In [0]:

```
# Importing Libraries
```

In [0]:

```
import pandas as pd
import numpy as np
np.random.seed(42)
```

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'])
```

In [3]:

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

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=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%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly

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

Data

In [0]:

```
# Data directory
DATADIR = '/content/drive/My Drive/HAR/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/HAR/UCI_HAR_Dataset/{subset}/Inertial Signals/{signal}_{su
bset}.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 [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/HAR/UCI_HAR_Dataset/{subset}/y_{subset}.txt'
    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 [9]:

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

```
The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via the %tensorflow_version 1.x magic: more info.
```

In [0]:

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

In [11]:

```
# 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 [0]:

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

In [0]:

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

In [15]:

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

In [16]:

```
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
```

In [0]:

```
data_disp = {}
```

- Defining the Architecture of LSTM

One LSTM + Dropout(50%)

In [96]:

In [96]:

```
# Initiliazing the sequential model
model = Sequential()
# Configuring the parameters
model.add(LSTM(40, 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='softmax'))
# Compiling the model
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
model.summary()
```

```
Model: "sequential_30"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_47 (LSTM)               (None, 40)                8000
_________________________________________________________________
dropout_46 (Dropout)         (None, 40)                0
_________________________________________________________________
dense_30 (Dense)             (None, 6)                 246
=================================================================
Total params: 8,246
Trainable params: 8,246
Non-trainable params: 0
_________________________________________________________________
```

Batch Size = 8

In [83]:

```
# Training the model
model.fit(X_train,
          Y_train,
          batch_size=8,
          validation_data=(X_test, Y_test),
          epochs=30)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 55s 8ms/step - loss: 1.2409 - acc: 0.4717 - val_loss: 1.32
14 - val_acc: 0.4411
Epoch 2/30
7352/7352 [==============================] - 49s 7ms/step - loss: 0.9766 - acc: 0.5857 - val_loss: 0.84
79 - val_acc: 0.6715
Epoch 3/30
7352/7352 [==============================] - 49s 7ms/step - loss: 0.7848 - acc: 0.7004 - val_loss: 1.07
93 - val_acc: 0.5697
Epoch 4/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.6543 - acc: 0.7364 - val_loss: 0.60
02 - val_acc: 0.7703
Epoch 5/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.4717 - acc: 0.8172 - val_loss: 0.65
88 - val_acc: 0.7475
Epoch 6/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.4069 - acc: 0.8513 - val_loss: 0.48
68 - val_acc: 0.8320
Epoch 7/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.3893 - acc: 0.8754 - val_loss: 1.15
11 - val_acc: 0.7014
Epoch 8/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.4311 - acc: 0.8615 - val_loss: 0.35
24 - val_acc: 0.8751
Epoch 9/30
7352/7352 [==============================] - 52s 7ms/step - loss: 0.2553 - acc: 0.9189 - val_loss: 0.48
64 - val_acc: 0.8592
Epoch 10/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.2466 - acc: 0.9208 - val_loss: 0.54
07 - val_acc: 0.8554
```

```
Epoch 11/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.2248 - acc: 0.9268 - val_loss: 0.40
33 - val_acc: 0.8792
Epoch 12/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.2375 - acc: 0.9202 - val_loss: 0.43
92 - val_acc: 0.8507
Epoch 13/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.2103 - acc: 0.9310 - val_loss: 0.46
23 - val_acc: 0.8853
Epoch 14/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.2379 - acc: 0.9257 - val_loss: 0.43
98 - val_acc: 0.8510
Epoch 15/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.2502 - acc: 0.9184 - val_loss: 0.38
49 - val_acc: 0.8819
Epoch 16/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.1940 - acc: 0.9319 - val_loss: 0.35
32 - val_acc: 0.8768
Epoch 17/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1691 - acc: 0.9373 - val_loss: 0.32
91 - val_acc: 0.8962
Epoch 18/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.2231 - acc: 0.9264 - val_loss: 0.45
58 - val_acc: 0.8785
Epoch 19/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1619 - acc: 0.9408 - val_loss: 0.24
97 - val_acc: 0.9074
Epoch 20/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.2270 - acc: 0.9237 - val_loss: 0.39
44 - val_acc: 0.8924
Epoch 21/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1897 - acc: 0.9354 - val_loss: 0.29
93 - val_acc: 0.8992
Epoch 22/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1541 - acc: 0.9448 - val_loss: 0.27
86 - val_acc: 0.8979
Epoch 23/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1670 - acc: 0.9395 - val_loss: 0.38
42 - val_acc: 0.8992
Epoch 24/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1435 - acc: 0.9461 - val_loss: 0.34
59 - val_acc: 0.9084
Epoch 25/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1541 - acc: 0.9421 - val_loss: 0.37
57 - val_acc: 0.9087
Epoch 26/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1300 - acc: 0.9480 - val_loss: 0.38
04 - val_acc: 0.9097
Epoch 27/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.4083 - acc: 0.9081 - val_loss: 1.55
21 - val_acc: 0.6834
Epoch 28/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.3512 - acc: 0.8927 - val_loss: 0.38
44 - val_acc: 0.8700
Epoch 29/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1795 - acc: 0.9370 - val_loss: 0.30
55 - val_acc: 0.9111
Epoch 30/30
7352/7352 [==============================] - 49s 7ms/step - loss: 0.1461 - acc: 0.9444 - val_loss: 0.27
27 - val_acc: 0.9175
```

Out[83]:

```
<keras.callbacks.History at 0x7efc6bdb1240>
```

In [84]:

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

Out[84]:

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS

Pred \ True	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	537	0	0	0	0	0
SITTING	5	359	124	1	0	2
STANDING	0	48	478	6	0	0
WALKING	0	0	1	467	16	12
WALKING_DOWNSTAIRS	0	0	6	4	406	4
WALKING_UPSTAIRS	0	1	5	7	1	457

In [85]:

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

```
2947/2947 [==============================] - 1s 316us/step
```

Out[85]:

```
[0.2727017437300063, 0.9175432643366135]
```

In [0]:

```
data_disp['1LSTM_8batch'] = score
```

Batch SIze = 16

In [87]:

```
# Training the model
model.fit(X_train,
          Y_train,
          batch_size=16,
          validation_data=(X_test, Y_test),
          epochs=30)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 34s 5ms/step - loss: 1.2441 - acc: 0.4782 - val_loss: 1.00
98 - val_acc: 0.5948
Epoch 2/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.9204 - acc: 0.6260 - val_loss: 0.86
82 - val_acc: 0.6206
Epoch 3/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.6669 - acc: 0.7368 - val_loss: 0.97
17 - val_acc: 0.5738
Epoch 4/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.6337 - acc: 0.7775 - val_loss: 0.62
09 - val_acc: 0.7940
Epoch 5/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.4617 - acc: 0.8497 - val_loss: 0.54
00 - val_acc: 0.8178
Epoch 6/30
7352/7352 [==============================] - 28s 4ms/step - loss: 0.3130 - acc: 0.9013 - val_loss: 0.44
20 - val_acc: 0.8599
Epoch 7/30
7352/7352 [==============================] - 28s 4ms/step - loss: 0.3142 - acc: 0.8955 - val_loss: 0.42
08 - val_acc: 0.8541
Epoch 8/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.2637 - acc: 0.9170 - val_loss: 0.35
91 - val_acc: 0.8728
Epoch 9/30
7352/7352 [==============================] - 28s 4ms/step - loss: 0.2454 - acc: 0.9219 - val_loss: 0.42
92 - val_acc: 0.8656
Epoch 10/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.2080 - acc: 0.9302 - val_loss: 0.57
65 - val_acc: 0.8504
Epoch 11/30
```

```
7352/7352 [==============================] - 27s 4ms/step - loss: 0.2986 - acc: 0.9025 - val_loss: 0.43
67 - val_acc: 0.8666
Epoch 12/30
7352/7352 [==============================] - 28s 4ms/step - loss: 0.2277 - acc: 0.9211 - val_loss: 0.33
87 - val_acc: 0.8856
Epoch 13/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.2001 - acc: 0.9325 - val_loss: 0.29
04 - val_acc: 0.8941
Epoch 14/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1893 - acc: 0.9319 - val_loss: 0.28
05 - val_acc: 0.8921
Epoch 15/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1708 - acc: 0.9399 - val_loss: 0.29
50 - val_acc: 0.8965
Epoch 16/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1625 - acc: 0.9438 - val_loss: 0.28
96 - val_acc: 0.9067
Epoch 17/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.2056 - acc: 0.9229 - val_loss: 0.25
78 - val_acc: 0.9019
Epoch 18/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1781 - acc: 0.9373 - val_loss: 0.29
34 - val_acc: 0.9030
Epoch 19/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1560 - acc: 0.9427 - val_loss: 0.28
41 - val_acc: 0.9084
Epoch 20/30
7352/7352 [==============================] - 26s 4ms/step - loss: 0.1751 - acc: 0.9317 - val_loss: 0.25
92 - val_acc: 0.9067
Epoch 21/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.3172 - acc: 0.8945 - val_loss: 0.28
15 - val_acc: 0.9030
Epoch 22/30
7352/7352 [==============================] - 26s 4ms/step - loss: 0.1688 - acc: 0.9366 - val_loss: 0.32
57 - val_acc: 0.9094
Epoch 23/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.2176 - acc: 0.9240 - val_loss: 0.27
12 - val_acc: 0.9046
Epoch 24/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1559 - acc: 0.9402 - val_loss: 0.26
78 - val_acc: 0.9141
Epoch 25/30
7352/7352 [==============================] - 26s 4ms/step - loss: 0.1897 - acc: 0.9305 - val_loss: 0.34
27 - val_acc: 0.8358
Epoch 26/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.2753 - acc: 0.9061 - val_loss: 0.31
65 - val_acc: 0.8999
Epoch 27/30
7352/7352 [==============================] - 26s 4ms/step - loss: 0.1720 - acc: 0.9372 - val_loss: 0.25
76 - val_acc: 0.9074
Epoch 28/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1519 - acc: 0.9472 - val_loss: 0.24
06 - val_acc: 0.9094
Epoch 29/30
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1921 - acc: 0.9305 - val_loss: 0.22
64 - val_acc: 0.9121
Epoch 30/30
7352/7352 [==============================] - 25s 3ms/step - loss: 0.1429 - acc: 0.9468 - val_loss: 0.25
32 - val_acc: 0.9070
```

Out[87]:

```
<keras.callbacks.History at 0x7efc6bd1ec18>
```

In [91]:

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

Out[91]:

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
True						

Pred / True	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	536	0	0	0	0	1
SITTING	3	419	68	1	0	0
STANDING	0	132	399	1	0	0
WALKING	0	0	0	468	8	20
WALKING_DOWNSTAIRS	0	0	0	0	398	22
WALKING_UPSTAIRS	0	0	0	18	0	453

In [92]:

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

```
2947/2947 [==============================] - 1s 301us/step
```

Out[92]:

```
[0.2531991467388801, 0.9070240922972514]
```

In [0]:

```
data_disp['1LSTM_16batch'] = score
```

Batch Size = 32

In [97]:

```
# Training the model
model.fit(X_train,
          Y_train,
          batch_size=32,
          validation_data=(X_test, Y_test),
          epochs=30)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 21s 3ms/step - loss: 1.3205 - acc: 0.4430 - val_loss: 1.21
47 - val_acc: 0.4978
Epoch 2/30
7352/7352 [==============================] - 15s 2ms/step - loss: 1.1238 - acc: 0.5329 - val_loss: 1.07
49 - val_acc: 0.5453
Epoch 3/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.8078 - acc: 0.6511 - val_loss: 0.77
40 - val_acc: 0.6692
Epoch 4/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.7331 - acc: 0.6863 - val_loss: 1.02
61 - val_acc: 0.6033
Epoch 5/30
7352/7352 [==============================] - 15s 2ms/step - loss: 1.0161 - acc: 0.6083 - val_loss: 0.90
81 - val_acc: 0.6994
Epoch 6/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.8597 - acc: 0.6250 - val_loss: 0.85
15 - val_acc: 0.6729
Epoch 7/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.6627 - acc: 0.7150 - val_loss: 0.64
39 - val_acc: 0.7421
Epoch 8/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.5664 - acc: 0.7609 - val_loss: 0.70
44 - val_acc: 0.7333
Epoch 9/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.5987 - acc: 0.7572 - val_loss: 0.64
65 - val_acc: 0.7445
Epoch 10/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.5247 - acc: 0.8029 - val_loss: 0.53
08 - val_acc: 0.8215
Epoch 11/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.3928 - acc: 0.8760 - val_loss: 0.50
```

```
11 - val_acc: 0.8402
Epoch 12/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.4280 - acc: 0.8630 - val_loss: 0.49
45 - val_acc: 0.8432
Epoch 13/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.3259 - acc: 0.8977 - val_loss: 0.58
87 - val_acc: 0.8269
Epoch 14/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.3082 - acc: 0.9041 - val_loss: 0.51
44 - val_acc: 0.8405
Epoch 15/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.3218 - acc: 0.8936 - val_loss: 0.49
50 - val_acc: 0.8571
Epoch 16/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.2495 - acc: 0.9251 - val_loss: 0.43
07 - val_acc: 0.8806
Epoch 17/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.5453 - acc: 0.8020 - val_loss: 0.48
77 - val_acc: 0.8208
Epoch 18/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.3110 - acc: 0.8954 - val_loss: 0.40
42 - val_acc: 0.8575
Epoch 19/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.2791 - acc: 0.9157 - val_loss: 0.31
72 - val_acc: 0.8806
Epoch 20/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.2350 - acc: 0.9223 - val_loss: 0.31
13 - val_acc: 0.8884
Epoch 21/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.2125 - acc: 0.9312 - val_loss: 0.61
61 - val_acc: 0.7974
Epoch 22/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.3654 - acc: 0.8709 - val_loss: 0.33
78 - val_acc: 0.8816
Epoch 23/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.2110 - acc: 0.9280 - val_loss: 0.34
53 - val_acc: 0.8945
Epoch 24/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.1825 - acc: 0.9353 - val_loss: 0.35
45 - val_acc: 0.8914
Epoch 25/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.1703 - acc: 0.9395 - val_loss: 0.27
07 - val_acc: 0.9030
Epoch 26/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.1614 - acc: 0.9407 - val_loss: 0.27
02 - val_acc: 0.9050
Epoch 27/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.1544 - acc: 0.9434 - val_loss: 0.26
27 - val_acc: 0.9141
Epoch 28/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.1592 - acc: 0.9430 - val_loss: 0.47
44 - val_acc: 0.8711
Epoch 29/30
7352/7352 [==============================] - 15s 2ms/step - loss: 0.1813 - acc: 0.9403 - val_loss: 0.35
23 - val_acc: 0.8945
Epoch 30/30
7352/7352 [==============================] - 14s 2ms/step - loss: 0.2280 - acc: 0.9162 - val_loss: 0.31
89 - val_acc: 0.8809
```

Out[97]:

```
<keras.callbacks.History at 0x7efc6b16ada0>
```

In [98]:

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

Out[98]:

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
True						
LAYING	518	0	0	16	0	3

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
SITTING	2	313	173	0	0	3
True STANDING	0	39	491	2	0	0
WALKING	0	2	18	441	9	26
WALKING_DOWNSTAIRS	0	0	0	1	379	40
WALKING_UPSTAIRS	0	0	0	14	3	454

In [99]:

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

```
2947/2947 [==============================] - 1s 319us/step
```

Out[99]:

```
[0.31892726644301617, 0.8808958262639973]
```

In [0]:

```
data_disp['1LSTM_32batch'] = score
```

One LSTM + Dropout(30%)

In [115]:

```
# Initiliazing the sequential model
model = Sequential()
# Configuring the parameters
model.add(LSTM(36, input_shape=(timesteps, input_dim)))
# Adding a dropout layer
model.add(Dropout(0.3))

# Adding a dense output layer with sigmoid activation
model.add(Dense(n_classes, activation='softmax'))
# Compiling the model
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
model.summary()
```

```
Model: "sequential_36"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_57 (LSTM)               (None, 36)                6624
_________________________________________________________________
dropout_56 (Dropout)         (None, 36)                0
_________________________________________________________________
dense_37 (Dense)             (None, 6)                 222
=================================================================
Total params: 6,846
Trainable params: 6,846
Non-trainable params: 0
_________________________________________________________________
```

Batch Size = 8

In [102]:

```
# Training the model
model.fit(X_train,
          Y_train,
          batch_size=8,
          validation_data=(X_test, Y_test),
          epochs=30)
```

```
          epochs=30)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 56s 8ms/step - loss: 1.0537 - acc: 0.5699 - val_loss: 0.80
30 - val_acc: 0.6885
Epoch 2/30
7352/7352 [==============================] - 49s 7ms/step - loss: 0.6486 - acc: 0.7432 - val_loss: 0.82
88 - val_acc: 0.6529
Epoch 3/30
7352/7352 [==============================] - 49s 7ms/step - loss: 0.5803 - acc: 0.7854 - val_loss: 0.61
77 - val_acc: 0.7662
Epoch 4/30
7352/7352 [==============================] - 47s 6ms/step - loss: 0.5569 - acc: 0.8096 - val_loss: 0.52
05 - val_acc: 0.8073
Epoch 5/30
7352/7352 [==============================] - 47s 6ms/step - loss: 0.3660 - acc: 0.8769 - val_loss: 0.41
47 - val_acc: 0.8419
Epoch 6/30
7352/7352 [==============================] - 48s 7ms/step - loss: 0.2758 - acc: 0.9108 - val_loss: 0.34
30 - val_acc: 0.8863
Epoch 7/30
7352/7352 [==============================] - 47s 6ms/step - loss: 0.2342 - acc: 0.9206 - val_loss: 0.33
04 - val_acc: 0.8843
Epoch 8/30
7352/7352 [==============================] - 47s 6ms/step - loss: 0.1958 - acc: 0.9306 - val_loss: 0.37
89 - val_acc: 0.8965
Epoch 9/30
7352/7352 [==============================] - 49s 7ms/step - loss: 0.2084 - acc: 0.9242 - val_loss: 0.29
48 - val_acc: 0.8938
Epoch 10/30
7352/7352 [==============================] - 49s 7ms/step - loss: 0.2510 - acc: 0.9089 - val_loss: 0.64
76 - val_acc: 0.7693
Epoch 11/30
7352/7352 [==============================] - 49s 7ms/step - loss: 0.2057 - acc: 0.9252 - val_loss: 0.28
28 - val_acc: 0.9009
Epoch 12/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.2548 - acc: 0.9158 - val_loss: 0.32
24 - val_acc: 0.8951
Epoch 13/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.1888 - acc: 0.9344 - val_loss: 0.26
92 - val_acc: 0.9026
Epoch 14/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1650 - acc: 0.9370 - val_loss: 0.27
90 - val_acc: 0.8982
Epoch 15/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1571 - acc: 0.9373 - val_loss: 0.25
93 - val_acc: 0.9128
Epoch 16/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.2064 - acc: 0.9241 - val_loss: 0.24
50 - val_acc: 0.9125
Epoch 17/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1707 - acc: 0.9388 - val_loss: 0.29
76 - val_acc: 0.8924
Epoch 18/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1573 - acc: 0.9407 - val_loss: 0.31
90 - val_acc: 0.8951
Epoch 19/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1459 - acc: 0.9421 - val_loss: 0.27
51 - val_acc: 0.9155
Epoch 20/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1399 - acc: 0.9441 - val_loss: 0.31
08 - val_acc: 0.9097
Epoch 21/30
7352/7352 [==============================] - 48s 7ms/step - loss: 0.1383 - acc: 0.9471 - val_loss: 0.21
70 - val_acc: 0.9179
Epoch 22/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.1793 - acc: 0.9376 - val_loss: 0.21
85 - val_acc: 0.9196
Epoch 23/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1519 - acc: 0.9450 - val_loss: 0.22
52 - val_acc: 0.9155
Epoch 24/30
7352/7352 [==============================] - 51s 7ms/step - loss: 0.1500 - acc: 0.9450 - val_loss: 0.20
66 - val_acc: 0.9281
Epoch 25/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1414 - acc: 0.9463 - val_loss: 0.27
```

```
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1414 - acc: 0.9463 - val_loss: 0.27
53 - val_acc: 0.9019
Epoch 26/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1571 - acc: 0.9387 - val_loss: 0.24
24 - val_acc: 0.9128
Epoch 27/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1487 - acc: 0.9397 - val_loss: 0.22
21 - val_acc: 0.9213
Epoch 28/30
7352/7352 [==============================] - 50s 7ms/step - loss: 0.1359 - acc: 0.9426 - val_loss: 0.24
21 - val_acc: 0.9196
Epoch 29/30
7352/7352 [==============================] - 49s 7ms/step - loss: 0.1220 - acc: 0.9465 - val_loss: 0.28
09 - val_acc: 0.8982
Epoch 30/30
7352/7352 [==============================] - 49s 7ms/step - loss: 0.1397 - acc: 0.9450 - val_loss: 0.25
64 - val_acc: 0.9050
```

Out[102]:

```
<keras.callbacks.History at 0x7efc6b12fd68>
```

In [103]:

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

Out[103]:

Pred True	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	517	0	0	0	0	20
SITTING	0	376	107	1	0	7
STANDING	0	96	430	3	0	3
WALKING	0	0	0	460	12	24
WALKING_DOWNSTAIRS	0	0	0	0	415	5
WALKING_UPSTAIRS	0	0	0	2	0	469

In [104]:

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

```
2947/2947 [==============================] - 1s 319us/step
```

Out[104]:

```
[0.25635178967629957, 0.9049881235154394]
```

In [0]:

```
data_disp['1LSTM_Drop3_8batch'] = score
```

Batch size = 16

In [116]:

```
# Training the model
model.fit(X_train,
          Y_train,
          batch_size=16,
          validation_data=(X_test, Y_test),
          epochs=30)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 33s 4ms/step - loss: 1.2467 - acc: 0.4747 - val_loss: 1.22
40 - val_acc: 0.5039
Epoch 2/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.8625 - acc: 0.6167 - val_loss: 0.82
23 - val_acc: 0.6563
Epoch 3/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.6891 - acc: 0.7261 - val_loss: 0.66
78 - val_acc: 0.7499
Epoch 4/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.5248 - acc: 0.8067 - val_loss: 0.52
43 - val_acc: 0.8263
Epoch 5/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.3606 - acc: 0.8796 - val_loss: 0.49
09 - val_acc: 0.8381
Epoch 6/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.2804 - acc: 0.9097 - val_loss: 0.37
92 - val_acc: 0.8772
Epoch 7/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.2696 - acc: 0.9076 - val_loss: 0.37
39 - val_acc: 0.8809
Epoch 8/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.3297 - acc: 0.8864 - val_loss: 0.38
17 - val_acc: 0.8826
Epoch 9/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.2932 - acc: 0.9042 - val_loss: 0.39
05 - val_acc: 0.8687
Epoch 10/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1986 - acc: 0.9343 - val_loss: 0.29
68 - val_acc: 0.8941
Epoch 11/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1901 - acc: 0.9368 - val_loss: 0.29
01 - val_acc: 0.9009
Epoch 12/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1905 - acc: 0.9363 - val_loss: 0.31
57 - val_acc: 0.9006
Epoch 13/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1821 - acc: 0.9382 - val_loss: 0.37
45 - val_acc: 0.8799
Epoch 14/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1832 - acc: 0.9372 - val_loss: 0.27
23 - val_acc: 0.9033
Epoch 15/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1698 - acc: 0.9402 - val_loss: 0.27
25 - val_acc: 0.9060
Epoch 16/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1674 - acc: 0.9404 - val_loss: 0.32
61 - val_acc: 0.8877
Epoch 17/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1554 - acc: 0.9461 - val_loss: 0.24
85 - val_acc: 0.9067
Epoch 18/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1611 - acc: 0.9396 - val_loss: 0.28
34 - val_acc: 0.8975
Epoch 19/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1547 - acc: 0.9426 - val_loss: 0.25
68 - val_acc: 0.9053
Epoch 20/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1544 - acc: 0.9392 - val_loss: 0.31
88 - val_acc: 0.8816
Epoch 21/30
7352/7352 [==============================] - 24s 3ms/step - loss: 0.1677 - acc: 0.9373 - val_loss: 0.26
78 - val_acc: 0.8941
Epoch 22/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1285 - acc: 0.9479 - val_loss: 0.21
86 - val_acc: 0.9138
Epoch 23/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1400 - acc: 0.9440 - val_loss: 0.24
65 - val_acc: 0.9057
Epoch 24/30
7352/7352 [==============================] - 23s 3ms/step - loss: 0.1372 - acc: 0.9461 - val_loss: 0.22
93 - val_acc: 0.9155
Epoch 25/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1319 - acc: 0.9497 - val_loss: 0.23
89 - val_acc: 0.9125
```

```
Epoch 26/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1251 - acc: 0.9510 - val_loss: 0.23
90 - val_acc: 0.9101
Epoch 27/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1224 - acc: 0.9501 - val_loss: 0.31
66 - val_acc: 0.9026
Epoch 28/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1796 - acc: 0.9376 - val_loss: 0.54
31 - val_acc: 0.8151
Epoch 29/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.2016 - acc: 0.9293 - val_loss: 0.28
99 - val_acc: 0.9013
Epoch 30/30
7352/7352 [==============================] - 22s 3ms/step - loss: 0.1243 - acc: 0.9465 - val_loss: 0.37
18 - val_acc: 0.8948
```

Out[116]:

```
<keras.callbacks.History at 0x7efc6a6c8e80>
```

In [117]:

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

Out[117]:

Pred True	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	536	0	1	0	0	0
SITTING	0	385	102	1	0	3
STANDING	0	99	424	9	0	0
WALKING	0	0	0	467	24	5
WALKING_DOWNSTAIRS	0	0	0	2	418	0
WALKING_UPSTAIRS	0	0	0	23	41	407

In [118]:

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

```
2947/2947 [==============================] - 1s 296us/step
```

Out[118]:

```
[0.3718474003477882, 0.8948082796063793]
```

In [0]:

```
data_disp['1LSTM_Drop3_16batch'] = score
```

Two LSTM + Dropout

In [125]:

```
# Initiliazing the sequential model
model = Sequential()
# Configuring the parameters
model.add(LSTM(20, input_shape=(timesteps, input_dim), return_sequences=True))
# Adding a dropout layer
model.add(Dropout(0.3))

model.add(LSTM(20))
```

```
# Adding a dropout layer
model.add(Dropout(0.5))

# Adding a dense output layer with sigmoid activation
model.add(Dense(n_classes, activation='softmax'))
# Compiling the model
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
model.summary()
```

```
Model: "sequential_38"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_60 (LSTM)               (None, 128, 20)           2400
_________________________________________________________________
dropout_59 (Dropout)         (None, 128, 20)           0
_________________________________________________________________
lstm_61 (LSTM)               (None, 20)                3280
_________________________________________________________________
dropout_60 (Dropout)         (None, 20)                0
_________________________________________________________________
dense_39 (Dense)             (None, 6)                 126
=================================================================
Total params: 5,806
Trainable params: 5,806
Non-trainable params: 0
_________________________________________________________________
```

Batch Size = 8

In [121]:

```
# Training the model
model.fit(X_train,
          Y_train,
          batch_size=8,
          validation_data=(X_test, Y_test),
          epochs=30)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 91s 12ms/step - loss: 1.1299 - acc: 0.5276 - val_loss: 0.8
241 - val_acc: 0.6637
Epoch 2/30
7352/7352 [==============================] - 84s 11ms/step - loss: 0.7477 - acc: 0.7175 - val_loss: 0.7
358 - val_acc: 0.7363
Epoch 3/30
7352/7352 [==============================] - 84s 11ms/step - loss: 0.5924 - acc: 0.7884 - val_loss: 0.6
664 - val_acc: 0.7648
Epoch 4/30
7352/7352 [==============================] - 84s 11ms/step - loss: 0.4982 - acc: 0.8268 - val_loss: 0.7
771 - val_acc: 0.7628
Epoch 5/30
7352/7352 [==============================] - 84s 11ms/step - loss: 0.4058 - acc: 0.8651 - val_loss: 0.6
789 - val_acc: 0.7974
Epoch 6/30
7352/7352 [==============================] - 84s 11ms/step - loss: 0.3733 - acc: 0.8796 - val_loss: 0.6
812 - val_acc: 0.8012
Epoch 7/30
7352/7352 [==============================] - 85s 12ms/step - loss: 0.2804 - acc: 0.9123 - val_loss: 0.4
393 - val_acc: 0.8636
Epoch 8/30
7352/7352 [==============================] - 84s 11ms/step - loss: 0.2500 - acc: 0.9187 - val_loss: 0.4
697 - val_acc: 0.8677
Epoch 9/30
7352/7352 [==============================] - 84s 11ms/step - loss: 0.2355 - acc: 0.9251 - val_loss: 0.4
339 - val_acc: 0.8863
Epoch 10/30
7352/7352 [==============================] - 84s 11ms/step - loss: 0.2585 - acc: 0.9203 - val_loss: 0.3
953 - val_acc: 0.8846
Epoch 11/30
```

```
7352/7352 [==============================] - 84s 11ms/step - loss: 0.1999 - acc: 0.9335 - val_loss: 0.5
226 - val_acc: 0.8683
Epoch 12/30
7352/7352 [==============================] - 84s 11ms/step - loss: 0.2952 - acc: 0.9060 - val_loss: 0.5
140 - val_acc: 0.8880
Epoch 13/30
7352/7352 [==============================] - 83s 11ms/step - loss: 0.2307 - acc: 0.9319 - val_loss: 0.3
999 - val_acc: 0.8989
Epoch 14/30
7352/7352 [==============================] - 84s 11ms/step - loss: 0.2682 - acc: 0.9154 - val_loss: 0.3
859 - val_acc: 0.9033
Epoch 15/30
7352/7352 [==============================] - 85s 12ms/step - loss: 0.1944 - acc: 0.9385 - val_loss: 0.2
856 - val_acc: 0.9111
Epoch 16/30
7352/7352 [==============================] - 84s 11ms/step - loss: 0.2033 - acc: 0.9286 - val_loss: 0.3
090 - val_acc: 0.9141
Epoch 17/30
7352/7352 [==============================] - 117s 16ms/step - loss: 0.1698 - acc: 0.9368 - val_loss: 0.
3159 - val_acc: 0.9189
Epoch 18/30
7352/7352 [==============================] - 126s 17ms/step - loss: 0.2682 - acc: 0.9143 - val_loss: 0.
3977 - val_acc: 0.8948
Epoch 19/30
7352/7352 [==============================] - 126s 17ms/step - loss: 0.1796 - acc: 0.9376 - val_loss: 0.
3634 - val_acc: 0.9013
Epoch 20/30
7352/7352 [==============================] - 123s 17ms/step - loss: 0.1490 - acc: 0.9470 - val_loss: 0.
3578 - val_acc: 0.9091
Epoch 21/30
7352/7352 [==============================] - 122s 17ms/step - loss: 0.1418 - acc: 0.9474 - val_loss: 0.
3851 - val_acc: 0.9053
Epoch 22/30
7352/7352 [==============================] - 120s 16ms/step - loss: 0.1747 - acc: 0.9373 - val_loss: 0.
4205 - val_acc: 0.8989
Epoch 23/30
7352/7352 [==============================] - 119s 16ms/step - loss: 0.1723 - acc: 0.9404 - val_loss: 0.
5246 - val_acc: 0.8979
Epoch 24/30
7352/7352 [==============================] - 119s 16ms/step - loss: 0.1572 - acc: 0.9509 - val_loss: 0.
3875 - val_acc: 0.9026
Epoch 25/30
7352/7352 [==============================] - 122s 17ms/step - loss: 0.1559 - acc: 0.9434 - val_loss: 0.
4701 - val_acc: 0.9114
Epoch 26/30
7352/7352 [==============================] - 122s 17ms/step - loss: 0.1617 - acc: 0.9433 - val_loss: 0.
5362 - val_acc: 0.9036
Epoch 27/30
7352/7352 [==============================] - 124s 17ms/step - loss: 0.1786 - acc: 0.9388 - val_loss: 0.
3491 - val_acc: 0.9203
Epoch 28/30
7352/7352 [==============================] - 125s 17ms/step - loss: 0.1424 - acc: 0.9471 - val_loss: 0.
4278 - val_acc: 0.9094
Epoch 29/30
7352/7352 [==============================] - 126s 17ms/step - loss: 0.1501 - acc: 0.9441 - val_loss: 0.
4421 - val_acc: 0.9033
Epoch 30/30
7352/7352 [==============================] - 129s 17ms/step - loss: 0.1416 - acc: 0.9510 - val_loss: 0.
3962 - val_acc: 0.9172
```

Out[121]:

```
<keras.callbacks.History at 0x7efc6a376358>
```

In [122]:

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

Out[122]:

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
True						

Pred / True	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	537	0	0	0	0	0
SITTING	6	395	70	2	0	18
STANDING	0	104	425	1	0	2
WALKING	0	0	0	467	27	2
WALKING_DOWNSTAIRS	0	0	0	1	417	2
WALKING_UPSTAIRS	0	0	1	2	6	462

In [123]:

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

```
2947/2947 [==============================] - 2s 702us/step
```

Out[123]:

```
[0.3962146586148283, 0.9172039362063115]
```

In [0]:

```
data_disp['2LSTM_8batch'] = score
```

Batch Size = 16

In [126]:

```
# Training the model
model.fit(X_train,
          Y_train,
          batch_size=16,
          validation_data=(X_test, Y_test),
          epochs=30)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 76s 10ms/step - loss: 1.1923 - acc: 0.4792 - val_loss: 1.0
667 - val_acc: 0.5056
Epoch 2/30
7352/7352 [==============================] - 65s 9ms/step - loss: 0.7928 - acc: 0.6337 - val_loss: 0.76
68 - val_acc: 0.6128
Epoch 3/30
7352/7352 [==============================] - 65s 9ms/step - loss: 0.7390 - acc: 0.6465 - val_loss: 0.77
04 - val_acc: 0.6037
Epoch 4/30
7352/7352 [==============================] - 64s 9ms/step - loss: 0.6943 - acc: 0.6518 - val_loss: 0.72
45 - val_acc: 0.6261
Epoch 5/30
7352/7352 [==============================] - 64s 9ms/step - loss: 0.6670 - acc: 0.6593 - val_loss: 0.72
33 - val_acc: 0.6593
Epoch 6/30
7352/7352 [==============================] - 63s 9ms/step - loss: 0.6417 - acc: 0.6895 - val_loss: 0.70
56 - val_acc: 0.6905
Epoch 7/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.5903 - acc: 0.7428 - val_loss: 0.59
78 - val_acc: 0.7628
Epoch 8/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.4646 - acc: 0.8101 - val_loss: 0.57
37 - val_acc: 0.7954
Epoch 9/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.3741 - acc: 0.8679 - val_loss: 0.43
15 - val_acc: 0.8514
Epoch 10/30
7352/7352 [==============================] - 61s 8ms/step - loss: 0.3605 - acc: 0.8815 - val_loss: 0.40
25 - val_acc: 0.8697
Epoch 11/30
7352/7352 [==============================] - 61s 8ms/step - loss: 0.2802 - acc: 0.9112 - val_loss: 0.43
```

```
28 - val_acc: 0.8612
Epoch 12/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.3299 - acc: 0.8972 - val_loss: 0.41
56 - val_acc: 0.8775
Epoch 13/30
7352/7352 [==============================] - 61s 8ms/step - loss: 0.2398 - acc: 0.9251 - val_loss: 0.41
12 - val_acc: 0.8778
Epoch 14/30
7352/7352 [==============================] - 61s 8ms/step - loss: 0.2130 - acc: 0.9309 - val_loss: 0.50
10 - val_acc: 0.8510
Epoch 15/30
7352/7352 [==============================] - 60s 8ms/step - loss: 0.2135 - acc: 0.9342 - val_loss: 0.45
36 - val_acc: 0.8741
Epoch 16/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.2025 - acc: 0.9312 - val_loss: 0.48
08 - val_acc: 0.8751
Epoch 17/30
7352/7352 [==============================] - 61s 8ms/step - loss: 0.2289 - acc: 0.9211 - val_loss: 0.37
30 - val_acc: 0.8833
Epoch 18/30
7352/7352 [==============================] - 60s 8ms/step - loss: 0.2704 - acc: 0.9101 - val_loss: 0.34
25 - val_acc: 0.8870
Epoch 19/30
7352/7352 [==============================] - 60s 8ms/step - loss: 0.1712 - acc: 0.9387 - val_loss: 0.39
96 - val_acc: 0.8846
Epoch 20/30
7352/7352 [==============================] - 62s 8ms/step - loss: 0.1632 - acc: 0.9452 - val_loss: 0.38
19 - val_acc: 0.8975
Epoch 21/30
7352/7352 [==============================] - 63s 9ms/step - loss: 0.1644 - acc: 0.9440 - val_loss: 0.33
79 - val_acc: 0.9084
Epoch 22/30
7352/7352 [==============================] - 64s 9ms/step - loss: 0.2306 - acc: 0.9276 - val_loss: 0.49
73 - val_acc: 0.8534
Epoch 23/30
7352/7352 [==============================] - 63s 9ms/step - loss: 0.1634 - acc: 0.9465 - val_loss: 0.33
06 - val_acc: 0.8985
Epoch 24/30
7352/7352 [==============================] - 63s 9ms/step - loss: 0.1400 - acc: 0.9495 - val_loss: 0.35
54 - val_acc: 0.8968
Epoch 25/30
7352/7352 [==============================] - 64s 9ms/step - loss: 0.1439 - acc: 0.9483 - val_loss: 0.37
36 - val_acc: 0.8958
Epoch 26/30
7352/7352 [==============================] - 64s 9ms/step - loss: 0.1563 - acc: 0.9448 - val_loss: 0.38
10 - val_acc: 0.8911
Epoch 27/30
7352/7352 [==============================] - 63s 9ms/step - loss: 0.2519 - acc: 0.9208 - val_loss: 0.38
19 - val_acc: 0.8890
Epoch 28/30
7352/7352 [==============================] - 63s 9ms/step - loss: 0.1576 - acc: 0.9459 - val_loss: 0.39
80 - val_acc: 0.8863
Epoch 29/30
7352/7352 [==============================] - 65s 9ms/step - loss: 0.1519 - acc: 0.9421 - val_loss: 0.33
94 - val_acc: 0.9057
Epoch 30/30
7352/7352 [==============================] - 66s 9ms/step - loss: 0.1827 - acc: 0.9385 - val_loss: 0.46
90 - val_acc: 0.8924
```

Out[126]:

```
<keras.callbacks.History at 0x7efc695e8a20>
```

In [127]:

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

Out[127]:

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
True						
LAYING	521	0	0	0	0	16

Pred / True	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	521	0	0	0	0	16
SITTING	5	401	64	1	0	20
STANDING	0	96	391	0	0	45
WALKING	0	0	0	460	29	7
WALKING_DOWNSTAIRS	0	0	0	6	411	3
WALKING_UPSTAIRS	0	0	0	21	4	446

In [128]:

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

```
2947/2947 [==============================] - 2s 679us/step
```

Out[128]:

```
[0.46896032411779726, 0.8924329826942654]
```

In [0]:

```
data_disp['2LSTM_16batch'] = score
```

Conclusion

1. Obtain and Load Data from UCI
2. Data Cleaning
3. Check for imbalance data
4. Changing feature names (Just remove '-' and replace with empty space
5. EDA: Stationary and Moving Activities
6. EDA: Magnitude of an acceleration (BoxPlot)
7. EDA: Position of GravityAccelerationComponants
8. Perform t-SNE with different perplexity
9. Based on Feature Engineering done by people (contain 561 features), we perform Classical ML (which achieved 96%)
10. Based on Acceleratometer and Gyro Sensor (contain 9 features), We perform hyperparameter on LSTM (which achieved 92%)
11. Used Divide and Conquer CNN

In [132]:

```
from prettytable import PrettyTable
d = PrettyTable()
d.field_names = ['Model', 'Test Loss', 'Test Accuracy %']
for i in data_disp.keys():
    d.add_row([i,data_disp.get(i)[0],data_disp.get(i)[1]*100])
print(d)
```

```
+--------------------+--------------------+--------------------+
|       Model        |     Test Loss      |  Test Accuracy %   |
+--------------------+--------------------+--------------------+
|    1LSTM_8batch    |  0.2727017437300063 | 91.75432643366135 |
|   1LSTM_16batch    |  0.2531991467388801 | 90.70240922972515 |
|   1LSTM_32batch    | 0.3189272664430161 | 88.08958262639973 |
| 1LSTM_Drop3_8batch | 0.25635178967629957 | 90.49881235154395 |
| 1LSTM_Drop3_16batch |  0.3718474003477882 | 89.48082796063794 |
|    2LSTM_8batch    |  0.3962146586148283 | 91.72039362063114 |
|   2LSTM_16batch    | 0.46896032411779726 | 89.24329826942655 |
+--------------------+--------------------+--------------------+
```

In [2]:

```
from prettytable import PrettyTable
print('Please go to sahiltinky94@gmail.com_2.ipynb to see the result. (Used Divide and Conquer CNN)')
d = PrettyTable()
```

```
d.field_names = ['Model', 'Test Loss', 'Test Accuracy %']
d.add_row(['Dvide and Conquer CNN (Static)',0.288,92.7])
d.add_row(['Dvide and Conquer CNN (Dynamic)',0.205,96.4])
d.add_row(['Dvide and Conquer CNN (On test stage after merging)','-',94.43])
print(d)
```

```
Please go to sahiltinky94@gmail.com_2.ipynb to see the result. (Used Divide and Conquer CNN)
+-----------------------------------------------------+-----------+-----------------+
|                        Model                        | Test Loss | Test Accuracy % |
+-----------------------------------------------------+-----------+-----------------+
|            Dvide and Conquer CNN (Static)           |   0.288   |       92.7      |
|           Dvide and Conquer CNN (Dynamic)           |   0.205   |       96.4      |
| Dvide and Conquer CNN (On test stage after merging) |     -     |      94.43      |
+-----------------------------------------------------+-----------+-----------------+
```

In [ ]:

```
```