Document Created date:19-01-2020

Human Activity Recognition

Understanding Data

This project is to build a model that predicts the human activities such as Walking, Walking_Upstairs, Walking_Downstairs, Sitting, Standing or Laying.

This dataset is collected from 30 persons(referred as subjects in this dataset), performing different activities with a smartphone to their waists. The data is recorded with the help of sensors (accelerometer and Gyroscope) in that smartphone. This experiment was video recorded to label the data manually.

How data was recorded

By using the sensors(Gyroscope and accelerometer) in a smartphone, they have captured '3-axial linear acceleration'(*tAcc-XYZ*) from accelerometer and '3-axial angular velocity' (*tGyro-XYZ*) from Gyroscope with several variations.

> prefix 't' in those metrics denotes time.

> suffix 'XYZ' represents 3-axial signals in X , Y, and Z directions.

Y_Labels(Encoded)

- In the dataset, Y_labels are represented as numbers from 1 to 6 as their identifiers.
  - WALKING as **1**

- WALKING_UPSTAIRS as **2**
- WALKING_DOWNSTAIRS as **3**
- SITTING as **4**
- STANDING as **5**
- LAYING as **6**

Train and test data were saperated

- The readings from ***70%*** of the volunteers were taken as ***trianing data*** and remaining ***30%*** subjects recordings were taken for ***test data***

Data

- All the data is present in 'UCI_HAR_dataset/' folder in present working directory.
  - Feature names are present in 'UCI_HAR_dataset/features.txt'
  - ***Train Data***
    - 'UCI_HAR_dataset/train/X_train.txt'
    - 'UCI_HAR_dataset/train/subject_train.txt'
    - 'UCI_HAR_dataset/train/y_train.txt'
  - ***Test Data***
    - 'UCI_HAR_dataset/test/X_test.txt'
    - 'UCI_HAR_dataset/test/subject_test.txt'
    - 'UCI_HAR_dataset/test/y_test.txt'

Data Size :

```
27 MB
```

- Accelerometer and Gyroscope readings are taken from 30 volunteers(referred as subjects) while performing the following 6 Activities.

  1. Walking
  2. WalkingUpstairs
  3. WalkingDownstairs
  4. Standing
  5. Sitting
  6. Lying.

- Readings are divided into a window of 2.56 seconds with 50% overlapping.
- Accelerometer readings are divided into gravity acceleration and body acceleration readings, which has x,y and z components each.
- Gyroscope readings are the measure of angular velocities which has x,y and z components.
- Jerk signals are calculated for BodyAcceleration readings.
- Fourier Transforms are made on the above time readings to obtain frequency readings.
- Now, on all the base signal readings., mean, max, mad, sma, arcoefficient, engerybands,entropy etc., are calculated for each window.
- We get a feature vector of 561 features and these features are given in the dataset.
- Each window of readings is a datapoint of 561 features.

Problem Framework

Problem Framework

- 30 subjects(volunteers) data is randomly split to 70%(21) test and 30%(7) train data.
- Each datapoint corresponds one of the 6 Activities.

Problem Statement

- Given a new datapoint we have to predict the Activity

Exercise : HAR_LSTM HYPERPARATEMER TUNING

In [20]:

```
# Mounting Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?cli
ent_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleuserconten
t.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&response_type=code&
scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%
3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.c
om%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2faut
h%2fpeopleapi.readonly (https://accounts.google.com/o/oauth2/auth?client_i
d=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com
&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&response_type=code&scope
=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2
f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2
fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2f
peopleapi.readonly)

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

In [0]:

```
# Unzip the Data
import zipfile
zipfile_from = zipfile.ZipFile('/content/drive/My Drive/HumanActivityRecognition.zip','
zipfile_from.extractall('/content/drive/My Drive/HumanActivityrecognition')
zipfile_from.close()
```

In [0]:

```
# Importing Libraries
import pandas as pd
import numpy as np
```

In [0]:

```
# 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 directory

In [0]:

```
# Data directory
DATADIR = 'UCI_HAR_Dataset'
```

In [0]:

```
# 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 [0]:

```
# 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'/content/drive/My Drive/HumanActivityrecognition/HAR/UCI_HAR_Datas
        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 [0]:

```

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'/content/drive/My Drive/HumanActivityrecognition/HAR/UCI_HAR_Dataset/{
    y = _read_csv(filename)[0]

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

In [0]:

```
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 [0]:

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

In [0]:

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

In [0]:

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

In [0]:

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

In [0]:

```
# Initializing parameters
epochs = 30
batch_size = 16
n_hidden = 32
```

In [0]:

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

In [35]:

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

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:11: FutureWar
ning: Method .as_matrix will be removed in a future version. Use .values i
nstead.
  # This is added back by InteractiveShellApp.init_path()
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:12: FutureWar
ning: Method .as_matrix will be removed in a future version. Use .values i
nstead.
  if sys.path[0] == '':
```

In [36]:

```
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
```

Model :: Defining the Architecture of LSTM(32) + Dropout(0.5) + rmsprop + sigmoid

In [19]:

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

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:541: The name tf.placeholder is deprecated. Pleas
e use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:4432: The name tf.random_uniform is deprecated. P
lease use tf.random.uniform instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:148: The name tf.placeholder_with_default is depr
ecated. Please use tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:3733: calling dropout (from tensorflow.python.op
s.nn_ops) with keep_prob is deprecated and will be removed in a future ver
sion.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1
- keep_prob`.
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 32)                5376
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 198
=================================================================
Total params: 5,574
Trainable params: 5,574
Non-trainable params: 0
_________________________________________________________________
```

In [20]:

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

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optim
izers.py:793: The name tf.train.Optimizer is deprecated. Please use tf.com
pat.v1.train.Optimizer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:3576: The name tf.log is deprecated. Please use t
f.math.log instead.
```

In [21]:

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

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflo
w_core/python/ops/math_grad.py:1424: where (from tensorflow.python.ops.a
rray_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/bac
kend/tensorflow_backend.py:1033: The name tf.assign_add is deprecated. P
lease use tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/bac
kend/tensorflow_backend.py:1020: The name tf.assign is deprecated. Pleas
e use tf.compat.v1.assign instead.

Train on 7352 samples, validate on 2947 samples
Epoch 1/30
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/bac
kend/tensorflow_backend.py:190: The name tf.get_default_session is depre
cated. Please use tf.compat.v1.get_default_session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/bac
kend/tensorflow_backend.py:207: The name tf.global_variables is deprecat
ed. Please use tf.compat.v1.global_variables instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/bac
kend/tensorflow_backend.py:216: The name tf.is_variable_initialized is d
eprecated. Please use tf.compat.v1.is_variable_initialized instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/bac
kend/tensorflow_backend.py:223: The name tf.variables_initializer is dep
recated. Please use tf.compat.v1.variables_initializer instead.

7352/7352 [==============================] - 36s 5ms/step - loss: 1.3117
- acc: 0.4410 - val_loss: 1.1254 - val_acc: 0.4730
Epoch 2/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.9697
- acc: 0.5919 - val_loss: 0.8992 - val_acc: 0.5931
Epoch 3/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.7834
- acc: 0.6532 - val_loss: 0.7576 - val_acc: 0.6159
Epoch 4/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.6817
- acc: 0.6678 - val_loss: 0.6802 - val_acc: 0.6210
Epoch 5/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.6346
- acc: 0.6902 - val_loss: 0.6976 - val_acc: 0.6722
Epoch 6/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.6127
- acc: 0.7087 - val_loss: 0.7093 - val_acc: 0.7041
Epoch 7/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.5647
- acc: 0.7561 - val_loss: 0.6837 - val_acc: 0.7475
Epoch 8/30
```

```
7352/7352 [==============================] - 35s 5ms/step - loss: 0.5169
- acc: 0.7837 - val_loss: 0.6575 - val_acc: 0.7370
Epoch 9/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.4744
- acc: 0.7953 - val_loss: 0.6772 - val_acc: 0.7435
Epoch 10/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.4525
- acc: 0.8051 - val_loss: 0.6664 - val_acc: 0.7075
Epoch 11/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.4324
- acc: 0.8275 - val_loss: 0.6367 - val_acc: 0.7900
Epoch 12/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.3932
- acc: 0.8520 - val_loss: 0.5087 - val_acc: 0.8656
Epoch 13/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.3484
- acc: 0.8913 - val_loss: 0.4076 - val_acc: 0.8799
Epoch 14/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.3086
- acc: 0.9116 - val_loss: 0.4920 - val_acc: 0.8680
Epoch 15/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2776
- acc: 0.9252 - val_loss: 0.9069 - val_acc: 0.7998
Epoch 16/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2680
- acc: 0.9229 - val_loss: 0.4340 - val_acc: 0.8724
Epoch 17/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2403
- acc: 0.9300 - val_loss: 0.4533 - val_acc: 0.8894
Epoch 18/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2587
- acc: 0.9236 - val_loss: 0.4504 - val_acc: 0.8968
Epoch 19/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1937
- acc: 0.9418 - val_loss: 0.5122 - val_acc: 0.8795
Epoch 20/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1992
- acc: 0.9378 - val_loss: 0.4035 - val_acc: 0.8931
Epoch 21/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1899
- acc: 0.9402 - val_loss: 0.4522 - val_acc: 0.8958
Epoch 22/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2051
- acc: 0.9382 - val_loss: 0.5021 - val_acc: 0.8894
Epoch 23/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1791
- acc: 0.9404 - val_loss: 0.4781 - val_acc: 0.8951
Epoch 24/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1746
- acc: 0.9456 - val_loss: 0.4376 - val_acc: 0.8989
Epoch 25/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1961
- acc: 0.9384 - val_loss: 0.4530 - val_acc: 0.8880
Epoch 26/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1802
- acc: 0.9452 - val_loss: 0.4403 - val_acc: 0.8951
Epoch 27/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1604
- acc: 0.9452 - val_loss: 0.4152 - val_acc: 0.8989
Epoch 28/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1552
```

```
- acc: 0.9455 - val_loss: 0.4937 - val_acc: 0.8948
Epoch 29/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1627
- acc: 0.9472 - val_loss: 0.5597 - val_acc: 0.8904
Epoch 30/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1748
- acc: 0.9433 - val_loss: 0.5089 - val_acc: 0.8958
```

Out[21]:

```
<keras.callbacks.History at 0x7f182097bf28>
```

In [22]:

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

```
Pred                LAYING  SITTING  ...  WALKING_DOWNSTAIRS  WALKING_UP
STAIRS
True                                 ...
LAYING                 510        0  ...                   0
26
SITTING                  0      376  ...                   2
1
STANDING                 0       82  ...                   0
1
WALKING                  0        0  ...                  28
5
WALKING_DOWNSTAIRS       0        0  ...                 412
8
WALKING_UPSTAIRS         0        0  ...                  24
430

[6 rows x 6 columns]
```

In [23]:

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

```
2947/2947 [==============================] - 1s 367us/step
```

In [24]:

```
score
```

Out[24]:

```
[0.5088777291587507, 0.8958262639972854]
```

- With a simple 2 layer architecture we got 90.09% accuracy and a loss of 0.30
- We can further imporve the performace with Hyperparameter tuning

Model1:: Architecture of LSTM(32) + Dropout(0.5) + rmsprop + sigmoid

In [25]:

```
# Initiliazing the sequential model
model1 = Sequential()
# Configuring the parameters
model1.add(LSTM(n_hidden, input_shape=(timesteps, input_dim)))
# Adding a dropout layer
model1.add(Dropout(0.5))
# Adding a dense output layer with sigmoid activation
model1.add(Dense(n_classes, activation='sigmoid'))
model1.summary()

# Compiling the model
model1.compile(loss='categorical_crossentropy',optimizer='rmsprop',metrics=['accuracy']

# Training the model
history1 = model1.fit(X_train, Y_train, batch_size=batch_size,validation_data=(X_test, 
```

```
Model: "sequential_2"
_________________________________________________________________
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
_________________________________________________________________
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 37s 5ms/step - loss: 1.3362 -
acc: 0.4445 - val_loss: 1.2370 - val_acc: 0.5131
Epoch 2/30
7352/7352 [==============================] - 35s 5ms/step - loss: 1.0587 -
acc: 0.5453 - val_loss: 1.0204 - val_acc: 0.5080
Epoch 3/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.8865 -
acc: 0.5990 - val_loss: 1.0198 - val_acc: 0.5860
Epoch 4/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.8135 -
acc: 0.6125 - val_loss: 0.9102 - val_acc: 0.6033
Epoch 5/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.7604 -
acc: 0.6496 - val_loss: 0.8154 - val_acc: 0.6393
Epoch 6/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.7038 -
acc: 0.6790 - val_loss: 0.7573 - val_acc: 0.7170
Epoch 7/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.6713 -
acc: 0.7021 - val_loss: 0.8712 - val_acc: 0.6610
Epoch 8/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.6434 -
acc: 0.7469 - val_loss: 0.8214 - val_acc: 0.7340
Epoch 9/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.5610 -
acc: 0.7980 - val_loss: 0.7142 - val_acc: 0.7553
```

```
Epoch 10/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.4465 -
acc: 0.8572 - val_loss: 0.6353 - val_acc: 0.8144
Epoch 11/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.3947 -
acc: 0.8799 - val_loss: 0.6083 - val_acc: 0.8090
Epoch 12/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.3477 -
acc: 0.8965 - val_loss: 0.6552 - val_acc: 0.8232
Epoch 13/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2979 -
acc: 0.9093 - val_loss: 0.4861 - val_acc: 0.8666
Epoch 14/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.3046 -
acc: 0.9087 - val_loss: 0.5533 - val_acc: 0.8510
Epoch 15/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2699 -
acc: 0.9180 - val_loss: 0.4373 - val_acc: 0.8860
Epoch 16/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2486 -
acc: 0.9253 - val_loss: 0.4164 - val_acc: 0.8850
Epoch 17/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2335 -
acc: 0.9286 - val_loss: 0.7569 - val_acc: 0.8219
Epoch 18/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2339 -
acc: 0.9266 - val_loss: 0.4528 - val_acc: 0.8744
Epoch 19/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2290 -
acc: 0.9261 - val_loss: 0.3914 - val_acc: 0.8924
Epoch 20/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.2229 -
acc: 0.9304 - val_loss: 0.4858 - val_acc: 0.8823
Epoch 21/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2012 -
acc: 0.9329 - val_loss: 0.3812 - val_acc: 0.8914
Epoch 22/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1896 -
acc: 0.9373 - val_loss: 0.4334 - val_acc: 0.8911
Epoch 23/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1898 -
acc: 0.9377 - val_loss: 0.4008 - val_acc: 0.8887
Epoch 24/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1890 -
acc: 0.9391 - val_loss: 0.4503 - val_acc: 0.8724
Epoch 25/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1855 -
acc: 0.9396 - val_loss: 0.3993 - val_acc: 0.8918
Epoch 26/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1810 -
acc: 0.9421 - val_loss: 0.3557 - val_acc: 0.9002
Epoch 27/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.2109 -
acc: 0.9361 - val_loss: 0.5865 - val_acc: 0.8924
Epoch 28/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1622 -
acc: 0.9446 - val_loss: 0.3609 - val_acc: 0.9002
Epoch 29/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1613 -
acc: 0.9430 - val_loss: 0.4424 - val_acc: 0.8985
Epoch 30/30
```

```
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1648 -
acc: 0.9489 - val_loss: 0.3348 - val_acc: 0.9155
```

In [26]:

```
# Confusion Matrix
print(confusion_matrix(Y_test, model1.predict(X_test)))
```

```
Pred                LAYING  SITTING  ...  WALKING_DOWNSTAIRS  WALKING_UP
STAIRS
True                                 ...
LAYING                 537        0  ...                   0
0
SITTING                  0      380  ...                   0
3
STANDING                 0       59  ...                   0
0
WALKING                  0        0  ...                  14
14
WALKING_DOWNSTAIRS       0        0  ...                 413
3
WALKING_UPSTAIRS         0        0  ...                  16
434

[6 rows x 6 columns]
```

In [27]:

```
score1 = model1.evaluate(X_test, Y_test)
```

```
2947/2947 [==============================] - 1s 383us/step
```

In [28]:

```
score1
```

Out[28]:

```
[0.33480536445324116, 0.9155072955548015]
```

Model2:: Architecture of LSTM(42) + Dropout(0.5) + rmsprop + sigmoid

In [29]:

```
# Initiliazing the sequential model
model2 = Sequential()
# Configuring the parameters
model2.add(LSTM(42, input_shape=(timesteps, input_dim)))
# Adding a dropout layer
model2.add(Dropout(0.5))
# Adding a dense output layer with sigmoid activation
model2.add(Dense(n_classes, activation='sigmoid'))
print(model2.summary())

# Compiling the model
model2.compile(loss='categorical_crossentropy',optimizer='rmsprop',metrics=['accuracy']

# Training the model
history2 = model2.fit(X_train,Y_train,batch_size=batch_size,validation_data=(X_test, Y_
```

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_3 (LSTM)                (None, 42)                8736
_________________________________________________________________
dropout_3 (Dropout)          (None, 42)                0
_________________________________________________________________
dense_3 (Dense)              (None, 6)                 258
=================================================================
Total params: 8,994
Trainable params: 8,994
Non-trainable params: 0
_________________________________________________________________
None
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 39s 5ms/step - loss: 1.3137
- acc: 0.4222 - val_loss: 1.2687 - val_acc: 0.4136
Epoch 2/30
7352/7352 [==============================] - 37s 5ms/step - loss: 1.0355
- acc: 0.5498 - val_loss: 1.0039 - val_acc: 0.5416
Epoch 3/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.9231
- acc: 0.6034 - val_loss: 0.8401 - val_acc: 0.6345
Epoch 4/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.7992
- acc: 0.6453 - val_loss: 0.9690 - val_acc: 0.6495
Epoch 5/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.6993
- acc: 0.7014 - val_loss: 0.8239 - val_acc: 0.6922
Epoch 6/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.6378
- acc: 0.7333 - val_loss: 0.7043 - val_acc: 0.7509
Epoch 7/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.5463
- acc: 0.7994 - val_loss: 0.5985 - val_acc: 0.7974
Epoch 8/30
7352/7352 [==============================] - 38s 5ms/step - loss: 0.4979
- acc: 0.8319 - val_loss: 0.6876 - val_acc: 0.7822
Epoch 9/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.5142
- acc: 0.8229 - val_loss: 0.6982 - val_acc: 0.7828
```

```
Epoch 10/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.4176
- acc: 0.8653 - val_loss: 0.4709 - val_acc: 0.8354
Epoch 11/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.3551
- acc: 0.8925 - val_loss: 0.5235 - val_acc: 0.8354
Epoch 12/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.2898
- acc: 0.9154 - val_loss: 0.4995 - val_acc: 0.8588
Epoch 13/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.4432
- acc: 0.8730 - val_loss: 0.4778 - val_acc: 0.8303
Epoch 14/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.2852
- acc: 0.9117 - val_loss: 0.4824 - val_acc: 0.8717
Epoch 15/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.2360
- acc: 0.9279 - val_loss: 0.4126 - val_acc: 0.8778
Epoch 16/30
7352/7352 [==============================] - 38s 5ms/step - loss: 0.2132
- acc: 0.9359 - val_loss: 0.4838 - val_acc: 0.8649
Epoch 17/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.2156
- acc: 0.9347 - val_loss: 0.5602 - val_acc: 0.8605
Epoch 18/30
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1876
- acc: 0.9399 - val_loss: 0.4132 - val_acc: 0.8826
Epoch 19/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1852
- acc: 0.9377 - val_loss: 0.3586 - val_acc: 0.8877
Epoch 20/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.2028
- acc: 0.9389 - val_loss: 0.3264 - val_acc: 0.9016
Epoch 21/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1925
- acc: 0.9418 - val_loss: 0.4438 - val_acc: 0.8989
Epoch 22/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1725
- acc: 0.9437 - val_loss: 0.2788 - val_acc: 0.9087
Epoch 23/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1867
- acc: 0.9355 - val_loss: 0.2854 - val_acc: 0.9006
Epoch 24/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1755
- acc: 0.9391 - val_loss: 0.2862 - val_acc: 0.9135
Epoch 25/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1634
- acc: 0.9427 - val_loss: 0.4857 - val_acc: 0.8843
Epoch 26/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1557
- acc: 0.9446 - val_loss: 0.3935 - val_acc: 0.8945
Epoch 27/30
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1572
- acc: 0.9498 - val_loss: 0.4415 - val_acc: 0.8945
Epoch 28/30
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1680
- acc: 0.9464 - val_loss: 0.4496 - val_acc: 0.8996
Epoch 29/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1592
- acc: 0.9463 - val_loss: 0.4654 - val_acc: 0.8860
Epoch 30/30
```

```
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1833
- acc: 0.9437 - val_loss: 0.8907 - val_acc: 0.8493
```

In [30]:

```
score2 = model2.evaluate(X_test, Y_test)
```

```
2947/2947 [==============================] - 1s 432us/step
```

In [31]:

```
score2
```

Out[31]:

```
[0.8907021589124672, 0.8493383101459111]
```

Model3:: Architecture of LSTM(80) + Dropout(0.25) + adam + sigmoid

In [32]:

```
# With One LSTM Layer Model 1  #
n_hidden = 80

model3 = Sequential()

# 1 LSTM layer
model3.add(LSTM(n_hidden, input_shape = (timesteps, input_dim)))     # 1 LSTM

model3.add(Dropout(0.25))
model3.add(Dense(n_classes, activation = 'sigmoid'))
model3.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics = ['accuracy']
print(model3.summary())
```

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_4 (LSTM)                (None, 80)                28800
_________________________________________________________________
dropout_4 (Dropout)          (None, 80)                0
_________________________________________________________________
dense_4 (Dense)              (None, 6)                 486
=================================================================
Total params: 29,286
Trainable params: 29,286
Non-trainable params: 0
_________________________________________________________________
None
```

In [33]:

```
# Training the model
history3 = model3.fit(X_train,
          Y_train,
          batch_size=64,
          validation_data=(X_test, Y_test),
          epochs=epochs)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 20s 3ms/step - loss: 0.4040 -
acc: 0.8510 - val_loss: 0.3652 - val_acc: 0.8564
Epoch 2/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.3532 -
acc: 0.8625 - val_loss: 0.3597 - val_acc: 0.8598
Epoch 3/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.3647 -
acc: 0.8567 - val_loss: 0.3311 - val_acc: 0.8608
Epoch 4/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.3385 -
acc: 0.8587 - val_loss: 0.3191 - val_acc: 0.8702
Epoch 5/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.3059 -
acc: 0.8739 - val_loss: 0.2738 - val_acc: 0.8756
Epoch 6/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.2742 -
acc: 0.8867 - val_loss: 0.2484 - val_acc: 0.8894
Epoch 7/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.2912 -
acc: 0.8795 - val_loss: 0.2946 - val_acc: 0.8738
Epoch 8/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.2614 -
acc: 0.8816 - val_loss: 0.2477 - val_acc: 0.8879
Epoch 9/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.2149 -
acc: 0.8987 - val_loss: 0.2236 - val_acc: 0.8950
Epoch 10/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.1992 -
acc: 0.9033 - val_loss: 0.2206 - val_acc: 0.8931
Epoch 11/30
7352/7352 [==============================] - 17s 2ms/step - loss: 0.1886 -
acc: 0.9091 - val_loss: 0.2382 - val_acc: 0.8794
Epoch 12/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.2300 -
acc: 0.8969 - val_loss: 0.2463 - val_acc: 0.8854
Epoch 13/30
7352/7352 [==============================] - 17s 2ms/step - loss: 0.1996 -
acc: 0.9064 - val_loss: 0.2107 - val_acc: 0.9119
Epoch 14/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.1500 -
acc: 0.9381 - val_loss: 0.1855 - val_acc: 0.9243
Epoch 15/30
7352/7352 [==============================] - 17s 2ms/step - loss: 0.1398 -
acc: 0.9454 - val_loss: 0.1801 - val_acc: 0.9334
Epoch 16/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.1314 -
acc: 0.9518 - val_loss: 0.1750 - val_acc: 0.9311
Epoch 17/30
7352/7352 [==============================] - 17s 2ms/step - loss: 0.1084 -
acc: 0.9617 - val_loss: 0.1380 - val_acc: 0.9495
```

```
Epoch 18/30
7352/7352 [==============================] - 17s 2ms/step - loss: 0.0795 -
acc: 0.9739 - val_loss: 0.1206 - val_acc: 0.9585
Epoch 19/30
7352/7352 [==============================] - 17s 2ms/step - loss: 0.1009 -
acc: 0.9634 - val_loss: 0.1540 - val_acc: 0.9527
Epoch 20/30
7352/7352 [==============================] - 17s 2ms/step - loss: 0.2128 -
acc: 0.9246 - val_loss: 0.1639 - val_acc: 0.9453
Epoch 21/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.1458 -
acc: 0.9471 - val_loss: 0.1596 - val_acc: 0.9494
Epoch 22/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.1169 -
acc: 0.9602 - val_loss: 0.1271 - val_acc: 0.9566
Epoch 23/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.0853 -
acc: 0.9730 - val_loss: 0.1216 - val_acc: 0.9591
Epoch 24/30
7352/7352 [==============================] - 17s 2ms/step - loss: 0.0954 -
acc: 0.9663 - val_loss: 0.1185 - val_acc: 0.9603
Epoch 25/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.0889 -
acc: 0.9685 - val_loss: 0.1252 - val_acc: 0.9544
Epoch 26/30
7352/7352 [==============================] - 17s 2ms/step - loss: 0.0800 -
acc: 0.9722 - val_loss: 0.1241 - val_acc: 0.9607
Epoch 27/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.0635 -
acc: 0.9782 - val_loss: 0.1190 - val_acc: 0.9640
Epoch 28/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.0670 -
acc: 0.9772 - val_loss: 0.1224 - val_acc: 0.9632
Epoch 29/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.0575 -
acc: 0.9796 - val_loss: 0.1246 - val_acc: 0.9597
Epoch 30/30
7352/7352 [==============================] - 18s 2ms/step - loss: 0.0655 -
acc: 0.9752 - val_loss: 0.1164 - val_acc: 0.9629
```

In [34]:

```
score3 = model3.evaluate(X_test, Y_test)
print(score)
```

```
2947/2947 [==============================] - 2s 570us/step
[0.5088777291587507, 0.8958262639972854]
```

In [35]:

```
score3
```

Out[35]:

```
[0.11640074694648317, 0.9629001287350948]
```

In [36]:

```
# Confusion Matrix
print(confusion_matrix(Y_test, model3.predict(X_test)))
```

```
Pred                LAYING  SITTING  ...  WALKING_DOWNSTAIRS  WALKING_UPST
AIRS
True                                 ...
LAYING                 510        0  ...                   0
27
SITTING                  0      371  ...                   2
7
STANDING                 0       81  ...                   0
12
WALKING                  0        0  ...                   6
4
WALKING_DOWNSTAIRS       0        0  ...                 416
1
WALKING_UPSTAIRS         0        0  ...                   9
405

[6 rows x 6 columns]
```

Model4 :: Architecture of LSTM(80) + Dropout(0.25) + rmsprop + sigmoid

In [37]:

```
# With One LSTM Layer Model 1  #
n_hidden = 80

model4 = Sequential()

# 1 LSTM layer
model4.add(LSTM(n_hidden, input_shape = (timesteps, input_dim), return_sequences = True

model4.add(Dropout(0.25))
model4.add(LSTM(n_hidden))
model4.add(Dense(n_classes, activation = 'sigmoid'))

model4.compile(loss = 'binary_crossentropy', optimizer = 'rmsprop', metrics = ['accuracy
print(model4.summary())
```

```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_5 (LSTM)                (None, 128, 80)           28800
_________________________________________________________________
dropout_5 (Dropout)          (None, 128, 80)           0
_________________________________________________________________
lstm_6 (LSTM)                (None, 80)                51520
_________________________________________________________________
dense_5 (Dense)              (None, 6)                 486
=================================================================
Total params: 80,806
Trainable params: 80,806
Non-trainable params: 0
_________________________________________________________________
None
```

In [38]:

```
%%time
# Training the model
history4 = model4.fit(X_train,
          Y_train,
          batch_size= 64,
          validation_data=(X_test, Y_test),
          epochs=epochs)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 47s 6ms/step - loss: 0.3401 -
acc: 0.8628 - val_loss: 0.3149 - val_acc: 0.8722
Epoch 2/30
7352/7352 [==============================] - 44s 6ms/step - loss: 0.2328 -
acc: 0.8961 - val_loss: 0.2118 - val_acc: 0.9044
Epoch 3/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.1679 -
acc: 0.9306 - val_loss: 0.1756 - val_acc: 0.9288
Epoch 4/30
7352/7352 [==============================] - 44s 6ms/step - loss: 0.1274 -
acc: 0.9525 - val_loss: 0.1292 - val_acc: 0.9493
Epoch 5/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.1017 -
acc: 0.9632 - val_loss: 0.1161 - val_acc: 0.9540
Epoch 6/30
7352/7352 [==============================] - 44s 6ms/step - loss: 0.0801 -
acc: 0.9708 - val_loss: 0.1106 - val_acc: 0.9562
Epoch 7/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0653 -
acc: 0.9759 - val_loss: 0.1052 - val_acc: 0.9612
Epoch 8/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0647 -
acc: 0.9770 - val_loss: 0.1008 - val_acc: 0.9672
Epoch 9/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0494 -
acc: 0.9816 - val_loss: 0.1554 - val_acc: 0.9529
Epoch 10/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0516 -
acc: 0.9802 - val_loss: 0.1073 - val_acc: 0.9642
Epoch 11/30
7352/7352 [==============================] - 44s 6ms/step - loss: 0.0466 -
acc: 0.9823 - val_loss: 0.0898 - val_acc: 0.9687
Epoch 12/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0449 -
acc: 0.9820 - val_loss: 0.0831 - val_acc: 0.9756
Epoch 13/30
7352/7352 [==============================] - 44s 6ms/step - loss: 0.0444 -
acc: 0.9829 - val_loss: 0.0865 - val_acc: 0.9690
Epoch 14/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0419 -
acc: 0.9835 - val_loss: 0.0989 - val_acc: 0.9684
Epoch 15/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0467 -
acc: 0.9832 - val_loss: 0.0985 - val_acc: 0.9681
Epoch 16/30
7352/7352 [==============================] - 44s 6ms/step - loss: 0.0443 -
acc: 0.9844 - val_loss: 0.0923 - val_acc: 0.9714
Epoch 17/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0470 -
```

```
acc: 0.9827 - val_loss: 0.1032 - val_acc: 0.9675
Epoch 18/30
7352/7352 [==============================] - 44s 6ms/step - loss: 0.0423 -
acc: 0.9836 - val_loss: 0.0787 - val_acc: 0.9728
Epoch 19/30
7352/7352 [==============================] - 44s 6ms/step - loss: 0.0412 -
acc: 0.9831 - val_loss: 0.0804 - val_acc: 0.9699
Epoch 20/30
7352/7352 [==============================] - 44s 6ms/step - loss: 0.0411 -
acc: 0.9828 - val_loss: 0.0803 - val_acc: 0.9726
Epoch 21/30
7352/7352 [==============================] - 44s 6ms/step - loss: 0.0393 -
acc: 0.9832 - val_loss: 0.1046 - val_acc: 0.9693
Epoch 22/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0384 -
acc: 0.9846 - val_loss: 0.0934 - val_acc: 0.9704
Epoch 23/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0401 -
acc: 0.9851 - val_loss: 0.1119 - val_acc: 0.9693
Epoch 24/30
7352/7352 [==============================] - 44s 6ms/step - loss: 0.0360 -
acc: 0.9846 - val_loss: 0.0837 - val_acc: 0.9764
Epoch 25/30
7352/7352 [==============================] - 44s 6ms/step - loss: 0.0387 -
acc: 0.9840 - val_loss: 0.0899 - val_acc: 0.9737
Epoch 26/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0365 -
acc: 0.9845 - val_loss: 0.0940 - val_acc: 0.9665
Epoch 27/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0366 -
acc: 0.9849 - val_loss: 0.1040 - val_acc: 0.9718
Epoch 28/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0362 -
acc: 0.9853 - val_loss: 0.1182 - val_acc: 0.9647
Epoch 29/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0371 -
acc: 0.9849 - val_loss: 0.0832 - val_acc: 0.9703
Epoch 30/30
7352/7352 [==============================] - 43s 6ms/step - loss: 0.0334 -
acc: 0.9852 - val_loss: 0.0955 - val_acc: 0.9709
CPU times: user 21min 25s, sys: 19.7 s, total: 21min 45s
Wall time: 21min 49s
```

In [39]:

```
score4 = model4.evaluate(X_test, Y_test)
print(score4)
```

```
2947/2947 [==============================] - 4s 1ms/step
[0.09553245016018255, 0.9708743404444088]
```

In [40]:

```
# Confusion Matrix
print(confusion_matrix(Y_test, model4.predict(X_test)))
```

```
Pred                LAYING  SITTING  ...  WALKING_DOWNSTAIRS  WALKING_UPST
AIRS
True                                 ...
LAYING                 535        0  ...                   0
2
SITTING                  3      414  ...                   0
2
STANDING                 0       80  ...                   0
0
WALKING                  0        0  ...                  25
12
WALKING_DOWNSTAIRS       0        0  ...                 418
1
WALKING_UPSTAIRS         0        0  ...                   5
466

[6 rows x 6 columns]
```

Model5:: Architecture of LSTM(120) + Dropout(0.2) + l2 reg + rmsprop + sigmoid

In [38]:

```
# With One LSTM Layer Model 1  #
from keras import regularizers
n_hidden = 120

model5 = Sequential()

# 1 LSTM layer
model5.add(LSTM(n_hidden, input_shape = (timesteps, input_dim),kernel_regularizer=regul
model5.add(Dropout(0.2))
model5.add(LSTM(n_hidden))
model5.add(Dense(n_classes, activation = 'sigmoid'))

model5.compile(loss = 'binary_crossentropy', optimizer = 'rmsprop', metrics = ['accurac
print(model5.summary())
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:541: The name tf.placeholder is deprecated. Pleas
e use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:4432: The name tf.random_uniform is deprecated. P
lease use tf.random.uniform instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:148: The name tf.placeholder_with_default is depr
ecated. Please use tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:3733: calling dropout (from tensorflow.python.op
s.nn_ops) with keep_prob is deprecated and will be removed in a future ver
sion.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1
- keep_prob`.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optim
izers.py:793: The name tf.train.Optimizer is deprecated. Please use tf.com
pat.v1.train.Optimizer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:3657: The name tf.log is deprecated. Please use t
f.math.log instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_
core/python/ops/nn_impl.py:183: where (from tensorflow.python.ops.array_op
s) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 128, 120)          62400
_________________________________________________________________
dropout_1 (Dropout)          (None, 128, 120)          0
_________________________________________________________________
lstm_2 (LSTM)                (None, 120)               115680
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 726
```

```
==========================================================================
Total params: 178,806
Trainable params: 178,806
Non-trainable params: 0

_________________________________________________________________
None
```

In [39]:

```
# Training the model
history5 = model5.fit(X_train,
          Y_train,
          batch_size= 64,
          validation_data=(X_test, Y_test),
          epochs=epochs)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/bac
kend/tensorflow_backend.py:1033: The name tf.assign_add is deprecated. P
lease use tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/bac
kend/tensorflow_backend.py:1020: The name tf.assign is deprecated. Pleas
e use tf.compat.v1.assign instead.

Train on 7352 samples, validate on 2947 samples
Epoch 1/30
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/bac
kend/tensorflow_backend.py:190: The name tf.get_default_session is depre
cated. Please use tf.compat.v1.get_default_session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/bac
kend/tensorflow_backend.py:207: The name tf.global_variables is deprecat
ed. Please use tf.compat.v1.global_variables instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/bac
kend/tensorflow_backend.py:216: The name tf.is_variable_initialized is d
eprecated. Please use tf.compat.v1.is_variable_initialized instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/bac
kend/tensorflow_backend.py:223: The name tf.variables_initializer is dep
recated. Please use tf.compat.v1.variables_initializer instead.

7352/7352 [==============================] - 64s 9ms/step - loss: 0.3401
- acc: 0.8633 - val_loss: 0.3031 - val_acc: 0.8701
Epoch 2/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.2371
- acc: 0.8936 - val_loss: 0.2034 - val_acc: 0.9147
Epoch 3/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.1735
- acc: 0.9275 - val_loss: 0.1988 - val_acc: 0.9175
Epoch 4/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.1435
- acc: 0.9455 - val_loss: 0.1584 - val_acc: 0.9384
Epoch 5/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.1283
- acc: 0.9526 - val_loss: 0.1741 - val_acc: 0.9325
Epoch 6/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.1031
- acc: 0.9649 - val_loss: 0.1710 - val_acc: 0.9416
Epoch 7/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0879
- acc: 0.9700 - val_loss: 0.1748 - val_acc: 0.9423
Epoch 8/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0781
- acc: 0.9739 - val_loss: 0.1208 - val_acc: 0.9598
Epoch 9/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0643
- acc: 0.9782 - val_loss: 0.2129 - val_acc: 0.9399
```

```
Epoch 10/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0662
- acc: 0.9771 - val_loss: 0.2555 - val_acc: 0.9295
Epoch 11/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0601
- acc: 0.9795 - val_loss: 0.1097 - val_acc: 0.9628
Epoch 12/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0548
- acc: 0.9810 - val_loss: 0.1137 - val_acc: 0.9641
Epoch 13/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0543
- acc: 0.9807 - val_loss: 0.1084 - val_acc: 0.9643
Epoch 14/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0523
- acc: 0.9810 - val_loss: 0.1178 - val_acc: 0.9640
Epoch 15/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0566
- acc: 0.9806 - val_loss: 0.0952 - val_acc: 0.9683
Epoch 16/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0532
- acc: 0.9813 - val_loss: 0.1558 - val_acc: 0.9587
Epoch 17/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0535
- acc: 0.9807 - val_loss: 0.1690 - val_acc: 0.9567
Epoch 18/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0504
- acc: 0.9819 - val_loss: 0.1125 - val_acc: 0.9676
Epoch 19/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0510
- acc: 0.9821 - val_loss: 0.1061 - val_acc: 0.9696
Epoch 20/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0545
- acc: 0.9815 - val_loss: 0.1068 - val_acc: 0.9661
Epoch 21/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0485
- acc: 0.9832 - val_loss: 0.0999 - val_acc: 0.9738
Epoch 22/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0482
- acc: 0.9829 - val_loss: 0.1025 - val_acc: 0.9691
Epoch 23/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0515
- acc: 0.9825 - val_loss: 0.1023 - val_acc: 0.9683
Epoch 24/30
7352/7352 [==============================] - 63s 9ms/step - loss: 0.0480
- acc: 0.9835 - val_loss: 0.1004 - val_acc: 0.9680
Epoch 25/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0460
- acc: 0.9836 - val_loss: 0.0954 - val_acc: 0.9720
Epoch 26/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0446
- acc: 0.9836 - val_loss: 0.1036 - val_acc: 0.9695
Epoch 27/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0436
- acc: 0.9843 - val_loss: 0.1164 - val_acc: 0.9654
Epoch 28/30
7352/7352 [==============================] - 63s 9ms/step - loss: 0.0443
- acc: 0.9842 - val_loss: 0.0846 - val_acc: 0.9721
Epoch 29/30
7352/7352 [==============================] - 63s 9ms/step - loss: 0.0426
- acc: 0.9830 - val_loss: 0.0901 - val_acc: 0.9746
Epoch 30/30
```

```
7352/7352 [==============================] - 62s 8ms/step - loss: 0.0439
- acc: 0.9830 - val_loss: 0.0927 - val_acc: 0.9678
```

In [40]:

```
score5 = model5.evaluate(X_test, Y_test)
score5
```

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

Out[40]:

```
[0.09267068984828229, 0.9678203892133419]
```

In [41]:

```
# Confusion Matrix
print(confusion_matrix(Y_test, model5.predict(X_test)))
```

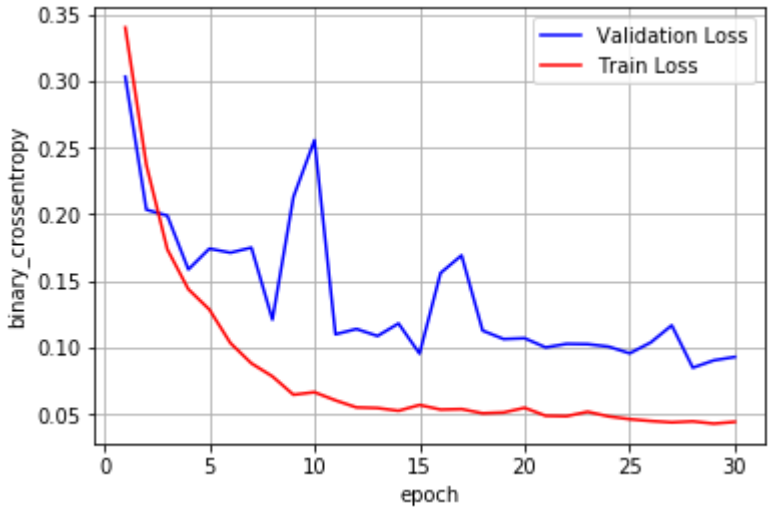
```
Pred                LAYING  SITTING  ...  WALKING_DOWNSTAIRS  WALKING_UP
STAIRS
True                                 ...
LAYING                 502       35  ...                   0
0
SITTING                  2      417  ...                   0
6
STANDING                 0      112  ...                   0
3
WALKING                  0        3  ...                  14
12
WALKING_DOWNSTAIRS       0        0  ...                 415
0
WALKING_UPSTAIRS         0        0  ...                   2
441

[6 rows x 6 columns]
```

In [0]:

```
# utility function
import matplotlib.pyplot as plt
import numpy as np
import time
# https://gist.github.com/greydanus/f6eee59eaf1d90fcb3b534a25362cea4
# https://stackoverflow.com/a/14434334
# this function is used to update the plots for each epoch and error
def plt_dynamic(x, vy, ty, ax, colors=['b']):
    ax.plot(x, vy, 'b', label="Validation Loss")
    ax.plot(x, ty, 'r', label="Train Loss")
    plt.legend()
    plt.grid()
    fig.canvas.draw()
```

In [50]:

```
fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('binary_crossentropy')

# list of epoch numbers
x = list(range(1,epochs+1))

# print(history.history.keys())
# dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
# history = model_drop.fit(X_train, Y_train, batch_size=batch_size, epochs=nb_epoch, ve

# we will get val_loss and val_acc only when you pass the paramter validation_data
# val_loss : validation loss
# val_acc : validation accuracy

# loss : training loss
# acc : train accuracy
# for each key in histrory.histrory we will have a list of length equal to number of ep

vy = history5.history['val_loss']
ty = history5.history['loss']
plt_dynamic(x, vy, ty, ax)
```



Model6 :: Architecture of LSTM(120) + Dropout(0.7) + l2 reg + rmsprop + sigmoid

In [52]:

```
# With One LSTM Layer Model 1  #
n_hidden = 120
# Initiliazing the sequential model
model6 = Sequential()

# Configuring the parameters
model6.add(LSTM(n_hidden, input_shape = (timesteps, input_dim),kernel_regularizer=regul
model6.add(Dropout(0.7)) # Adding a dropout layer
model6.add(LSTM(n_hidden)) # Configuring the parameters
model6.add(Dropout(0.7)) # Adding a dropout layer
model6.add(Dense(n_classes, activation = 'sigmoid')) # Adding a dense output layer with
# Compiling the model
model6.compile(loss = 'binary_crossentropy', optimizer = 'rmsprop', metrics = ['accurac
print(model6.summary())
# Training the model
history6 = model6.fit(X_train,Y_train,batch_size=batch_size,validation_data=(X_test, Y_
```

```
WARNING:tensorflow:Large dropout rate: 0.7 (>0.5). In TensorFlow 2.x, dr
opout() uses dropout rate instead of keep_prob. Please ensure that this
is intended.
WARNING:tensorflow:Large dropout rate: 0.7 (>0.5). In TensorFlow 2.x, dr
opout() uses dropout rate instead of keep_prob. Please ensure that this
is intended.
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_3 (LSTM)                (None, 128, 120)          62400
_________________________________________________________________
dropout_2 (Dropout)          (None, 128, 120)          0
_________________________________________________________________
lstm_4 (LSTM)                (None, 120)               115680
_________________________________________________________________
dropout_3 (Dropout)          (None, 120)               0
_________________________________________________________________
dense_2 (Dense)              (None, 6)                 726
```

In [53]:

```
score6 = model6.evaluate(X_test, Y_test)
score6
```

```
2947/2947 [==============================] - 6s 2ms/step
```

Out[53]:

```
[0.09851918673862693, 0.9741545128021396]
```

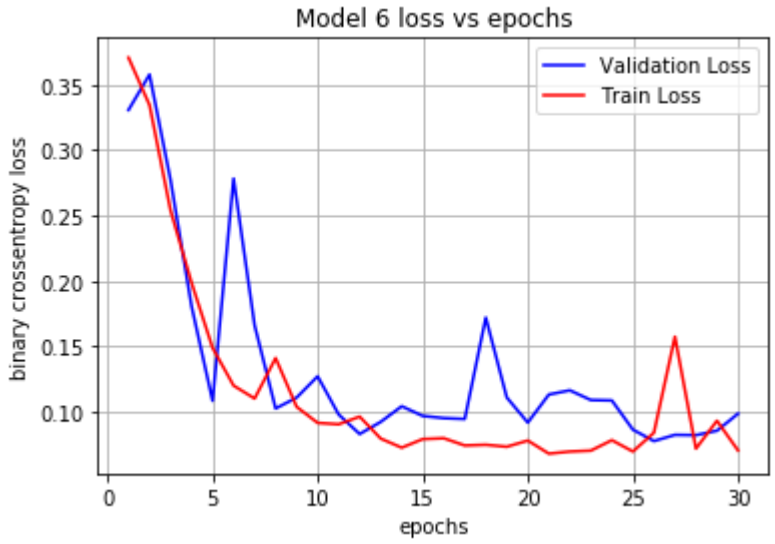
In [54]:

```
# Confusion MAtrix
print(confusion_matrix(Y_test, model6.predict(X_test)))
```

```
Pred                 LAYING  SITTING  ...  WALKING_DOWNSTAIRS  WALKING_UPST
AIRS
True                                  ...
LAYING                  537        0  ...                   0
0
SITTING                   1      376  ...                   0
0
STANDING                  0       74  ...                   0
0
WALKING                   0        0  ...                   8
16
WALKING_DOWNSTAIRS        0        0  ...                 417
3
WALKING_UPSTAIRS          0        0  ...                  16
455

[6 rows x 6 columns]
```

In [57]:

```
fig, ax = plt.subplots(1,1)
ax.set_xlabel('epochs');ax.set_ylabel('binary crossentropy loss');ax.set_title("Model 6
x = list(range(1,31))

vy = history6.history['val_loss']
ty = history6.history['loss']

plt_dynamic(x, vy, ty, ax)
```



Conclusion::

In [1]:

```
from prettytable import PrettyTable

p = PrettyTable()
p.field_names = ["Model","Hidden layer","epochs", "batch_size","activation","Optimizer"
p.add_row(['LSTM Model',  32, 30, 16, 'Sigmoid', 'rmsprop', 0.5, 0.5088, 0.8958])
p.add_row(['LSTM Model1', 32, 30, 16, 'Sigmoid', 'rmsprop', 0.5, 0.3348, 0.9155])
p.add_row(['LSTM Model2', 42, 30, 16, 'Sigmoid', 'rmsprop', 0.5, 0.8907, 0.8493])
p.add_row(['LSTM Model3', 80, 30, 64, 'Sigmoid', 'adam',   0.25, 0.1164, 0.9629])
p.add_row(['LSTM Model4', 80, 30, 64, 'Sigmoid', 'rmsprop', 0.25, 0.0955, 0.9708])
p.add_row(['LSTM Model5 with l2 Reg', 120, 30, 64, 'Sigmoid', 'rmsprop', 0.2, 0.0926, 0
p.add_row(['LSTM Model6 with l2 Reg', 120, 30, 16, 'Sigmoid', 'rmsprop', 0.7, 0.0985, 0
print(p)
```

```
+-------------------------+-------------+--------+-----------+----------
--+----------+---------+----------+-----------+
|          Model          | Hidden layer | epochs | batch_size | activatio
n | Optimizer | Dropout | Val_Loss | Val_acc % |
+-------------------------+-------------+--------+-----------+----------
--+----------+---------+----------+-----------+
|        LSTM Model       |      32      |   30   |     16     |  Sigmoid
|  rmsprop  |   0.5   |  0.5088  |   0.8958  |
|       LSTM Model1       |      32      |   30   |     16     |  Sigmoid
|  rmsprop  |   0.5   |  0.3348  |   0.9155  |
|       LSTM Model2       |      42      |   30   |     16     |  Sigmoid
|  rmsprop  |   0.5   |  0.8907  |   0.8493  |
|       LSTM Model3       |      80      |   30   |     64     |  Sigmoid
|    adam   |   0.25  |  0.1164  |   0.9629  |
|       LSTM Model4       |      80      |   30   |     64     |  Sigmoid
|  rmsprop  |   0.25  |  0.0955  |   0.9708  |
| LSTM Model5 with l2 Reg |     120      |   30   |     64     |  Sigmoid
|  rmsprop  |   0.2   |  0.0926  |   0.9678  |
| LSTM Model6 with l2 Reg |     120      |   30   |     16     |  Sigmoid
|  rmsprop  |   0.7   |  0.0985  |   0.9741  |
+-------------------------+-------------+--------+-----------+----------
--+----------+---------+----------+-----------+
```

Observations ::

- By increasing the hidden layers to 80 we got a very good result of 0.9629 % with "adam" optimzier .
- Lstm + dropout(0.25) model ,with 80 hidden layers with "rmsprop" optimizer we got a test acccuracy of 0.9708 %.
- Lstm + Large dropout(0.7) model ,with 120 hidden layers with "sigmoid" activation function we got a test acccuracy of 0.9741%.

- Cite : https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-i-hyper-parameter-8129009f131b (https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-i-hyper-parameter-8129009f131b)
- Cite : https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/ (https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/)

Thank You

Sign Off RAMESH BATTU (https://www.linkedin.com/in/rameshbattuai/)