Human Activity Recognition

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
# Activities are the class labels
# It is a 6 class classification
ACTIVITIES = {
    0: 'WALKING',
    1: 'WALKING_UPSTAIRS',
    2: 'WALKING_DOWNSTAIRS',
    3: 'SITTING',
    4: 'STANDING',
    5: 'LAYING',
}

# Utility function to print the confusion matrix
def confusion_matrix(Y_true, Y_pred):
    Y_true = pd.Series([ACTIVITIES[y] for y in np.argmax(Y_true, axis=1)])
    Y_pred = pd.Series([ACTIVITIES[y] for y in np.argmax(Y_pred, axis=1)])

    return pd.crosstab(Y_true, Y_pred, rownames=['True'], colnames=['Pred'])
```

Data

In [4]:

```
# Raw data signals
# Signals are from Accelerometer and Gyroscope
# The signals are in x,y,z directions
# Sensor signals are filtered to have only body acceleration
# excluding the acceleration due to gravity
# Triaxial acceleration from the accelerometer is total acceleration
SIGNALS = [
    "body_acc_x",
    "body_acc_y",
    "body_acc_z",
    "body_gyro_x",
    "body_gyro_y",
    "body_gyro_z",
    "total_acc_x",
    "total_acc_y",
    "total_acc_z"
]
```

In [4]:

```
'''

# Code to read csv file into Colaboratory from google drive:

!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials

# Authenticate and create the PyDrive client.
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)

'''
```

Out[4]:

```
'\n\n# Code to read csv file into Colaboratory from google drive:\n\n!pip
install -U -q PyDrive\nfrom pydrive.auth import GoogleAuth\nfrom pydrive.d
rive import GoogleDrive\nfrom google.colab import auth\nfrom oauth2client.
client import GoogleCredentials\n\n# Authenticate and create the PyDrive c
lient.\nauth.authenticate_user()\ngauth = GoogleAuth()\ngauth.credentials
= GoogleCredentials.get_application_default()\ndrive = GoogleDrive(gauth)
\n\n'
```

In [0]:

```
'''
link = 'https://drive.google.com/open?id=1Kg_2TU9eHe-oxO3vZdz-d5QFUx3vcTS4' #X_train
link1 = 'https://drive.google.com/open?id=12S6ntD_pIBvPNVsAlxmSDeEbvi2YGnrd' # X_test
link2 = 'https://drive.google.com/open?id=1IuMv7H7XPxp7JrGixKW4JcXJ3PLvMGh6' #y_train
link3 = 'https://drive.google.com/open?id=1oDp9e3Pf1kmCnfgAzL0xTQgCsPw8nB_B' #y_test

'''
```

In [17]:

```
#fluff, id = link3.split('=')

#print (id)
```

```
1oDp9e3Pf1kmCnfgAzL0xTQgCsPw8nB_B
```

In [0]:

```
#downloaded = drive.CreateFile({'id':id})
#downloaded.GetContentFile('X_train.txt')
```

In [5]:

```
# Utility function to read the data from csv file
def _read_csv(filename):
    return pd.read_csv(filename, delim_whitespace=True, header=None)

# Utility function to load the load
def load_signals(subset):
    signals_data = []

    for signal in SIGNALS:
        filename = f'UCI_HAR_Dataset/{subset}/Inertial Signals/{signal}_{subset}.txt'
        signals_data.append(
            _read_csv(filename).as_matrix()
        )

    # Transpose is used to change the dimensionality of the output,
    # aggregating the signals by combination of sample/timestep.
    # Resultant shape is (7352 train/2947 test samples, 128 timesteps, 9 signals)
    return np.transpose(signals_data, (1, 2, 0))
```

In [6]:

```
def load_y(subset):
    """
    The objective that we are trying to predict is a integer, from 1 to 6,
    that represents a human activity. We return a binary representation of
    every sample objective as a 6 bits vector using One Hot Encoding
    (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.get_dummies.html)
    """
    filename = f'UCI_HAR_Dataset/{subset}/y_{subset}.txt'
    y = _read_csv(filename)[0]

    return pd.get_dummies(y).as_matrix()
```

In [7]:

```
def load_data():
    """
    Obtain the dataset from multiple files.
    Returns: X_train, X_test, y_train, y_test
    """
    X_train, X_test = load_signals('train'), load_signals('test')
    y_train, y_test = load_y('train'), load_y('test')

    return X_train, X_test, y_train, y_test
```

In [52]:

```
# Importing tensorflow
np.random.seed(42)
import tensorflow as tf
tf.set_random_seed(42)
```

In [9]:

```
# Configuring a session
session_conf = tf.ConfigProto(
    intra_op_parallelism_threads=1,
    inter_op_parallelism_threads=1
)
```

In [10]:

```
# Import Keras
from keras import backend as K
sess = tf.Session(graph=tf.get_default_graph(), config=session_conf)
K.set_session(sess)
```

```
Using TensorFlow backend.
```

In [11]:

```
# Importing libraries
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers.core import Dense, Dropout
```

In [13]:

```
# Initializing parameters
epochs = 20
batch_size = 32
n_hidden = 32
```

In [14]:

```
# Utility function to count the number of classes
def _count_classes(y):
    return len(set([tuple(category) for category in y]))
```

In [64]:

```
# Loading the train and test data
X_train, X_test, Y_train, Y_test = load_data()
```

```
C:\Users\Shamim Ahmed\Anaconda3\lib\site-packages\ipykernel_launcher.py:1
2: FutureWarning: Method .as_matrix will be removed in a future version. U
se .values instead.
  if sys.path[0] == '':
C:\Users\Shamim Ahmed\Anaconda3\lib\site-packages\ipykernel_launcher.py:1
1: FutureWarning: Method .as_matrix will be removed in a future version. U
se .values instead.
  # This is added back by InteractiveShellApp.init_path()
```

In [65]:

```
timesteps = len(X_train[0])
input_dim = len(X_train[0][0])
n_classes = _count_classes(Y_train)

print(timesteps)
print(input_dim)
print(len(X_train))
```

```
128
9
7352
```

Defining the Architecture of LSTM

In [17]:

```
# Initialization
model = Sequential()
# configuring with parameters
model.add(LSTM(n_hidden, input_shape=(timesteps, input_dim)))
#Adding a dropout layer
model.add(Dropout(0.5))
#Adding dense layer
model.add(Dense(n_classes, activation = 'sigmoid'))
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_2 (LSTM)                (None, 32)                5376
_________________________________________________________________
dropout_2 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_2 (Dense)              (None, 6)                 198
=================================================================
Total params: 5,574
Trainable params: 5,574
Non-trainable params: 0
_________________________________________________________________
```

In [18]:

```
# Compiling the model
model.compile(loss='categorical_crossentropy',
              optimizer='rmsprop',
              metrics=['accuracy'])
```

In [19]:

```
# Training the model
model.fit(X_train,
          Y_train,
          batch_size=batch_size,
          validation_data=(X_test, Y_test),
          epochs=epochs)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 48s 6ms/step - loss: 1.3960 -
acc: 0.3681 - val_loss: 1.2606 - val_acc: 0.4337
Epoch 2/30
7352/7352 [==============================] - 45s 6ms/step - loss: 1.1265 -
acc: 0.4986 - val_loss: 1.1264 - val_acc: 0.4727
Epoch 3/30
7352/7352 [==============================] - 85s 12ms/step - loss: 1.0019
- acc: 0.5449 - val_loss: 0.9544 - val_acc: 0.5741
Epoch 4/30
7352/7352 [==============================] - 124s 17ms/step - loss: 0.8440
- acc: 0.6209 - val_loss: 0.8784 - val_acc: 0.5914
Epoch 5/30
7352/7352 [==============================] - 124s 17ms/step - loss: 0.7386
- acc: 0.6481 - val_loss: 0.8611 - val_acc: 0.5935
Epoch 6/30
7352/7352 [==============================] - 124s 17ms/step - loss: 0.7624
- acc: 0.6353 - val_loss: 0.9017 - val_acc: 0.5864
Epoch 7/30
7352/7352 [==============================] - 127s 17ms/step - loss: 0.6645
- acc: 0.6640 - val_loss: 0.6681 - val_acc: 0.6162
Epoch 8/30
7352/7352 [==============================] - 129s 18ms/step - loss: 0.6460
- acc: 0.6851 - val_loss: 0.6809 - val_acc: 0.6135
Epoch 9/30
7352/7352 [==============================] - 128s 17ms/step - loss: 0.5765
- acc: 0.7121 - val_loss: 0.6124 - val_acc: 0.6807
Epoch 10/30
7352/7352 [==============================] - 127s 17ms/step - loss: 0.5393
- acc: 0.7601 - val_loss: 0.5380 - val_acc: 0.7397
Epoch 11/30
7352/7352 [==============================] - 127s 17ms/step - loss: 0.4637
- acc: 0.7975 - val_loss: 0.6335 - val_acc: 0.7238
Epoch 12/30
7352/7352 [==============================] - 127s 17ms/step - loss: 0.4141
- acc: 0.8252 - val_loss: 0.7027 - val_acc: 0.7570
Epoch 13/30
7352/7352 [==============================] - 127s 17ms/step - loss: 0.3732
- acc: 0.8794 - val_loss: 0.4991 - val_acc: 0.8544
Epoch 14/30
7352/7352 [==============================] - 127s 17ms/step - loss: 0.3349
- acc: 0.9029 - val_loss: 0.3707 - val_acc: 0.8602
Epoch 15/30
7352/7352 [==============================] - 129s 17ms/step - loss: 0.2677
- acc: 0.9163 - val_loss: 0.3907 - val_acc: 0.8690
Epoch 16/30
7352/7352 [==============================] - 127s 17ms/step - loss: 0.2882
- acc: 0.9135 - val_loss: 0.3232 - val_acc: 0.8839
Epoch 17/30
7352/7352 [==============================] - 129s 17ms/step - loss: 0.2404
- acc: 0.9249 - val_loss: 0.3698 - val_acc: 0.8656
Epoch 18/30
7352/7352 [==============================] - 131s 18ms/step - loss: 0.2267
- acc: 0.9347 - val_loss: 0.3346 - val_acc: 0.8873
Epoch 19/30
7352/7352 [==============================] - 126s 17ms/step - loss: 0.2111
- acc: 0.9314 - val_loss: 0.3247 - val_acc: 0.8901
Epoch 20/30
7352/7352 [==============================] - 126s 17ms/step - loss: 0.1960
- acc: 0.9348 - val_loss: 0.4490 - val_acc: 0.8880
```

```
Epoch 21/30
7352/7352 [==============================] - 127s 17ms/step - loss: 0.2113
- acc: 0.9377 - val_loss: 0.2936 - val_acc: 0.8955
Epoch 22/30
7352/7352 [==============================] - 128s 17ms/step - loss: 0.2029
- acc: 0.9365 - val_loss: 0.3299 - val_acc: 0.8948
Epoch 23/30
7352/7352 [==============================] - 129s 18ms/step - loss: 0.2018
- acc: 0.9385 - val_loss: 0.3521 - val_acc: 0.8846
Epoch 24/30
7352/7352 [==============================] - 130s 18ms/step - loss: 0.1832
- acc: 0.9384 - val_loss: 0.3525 - val_acc: 0.9006
Epoch 25/30
7352/7352 [==============================] - 130s 18ms/step - loss: 0.1821
- acc: 0.9429 - val_loss: 0.2901 - val_acc: 0.8907
Epoch 26/30
7352/7352 [==============================] - 132s 18ms/step - loss: 0.1741
- acc: 0.9450 - val_loss: 0.3229 - val_acc: 0.8938
Epoch 27/30
7352/7352 [==============================] - 130s 18ms/step - loss: 0.1603
- acc: 0.9465 - val_loss: 0.3957 - val_acc: 0.8853
Epoch 28/30
7352/7352 [==============================] - 133s 18ms/step - loss: 0.1709
- acc: 0.9442 - val_loss: 0.3341 - val_acc: 0.8938
Epoch 29/30
7352/7352 [==============================] - 142s 19ms/step - loss: 0.1918
- acc: 0.9382 - val_loss: 0.3940 - val_acc: 0.8850
Epoch 30/30
7352/7352 [==============================] - 130s 18ms/step - loss: 0.1698
- acc: 0.9450 - val_loss: 0.3981 - val_acc: 0.8904
```

Out[19]:

```
<keras.callbacks.History at 0xebcabf3908>
```

In [22]:

```
score = model.evaluate(X_test, Y_test)
score
```

```
2947/2947 [==============================] - 5s 2ms/step
```

Out[22]:

```
[0.39807603972119654, 0.8903970139124533]
```

In [20]:

```
# Confusion Matrix
print(confusion_matrix(Y_test, model.predict(X_test)))
```

```
Pred                LAYING  SITTING  STANDING  WALKING  WALKING_DOWNSTAIRS
\
True
LAYING                 510        0        27        0                   0
SITTING                  2      377       108        0                   3
STANDING                 0       84       448        0                   0
WALKING                  0        0         0      450                  10
WALKING_DOWNSTAIRS       0        0         0        5                 406
WALKING_UPSTAIRS         0        0         0       25                  13

Pred                WALKING_UPSTAIRS
True
LAYING                             0
SITTING                            1
STANDING                           0
WALKING                           36
WALKING_DOWNSTAIRS                 9
WALKING_UPSTAIRS                 433
```

Using LSTM units and adam optimizer:

In [18]:

```
# Initialization
model = Sequential()
# configuring with parameters
model.add(LSTM(units=50, input_shape=(timesteps, input_dim)))
#Adding a dropout layer
model.add(Dropout(0.6))
#Adding dense layer
model.add(Dense(n_classes, activation = 'sigmoid'))
model.summary()
# Compiling the model
model.compile(loss='categorical_crossentropy',optimizer='adam', metrics=['accuracy'])
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 50)                12000
_________________________________________________________________
dropout_1 (Dropout)          (None, 50)                0
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 306
=================================================================
Total params: 12,306
Trainable params: 12,306
Non-trainable params: 0
_________________________________________________________________
```

In [20]:

```
history = model.fit(X_train, Y_train, batch_size=batch_size, validation_data=(X_test, Y
_test), epochs=epochs)
score = model.evaluate(X_test, Y_test)
print(score)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.2630 -
acc: 0.4274 - val_loss: 1.4146 - val_acc: 0.2582
Epoch 2/20
7352/7352 [==============================] - 28s 4ms/step - loss: 1.2912 -
acc: 0.3980 - val_loss: 1.3180 - val_acc: 0.4628
Epoch 3/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.2529 -
acc: 0.4282 - val_loss: 1.3501 - val_acc: 0.4201
Epoch 4/20
7352/7352 [==============================] - 26s 4ms/step - loss: 1.3423 -
acc: 0.4321 - val_loss: 1.2126 - val_acc: 0.5497
Epoch 5/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.2102 -
acc: 0.4778 - val_loss: 1.1773 - val_acc: 0.4805
Epoch 6/20
7352/7352 [==============================] - 29s 4ms/step - loss: 1.1754 -
acc: 0.4799 - val_loss: 1.3501 - val_acc: 0.3627
Epoch 7/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.2842 -
acc: 0.4361 - val_loss: 1.2743 - val_acc: 0.5076
Epoch 8/20
7352/7352 [==============================] - 26s 4ms/step - loss: 1.2050 -
acc: 0.5046 - val_loss: 1.1863 - val_acc: 0.5127
Epoch 9/20
7352/7352 [==============================] - 26s 4ms/step - loss: 1.0916 -
acc: 0.5313 - val_loss: 1.0969 - val_acc: 0.5090
Epoch 10/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.0108 -
acc: 0.5507 - val_loss: 1.0445 - val_acc: 0.5209
Epoch 11/20
7352/7352 [==============================] - 28s 4ms/step - loss: 0.9969 -
acc: 0.5631 - val_loss: 1.0500 - val_acc: 0.5100
Epoch 12/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.2033 -
acc: 0.5045 - val_loss: 1.1557 - val_acc: 0.4578
Epoch 13/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.1006 -
acc: 0.5107 - val_loss: 1.1439 - val_acc: 0.4798
Epoch 14/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.1357 -
acc: 0.5165 - val_loss: 1.0948 - val_acc: 0.4995
Epoch 15/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.3498 -
acc: 0.4109 - val_loss: 1.2969 - val_acc: 0.4503
Epoch 16/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.2417 -
acc: 0.4706 - val_loss: 1.2664 - val_acc: 0.5066
Epoch 17/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.2007 -
acc: 0.4967 - val_loss: 1.1903 - val_acc: 0.5711
Epoch 18/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.1122 -
acc: 0.5331 - val_loss: 1.0843 - val_acc: 0.5894
Epoch 19/20
7352/7352 [==============================] - 27s 4ms/step - loss: 1.0250 -
acc: 0.5543 - val_loss: 1.0557 - val_acc: 0.5782
Epoch 20/20
7352/7352 [==============================] - 27s 4ms/step - loss: 0.9277 -
acc: 0.5687 - val_loss: 0.9259 - val_acc: 0.5999
```

```
2947/2947 [==============================] - 2s 657us/step
[0.9259312011688331, 0.5999321343941652]
```

In [25]:

```
# Initialization
model = Sequential()
# configuring with parameters
model.add(LSTM(units=32, input_shape=(timesteps, input_dim)))
#Adding a dropout layer
model.add(Dropout(0.4))
#Adding dense layer
model.add(Dense(n_classes, activation = 'sigmoid'))
model.summary()
# Compiling the model
model.compile(loss='categorical_crossentropy',optimizer='adam', metrics=['accuracy'])
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_4 (LSTM)                (None, 32)                5376
_________________________________________________________________
dropout_4 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_4 (Dense)              (None, 6)                 198
=================================================================
Total params: 5,574
Trainable params: 5,574
Non-trainable params: 0
_________________________________________________________________
```

In [23]:

```
history = model.fit(X_train, Y_train, batch_size=batch_size, validation_data=(X_test, Y
_test), epochs=epochs)
score = model.evaluate(X_test, Y_test)
print(score)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
7352/7352 [==============================] - 24s 3ms/step - loss: 1.4281 -
acc: 0.4030 - val_loss: 1.2541 - val_acc: 0.4581
Epoch 2/20
7352/7352 [==============================] - 23s 3ms/step - loss: 1.2154 -
acc: 0.4491 - val_loss: 1.2486 - val_acc: 0.4818
Epoch 3/20
7352/7352 [==============================] - 23s 3ms/step - loss: 1.2854 -
acc: 0.4162 - val_loss: 1.3135 - val_acc: 0.3950
Epoch 4/20
7352/7352 [==============================] - 24s 3ms/step - loss: 1.1853 -
acc: 0.4716 - val_loss: 1.4516 - val_acc: 0.3437
Epoch 5/20
7352/7352 [==============================] - 22s 3ms/step - loss: 1.2143 -
acc: 0.4548 - val_loss: 1.4385 - val_acc: 0.3933
Epoch 6/20
7352/7352 [==============================] - 24s 3ms/step - loss: 1.2688 -
acc: 0.4211 - val_loss: 1.1763 - val_acc: 0.5070
Epoch 7/20
7352/7352 [==============================] - 24s 3ms/step - loss: 1.0020 -
acc: 0.5256 - val_loss: 1.0492 - val_acc: 0.5059
Epoch 8/20
7352/7352 [==============================] - 24s 3ms/step - loss: 1.0546 -
acc: 0.5049 - val_loss: 1.5267 - val_acc: 0.3478
Epoch 9/20
7352/7352 [==============================] - 23s 3ms/step - loss: 1.3753 -
acc: 0.3478 - val_loss: 1.3442 - val_acc: 0.3444
Epoch 10/20
7352/7352 [==============================] - 22s 3ms/step - loss: 1.2389 -
acc: 0.4430 - val_loss: 1.1910 - val_acc: 0.5025
Epoch 11/20
7352/7352 [==============================] - 22s 3ms/step - loss: 0.9967 -
acc: 0.5405 - val_loss: 0.9318 - val_acc: 0.5124
Epoch 12/20
7352/7352 [==============================] - 23s 3ms/step - loss: 0.8580 -
acc: 0.5966 - val_loss: 0.8257 - val_acc: 0.5745
Epoch 13/20
7352/7352 [==============================] - 22s 3ms/step - loss: 0.8596 -
acc: 0.5790 - val_loss: 1.2162 - val_acc: 0.4608
Epoch 14/20
7352/7352 [==============================] - 22s 3ms/step - loss: 0.8357 -
acc: 0.5949 - val_loss: 0.7822 - val_acc: 0.6176
Epoch 15/20
7352/7352 [==============================] - 24s 3ms/step - loss: 0.7465 -
acc: 0.6288 - val_loss: 0.7828 - val_acc: 0.6020
Epoch 16/20
7352/7352 [==============================] - 23s 3ms/step - loss: 0.7160 -
acc: 0.6469 - val_loss: 0.7577 - val_acc: 0.6094
Epoch 17/20
7352/7352 [==============================] - 22s 3ms/step - loss: 0.7065 -
acc: 0.6613 - val_loss: 0.7633 - val_acc: 0.6074
Epoch 18/20
7352/7352 [==============================] - 22s 3ms/step - loss: 0.7551 -
acc: 0.6266 - val_loss: 0.7759 - val_acc: 0.6281
Epoch 19/20
7352/7352 [==============================] - 23s 3ms/step - loss: 0.7021 -
acc: 0.6710 - val_loss: 0.7382 - val_acc: 0.6624
Epoch 20/20
7352/7352 [==============================] - 22s 3ms/step - loss: 0.6739 -
acc: 0.6771 - val_loss: 0.7109 - val_acc: 0.6542
```

```
2947/2947 [==============================] - 2s 514us/step
[0.7109381460844155, 0.6542246352323727]
```

In [26]:

```
# Compiling the model
model.compile(loss='categorical_crossentropy',optimizer='rmsprop', metrics=['accuracy'
])
```

In [28]:

```
history = model.fit(X_train, Y_train, batch_size=batch_size, validation_data=(X_test, Y
_test), epochs=30)
score = model.evaluate(X_test, Y_test)
print(score)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2323 -
acc: 0.9214 - val_loss: 0.5842 - val_acc: 0.8605
Epoch 2/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2577 -
acc: 0.9227 - val_loss: 0.5204 - val_acc: 0.8690
Epoch 3/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2235 -
acc: 0.9320 - val_loss: 0.6263 - val_acc: 0.8626
Epoch 4/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1926 -
acc: 0.9361 - val_loss: 0.4435 - val_acc: 0.8812
Epoch 5/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1970 -
acc: 0.9339 - val_loss: 0.5430 - val_acc: 0.8700
Epoch 6/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2107 -
acc: 0.9336 - val_loss: 0.6455 - val_acc: 0.8622
Epoch 7/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2114 -
acc: 0.9389 - val_loss: 0.5947 - val_acc: 0.8772
Epoch 8/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1808 -
acc: 0.9406 - val_loss: 0.5029 - val_acc: 0.8772
Epoch 9/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2043 -
acc: 0.9306 - val_loss: 0.6877 - val_acc: 0.8619
Epoch 10/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1691 -
acc: 0.9426 - val_loss: 0.4956 - val_acc: 0.8819
Epoch 11/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1725 -
acc: 0.9406 - val_loss: 0.5153 - val_acc: 0.8812
Epoch 12/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1740 -
acc: 0.9426 - val_loss: 0.4900 - val_acc: 0.8755
Epoch 13/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1719 -
acc: 0.9430 - val_loss: 0.4591 - val_acc: 0.8687
Epoch 14/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1781 -
acc: 0.9385 - val_loss: 0.4842 - val_acc: 0.8843
Epoch 15/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1634 -
acc: 0.9460 - val_loss: 0.5345 - val_acc: 0.8738
Epoch 16/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1820 -
acc: 0.9400 - val_loss: 0.4031 - val_acc: 0.8863
Epoch 17/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1491 -
acc: 0.9475 - val_loss: 0.4113 - val_acc: 0.8935
Epoch 18/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1466 -
acc: 0.9484 - val_loss: 0.4702 - val_acc: 0.8911
Epoch 19/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1497 -
acc: 0.9482 - val_loss: 0.3895 - val_acc: 0.9009
Epoch 20/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1731 -
acc: 0.9433 - val_loss: 0.4073 - val_acc: 0.9016
```

```
Epoch 21/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1442 -
acc: 0.9483 - val_loss: 0.4207 - val_acc: 0.8918
Epoch 22/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1609 -
acc: 0.9486 - val_loss: 0.3716 - val_acc: 0.8985
Epoch 23/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1494 -
acc: 0.9502 - val_loss: 0.5822 - val_acc: 0.8809
Epoch 24/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1404 -
acc: 0.9486 - val_loss: 0.5269 - val_acc: 0.8843
Epoch 25/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1399 -
acc: 0.9489 - val_loss: 0.4493 - val_acc: 0.8911
Epoch 26/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1535 -
acc: 0.9483 - val_loss: 0.4404 - val_acc: 0.8850
Epoch 27/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1495 -
acc: 0.9480 - val_loss: 0.4424 - val_acc: 0.8962
Epoch 28/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1505 -
acc: 0.9448 - val_loss: 0.5367 - val_acc: 0.8843
Epoch 29/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1977 -
acc: 0.9479 - val_loss: 0.4301 - val_acc: 0.8955
Epoch 30/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2019 -
acc: 0.9415 - val_loss: 0.4889 - val_acc: 0.8911
2947/2947 [==============================] - 2s 520us/step
[0.48891396308967605, 0.8910756701730573]
```

Dropout = 60%

In [30]:

```
# Initialization
model = Sequential()
# configuring with parameters
model.add(LSTM(units=32, input_shape=(timesteps, input_dim)))
#Adding a dropout layer
model.add(Dropout(0.6))
#Adding dense layer
model.add(Dense(n_classes, activation = 'sigmoid'))
model.summary()
# Compiling the model
model.compile(loss='categorical_crossentropy',optimizer='rmsprop', metrics=['accuracy'
])
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_5 (LSTM)                (None, 32)                5376
_________________________________________________________________
dropout_5 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_5 (Dense)              (None, 6)                 198
=================================================================
Total params: 5,574
Trainable params: 5,574
Non-trainable params: 0
_________________________________________________________________
```

In [32]:

```
history = model.fit(X_train, Y_train, batch_size=batch_size, validation_data=(X_test, Y
_test), epochs=30)
score = model.evaluate(X_test, Y_test)
print(score)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2811 -
acc: 0.9245 - val_loss: 0.4400 - val_acc: 0.8829
Epoch 2/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2721 -
acc: 0.9244 - val_loss: 0.6199 - val_acc: 0.8483
Epoch 3/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2584 -
acc: 0.9260 - val_loss: 0.4323 - val_acc: 0.8758
Epoch 4/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.4427 -
acc: 0.8932 - val_loss: 0.4200 - val_acc: 0.8823
Epoch 5/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2506 -
acc: 0.9298 - val_loss: 0.4346 - val_acc: 0.8711
Epoch 6/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2438 -
acc: 0.9298 - val_loss: 0.4508 - val_acc: 0.8799
Epoch 7/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2356 -
acc: 0.9314 - val_loss: 0.4223 - val_acc: 0.8802
Epoch 8/30
7352/7352 [==============================] - 26s 4ms/step - loss: 0.2915 -
acc: 0.9253 - val_loss: 0.3917 - val_acc: 0.8873
Epoch 9/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.2431 -
acc: 0.9271 - val_loss: 0.5494 - val_acc: 0.8252
Epoch 10/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.2275 -
acc: 0.9339 - val_loss: 0.4168 - val_acc: 0.8843
Epoch 11/30
7352/7352 [==============================] - 26s 4ms/step - loss: 0.2252 -
acc: 0.9298 - val_loss: 0.3852 - val_acc: 0.8887
Epoch 12/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.2333 -
acc: 0.9351 - val_loss: 0.4956 - val_acc: 0.8870
Epoch 13/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.2080 -
acc: 0.9414 - val_loss: 0.4059 - val_acc: 0.8901
Epoch 14/30
7352/7352 [==============================] - 25s 3ms/step - loss: 0.2685 -
acc: 0.9305 - val_loss: 0.4207 - val_acc: 0.8884
Epoch 15/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.2119 -
acc: 0.9347 - val_loss: 0.5143 - val_acc: 0.8731
Epoch 16/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1863 -
acc: 0.9410 - val_loss: 0.3894 - val_acc: 0.8962
Epoch 17/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1908 -
acc: 0.9411 - val_loss: 0.3776 - val_acc: 0.8792
Epoch 18/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1898 -
acc: 0.9418 - val_loss: 0.3793 - val_acc: 0.8975
Epoch 19/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1964 -
acc: 0.9396 - val_loss: 0.5139 - val_acc: 0.8775
Epoch 20/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1911 -
acc: 0.9427 - val_loss: 0.4563 - val_acc: 0.8812
```

```
Epoch 21/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1934 -
acc: 0.9415 - val_loss: 0.3890 - val_acc: 0.8989
Epoch 22/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1850 -
acc: 0.9442 - val_loss: 0.4422 - val_acc: 0.8989
Epoch 23/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.2239 -
acc: 0.9385 - val_loss: 0.4873 - val_acc: 0.8938
Epoch 24/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.2082 -
acc: 0.9423 - val_loss: 0.3743 - val_acc: 0.9019
Epoch 25/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1725 -
acc: 0.9431 - val_loss: 0.6799 - val_acc: 0.8470
Epoch 26/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1925 -
acc: 0.9460 - val_loss: 0.3970 - val_acc: 0.8809
Epoch 27/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1831 -
acc: 0.9431 - val_loss: 0.5824 - val_acc: 0.8622
Epoch 28/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1657 -
acc: 0.9479 - val_loss: 0.4361 - val_acc: 0.8907
Epoch 29/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1937 -
acc: 0.9446 - val_loss: 0.3363 - val_acc: 0.9002
Epoch 30/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1810 -
acc: 0.9440 - val_loss: 0.3395 - val_acc: 0.8941
2947/2947 [==============================] - 2s 511us/step
[0.33953921478739224, 0.8941296233457754]
```

2 Layer LSTM:

In [62]:

```
# Initialization
model = Sequential()
model.add(LSTM(units=32, input_shape=(timesteps, input_dim), return_sequences=True))
model.add(Dropout(0.5))

model.add(LSTM(units= 32))
model.add(Dropout(0.5))

model.add(Dense(n_classes, activation = 'sigmoid'))
model.summary()
# Compiling the model
model.compile(loss='categorical_crossentropy',optimizer='rmsprop', metrics=['accuracy'
])
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_46 (LSTM)               (None, 128, 32)           5376
_________________________________________________________________
dropout_22 (Dropout)         (None, 128, 32)           0
_________________________________________________________________
lstm_47 (LSTM)               (None, 32)                8320
_________________________________________________________________
dropout_23 (Dropout)         (None, 32)                0
_________________________________________________________________
dense_10 (Dense)             (None, 6)                 198
=================================================================
Total params: 13,894
Trainable params: 13,894
Non-trainable params: 0
_________________________________________________________________
```

In [63]:

```
history = model.fit(X_train, Y_train, batch_size=batch_size, validation_data=(X_test, Y
_test), epochs=30)
score = model.evaluate(X_test, Y_test)
print(score)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 57s 8ms/step - loss: 1.3566 -
acc: 0.4430 - val_loss: 1.1989 - val_acc: 0.4503
Epoch 2/30
7352/7352 [==============================] - 51s 7ms/step - loss: 1.0793 -
acc: 0.5257 - val_loss: 1.2094 - val_acc: 0.5046
Epoch 3/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.9271 -
acc: 0.6281 - val_loss: 0.8360 - val_acc: 0.6973
Epoch 4/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.7318 -
acc: 0.7189 - val_loss: 0.6465 - val_acc: 0.7340
Epoch 5/30
7352/7352 [==============================] - 52s 7ms/step - loss: 0.6495 -
acc: 0.7421 - val_loss: 0.6039 - val_acc: 0.7652
Epoch 6/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.5609 -
acc: 0.7637 - val_loss: 0.6557 - val_acc: 0.7503
Epoch 7/30
7352/7352 [==============================] - 136s 19ms/step - loss: 0.5716
- acc: 0.7567 - val_loss: 0.5240 - val_acc: 0.7689
Epoch 8/30
7352/7352 [==============================] - 187s 25ms/step - loss: 0.4841
- acc: 0.7947 - val_loss: 0.4357 - val_acc: 0.8208
Epoch 9/30
7352/7352 [==============================] - 152s 21ms/step - loss: 0.4584
- acc: 0.8283 - val_loss: 0.4714 - val_acc: 0.8364
Epoch 10/30
7352/7352 [==============================] - 66s 9ms/step - loss: 0.4124 -
acc: 0.8755 - val_loss: 0.4248 - val_acc: 0.8619
Epoch 11/30
7352/7352 [==============================] - 56s 8ms/step - loss: 0.3406 -
acc: 0.9105 - val_loss: 0.5061 - val_acc: 0.8714
Epoch 12/30
7352/7352 [==============================] - 54s 7ms/step - loss: 0.2834 -
acc: 0.9260 - val_loss: 0.3291 - val_acc: 0.9023
Epoch 13/30
7352/7352 [==============================] - 55s 7ms/step - loss: 0.2911 -
acc: 0.9189 - val_loss: 0.3876 - val_acc: 0.8856
Epoch 14/30
7352/7352 [==============================] - 55s 7ms/step - loss: 0.2416 -
acc: 0.9319 - val_loss: 0.3373 - val_acc: 0.8999
Epoch 15/30
7352/7352 [==============================] - 56s 8ms/step - loss: 0.2189 -
acc: 0.9353 - val_loss: 0.4129 - val_acc: 0.8914
Epoch 16/30
7352/7352 [==============================] - 56s 8ms/step - loss: 0.1992 -
acc: 0.9410 - val_loss: 0.3357 - val_acc: 0.9040
Epoch 17/30
7352/7352 [==============================] - 56s 8ms/step - loss: 0.2144 -
acc: 0.9408 - val_loss: 0.3307 - val_acc: 0.8951
Epoch 18/30
7352/7352 [==============================] - 56s 8ms/step - loss: 0.2022 -
acc: 0.9425 - val_loss: 0.3591 - val_acc: 0.9019
Epoch 19/30
7352/7352 [==============================] - 55s 8ms/step - loss: 0.1902 -
acc: 0.9444 - val_loss: 0.3207 - val_acc: 0.9087
Epoch 20/30
7352/7352 [==============================] - 56s 8ms/step - loss: 0.1776 -
acc: 0.9472 - val_loss: 0.3916 - val_acc: 0.9084
```

```
Epoch 21/30
7352/7352 [==============================] - 55s 7ms/step - loss: 0.1702 -
acc: 0.9459 - val_loss: 0.3491 - val_acc: 0.9067
Epoch 22/30
7352/7352 [==============================] - 52s 7ms/step - loss: 0.1627 -
acc: 0.9459 - val_loss: 0.3141 - val_acc: 0.9128
Epoch 23/30
7352/7352 [==============================] - 56s 8ms/step - loss: 0.1637 -
acc: 0.9490 - val_loss: 0.3061 - val_acc: 0.9060
Epoch 24/30
7352/7352 [==============================] - 56s 8ms/step - loss: 0.1573 -
acc: 0.9480 - val_loss: 0.3335 - val_acc: 0.9087
Epoch 25/30
7352/7352 [==============================] - 52s 7ms/step - loss: 0.1653 -
acc: 0.9456 - val_loss: 0.3352 - val_acc: 0.9101
Epoch 26/30
7352/7352 [==============================] - 54s 7ms/step - loss: 0.1668 -
acc: 0.9457 - val_loss: 0.4254 - val_acc: 0.9030
Epoch 27/30
7352/7352 [==============================] - 57s 8ms/step - loss: 0.1456 -
acc: 0.9536 - val_loss: 0.3417 - val_acc: 0.9077
Epoch 28/30
7352/7352 [==============================] - 52s 7ms/step - loss: 0.1411 -
acc: 0.9527 - val_loss: 0.3265 - val_acc: 0.9087
Epoch 29/30
7352/7352 [==============================] - 52s 7ms/step - loss: 0.1787 -
acc: 0.9463 - val_loss: 0.3202 - val_acc: 0.9131
Epoch 30/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.1393 -
acc: 0.9510 - val_loss: 0.3888 - val_acc: 0.9043
2947/2947 [==============================] - 3s 1ms/step
[0.3888279681393048, 0.9043094672548354]
```

2 Layer LSTM with 70% Dropout:

In [69]:

```
# Initialization
model = Sequential()
model.add(LSTM(units=32, input_shape=(timesteps, input_dim), return_sequences=True))
model.add(Dropout(0.7))

model.add(LSTM(units= 32))
model.add(Dropout(0.7))

model.add(Dense(n_classes, activation = 'sigmoid'))
model.summary()
# Compiling the model
model.compile(loss='categorical_crossentropy',optimizer='rmsprop', metrics=['accuracy'
])
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_50 (LSTM)               (None, 128, 32)           5376
_________________________________________________________________
dropout_26 (Dropout)         (None, 128, 32)           0
_________________________________________________________________
lstm_51 (LSTM)               (None, 32)                8320
_________________________________________________________________
dropout_27 (Dropout)         (None, 32)                0
_________________________________________________________________
dense_12 (Dense)             (None, 6)                 198
=================================================================
Total params: 13,894
Trainable params: 13,894
Non-trainable params: 0
_________________________________________________________________
```

In [71]:

```
history = model.fit(X_train, Y_train, batch_size=batch_size, validation_data=(X_test, Y
_test), epochs=40)
score = model.evaluate(X_test, Y_test)
print(score)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/40
7352/7352 [==============================] - 54s 7ms/step - loss: 1.2440 -
acc: 0.5027 - val_loss: 1.0186 - val_acc: 0.5382
Epoch 2/40
7352/7352 [==============================] - 65s 9ms/step - loss: 0.9628 -
acc: 0.5875 - val_loss: 0.8396 - val_acc: 0.6308
Epoch 3/40
7352/7352 [==============================] - 64s 9ms/step - loss: 0.8538 -
acc: 0.6249 - val_loss: 0.7622 - val_acc: 0.6481
Epoch 4/40
7352/7352 [==============================] - 65s 9ms/step - loss: 0.7820 -
acc: 0.6579 - val_loss: 0.6884 - val_acc: 0.6498
Epoch 5/40
7352/7352 [==============================] - 66s 9ms/step - loss: 0.7334 -
acc: 0.6766 - val_loss: 0.7790 - val_acc: 0.6630
Epoch 6/40
7352/7352 [==============================] - 72s 10ms/step - loss: 0.6670
- acc: 0.7101 - val_loss: 0.6182 - val_acc: 0.7296
Epoch 7/40
7352/7352 [==============================] - 67s 9ms/step - loss: 0.6133 -
acc: 0.7397 - val_loss: 0.6160 - val_acc: 0.7180
Epoch 8/40
7352/7352 [==============================] - 67s 9ms/step - loss: 0.5662 -
acc: 0.7662 - val_loss: 0.5968 - val_acc: 0.7224
Epoch 9/40
7352/7352 [==============================] - 59s 8ms/step - loss: 0.5093 -
acc: 0.7888 - val_loss: 0.6639 - val_acc: 0.7092
Epoch 10/40
7352/7352 [==============================] - 63s 9ms/step - loss: 0.4953 -
acc: 0.7964 - val_loss: 0.4795 - val_acc: 0.7536
Epoch 11/40
7352/7352 [==============================] - 55s 7ms/step - loss: 0.4590 -
acc: 0.8283 - val_loss: 0.5342 - val_acc: 0.8426
Epoch 12/40
7352/7352 [==============================] - 62s 8ms/step - loss: 0.4296 -
acc: 0.8489 - val_loss: 0.7624 - val_acc: 0.8212
Epoch 13/40
7352/7352 [==============================] - 66s 9ms/step - loss: 0.4182 -
acc: 0.8658 - val_loss: 0.5040 - val_acc: 0.8809
Epoch 14/40
7352/7352 [==============================] - 61s 8ms/step - loss: 0.3962 -
acc: 0.8754 - val_loss: 0.5703 - val_acc: 0.8507
Epoch 15/40
7352/7352 [==============================] - 58s 8ms/step - loss: 0.3489 -
acc: 0.8961 - val_loss: 0.5582 - val_acc: 0.8697
Epoch 16/40
7352/7352 [==============================] - 64s 9ms/step - loss: 0.3322 -
acc: 0.9095 - val_loss: 0.4556 - val_acc: 0.8863
Epoch 17/40
7352/7352 [==============================] - 56s 8ms/step - loss: 0.2963 -
acc: 0.9158 - val_loss: 0.4575 - val_acc: 0.8972
Epoch 18/40
7352/7352 [==============================] - 61s 8ms/step - loss: 0.2793 -
acc: 0.9214 - val_loss: 0.4377 - val_acc: 0.8928
Epoch 19/40
7352/7352 [==============================] - 62s 8ms/step - loss: 0.2545 -
acc: 0.9255 - val_loss: 0.5380 - val_acc: 0.8870
Epoch 20/40
7352/7352 [==============================] - 66s 9ms/step - loss: 0.2603 -
acc: 0.9251 - val_loss: 0.4901 - val_acc: 0.8948
```

```
Epoch 21/40
7352/7352 [==============================] - 69s 9ms/step - loss: 0.2383 -
acc: 0.9306 - val_loss: 0.3836 - val_acc: 0.9104
Epoch 22/40
7352/7352 [==============================] - 64s 9ms/step - loss: 0.2272 -
acc: 0.9314 - val_loss: 0.4411 - val_acc: 0.8975
Epoch 23/40
7352/7352 [==============================] - 62s 8ms/step - loss: 0.2084 -
acc: 0.9336 - val_loss: 0.3428 - val_acc: 0.9087
Epoch 24/40
7352/7352 [==============================] - 63s 9ms/step - loss: 0.2772 -
acc: 0.9294 - val_loss: 0.6017 - val_acc: 0.8856
Epoch 25/40
7352/7352 [==============================] - 65s 9ms/step - loss: 0.2116 -
acc: 0.9365 - val_loss: 0.4807 - val_acc: 0.8931
Epoch 26/40
7352/7352 [==============================] - 63s 9ms/step - loss: 0.2083 -
acc: 0.9402 - val_loss: 0.4079 - val_acc: 0.9063
Epoch 27/40
7352/7352 [==============================] - 66s 9ms/step - loss: 0.2143 -
acc: 0.9365 - val_loss: 0.3932 - val_acc: 0.9013
Epoch 28/40
7352/7352 [==============================] - 69s 9ms/step - loss: 0.1989 -
acc: 0.9362 - val_loss: 0.4138 - val_acc: 0.9040
Epoch 29/40
7352/7352 [==============================] - 70s 10ms/step - loss: 0.2029
- acc: 0.9357 - val_loss: 0.4971 - val_acc: 0.8992
Epoch 30/40
7352/7352 [==============================] - 74s 10ms/step - loss: 0.1980
- acc: 0.9365 - val_loss: 0.3845 - val_acc: 0.9016
Epoch 31/40
7352/7352 [==============================] - 63s 9ms/step - loss: 0.2150 -
acc: 0.9354 - val_loss: 0.3216 - val_acc: 0.9145
Epoch 32/40
7352/7352 [==============================] - 65s 9ms/step - loss: 0.1939 -
acc: 0.9389 - val_loss: 0.4297 - val_acc: 0.8996
Epoch 33/40
7352/7352 [==============================] - 68s 9ms/step - loss: 0.1944 -
acc: 0.9361 - val_loss: 0.3912 - val_acc: 0.9125
Epoch 34/40
7352/7352 [==============================] - 64s 9ms/step - loss: 0.1848 -
acc: 0.9418 - val_loss: 0.4627 - val_acc: 0.9019
Epoch 35/40
7352/7352 [==============================] - 72s 10ms/step - loss: 0.1801
- acc: 0.9416 - val_loss: 0.5910 - val_acc: 0.8945
Epoch 36/40
7352/7352 [==============================] - 62s 8ms/step - loss: 0.1981 -
acc: 0.9407 - val_loss: 0.4079 - val_acc: 0.9091
Epoch 37/40
7352/7352 [==============================] - 63s 9ms/step - loss: 0.1946 -
acc: 0.9365 - val_loss: 0.4113 - val_acc: 0.9155
Epoch 38/40
7352/7352 [==============================] - 60s 8ms/step - loss: 0.1764 -
acc: 0.9402 - val_loss: 0.4933 - val_acc: 0.9087
Epoch 39/40
7352/7352 [==============================] - 60s 8ms/step - loss: 0.1854 -
acc: 0.9385 - val_loss: 0.5461 - val_acc: 0.9057
Epoch 40/40
7352/7352 [==============================] - 55s 7ms/step - loss: 0.1949 -
acc: 0.9366 - val_loss: 0.5007 - val_acc: 0.9087
```

```
2947/2947 [==============================] - 4s 1ms/step
[0.5006568275242099, 0.9087207329487614]
```

2 layer LSTM with 64 units and 70 % Dropout:

In [72]:

```
# Initialization
model = Sequential()
model.add(LSTM(units=64, input_shape=(timesteps, input_dim), return_sequences=True))
model.add(Dropout(0.7))

model.add(LSTM(units= 32))
model.add(Dropout(0.7))

model.add(Dense(n_classes, activation = 'sigmoid'))
model.summary()
# Compiling the model
model.compile(loss='categorical_crossentropy',optimizer='rmsprop', metrics=['accuracy'
])
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_52 (LSTM)               (None, 128, 64)           18944
_________________________________________________________________
dropout_28 (Dropout)         (None, 128, 64)           0
_________________________________________________________________
lstm_53 (LSTM)               (None, 32)                12416
_________________________________________________________________
dropout_29 (Dropout)         (None, 32)                0
_________________________________________________________________
dense_13 (Dense)             (None, 6)                 198
=================================================================
Total params: 31,558
Trainable params: 31,558
Non-trainable params: 0
_________________________________________________________________
```

In [73]:

```
history = model.fit(X_train, Y_train, batch_size=batch_size, validation_data=(X_test, Y
_test), epochs=30)
score = model.evaluate(X_test, Y_test)
print(score)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 78s 11ms/step - loss: 1.3609
- acc: 0.4445 - val_loss: 1.1298 - val_acc: 0.4751
Epoch 2/30
7352/7352 [==============================] - 69s 9ms/step - loss: 1.0889 -
acc: 0.5419 - val_loss: 0.9166 - val_acc: 0.5599
Epoch 3/30
7352/7352 [==============================] - 77s 10ms/step - loss: 0.9179
- acc: 0.6038 - val_loss: 0.8269 - val_acc: 0.5925
Epoch 4/30
7352/7352 [==============================] - 66s 9ms/step - loss: 0.9177 -
acc: 0.5967 - val_loss: 1.1375 - val_acc: 0.5053
Epoch 5/30
7352/7352 [==============================] - 64s 9ms/step - loss: 0.7997 -
acc: 0.6332 - val_loss: 0.8154 - val_acc: 0.6125
Epoch 6/30
7352/7352 [==============================] - 64s 9ms/step - loss: 0.7841 -
acc: 0.6427 - val_loss: 0.7719 - val_acc: 0.6132
Epoch 7/30
7352/7352 [==============================] - 64s 9ms/step - loss: 0.7570 -
acc: 0.6423 - val_loss: 0.7720 - val_acc: 0.6033
Epoch 8/30
7352/7352 [==============================] - 67s 9ms/step - loss: 0.8051 -
acc: 0.6454 - val_loss: 0.7194 - val_acc: 0.6159
Epoch 9/30
7352/7352 [==============================] - 64s 9ms/step - loss: 0.6825 -
acc: 0.6794 - val_loss: 0.6660 - val_acc: 0.6172
Epoch 10/30
7352/7352 [==============================] - 64s 9ms/step - loss: 0.6444 -
acc: 0.6934 - val_loss: 0.6641 - val_acc: 0.7112
Epoch 11/30
7352/7352 [==============================] - 64s 9ms/step - loss: 0.6325 -
acc: 0.7108 - val_loss: 0.6511 - val_acc: 0.7475
Epoch 12/30
7352/7352 [==============================] - 64s 9ms/step - loss: 0.6067 -
acc: 0.7363 - val_loss: 0.7106 - val_acc: 0.7530
Epoch 13/30
7352/7352 [==============================] - 70s 10ms/step - loss: 0.6442
- acc: 0.7402 - val_loss: 0.6517 - val_acc: 0.7268
Epoch 14/30
7352/7352 [==============================] - 72s 10ms/step - loss: 0.5429
- acc: 0.7745 - val_loss: 0.6877 - val_acc: 0.7414
Epoch 15/30
7352/7352 [==============================] - 69s 9ms/step - loss: 0.4863 -
acc: 0.7836 - val_loss: 0.5458 - val_acc: 0.7635
Epoch 16/30
7352/7352 [==============================] - 72s 10ms/step - loss: 0.4788
- acc: 0.7899 - val_loss: 0.5454 - val_acc: 0.7625
Epoch 17/30
7352/7352 [==============================] - 74s 10ms/step - loss: 0.4548
- acc: 0.7947 - val_loss: 0.6002 - val_acc: 0.7601
Epoch 18/30
7352/7352 [==============================] - 75s 10ms/step - loss: 0.4715
- acc: 0.7979 - val_loss: 0.5588 - val_acc: 0.7716
Epoch 19/30
7352/7352 [==============================] - 72s 10ms/step - loss: 0.4382
- acc: 0.8016 - val_loss: 0.6514 - val_acc: 0.7699
Epoch 20/30
7352/7352 [==============================] - 75s 10ms/step - loss: 0.4247
- acc: 0.8234 - val_loss: 0.5494 - val_acc: 0.7954
```

```
Epoch 21/30
7352/7352 [==============================] - 74s 10ms/step - loss: 0.4077
- acc: 0.8377 - val_loss: 0.4687 - val_acc: 0.8897
Epoch 22/30
7352/7352 [==============================] - 69s 9ms/step - loss: 0.3558 -
acc: 0.8943 - val_loss: 0.3861 - val_acc: 0.8979
Epoch 23/30
7352/7352 [==============================] - 67s 9ms/step - loss: 0.3354 -
acc: 0.9071 - val_loss: 0.4166 - val_acc: 0.9030
Epoch 24/30
7352/7352 [==============================] - 67s 9ms/step - loss: 0.3153 -
acc: 0.9106 - val_loss: 0.4584 - val_acc: 0.8731
Epoch 25/30
7352/7352 [==============================] - 67s 9ms/step - loss: 0.2710 -
acc: 0.9230 - val_loss: 0.3220 - val_acc: 0.9080
Epoch 26/30
7352/7352 [==============================] - 67s 9ms/step - loss: 0.2505 -
acc: 0.9210 - val_loss: 0.2856 - val_acc: 0.9138
Epoch 27/30
7352/7352 [==============================] - 67s 9ms/step - loss: 0.2474 -
acc: 0.9286 - val_loss: 0.2932 - val_acc: 0.9097
Epoch 28/30
7352/7352 [==============================] - 66s 9ms/step - loss: 0.2320 -
acc: 0.9298 - val_loss: 0.3437 - val_acc: 0.9026
Epoch 29/30
7352/7352 [==============================] - 67s 9ms/step - loss: 0.2474 -
acc: 0.9302 - val_loss: 0.3177 - val_acc: 0.9135
Epoch 30/30
7352/7352 [==============================] - 66s 9ms/step - loss: 0.2365 -
acc: 0.9312 - val_loss: 0.3265 - val_acc: 0.9169
2947/2947 [==============================] - 5s 2ms/step
[0.3265414497932831, 0.9168646080760094]
```

Conclusion:

In [ ]:

The best accuracy that we could get using LSTM model **is** 91.6 %. We got this using 2 lay
ered LSTM **with**