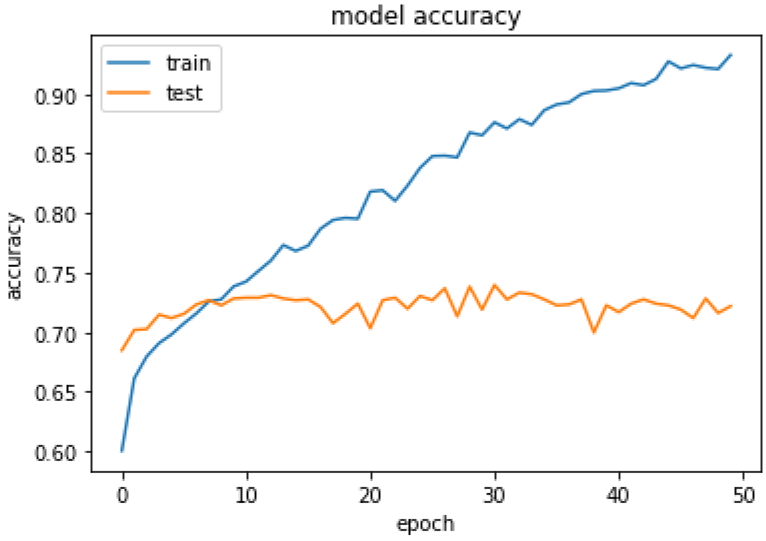
In [0]:

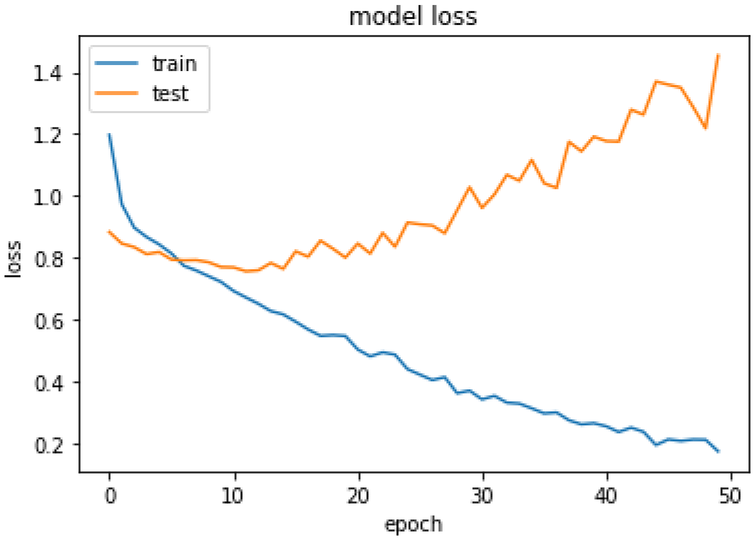
```
print(classification_report(test1, pred1))
```

```
              precision    recall  f1-score   support

           0       0.45      0.30      0.36        93
           1       0.44      0.55      0.49       164
           2       0.48      0.34      0.39       315
           3       0.20      0.05      0.07        44
           4       0.42      0.30      0.35       338
           5       0.81      0.90      0.85      2012
           6       0.42      0.36      0.39        39

    accuracy                           0.72      3005
   macro avg       0.46      0.40      0.42      3005
weighted avg       0.69      0.72      0.70      3005
```

In [0]:

```
# summarize history for accuracy
plt.plot(model1_history.history['acc'])
plt.plot(model1_history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# summarize history for loss

plt.plot(model1_history.history['loss'])
plt.plot(model1_history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```





In [0]:

```
classifier2 = Sequential()


# Adding the input layer and the three hidden layer
classifier2.add(Dense(100 ,input_dim = X_train.shape[1]))
classifier2.add(Activation('relu'))
classifier2.add(Dropout(0.3))
classifier2.add(Dense(100))
classifier2.add(Activation('relu'))
classifier2.add(Dropout(0.3))
classifier2.add(Dense(100))
classifier2.add(Activation('relu'))
classifier2.add(Dropout(0.3))
classifier2.add(Dense(100))
classifier2.add(Activation('relu'))
classifier2.add(Dropout(0.3))
classifier2.add(Dense(7, activation = 'softmax'))
classifier2.compile(optimizer = 'adam', loss = 'sparse_categorical_cro
ssentropy', metrics = ['accuracy'])
model2_history=classifier2.fit(X_train, y_train,validation_split=0.33,
batch_size = 128, nb_epoch = 50)
print(model2_history.history.keys())
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:18: Use
rWarning: The `nb_epoch` argument in `fit` has been renamed `epoch
s`.

Train on 4696 samples, validate on 2314 samples
Epoch 1/50
4696/4696 [==============================] - 2s 329us/step - loss:
1.3385 - acc: 0.5841 - val_loss: 0.9820 - val_acc: 0.6707
Epoch 2/50
4696/4696 [==============================] - 0s 57us/step - loss:
1.0601 - acc: 0.6469 - val_loss: 0.9583 - val_acc: 0.6560
Epoch 3/50
4696/4696 [==============================] - 0s 59us/step - loss:
1.0166 - acc: 0.6508 - val_loss: 0.9232 - val_acc: 0.6733
Epoch 4/50
4696/4696 [==============================] - 0s 59us/step - loss:
0.9859 - acc: 0.6625 - val_loss: 0.9037 - val_acc: 0.6819
Epoch 5/50
4696/4696 [==============================] - 0s 56us/step - loss:
0.9776 - acc: 0.6597 - val_loss: 0.9157 - val_acc: 0.6876
Epoch 6/50
4696/4696 [==============================] - 0s 58us/step - loss:
0.9449 - acc: 0.6650 - val_loss: 0.8776 - val_acc: 0.6884
Epoch 7/50
4696/4696 [==============================] - 0s 55us/step - loss:
0.9327 - acc: 0.6740 - val_loss: 0.9063 - val_acc: 0.6850
Epoch 8/50
4696/4696 [==============================] - 0s 60us/step - loss:
0.9164 - acc: 0.6714 - val_loss: 0.8981 - val_acc: 0.7061
Epoch 9/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.8910 - acc: 0.6812 - val_loss: 0.8618 - val_acc: 0.6901
Epoch 10/50
4696/4696 [==============================] - 0s 57us/step - loss:
```

```
4696/4696 [==============================] - 0s 57us/step - loss:
0.8883 - acc: 0.6823 - val_loss: 0.8466 - val_acc: 0.7083
Epoch 11/50

4696/4696 [==============================] - 0s 55us/step - loss:
0.8823 - acc: 0.6868 - val_loss: 0.8463 - val_acc: 0.7005
Epoch 12/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.8546 - acc: 0.6899 - val_loss: 0.8458 - val_acc: 0.7048
Epoch 13/50
4696/4696 [==============================] - 0s 55us/step - loss:
0.8516 - acc: 0.6910 - val_loss: 0.8341 - val_acc: 0.7122
Epoch 14/50
4696/4696 [==============================] - 0s 59us/step - loss:
0.8454 - acc: 0.6966 - val_loss: 0.8443 - val_acc: 0.7174
Epoch 15/50
4696/4696 [==============================] - 0s 59us/step - loss:
0.8274 - acc: 0.6919 - val_loss: 0.8106 - val_acc: 0.7135
Epoch 16/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.8185 - acc: 0.7002 - val_loss: 0.8222 - val_acc: 0.7105
Epoch 17/50
4696/4696 [==============================] - 0s 56us/step - loss:
0.8134 - acc: 0.7068 - val_loss: 0.8177 - val_acc: 0.7118
Epoch 18/50
4696/4696 [==============================] - 0s 60us/step - loss:
0.8159 - acc: 0.7023 - val_loss: 0.8062 - val_acc: 0.7118
Epoch 19/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.8049 - acc: 0.7098 - val_loss: 0.8318 - val_acc: 0.7113
Epoch 20/50
4696/4696 [==============================] - 0s 56us/step - loss:
0.8008 - acc: 0.7051 - val_loss: 0.8264 - val_acc: 0.7152
Epoch 21/50
4696/4696 [==============================] - 0s 55us/step - loss:
0.7879 - acc: 0.7119 - val_loss: 0.8167 - val_acc: 0.7131
Epoch 22/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.7685 - acc: 0.7198 - val_loss: 0.8097 - val_acc: 0.7092
Epoch 23/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.7778 - acc: 0.7189 - val_loss: 0.7976 - val_acc: 0.7191
Epoch 24/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.7594 - acc: 0.7221 - val_loss: 0.8054 - val_acc: 0.7165
Epoch 25/50
4696/4696 [==============================] - 0s 56us/step - loss:
0.7496 - acc: 0.7215 - val_loss: 0.7903 - val_acc: 0.7169
Epoch 26/50
4696/4696 [==============================] - 0s 69us/step - loss:
0.7434 - acc: 0.7270 - val_loss: 0.7868 - val_acc: 0.7178
Epoch 27/50
4696/4696 [==============================] - 0s 64us/step - loss:
0.7591 - acc: 0.7230 - val_loss: 0.8039 - val_acc: 0.7277
Epoch 28/50
4696/4696 [==============================] - 0s 61us/step - loss:
0.7341 - acc: 0.7302 - val_loss: 0.7983 - val_acc: 0.7226
```

```
Epoch 29/50
4696/4696 [==============================] - 0s 56us/step - loss:
0.7276 - acc: 0.7353 - val_loss: 0.7919 - val_acc: 0.7217

Epoch 30/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.7098 - acc: 0.7306 - val_loss: 0.7986 - val_acc: 0.7226
Epoch 31/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.7146 - acc: 0.7359 - val_loss: 0.8032 - val_acc: 0.7187
Epoch 32/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.7026 - acc: 0.7389 - val_loss: 0.8112 - val_acc: 0.7148
Epoch 33/50
4696/4696 [==============================] - 0s 58us/step - loss:
0.6875 - acc: 0.7474 - val_loss: 0.7853 - val_acc: 0.7247
Epoch 34/50
4696/4696 [==============================] - 0s 58us/step - loss:
0.6951 - acc: 0.7428 - val_loss: 0.7923 - val_acc: 0.7213
Epoch 35/50
4696/4696 [==============================] - 0s 58us/step - loss:
0.6787 - acc: 0.7487 - val_loss: 0.7741 - val_acc: 0.7234
Epoch 36/50
4696/4696 [==============================] - 0s 59us/step - loss:
0.6862 - acc: 0.7464 - val_loss: 0.8087 - val_acc: 0.7200
Epoch 37/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.6852 - acc: 0.7513 - val_loss: 0.7935 - val_acc: 0.7200
Epoch 38/50
4696/4696 [==============================] - 0s 56us/step - loss:
0.6520 - acc: 0.7598 - val_loss: 0.7808 - val_acc: 0.7273
Epoch 39/50
4696/4696 [==============================] - 0s 56us/step - loss:
0.6480 - acc: 0.7611 - val_loss: 0.7902 - val_acc: 0.7247
Epoch 40/50
4696/4696 [==============================] - 0s 61us/step - loss:
0.6511 - acc: 0.7587 - val_loss: 0.8000 - val_acc: 0.7213
Epoch 41/50
4696/4696 [==============================] - 0s 56us/step - loss:
0.6397 - acc: 0.7660 - val_loss: 0.7894 - val_acc: 0.7282
Epoch 42/50
4696/4696 [==============================] - 0s 58us/step - loss:
0.6499 - acc: 0.7666 - val_loss: 0.7727 - val_acc: 0.7282
Epoch 43/50
4696/4696 [==============================] - 0s 55us/step - loss:
0.6362 - acc: 0.7570 - val_loss: 0.7949 - val_acc: 0.7303
Epoch 44/50
4696/4696 [==============================] - 0s 56us/step - loss:
0.6210 - acc: 0.7666 - val_loss: 0.8119 - val_acc: 0.7174
Epoch 45/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.6259 - acc: 0.7609 - val_loss: 0.7901 - val_acc: 0.7269
Epoch 46/50
4696/4696 [==============================] - 0s 59us/step - loss:
0.6105 - acc: 0.7798 - val_loss: 0.7837 - val_acc: 0.7217
Epoch 47/50
4696/4696 [==============================] - 0s 57us/step - loss:
```

```
0.6164 - acc: 0.7777 - val_loss: 0.8037 - val_acc: 0.7264
Epoch 48/50
4696/4696 [==============================] - 0s 56us/step - loss:
0.5919 - acc: 0.7866 - val_loss: 0.8070 - val_acc: 0.7269
Epoch 49/50
4696/4696 [==============================] - 0s 57us/step - loss:
0.5926 - acc: 0.7792 - val_loss: 0.8001 - val_acc: 0.7221
Epoch 50/50
4696/4696 [==============================] - 0s 55us/step - loss:
0.5769 - acc: 0.7868 - val_loss: 0.8034 - val_acc: 0.7303
dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
```

In [0]:

```
# Predicting the Test set results
y_pred2 = classifier2.predict(X_test)
pred2 = list()
for i in range(len(y_pred2)):
    pred2.append(np.argmax(y_pred2[i]))
#Converting one hot encoded test label to label
test2 = list()
for i in range(len(y_test)):
    test2.append(y_test[i])
```

In [0]:

```
a2 = accuracy_score(pred2,test2)
print('Accuracy is:', a2*100)
```

```
Accuracy is: 71.48086522462562
```

In [0]:

```
# Making the Confusion Matrix
cm2= confusion_matrix(test2, pred2)
cm2
```

Out[0]:

```
array([[  21,   27,   20,    0,    1,   24,    0],
       [  11,   95,   22,    0,    0,   31,    5],
       [   6,   27,  131,    0,    9,  141,    1],
       [   3,   18,   13,    0,    0,   10,    0],
       [   5,   12,   66,    0,   28,  227,    0],
       [   3,   40,   84,    0,   14, 1869,    2],
       [   2,    8,    1,    0,    0,   24,    4]])
```

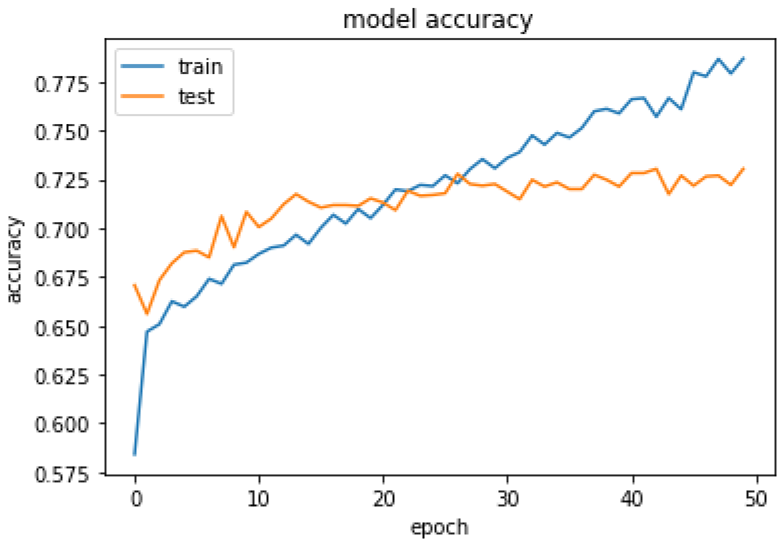
In [0]:

```
print(classification_report(test2, pred2))
```

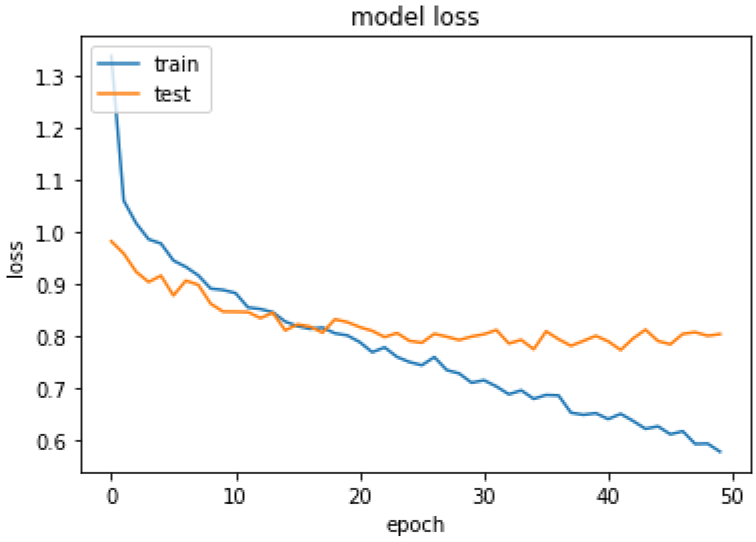
```
              precision    recall  f1-score   support

           0       0.41      0.23      0.29        93
           1       0.42      0.58      0.49       164
           2       0.39      0.42      0.40       315
           3       0.00      0.00      0.00        44
           4       0.54      0.08      0.14       338
           5       0.80      0.93      0.86      2012
           6       0.33      0.10      0.16        39

    accuracy                           0.71      3005
   macro avg       0.41      0.33      0.33      3005
weighted avg       0.68      0.71      0.67      3005


/usr/local/lib/python3.6/dist-packages/sklearn/metrics/_classificati
on.py:1272: UndefinedMetricWarning: Precision and F-score are ill-de
fined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [0]:

```
# summarize history for accuracy
plt.plot(model2_history.history['acc'])
plt.plot(model2_history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# summarize history for loss
plt.plot(model2_history.history['loss'])
plt.plot(model2_history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



model accuracy
train
test
accuracy
0.775
0.750
0.725
0.700
0.675
0.650
0.625
0.600
0.575
0
10
20
30
40
50
epoch



model loss
train
test
loss
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0
10
20
30
40
50
epoch

In [0]:

```python
classifier3 = Sequential()

# Adding the input layer and the three hidden layer
classifier3.add(Dense(100 ,input_dim = X_train.shape[1]))
classifier3.add(Activation('relu'))
classifier3.add(Dropout(0.3))
classifier3.add(Dense(100))
classifier3.add(Activation('relu'))
classifier3.add(Dropout(0.3))
classifier3.add(Dense(100))
classifier3.add(Activation('relu'))
classifier3.add(Dropout(0.3))
classifier3.add(Dense(100))
classifier3.add(Activation('relu'))
classifier3.add(Dropout(0.3))
classifier3.add(Dense(100))
classifier3.add(Activation('relu'))
classifier3.add(Dropout(0.3))
classifier3.add(Dense(7, activation = 'softmax'))
classifier3.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics = ['accuracy'])
model3_history=classifier3.fit(X_train, y_train,validation_split=0.33, batch_size = 128, nb_epoch = 50)
print(model3_history.history.keys())
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:21: UserWarning: The `nb_epoch` argument in `fit` has been renamed `epochs`.

Train on 4696 samples, validate on 2314 samples
Epoch 1/50
4696/4696 [==============================] - 2s 361us/step - loss: 1.2259 - acc: 0.6152 - val_loss: 0.9811 - val_acc: 0.6716
Epoch 2/50
4696/4696 [==============================] - 0s 60us/step - loss: 1.0467 - acc: 0.6491 - val_loss: 0.9677 - val_acc: 0.6711
Epoch 3/50
4696/4696 [==============================] - 0s 61us/step - loss: 1.0026 - acc: 0.6593 - val_loss: 0.9631 - val_acc: 0.6724
Epoch 4/50
4696/4696 [==============================] - 0s 65us/step - loss: 0.9913 - acc: 0.6589 - val_loss: 0.9669 - val_acc: 0.6780
Epoch 5/50
4696/4696 [==============================] - 0s 77us/step - loss: 0.9756 - acc: 0.6603 - val_loss: 0.9404 - val_acc: 0.6733
Epoch 6/50
4696/4696 [==============================] - 0s 66us/step - loss: 0.9533 - acc: 0.6650 - val_loss: 0.9263 - val_acc: 0.6832
Epoch 7/50
4696/4696 [==============================] - 0s 66us/step - loss: 0.9365 - acc: 0.6657 - val_loss: 0.9501 - val_acc: 0.6880
Epoch 8/50
4696/4696 [==============================] - 0s 59us/step - loss: 0.9331 - acc: 0.6727 - val_loss: 0.9345 - val_acc: 0.6871
Epoch 9/50
4696/4696 [==============================] - 0s 60us/step - loss:
```

```
0.9145 - acc: 0.6697 - val_loss: 0.9224 - val_acc: 0.6940
Epoch 10/50
4696/4696 [==============================] - 0s 59us/step - loss:
0.9056 - acc: 0.6655 - val_loss: 0.8952 - val_acc: 0.6923
Epoch 11/50
4696/4696 [==============================] - 0s 64us/step - loss:
0.8991 - acc: 0.6738 - val_loss: 0.9307 - val_acc: 0.6979
Epoch 12/50
4696/4696 [==============================] - 0s 61us/step - loss:
0.8787 - acc: 0.6789 - val_loss: 0.8843 - val_acc: 0.7018
Epoch 13/50
4696/4696 [==============================] - 0s 58us/step - loss:
0.8945 - acc: 0.6808 - val_loss: 0.9241 - val_acc: 0.7014
Epoch 14/50
4696/4696 [==============================] - 0s 58us/step - loss:
0.8591 - acc: 0.6851 - val_loss: 0.8768 - val_acc: 0.7087
Epoch 15/50
4696/4696 [==============================] - 0s 60us/step - loss:
0.8569 - acc: 0.6844 - val_loss: 0.8635 - val_acc: 0.7113
Epoch 16/50
4696/4696 [==============================] - 0s 59us/step - loss:

0.8432 - acc: 0.6910 - val_loss: 0.8621 - val_acc: 0.7057
Epoch 17/50
4696/4696 [==============================] - 0s 59us/step - loss:
0.8348 - acc: 0.6934 - val_loss: 0.8614 - val_acc: 0.7057
Epoch 18/50
4696/4696 [==============================] - 0s 66us/step - loss:
0.8446 - acc: 0.6966 - val_loss: 0.9033 - val_acc: 0.7031
Epoch 19/50
4696/4696 [==============================] - 0s 59us/step - loss:
0.8247 - acc: 0.6995 - val_loss: 0.8392 - val_acc: 0.7061
Epoch 20/50
4696/4696 [==============================] - 0s 64us/step - loss:
0.8260 - acc: 0.7070 - val_loss: 0.8547 - val_acc: 0.7074
Epoch 21/50
4696/4696 [==============================] - 0s 58us/step - loss:
0.8266 - acc: 0.6993 - val_loss: 0.8589 - val_acc: 0.7113
Epoch 22/50
4696/4696 [==============================] - 0s 58us/step - loss:
0.8052 - acc: 0.7072 - val_loss: 0.8743 - val_acc: 0.7139
Epoch 23/50
4696/4696 [==============================] - 0s 61us/step - loss:
0.8017 - acc: 0.7012 - val_loss: 0.8605 - val_acc: 0.7070
Epoch 24/50
4696/4696 [==============================] - 0s 65us/step - loss:
0.7987 - acc: 0.7036 - val_loss: 0.8622 - val_acc: 0.7165
Epoch 25/50
4696/4696 [==============================] - 0s 60us/step - loss:
0.7890 - acc: 0.7053 - val_loss: 0.8424 - val_acc: 0.7204
Epoch 26/50
4696/4696 [==============================] - 0s 62us/step - loss:
0.7732 - acc: 0.7117 - val_loss: 0.8353 - val_acc: 0.7165
Epoch 27/50
4696/4696 [==============================] - 0s 62us/step - loss:
0.7843 - acc: 0.7091 - val_loss: 0.8334 - val_acc: 0.7195
Epoch 28/50
Epoch 28/50
4696/4696 [==============================] - 0s 58us/step - loss:
0.7707 - acc: 0.7117 - val_loss: 0.8137 - val_acc: 0.7178
Epoch 29/50
4696/4696 [==============================] - 0s 60us/step - loss:
0.7661 - acc: 0.7164 - val_loss: 0.8303 - val_acc: 0.7096
Epoch 30/50
4696/4696 [==============================] - 0s 60us/step - loss:
0.7691 - acc: 0.7161 - val_loss: 0.8355 - val_acc: 0.7204
Epoch 31/50
4696/4696 [==============================] - 0s 65us/step - loss:
0.7475 - acc: 0.7238 - val_loss: 0.8031 - val_acc: 0.7260
Epoch 32/50
4696/4696 [==============================] - 0s 59us/step - loss:
0.7540 - acc: 0.7198 - val_loss: 0.8506 - val_acc: 0.7048
Epoch 33/50
4696/4696 [==============================] - 0s 60us/step - loss:
0.7327 - acc: 0.7306 - val_loss: 0.8310 - val_acc: 0.7165
Epoch 34/50
4696/4696 [==============================] - 0s 61us/step - loss:
0.7343 - acc: 0.7219 - val_loss: 0.8367 - val_acc: 0.7122
Epoch 35/50

4696/4696 [==============================] - 0s 63us/step - loss:
0.7312 - acc: 0.7334 - val_loss: 0.8246 - val_acc: 0.7234
Epoch 36/50
4696/4696 [==============================] - 0s 60us/step - loss:
0.7191 - acc: 0.7347 - val_loss: 0.8186 - val_acc: 0.7325
Epoch 37/50
4696/4696 [==============================] - 0s 62us/step - loss:
0.7208 - acc: 0.7325 - val_loss: 0.8244 - val_acc: 0.7178
Epoch 38/50
4696/4696 [==============================] - 0s 60us/step - loss:
0.7154 - acc: 0.7379 - val_loss: 0.8075 - val_acc: 0.7208
Epoch 39/50
4696/4696 [==============================] - 0s 63us/step - loss:
0.6991 - acc: 0.7428 - val_loss: 0.8306 - val_acc: 0.7169
Epoch 40/50
4696/4696 [==============================] - 0s 62us/step - loss:
0.7108 - acc: 0.7381 - val_loss: 0.8422 - val_acc: 0.7299
Epoch 41/50
4696/4696 [==============================] - 0s 63us/step - loss:
0.7110 - acc: 0.7323 - val_loss: 0.8205 - val_acc: 0.7122
Epoch 42/50
4696/4696 [==============================] - 0s 61us/step - loss:
0.6999 - acc: 0.7319 - val_loss: 0.8262 - val_acc: 0.7221
Epoch 43/50
4696/4696 [==============================] - 0s 62us/step - loss:
0.6671 - acc: 0.7543 - val_loss: 0.7978 - val_acc: 0.7204
Epoch 44/50
4696/4696 [==============================] - 0s 63us/step - loss:
0.6718 - acc: 0.7453 - val_loss: 0.8344 - val_acc: 0.7208
Epoch 45/50
4696/4696 [==============================] - 0s 60us/step - loss:
0.6717 - acc: 0.7477 - val_loss: 0.7988 - val_acc: 0.7273
Epoch 46/50
4696/4696 [==============================] - 0s 61us/step - loss:
0.6596 - acc: 0.7543 - val_loss: 0.8037 - val_acc: 0.7290
```

```
0.6596 - acc: 0.7543 - val_loss: 0.8037 - val_acc: 0.7290
Epoch 47/50
4696/4696 [==============================] - 0s 61us/step - loss:
0.6663 - acc: 0.7523 - val_loss: 0.8173 - val_acc: 0.7234
Epoch 48/50
4696/4696 [==============================] - 0s 64us/step - loss:
0.6519 - acc: 0.7474 - val_loss: 0.7846 - val_acc: 0.7252
Epoch 49/50
4696/4696 [==============================] - 0s 58us/step - loss:
0.6497 - acc: 0.7566 - val_loss: 0.8213 - val_acc: 0.7161
Epoch 50/50
4696/4696 [==============================] - 0s 61us/step - loss:
0.6418 - acc: 0.7621 - val_loss: 0.7987 - val_acc: 0.7165
dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
```

In [0]:

```
# Predicting the Test set results
y_pred3 = classifier2.predict(X_test)
pred3 = list()
for i in range(len(y_pred3)):
    pred3.append(np.argmax(y_pred3[i]))
#Converting one hot encoded test label to label
test3 = list()
for i in range(len(y_test)):
    test3.append(y_test[i])
```

In [0]:

```
a3 = accuracy_score(pred3,test3)
print('Accuracy is:', a3*100)
```

```
Accuracy is: 71.48086522462562
```

In [0]:

```
# Making the Confusion Matrix
cm3= confusion_matrix(test3, pred3)
cm3
```

Out[0]:

```
array([[  21,   27,   20,    0,    1,   24,    0],
       [  11,   95,   22,    0,    0,   31,    5],
       [   6,   27,  131,    0,    9,  141,    1],
       [   3,   18,   13,    0,    0,   10,    0],
       [   5,   12,   66,    0,   28,  227,    0],
       [   3,   40,   84,    0,   14, 1869,    2],
       [   2,    8,    1,    0,    0,   24,    4]])
```

In [0]:

```
print(classification_report(test3, pred3))
```

```
              precision    recall  f1-score   support

           0       0.41      0.23      0.29        93
           1       0.42      0.58      0.49       164
           2       0.39      0.42      0.40       315
           3       0.00      0.00      0.00        44
           4       0.54      0.08      0.14       338
           5       0.80      0.93      0.86      2012
           6       0.33      0.10      0.16        39

    accuracy                           0.71      3005
   macro avg       0.41      0.33      0.33      3005
weighted avg       0.68      0.71      0.67      3005


/usr/local/lib/python3.6/dist-packages/sklearn/metrics/_classificati
on.py:1272: UndefinedMetricWarning: Precision and F-score are ill-de
fined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [0]:

```

```