In [0]:

```
# Importing Libraries
```

In [0]:

```
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

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 [9]:

```
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=947318989803-6bn6
qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%
b&response_type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2
www.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly
ttps%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly
```

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

In [0]:

```python
# 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}
_{subset}.txt'
        signals_data.append(
            _read_csv(filename).as_matrix()
        )

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

In [0]:

```python
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]:

```python
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 [13]:

```python
# 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]:

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

In [15]:

```
# 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
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 [19]:

```
# 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 [20]:

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

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

```
128
9
7352
```

- Defining the Architecture of LSTM

In [0]:

```
# 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.add(Dense(n_classes, activation='sigmoid'))
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm_3 (LSTM)                (None, 32)                5376      
_________________________________________________________________
dropout_3 (Dropout)          (None, 32)                0         
_________________________________________________________________
dense_3 (Dense)              (None, 6)                 198       
=================================================================
Total params: 5,574
Trainable params: 5,574
Non-trainable params: 0
_________________________________________________________________
```

In [0]:

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

In [0]:

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

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 92s 13ms/step - loss: 1.3018 - acc: 0.4395 - val_loss
: 1.1254 - val_acc: 0.4662
Epoch 2/30
7352/7352 [==============================] - 94s 13ms/step - loss: 0.9666 - acc: 0.5880 - val_loss
: 0.9491 - val_acc: 0.5714
Epoch 3/30
7352/7352 [==============================] - 97s 13ms/step - loss: 0.7812 - acc: 0.6408 - val_loss
: 0.8286 - val_acc: 0.5850
Epoch 4/30
7352/7352 [==============================] - 95s 13ms/step - loss: 0.6941 - acc: 0.6574 - val_loss
: 0.7297 - val_acc: 0.6128
Epoch 5/30
7352/7352 [==============================] - 92s 13ms/step - loss: 0.6336 - acc: 0.6912 - val_loss
: 0.7359 - val_acc: 0.6787
Epoch 6/30
7352/7352 [==============================] - 94s 13ms/step - loss: 0.5859 - acc: 0.7134 - val_loss
: 0.7015 - val_acc: 0.6939
Epoch 7/30
7352/7352 [==============================] - 95s 13ms/step - loss: 0.5692 - acc: 0.7477 - val_loss
: 0.5995 - val_acc: 0.7387
Epoch 8/30
7352/7352 [==============================] - 96s 13ms/step - loss: 0.4899 - acc: 0.7809 - val_loss
: 0.5762 - val_acc: 0.7387
Epoch 9/30
7352/7352 [==============================] - 90s 12ms/step - loss: 0.4482 - acc: 0.7886 - val_loss
: 0.7413 - val_acc: 0.7126
Epoch 10/30
7352/7352 [==============================] - 90s 12ms/step - loss: 0.4132 - acc: 0.8077 - val_loss
: 0.5048 - val_acc: 0.7513
Epoch 11/30
7352/7352 [==============================] - 89s 12ms/step - loss: 0.3985 - acc: 0.8274 - val_loss
: 0.5234 - val_acc: 0.7452
Epoch 12/30
7352/7352 [==============================] - 91s 12ms/step - loss: 0.3378 - acc: 0.8638 - val_loss
: 0.4114 - val_acc: 0.8833
Epoch 13/30
7352/7352 [==============================] - 91s 12ms/step - loss: 0.2947 - acc: 0.9051 - val_loss
: 0.4386 - val_acc: 0.8731
Epoch 14/30
```

```
7352/7352 [==============================] - 90s 12ms/step - loss: 0.2448 - acc: 0.9291 - val_loss
: 0.3768 - val_acc: 0.8921
Epoch 15/30
7352/7352 [==============================] - 91s 12ms/step - loss: 0.2157 - acc: 0.9331 - val_loss
: 0.4441 - val_acc: 0.8931
Epoch 16/30
7352/7352 [==============================] - 90s 12ms/step - loss: 0.2053 - acc: 0.9366 - val_loss
: 0.4162 - val_acc: 0.8968
Epoch 17/30
7352/7352 [==============================] - 89s 12ms/step - loss: 0.2028 - acc: 0.9404 - val_loss
: 0.4538 - val_acc: 0.8962
Epoch 18/30
7352/7352 [==============================] - 93s 13ms/step - loss: 0.1911 - acc: 0.9419 - val_loss
: 0.3964 - val_acc: 0.8999
Epoch 19/30
7352/7352 [==============================] - 96s 13ms/step - loss: 0.1912 - acc: 0.9407 - val_loss
: 0.3165 - val_acc: 0.9030
Epoch 20/30
7352/7352 [==============================] - 96s 13ms/step - loss: 0.1732 - acc: 0.9446 - val_loss
: 0.4546 - val_acc: 0.8904
Epoch 21/30
7352/7352 [==============================] - 94s 13ms/step - loss: 0.1782 - acc: 0.9444 - val_loss
: 0.3346 - val_acc: 0.9063
Epoch 22/30
7352/7352 [==============================] - 95s 13ms/step - loss: 0.1812 - acc: 0.9418 - val_loss
: 0.8164 - val_acc: 0.8582
Epoch 23/30
7352/7352 [==============================] - 95s 13ms/step - loss: 0.1824 - acc: 0.9426 - val_loss
: 0.4240 - val_acc: 0.9036
Epoch 24/30
7352/7352 [==============================] - 94s 13ms/step - loss: 0.1726 - acc: 0.9429 - val_loss
: 0.4067 - val_acc: 0.9148
Epoch 25/30
7352/7352 [==============================] - 96s 13ms/step - loss: 0.1737 - acc: 0.9411 - val_loss
: 0.3396 - val_acc: 0.9074
Epoch 26/30
7352/7352 [==============================] - 96s 13ms/step - loss: 0.1650 - acc: 0.9461 - val_loss
: 0.3806 - val_acc: 0.9019
Epoch 27/30
7352/7352 [==============================] - 89s 12ms/step - loss: 0.1925 - acc: 0.9415 - val_loss
: 0.6464 - val_acc: 0.8850
Epoch 28/30
7352/7352 [==============================] - 91s 12ms/step - loss: 0.1965 - acc: 0.9425 - val_loss
: 0.3363 - val_acc: 0.9203
Epoch 29/30
7352/7352 [==============================] - 92s 12ms/step - loss: 0.1889 - acc: 0.9431 - val_loss
: 0.3737 - val_acc: 0.9158
Epoch 30/30
7352/7352 [==============================] - 95s 13ms/step - loss: 0.1945 - acc: 0.9414 - val_loss
: 0.3088 - val_acc: 0.9097
```

Out[0]:

```
<keras.callbacks.History at 0x29b5ee36a20>
```

In [0]:

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

```
Pred                LAYING  SITTING  STANDING  WALKING  WALKING_DOWNSTAIRS  \
True
LAYING                 512        0        25        0                   0
SITTING                  3      410        75        0                   0
STANDING                 0       87       445        0                   0
WALKING                  0        0         0      481                   2
WALKING_DOWNSTAIRS       0        0         0        0                 382
WALKING_UPSTAIRS         0        0         0        2                  18

Pred                WALKING_UPSTAIRS
True
LAYING                             0
SITTING                            3
STANDING                           0
WALKING                           13
```

```
WALKING_DOWNSTAIRS                  38
WALKING_UPSTAIRS                   451
```

In [0]:

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

```
2947/2947 [==============================] - 4s 2ms/step
```

In [0]:

```
score
```

Out[0]:

```
[0.3087582236972612, 0.9097387173396675]
```

- 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

In [50]:

```
fig , ax = plt.subplots(1,1)
```

In [0]:

```
%matplotlib notebook
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_plot(x, vy, ty, ax, colors=['b']):
    fig , ax = plt.subplots(1,1)
    ax.plot(x, vy, 'b', label="Validation Loss")
    ax.plot(x, ty, 'r', label="Train Loss")
    plt.legend()
    plt.grid()
    fig.canvas.draw()
```

In [0]:

```
# Initiliazing the sequential model

def create_LSTM_model(dropout = 0.1, optimizer ='rmsprop',n_hidden = 32):

    model = Sequential()
    # Configuring the parameters
    model.add(LSTM(n_hidden, input_shape=(timesteps, input_dim)))
    # Adding a dropout layer
    model.add(Dropout(dropout))
    # Adding a dense output layer with sigmoid activation
    model.add(Dense(n_classes, activation='sigmoid'))
    model.compile(loss='categorical_crossentropy',optimizer=optimizer,metrics=['accuracy'])
    return model
```

In [0]:

```
from sklearn.model_selection import GridSearchCV
from keras.layers import Dense, Input, Dropout
from keras import Sequential
from keras.wrappers.scikit_learn import KerasClassifier

model_hp = KerasClassifier(build_fn = create_LSTM_model, epochs=30, batch_size=16)
```

In [21]:

In [21]:

```
# define the grid search parameters
dropout = [0.3,0.4,0.5,0.6]
n_hidden = [35,40,52,56]


param_grid = dict( dropout = dropout, n_hidden = n_hidden)

model_grid = GridSearchCV(estimator=model_hp, param_grid=param_grid, n_jobs=-1, cv=3)
model_result = model_grid.fit(X_train,Y_train, batch_size=batch_size,validation_data=(X_test, Y_tes
t), epochs=epochs)

# summarize results
print("Best: %f using %s" % (model_result.best_score_, model_result.best_params_))
```

```
/usr/local/lib/python3.6/dist-packages/joblib/externals/loky/process_executor.py:706: UserWarning:
A worker stopped while some jobs were given to the executor. This can be caused by a too short wor
ker timeout or by a memory leak.
  "timeout or by a memory leak.", UserWarning
```

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

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

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

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:3733: calling dropout (from
tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future
version.
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/optimizers.py:793: The name t
f.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:3576: The name tf.log is deprecated. Please use tf.ma
th.log instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow_core/python/ops/math_grad.py:1424: where (from
tensorflow.python.ops.array_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/backend/tensorflow_backend.py:1033: The name tf.assign_add is deprecated. Please us
e tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:1020: The name tf.assign is deprecated. Please 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/backend/tensorflow_backend.py:190: The name tf.get_default_session is deprecated. P
lease use tf.compat.v1.get_default_session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:207: The name tf.global_variables is deprecated. Plea
se use tf.compat.v1.global_variables instead.

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

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

```
7352/7352 [==============================] - 36s 5ms/step - loss: 1.2580 - acc: 0.4491 - val_loss:
1.2065 - val_acc: 0.4483
Epoch 2/30
7352/7352 [==============================] - 37s 5ms/step - loss: 1.0177 - acc: 0.5400 - val_loss:
1.1445 - val_acc: 0.4951
Epoch 3/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.7743 - acc: 0.6473 - val_loss:
0.7985 - val_acc: 0.6162
Epoch 4/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.6574 - acc: 0.7148 - val_loss:
0.6960 - val_acc: 0.7068
Epoch 5/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.5237 - acc: 0.8164 - val_loss:
0.6128 - val_acc: 0.8419
Epoch 6/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.3897 - acc: 0.8777 - val_loss:
0.5130 - val_acc: 0.8432
Epoch 7/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.2768 - acc: 0.9106 - val_loss:
0.4408 - val_acc: 0.8826
Epoch 8/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.2349 - acc: 0.9242 - val_loss:
0.5436 - val_acc: 0.8266
Epoch 9/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.2027 - acc: 0.9328 - val_loss:
0.3406 - val_acc: 0.8894
Epoch 10/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1836 - acc: 0.9373 - val_loss:
0.4926 - val_acc: 0.8524
Epoch 11/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.2239 - acc: 0.9261 - val_loss:
0.4686 - val_acc: 0.8799
Epoch 12/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1913 - acc: 0.9336 - val_loss:
0.4525 - val_acc: 0.8721
Epoch 13/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1457 - acc: 0.9460 - val_loss:
0.4869 - val_acc: 0.8884
Epoch 14/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1675 - acc: 0.9434 - val_loss:
0.3264 - val_acc: 0.8955
Epoch 15/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1489 - acc: 0.9453 - val_loss:
0.3879 - val_acc: 0.9026
Epoch 16/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1462 - acc: 0.9489 - val_loss:
0.6029 - val_acc: 0.8931
Epoch 17/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1483 - acc: 0.9467 - val_loss:
0.5028 - val_acc: 0.9016
Epoch 18/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1583 - acc: 0.9482 - val_loss:
0.9213 - val_acc: 0.8297
Epoch 19/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1387 - acc: 0.9501 - val_loss:
0.3993 - val_acc: 0.8989
Epoch 20/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1370 - acc: 0.9483 - val_loss:
0.6000 - val_acc: 0.8938
Epoch 21/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1535 - acc: 0.9499 - val_loss:
0.4591 - val_acc: 0.8965
Epoch 22/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1437 - acc: 0.9493 - val_loss:
0.4688 - val_acc: 0.8968
Epoch 23/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1424 - acc: 0.9486 - val_loss:
0.5779 - val_acc: 0.9006
Epoch 24/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1263 - acc: 0.9516 - val_loss:
0.5373 - val_acc: 0.9077
Epoch 25/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1290 - acc: 0.9531 - val_loss:
0.5091 - val_acc: 0.9002
Epoch 26/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1311 - acc: 0.9512 - val_loss:
```

```
0.4648 - val_acc: 0.9125
Epoch 27/30
7352/7352 [==============================] - 35s 5ms/step - loss: 0.1294 - acc: 0.9523 - val_loss:
0.3985 - val_acc: 0.9053
Epoch 28/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1185 - acc: 0.9551 - val_loss:
0.4651 - val_acc: 0.9141
Epoch 29/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1462 - acc: 0.9517 - val_loss:
0.4613 - val_acc: 0.9043
Epoch 30/30
7352/7352 [==============================] - 34s 5ms/step - loss: 0.1217 - acc: 0.9547 - val_loss:
0.4660 - val_acc: 0.9104
Best: 0.928727 using {'dropout': 0.4, 'n_hidden': 52}
```

Run Train vs Test data on Best parameters

In [19]:

```
model_tuned = Sequential()
# Configuring the parameters
model_tuned.add(LSTM(52, input_shape=(timesteps, input_dim)))
# Adding a dropout layer
model_tuned.add(Dropout(0.4))
# Adding a dense output layer with sigmoid activation
model_tuned.add(Dense(n_classes, activation='sigmoid'))
```

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

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

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

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:3733: calling dropout (from
tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future
version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
```

In [20]:

```
model_tuned.compile(loss='categorical_crossentropy',optimizer='rmsprop',metrics=['accuracy'])

history_tuned = model_tuned.fit(X_train, Y_train,
          batch_size=16,
          epochs=30,
          verbose=1,
          validation_data=(X_test, Y_test))
score_tuned = model_tuned.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', score_tuned[0])
print('Test accuracy:', score_tuned[1])
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optimizers.py:793: The name t
f.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:3576: The name tf.log is deprecated. Please use tf.ma
th.log instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow_core/python/ops/math_grad.py:1424: where (from
tensorflow.python.ops.array_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/backend/tensorflow_backend.py:1033: The name tf.assign_add is deprecated. Please us
```

```
packages/keras/backend/tensorflow_backend.py:1033: The name tf.assign_add is deprecated. Please us
e tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:1020: The name tf.assign is deprecated. Please 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/backend/tensorflow_backend.py:190: The name tf.get_default_session is deprecated. P
lease use tf.compat.v1.get_default_session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:207: The name tf.global_variables is deprecated. Plea
se use tf.compat.v1.global_variables instead.

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

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

7352/7352 [==============================] - 38s 5ms/step - loss: 1.2325 - acc: 0.4581 - val_loss:
1.1053 - val_acc: 0.4941
Epoch 2/30
7352/7352 [==============================] - 37s 5ms/step - loss: 1.0265 - acc: 0.5284 - val_loss:
1.0074 - val_acc: 0.5786
Epoch 3/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.7745 - acc: 0.6387 - val_loss:
0.8072 - val_acc: 0.5938
Epoch 4/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.6556 - acc: 0.7067 - val_loss:
0.7486 - val_acc: 0.6960
Epoch 5/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.5630 - acc: 0.7745 - val_loss:
0.6779 - val_acc: 0.7486
Epoch 6/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.4793 - acc: 0.8060 - val_loss:
0.5241 - val_acc: 0.7869
Epoch 7/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.3840 - acc: 0.8720 - val_loss:
0.4650 - val_acc: 0.8497
Epoch 8/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.2915 - acc: 0.9097 - val_loss:
0.5165 - val_acc: 0.8595
Epoch 9/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.2590 - acc: 0.9176 - val_loss:
0.4385 - val_acc: 0.8578
Epoch 10/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.2359 - acc: 0.9241 - val_loss:
0.3673 - val_acc: 0.8806
Epoch 11/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1953 - acc: 0.9372 - val_loss:
0.3482 - val_acc: 0.8846
Epoch 12/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1731 - acc: 0.9399 - val_loss:
0.5497 - val_acc: 0.8758
Epoch 13/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1819 - acc: 0.9378 - val_loss:
0.3412 - val_acc: 0.9009
Epoch 14/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1719 - acc: 0.9391 - val_loss:
0.2591 - val_acc: 0.9091
Epoch 15/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1651 - acc: 0.9436 - val_loss:
0.2987 - val_acc: 0.8989
Epoch 16/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1611 - acc: 0.9421 - val_loss:
0.5175 - val_acc: 0.8711
Epoch 17/30
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1497 - acc: 0.9431 - val_loss:
0.2983 - val_acc: 0.9155
Epoch 18/30
7352/7352 [==============================] - 38s 5ms/step - loss: 0.1613 - acc: 0.9449 - val_loss:
0.3482 - val_acc: 0.9138
```

```
0.3482 - val_acc: 0.9138
Epoch 19/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1492 - acc: 0.9461 - val_loss:
0.3107 - val_acc: 0.9033
Epoch 20/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1460 - acc: 0.9461 - val_loss:
0.3883 - val_acc: 0.9033
Epoch 21/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1407 - acc: 0.9479 - val_loss:
0.3402 - val_acc: 0.9077
Epoch 22/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1417 - acc: 0.9498 - val_loss:
0.3626 - val_acc: 0.9097
Epoch 23/30
7352/7352 [==============================] - 36s 5ms/step - loss: 0.1504 - acc: 0.9467 - val_loss:
0.3377 - val_acc: 0.9145
Epoch 24/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1421 - acc: 0.9510 - val_loss:
0.4576 - val_acc: 0.9057
Epoch 25/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1402 - acc: 0.9493 - val_loss:
0.4428 - val_acc: 0.9097
Epoch 26/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1361 - acc: 0.9533 - val_loss:
0.3410 - val_acc: 0.9226
Epoch 27/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1395 - acc: 0.9479 - val_loss:
0.4976 - val_acc: 0.9128
Epoch 28/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1296 - acc: 0.9501 - val_loss:
0.3633 - val_acc: 0.9179
Epoch 29/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1398 - acc: 0.9482 - val_loss:
0.2531 - val_acc: 0.9165
Epoch 30/30
7352/7352 [==============================] - 37s 5ms/step - loss: 0.1261 - acc: 0.9505 - val_loss:
0.3882 - val_acc: 0.9196
Test loss: 0.3882474149415083
Test accuracy: 0.9195792331184255
```

In [2]:

```
pip install mpld3
```

```
Collecting mpld3
  Downloading
https://files.pythonhosted.org/packages/91/95/a52d3a83d0a29ba0d6898f6727e9858fe7a43f6c2ce81a5fe7e05
912/mpld3-0.3.tar.gz (788kB)
     |████████████████████████████████| 798kB 2.7MB/s
Building wheels for collected packages: mpld3
  Building wheel for mpld3 (setup.py) ... done
  Created wheel for mpld3: filename=mpld3-0.3-cp36-none-any.whl size=116679
sha256=c516cd230eec43a41dc0786c1400334ac90b3e603a0ab74aaff1f48d541387c3
  Stored in directory:
/root/.cache/pip/wheels/c0/47/fb/8a64f89aecfe0059830479308ad42d62e898a3e3cefdf6ba28
Successfully built mpld3
Installing collected packages: mpld3
Successfully installed mpld3-0.3
```

In [26]:

```
import mpld3
from mpld3 import plugins

score_tuned = model_tuned.evaluate( X_test,Y_test, verbose=0)

print('Test score:', score_tuned[0])
print('Test accuracy:', score_tuned[1])

ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

# list of epoch numbers
x_task1 = list(range(1,epochs+1))

vy_task1 = history_tuned.history['val_loss']
```

```
ty_task1 = history_tuned.history['loss']

print(vy_task1)
print(ty_task1)

plt_dynamic_plot(x_task1, vy_task1, ty_task1, ax)
mpld3.display()
```

```
Test score: 0.3882474149415083
Test accuracy: 0.9195792331184255
[1.1052516780093358, 1.007400548926826, 0.8072054080126607, 0.7485919309584699,
0.6778704413735224, 0.5241064832188393, 0.4650080417936101, 0.5164894072315553,
0.43848061577891306, 0.36728225362527905, 0.3482145374461637, 0.5497299465828398,
0.3411842788092275, 0.25906522794065157, 0.2986735072996452, 0.5174608449404123,
0.2983435132493801, 0.3482147149562622, 0.31072190842678293, 0.3882514895702614,
0.3401883760806517, 0.36255835237409495, 0.3376941309104325, 0.4575649724266741,
0.44280103576733826, 0.34100390364740557, 0.4976037994088862, 0.3633469000526815,
0.2531304164422931, 0.3882474233401916]
[1.2324611805987435, 1.0265450283025632, 0.7744624197677638, 0.65559397623251, 0.5630432712045925,
0.4792538316063056, 0.38396258797330357, 0.2915415139108799, 0.25903813445269736,
0.23592952726423028, 0.1952862721200023, 0.17313565549129564, 0.18192019666949444,
0.1718920935591393, 0.16507040469585973, 0.16106606266913293, 0.14973313726790824,
0.1613333808831765, 0.14916982826492034, 0.14599874252716932, 0.14065486409952732,
0.1416844128703514, 0.1503508756526516, 0.1420520100255129, 0.14024394817011365,
0.13612216549817324, 0.1394854156534523, 0.12955775405814865, 0.13980824824435772,
0.1260954074534867]
```

Out[26]:

In [71]:

```
confusion_matrix(Y_test, model_tuned.predict(X_test))
```

Out[71]:

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
True						
LAYING	537	0	0	0	0	0
SITTING	0	368	123	0	0	0
STANDING	0	60	472	0	0	0
WALKING	0	0	0	464	25	7
WALKING_DOWNSTAIRS	0	0	0	3	402	15
WALKING_UPSTAIRS	0	2	0	0	2	467

2 layer LSTM with large Dropouts

In [0]:

```
def create_2LSTM_model(dropout = 0.1, optimizer ='rmsprop',n_hidden = 32):

    model2 = Sequential()
    # Configuring the parameters
    model2.add(LSTM(n_hidden, return_sequences=True, input_shape=(timesteps, input_dim)))
    # Adding a dropout layer
    model2.add(Dropout(dropout))
    model2.add(LSTM(n_hidden, input_shape=(timesteps, input_dim)))
    model2.add(Dropout(dropout))
    # Adding a dense output layer with sigmoid activation
    model2.add(Dense(n_classes, activation='sigmoid'))
    model2.compile(loss='categorical_crossentropy',optimizer=optimizer,metrics=['accuracy'])
    return model2
```

In [0]:

```
model_2layerhp = KerasClassifier(build_fn = create_2LSTM_model, epochs=30, batch_size=16)
```

In [49]:

```python
dropout = [0.7,0.8,0.9]
n_hidden = [32,40,52]


param_grid = dict( dropout = dropout, n_hidden = n_hidden )

model_2layer = GridSearchCV(estimator=model_2layerhp, param_grid=param_grid, n_jobs=-1, cv= 2)
model_2layer_result = model_2layer.fit(X_train,Y_train,
batch_size=batch_size,validation_data=(X_test, Y_test), epochs=epochs)

# summarize results
print("Best: %f using %s" % (model_2layer_result.best_score_, model_2layer_result.best_params_))
```

```
/usr/local/lib/python3.6/dist-packages/joblib/externals/loky/process_executor.py:706: UserWarning:
A worker stopped while some jobs were given to the executor. This can be caused by a too short wor
ker timeout or by a memory leak.
  "timeout or by a memory leak.", UserWarning
```

```
WARNING:tensorflow:Large dropout rate: 0.8 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate
instead of keep_prob. Please ensure that this is intended.
WARNING:tensorflow:Large dropout rate: 0.8 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate
instead of keep_prob. Please ensure that this is intended.
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [==============================] - 81s 11ms/step - loss: 1.3162 - acc: 0.4475 - val_loss
: 1.0442 - val_acc: 0.5830
Epoch 2/30
7352/7352 [==============================] - 79s 11ms/step - loss: 0.9885 - acc: 0.5909 - val_loss
: 0.8353 - val_acc: 0.6047
Epoch 3/30
7352/7352 [==============================] - 80s 11ms/step - loss: 0.8702 - acc: 0.6209 - val_loss
: 0.7861 - val_acc: 0.6715
Epoch 4/30
7352/7352 [==============================] - 84s 11ms/step - loss: 0.7738 - acc: 0.6376 - val_loss
: 0.7156 - val_acc: 0.6335
Epoch 5/30
7352/7352 [==============================] - 85s 12ms/step - loss: 0.7533 - acc: 0.6598 - val_loss
: 0.7167 - val_acc: 0.6922
Epoch 6/30
7352/7352 [==============================] - 85s 12ms/step - loss: 0.7067 - acc: 0.6827 - val_loss
: 0.7629 - val_acc: 0.6105
Epoch 7/30
7352/7352 [==============================] - 82s 11ms/step - loss: 0.6698 - acc: 0.6965 - val_loss
: 0.7611 - val_acc: 0.6189
Epoch 8/30
7352/7352 [==============================] - 77s 11ms/step - loss: 0.6267 - acc: 0.7270 - val_loss
: 0.5586 - val_acc: 0.7516
Epoch 9/30
7352/7352 [==============================] - 78s 11ms/step - loss: 0.5484 - acc: 0.7670 - val_loss
: 0.5541 - val_acc: 0.7587
Epoch 10/30
7352/7352 [==============================] - 78s 11ms/step - loss: 0.5388 - acc: 0.7731 - val_loss
: 0.5615 - val_acc: 0.7642
Epoch 11/30
7352/7352 [==============================] - 78s 11ms/step - loss: 0.4751 - acc: 0.7867 - val_loss
: 0.5973 - val_acc: 0.7391
Epoch 12/30
7352/7352 [==============================] - 78s 11ms/step - loss: 0.4685 - acc: 0.7874 - val_loss
: 0.5206 - val_acc: 0.7591
Epoch 13/30
7352/7352 [==============================] - 78s 11ms/step - loss: 0.4504 - acc: 0.7943 - val_loss
: 0.5123 - val_acc: 0.7645
Epoch 14/30
7352/7352 [==============================] - 81s 11ms/step - loss: 0.4234 - acc: 0.7973 - val_loss
: 0.5507 - val_acc: 0.7743
Epoch 15/30
7352/7352 [==============================] - 80s 11ms/step - loss: 0.4195 - acc: 0.8086 - val_loss
: 0.4845 - val_acc: 0.8208
Epoch 16/30
7352/7352 [==============================] - 80s 11ms/step - loss: 0.3976 - acc: 0.8500 - val_loss
: 0.5655 - val_acc: 0.8375
Epoch 17/30
```

```
7352/7352 [==============================] - 77s 11ms/step - loss: 0.3272 - acc: 0.9010 - val_loss
: 0.3720 - val_acc: 0.8931
Epoch 18/30
7352/7352 [==============================] - 79s 11ms/step - loss: 0.3248 - acc: 0.9116 - val_loss
: 0.8055 - val_acc: 0.8633
Epoch 19/30
7352/7352 [==============================] - 78s 11ms/step - loss: 0.2640 - acc: 0.9187 - val_loss
: 0.4923 - val_acc: 0.8918
Epoch 20/30
7352/7352 [==============================] - 78s 11ms/step - loss: 0.2732 - acc: 0.9248 - val_loss
: 0.5299 - val_acc: 0.9033
Epoch 21/30
7352/7352 [==============================] - 77s 10ms/step - loss: nan - acc: 0.7461 - val_loss: n
an - val_acc: 0.1683
Epoch 22/30
7352/7352 [==============================] - 79s 11ms/step - loss: nan - acc: 0.1668 - val_loss: n
an - val_acc: 0.1683
Epoch 23/30
7352/7352 [==============================] - 79s 11ms/step - loss: nan - acc: 0.1668 - val_loss: n
an - val_acc: 0.1683
Epoch 24/30
7352/7352 [==============================] - 79s 11ms/step - loss: nan - acc: 0.1668 - val_loss: n
an - val_acc: 0.1683
Epoch 25/30
7352/7352 [==============================] - 78s 11ms/step - loss: nan - acc: 0.1668 - val_loss: n
an - val_acc: 0.1683
Epoch 26/30
7352/7352 [==============================] - 79s 11ms/step - loss: nan - acc: 0.1668 - val_loss: n
an - val_acc: 0.1683
Epoch 27/30
7352/7352 [==============================] - 79s 11ms/step - loss: nan - acc: 0.1668 - val_loss: n
an - val_acc: 0.1683
Epoch 28/30
7352/7352 [==============================] - 78s 11ms/step - loss: nan - acc: 0.1668 - val_loss: n
an - val_acc: 0.1683
Epoch 29/30
7352/7352 [==============================] - 78s 11ms/step - loss: nan - acc: 0.1668 - val_loss: n
an - val_acc: 0.1683
Epoch 30/30
7352/7352 [==============================] - 79s 11ms/step - loss: nan - acc: 0.1668 - val_loss: n
an - val_acc: 0.1683
Best: 0.875952 using {'dropout': 0.8, 'n_hidden': 52}
```

In [0]:

```
model_2ltuned = Sequential()
# Configuring the parameters
model_2ltuned.add(LSTM(80, return_sequences=True, kernel_initializer='glorot_uniform'))
# Adding a dropout layer
model_2ltuned.add(Dropout(0.8))
model_2ltuned.add(BatchNormalization())
model_2ltuned.add(LSTM(80, kernel_initializer='glorot_uniform'))
model_2ltuned.add(Dropout(0.8))
# Adding a dense output layer with sigmoid activation
model_2ltuned.add(Dense(n_classes, activation='softmax'))
```

In [45]:

```
model_2ltuned.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

history_2ltuned = model_2ltuned.fit(X_train, Y_train,
          batch_size=64,
          epochs=15,
          verbose=1,
          validation_data=(X_test, Y_test))
score_2ltuned = model_2ltuned.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', score_2ltuned[0])
print('Test accuracy:', score_2ltuned[1])
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/15
7352/7352 [==============================] - 45s 6ms/step - loss: 1.2259 - acc: 0.4908 - val_loss:
0.9020 - val_acc: 0.5965
Epoch 2/15
```

```
7352/7352 [==============================] - 41s 6ms/step - loss: 0.7379 - acc: 0.6945 - val_loss:
1.1543 - val_acc: 0.6474
Epoch 3/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.6200 - acc: 0.7622 - val_loss:
0.7348 - val_acc: 0.7479
Epoch 4/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.4551 - acc: 0.8555 - val_loss:
0.8013 - val_acc: 0.7313
Epoch 5/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.3104 - acc: 0.9025 - val_loss:
0.5455 - val_acc: 0.8273
Epoch 6/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.3582 - acc: 0.8882 - val_loss:
0.5165 - val_acc: 0.8100
Epoch 7/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.3394 - acc: 0.8958 - val_loss:
1.3822 - val_acc: 0.6647
Epoch 8/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.3736 - acc: 0.8855 - val_loss:
0.3532 - val_acc: 0.8914
Epoch 9/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.2404 - acc: 0.9270 - val_loss:
0.3296 - val_acc: 0.8890
Epoch 10/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.2258 - acc: 0.9278 - val_loss:
0.3260 - val_acc: 0.8985
Epoch 11/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1851 - acc: 0.9389 - val_loss:
0.3058 - val_acc: 0.8958
Epoch 12/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1819 - acc: 0.9389 - val_loss:
0.3053 - val_acc: 0.9043
Epoch 13/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1730 - acc: 0.9404 - val_loss:
0.3329 - val_acc: 0.8935
Epoch 14/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1609 - acc: 0.9419 - val_loss:
0.4368 - val_acc: 0.8744
Epoch 15/15
7352/7352 [==============================] - 41s 6ms/step - loss: 0.1695 - acc: 0.9388 - val_loss:
0.3620 - val_acc: 0.8846
Test loss: 0.3620372748806869
Test accuracy: 0.8846284356973193
```

In [52]:

```
import mpld3
from mpld3 import plugins

score_2ltuned = model_2ltuned.evaluate( X_test,Y_test, verbose=0)

print('Test score:', score_2ltuned[0])
print('Test accuracy:', score_2ltuned[1])

ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

# list of epoch numbers
x_task1 = list(range(1,15+1))

vy_task1 = history_2ltuned.history['val_loss']
ty_task1 = history_2ltuned.history['loss']

print(vy_task1)
print(ty_task1)

plt_dynamic_plot(x_task1, vy_task1, ty_task1, ax)
mpld3.display()
```

```
Test score: 0.3620372748806869
Test accuracy: 0.8846284356973193
[0.9019750010008565, 1.1543389822735315, 0.7348045028940798, 0.8012816614500134,
0.545484897131015, 0.5164960334737968, 1.382238187509882, 0.3531754784450986 7,
0.3296320338605188 7, 0.3259659292084561, 0.3057684504400075, 0.30525806530127597,
0.3328627586570912, 0.4367965132192891, 0.3620372790566273]
[1.225917573008885, 0.737897875129982, 0.6200193295476226, 0.4551241869052164 4,
0.3103685539227964, 0.3581659166026816, 0.33938817632859886, 0.3735972273531365
0.3103683339227964, 0.3381659166026816, 0.3395881763285988, 0.3735972273351363,
0.2404014403334899, 0.22583643132182277, 0.18512259257533734, 0.18192267088752576,
0.1730467584369232, 0.16085848828394603, 0.16950491364230527]
```

Out[52]:

In [70]:

```
confusion_matrix(Y_test, model_2ltuned.predict(X_test))
```

Out[70]:

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
True						
LAYING	510	0	0	0	0	27
SITTING	5	420	61	0	0	5
STANDING	0	116	415	1	0	0
WALKING	0	0	0	478	6	12
WALKING_DOWNSTAIRS	0	0	0	1	419	0
WALKING_UPSTAIRS	0	0	0	22	3	446

In [0]:

```
from keras.layers.normalization import BatchNormalization

model3_BN = Sequential()
model3_BN.add(LSTM(32, return_sequences=True,kernel_initializer='glorot_uniform'))
model3_BN.add(Dropout(0.5))
model3_BN.add(BatchNormalization())
model3_BN.add(LSTM(64,kernel_initializer='glorot_uniform'))
model3_BN.add(Dropout(0.5))
model3_BN.add(Dense(n_classes, activation='softmax'))
```

In [32]:

```
model3_BN.compile(loss='categorical_crossentropy',optimizer='Adam', metrics=['accuracy'])
history_model3 = model3_BN.fit(X_train,Y_train,batch_size=64,validation_data=(X_test, Y_test),epoch
s=15)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/15
7352/7352 [==============================] - 30s 4ms/step - loss: 1.0205 - acc: 0.5888 - val_loss:
0.8543 - val_acc: 0.6620
Epoch 2/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.4121 - acc: 0.8630 - val_loss:
0.6059 - val_acc: 0.7889
Epoch 3/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.2570 - acc: 0.9165 - val_loss:
0.3788 - val_acc: 0.8758
Epoch 4/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1973 - acc: 0.9328 - val_loss:
0.3123 - val_acc: 0.8931
Epoch 5/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1766 - acc: 0.9346 - val_loss:
0.6621 - val_acc: 0.8222
Epoch 6/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1907 - acc: 0.9327 - val_loss:
0.3763 - val_acc: 0.8795
Epoch 7/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1641 - acc: 0.9431 - val_loss:
0.4959 - val_acc: 0.8582
Epoch 8/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1442 - acc: 0.9461 - val_loss:
0.4006 - val_acc: 0.8860
Epoch 9/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1665 - acc: 0.9369 - val_loss:
0.3461 - val_acc: 0.9060
Epoch 10/15
Epoch 10/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1579 - acc: 0.9410 - val_loss:
0.2911 - val_acc: 0.9030
Epoch 11/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1537 - acc: 0.9434 - val_loss:
0.2785 - val_acc: 0.9070
Epoch 12/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1362 - acc: 0.9437 - val_loss:
0.4114 - val_acc: 0.8958
Epoch 13/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1418 - acc: 0.9446 - val_loss:
0.4063 - val_acc: 0.9043
Epoch 14/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1252 - acc: 0.9479 - val_loss:
0.3546 - val_acc: 0.9118
Epoch 15/15
7352/7352 [==============================] - 27s 4ms/step - loss: 0.1479 - acc: 0.9449 - val_loss:
0.2802 - val_acc: 0.9213
```

In [33]:

```
score_3BN = model3_BN.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', score_3BN[0])
print('Test accuracy:', score_3BN[1])
```

```
Test loss: 0.2801610008103335
Test accuracy: 0.9212758737699356
```

In [245]:

```
score_3BN = model3_BN.evaluate( X_test,Y_test, verbose=0)

print('Test score:', score_3BN[0])
print('Test accuracy:', score_3BN[1])

ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

# list of epoch numbers
x_task1 = list(range(1,15+1))

vy_task1 = history_model3.history['val_loss']
ty_task1 = history_model3.history['loss']

print(vy_task1)
print(ty_task1)

plt_dynamic_plot(x_task1, vy_task1, ty_task1, ax)
mpld3.display()
```

```
Test score: 0.28016100497819557
Test accuracy: 0.9212758737699356
[0.8543283758140799, 0.6059476831604919, 0.37879919630867526, 0.312258508172699,
0.6621226031931441, 0.37627733805544566, 0.49591911638005154, 0.4006004456980081,
0.3460746547072136, 0.2911300434518263, 0.2784883492619297, 0.41140616562724375,
0.4062874037279815, 0.35461832352162254, 0.28016100111103026]
[1.0204617922012902, 0.41205343347898116, 0.256961944151236, 0.19729011605561622,
0.17664646338651177, 0.19072281197118812, 0.16412355951048735, 0.144158141592252,
0.16654289470857842, 0.15793967101078427, 0.1536518697575205, 0.1362282255982678,
0.1418059280018552, 0.1252058780994327, 0.14789696409892983]
```

Out[245]:

In [246]:

```
confusion_matrix(Y_test, model3_BN.predict(X_test))
```

Out[246]:

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
True						

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
True						
LAYING	510	0	0	0	0	27
SITTING	5	397	74	0	0	15
STANDING	0	83	449	0	0	0
WALKING	0	1	7	482	3	3
WALKING_DOWNSTAIRS	0	0	0	0	420	0
WALKING_UPSTAIRS	0	1	0	12	1	457

Using CNN Architecture

In [0]:

```
from keras.layers.convolutional import Conv1D ,MaxPooling1D

model4_BN = Sequential()
model4_BN.add(Conv1D(32,3,activation='relu',padding='valid',input_shape=(timesteps, input_dim)))
model4_BN.add(BatchNormalization())
model4_BN.add(MaxPooling1D(pool_size=2))
model4_BN.add(Conv1D(48,3, padding='valid', activation='relu'))
model4_BN.add(BatchNormalization())
model4_BN.add(MaxPooling1D(pool_size=2))
model4_BN.add(Conv1D(64,3, padding='valid', activation='relu'))
model4_BN.add(BatchNormalization())
model4_BN.add(MaxPooling1D(pool_size=2))
model4_BN.add(Conv1D(128,5,padding='valid',activation='relu'))
model4_BN.add(MaxPooling1D(pool_size=4))
model4_BN.add(Flatten())
model4_BN.add(Dense(16, activation='relu'))
model4_BN.add(Dense(n_classes, activation='softmax'))
```

In [242]:

```
model4_BN.compile(loss='categorical_crossentropy',optimizer='Adam', metrics=['accuracy'])
history_model4 = model4_BN.fit(X_train,Y_train,batch_size=16,validation_data=(X_test, Y_test),epoch
s=15)
```

```
Train on 7352 samples, validate on 2947 samples
Epoch 1/15
7352/7352 [==============================] - 30s 4ms/step - loss: 0.2922 - acc: 0.8840 - val_loss:
0.2033 - val_acc: 0.9169
Epoch 2/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.1407 - acc: 0.9372 - val_loss:
0.1962 - val_acc: 0.9311
Epoch 3/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.1172 - acc: 0.9502 - val_loss:
0.2612 - val_acc: 0.9311
Epoch 4/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.1174 - acc: 0.9509 - val_loss:
0.2132 - val_acc: 0.9301
Epoch 5/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.1162 - acc: 0.9524 - val_loss:
0.2697 - val_acc: 0.9243
Epoch 6/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.1093 - acc: 0.9533 - val_loss:
0.1797 - val_acc: 0.9369
Epoch 7/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.0930 - acc: 0.9621 - val_loss:
0.4637 - val_acc: 0.9114
Epoch 8/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.0853 - acc: 0.9631 - val_loss:
0.2513 - val_acc: 0.9315
Epoch 9/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.0861 - acc: 0.9646 - val_loss:
0.2662 - val_acc: 0.9260
Epoch 10/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.0917 - acc: 0.9623 - val_loss:
0.2033 - val_acc: 0.9382
Epoch 11/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.0942 - acc: 0.9621 - val_loss:
0.3494 - val_acc: 0.9199
```

```
Epoch 12/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.0791 - acc: 0.9682 - val_loss:
0.3407 - val_acc: 0.9348
Epoch 13/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.0731 - acc: 0.9689 - val_loss:
0.2567 - val_acc: 0.9498
Epoch 14/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.0757 - acc: 0.9710 - val_loss:
0.3158 - val_acc: 0.9338
Epoch 15/15
7352/7352 [==============================] - 8s 1ms/step - loss: 0.0762 - acc: 0.9710 - val_loss:
0.1680 - val_acc: 0.9454
```

In [243]:

```
score_4BN = model4_BN.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', score_4BN[0])
print('Test accuracy:', score_4BN[1])
```

```
Test loss: 0.16795373193306978
Test accuracy: 0.9453681710213777
```

In [244]:

```
score_4BN = model4_BN.evaluate( X_test,Y_test, verbose=0)

print('Test score:', score_4BN[0])
print('Test accuracy:', score_4BN[1])

ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

# list of epoch numbers
x_task1 = list(range(1,15+1))

vy_task1 = history_model4.history['val_loss']
ty_task1 = history_model4.history['loss']

print(vy_task1)
print(ty_task1)

plt_dynamic_plot(x_task1, vy_task1, ty_task1, ax)
mpld3.display()
```

```
Test score: 0.16795373193306978
Test accuracy: 0.9453681710213777
[0.20325752102651423, 0.1961597157072711, 0.261181096223429, 0.21315644821775173,
0.26972441978058664, 0.17966743743169594, 0.46369743079874226, 0.25125248902523556,
0.2662349990351221, 0.20327006207220238, 0.3494389985284612, 0.340731804485431,
0.2567077164214705, 0.31580340727535877, 0.1679537306584709]
[0.2922106668194989, 0.1406654687882102, 0.11721167276041852, 0.11739735754122664,
0.11624899146758158, 0.10925040768504853, 0.09297219137807469, 0.08526058638243313,
0.0860902275434078, 0.09170510055125114, 0.09417734731288203, 0.07913021292558732,
0.07308445791184387, 0.07572169402036656, 0.076246753867482]
```

Out[244]:

In [247]:

```
confusion_matrix(Y_test, model4_BN.predict(X_test))
```

Out[247]:

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
True						
LAYING	537	0	0	0	0	0
SITTING	7	404	76	0	0	4
STANDING	0	34	498	0	0	0
WALKING	0	0	0	480	14	2

Pred	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
WALKING	0	0	0	[illegible]	11	2
WALKING_DOWNSTAIRS	0	0	0	0	420	0
True						
WALKING_UPSTAIRS	0	1	0	1	22	447

Conclusion

In [251]:

```
from prettytable import PrettyTable

x = PrettyTable()
x.field_names = ["Architecture", "Activation",  "Test score", "Test Accuracy"]


x.add_row(["1 Layer LSTM + Dropouts", "Sigmoid", 0.308, 0.9097 ])
x.add_row(["1 Layer LSTM + Dropouts", "Sigmoid", 0.388, 0.9195])
x.add_row(["2 Layer LSTM + Large Dropouts", "Sigmoid", 0.294, 0.9121])
x.add_row(["2 Layer LSTM + Dropouts + BN", "Sigmoid", 0.2801, 0.9212])
x.add_row(["4 Layer CNN + Maxpool + BN", "Relu",0.167, 0.9453])

print(x)
```

```
+-------------------------------+------------+------------+---------------+
|          Architecture         | Activation | Test score | Test Accuracy |
+-------------------------------+------------+------------+---------------+
|    1 Layer LSTM + Dropouts    |  Sigmoid   |   0.308    |     0.9097    |
|    1 Layer LSTM + Dropouts    |  Sigmoid   |   0.388    |     0.9195    |
| 2 Layer LSTM + Large Dropouts |  Sigmoid   |   0.294    |     0.9121    |
|  2 Layer LSTM + Dropouts + BN |  Sigmoid   |   0.2801   |     0.9212    |
|   4 Layer CNN + Maxpool + BN  |    Relu    |   0.167    |     0.9453    |
+-------------------------------+------------+------------+---------------+
```