

# HumanActivityRecognition

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

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

## How data was recorded

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

prefix 't' in those metrics denotes time.

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

## Feature names

1. These sensor signals are preprocessed by applying noise filters and then sampled in fixed-width windows(sliding windows) of 2.56 seconds each with 50% overlap. ie., each window has 128 readings.
2. From Each window, a feature vector was obtained by calculating variables from the time and frequency domain.

In our dataset, each datapoint represents a window with different readings

3. The acceleration signal was separated into Body and Gravity acceleration signals(**tBodyAcc-XYZ** and **tGravityAcc-XYZ**) using some low pass filter with corner frequency of 0.3Hz.
4. After that, the body linear acceleration and angular velocity were derived in time to obtain *jerk signals* (**tBodyAccJerk-XYZ** and **tBodyGyroJerk-XYZ**).
5. The magnitude of these 3-dimensional signals were calculated using the Euclidian norm. This magnitudes are represented as features with names like **tBodyAccMag**, **tGravityAccMag**, **tBodyAccJerkMag**, **tBodyGyroMag** and **tBodyGyroJerkMag**.
6. Finally, We've got frequency domain signals from some of the available signals by applying a FFT (Fast Fourier Transform). These signals obtained were labeled with **prefix 'f'** just like original signals with **prefix 't'**. These signals are labeled as **fBodyAcc-XYZ**, **fBodyGyroMag** etc.,
7. These are the signals that we got so far.

- tBodyAcc-XYZ
- tGravityAcc-XYZ
- tBodyAccJerk-XYZ
- tBodyGyro-XYZ
- tBodyGyroJerk-XYZ
- tBodyAccMag
- tGravityAccMag
- tBodyAccJerkMag
- tBodyGyroMag
- tBodyGyroJerkMag
- fBodyAcc-XYZ
- fBodyAccJerk-XYZ
- fBodyGyro-XYZ
- fBodyAccMag
- fBodyAccJerkMag
- fBodyGyroMag
- fBodyGyroJerkMag

8. We can estimate some set of variables from the above signals. ie., We will estimate the following properties on each and every signal that we recorded so far.

- **mean()**: Mean value
- **std()**: Standard deviation
- **mad()**: Median absolute deviation
- **max()**: Largest value in array

- **max()**: Largest value in array
- **min()**: Smallest value in array
- **sma()**: Signal magnitude area
- **energy()**: Energy measure. Sum of the squares divided by the number of values.
- **iqr()**: Interquartile range
- **entropy()**: Signal entropy
- **arCoeff()**: Autorregresion coefficients with Burg order equal to 4
- **correlation()**: correlation coefficient between two signals
- **maxInds()**: index of the frequency component with largest magnitude
- **meanFreq()**: Weighted average of the frequency components to obtain a mean frequency
- **skewness()**: skewness of the frequency domain signal
- **kurtosis()**: kurtosis of the frequency domain signal
- **bandsEnergy()**: Energy of a frequency interval within the 64 bins of the FFT of each window.
- **angle()**: Angle between to vectors.

9. We can obtain some other vectors by taking the average of signals in a single window sample. These are used on the angle() variable`

- gravityMean
- tBodyAccMean
- tBodyAccJerkMean
- tBodyGyroMean
- tBodyGyroJerkMean

## Y\_Labels(Encoded)

- In the dataset, Y\_labels are represented as numbers from 1 to 6 as their identifiers.
  - WALKING as **1**
  - WALKING\_UPSTAIRS as **2**
  - WALKING\_DOWNSTAIRS as **3**
  - SITTING as **4**
  - STANDING as **5**
  - LAYING as **6**

## Train and test data were saperated

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

## Data

- All the data is present in 'UCI\_HAR\_dataset/' folder in present working directory.
  - Feature names are present in 'UCI\_HAR\_dataset/features.txt'
  - **Train Data**
    - 'UCI\_HAR\_dataset/train/X\_train.txt'
    - 'UCI\_HAR\_dataset/train/subject\_train.txt'
    - 'UCI\_HAR\_dataset/train/y\_train.txt'
  - **Test Data**
    - 'UCI\_HAR\_dataset/test/X\_test.txt'
    - 'UCI\_HAR\_dataset/test/subject\_test.txt'
    - 'UCI\_HAR\_dataset/test/y\_test.txt'

## Data Size :

27 MB

## Quick overview of the dataset :

- Accelerometer and Gyroscope readings are taken from 30 volunteers(referred as subjects) while performing the following 6 Activities.
  1. Walking
  2. WalkingUpstairs
  3. WalkingDownstairs

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

## Problem Framework

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

## Mounting Drive

In [0]:

```
!kill -9 -1
```

In [1]:

```
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%b&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&response\\_type=code](https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%b&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&response_type=code)

Enter your authorization code:

.....

Mounted at /content/drive

In [2]:

```
!pwd
!ls
```

```
/content
drive sample_data
```

In [3]:

```
import os
PATH = os.getcwd()
print(PATH)
```

```
/content
```

In [4]:

```
data_path = PATH + '/drive/My Drive/AAIC/Case Studies/Human Activity Recognition/'
data_path
```

Out[4]:

```
'/content/drive/My Drive/AAIC/Case Studies/Human Activity Recognition/'
```

## Problem Statement

- Given a new datapoint we have to predict the Activity

In [0]:

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

In [0]:

```
# get the features from the file features.txt
features = list()
with open('UCI_HAR_Dataset/features.txt') as f:
    features = [line.split()[1] for line in f.readlines()]
print('No of Features: {}'.format(len(features)))
```

No of Features: 561

## Obtain the train data

In [0]:

```
# get the data from txt files to pandas dataffame
X_train = pd.read_csv('UCI_HAR_dataset/train/X_train.txt', delim_whitespace=True, header=None,
names=features)

# add subject column to the dataframe
X_train['subject'] = pd.read_csv('UCI_HAR_dataset/train/subject_train.txt', header=None,
squeeze=True)

y_train = pd.read_csv('UCI_HAR_dataset/train/y_train.txt', names=['Activity'], squeeze=True)
y_train_labels = y_train.map({1: 'WALKING', 2:'WALKING_UPSTAIRS', 3:'WALKING_DOWNSTAIRS', \
4:'SITTING', 5:'STANDING',6:'LAYING'})

# put all columns in a single dataframe
train = X_train
train['Activity'] = y_train
train['ActivityName'] = y_train_labels
train.sample()

D:\installed\Anaconda3\lib\site-packages\pandas\io\parsers.py:678: UserWarning: Duplicate names specified. This will raise an error in the future.
    return _read(filepath_or_buffer, kwds)
```

Out[0]:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc-mad()-Y	tBodyAcc-mad()-Z	tBodyAcc-max()-X	...	angle(tB
6015	0.2797	-0.004397	-0.10952	0.359081	0.119909	-0.177541	0.337963	0.066883	-0.221876	0.474093	...	

1 rows x 564 columns

In [0]:

```
train.shape
```

Out[0]:

(7352, 564)

## Obtain the test data

In [0]:

```
# get the data from txt files to pandas dataframe
X_test = pd.read_csv('UCI_HAR_dataset/test/X_test.txt', delim_whitespace=True, header=None, names=features)

# add subject column to the dataframe
X_test['subject'] = pd.read_csv('UCI_HAR_dataset/test/subject_test.txt', header=None, squeeze=True)

# get y labels from the txt file
y_test = pd.read_csv('UCI_HAR_dataset/test/y_test.txt', names=['Activity'], squeeze=True)
y_test_labels = y_test.map({1: 'WALKING', 2:'WALKING_UPSTAIRS',3:'WALKING_DOWNSTAIRS', \
                           4:'SITTING', 5:'STANDING',6:'LAYING'})

# put all columns in a single dataframe
test = X_test
test['Activity'] = y_test
test['ActivityName'] = y_test_labels
test.sample()

D:\installed\Anaconda3\lib\site-packages\pandas\io\parsers.py:678: UserWarning: Duplicate names specified. This will raise an error in the future.
    return _read(filepath_or_buffer, kwds)
```

Out[0]:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc-mad()-Y	tBodyAcc-mad()-Z	tBodyAcc-max()-X	tBodyAcc-max()-Y	tBodyAcc-max()-Z	...	angle(tB
2261	0.279196	-0.018261	-0.103376	-0.996955	-0.982959	-0.988239	-0.9972	-0.982509	-0.986964	-0.940634	...	...	...	...

1 rows × 564 columns

In [0]:

```
test.shape
```

Out[0]:

(2947, 564)

## Data Cleaning

### 1. Check for Duplicates

In [0]:

```
print('No of duplicates in train: {}'.format(sum(train.duplicated())))
print('No of duplicates in test : {}'.format(sum(test.duplicated())))
```

No of duplicates in train: 0  
No of duplicates in test : 0

### 2. Checking for NaN/null values

In [0]:

```
print('We have {} NaN/Null values in train'.format(train.isnull().values.sum()))
print('We have {} NaN/Null values in test'.format(test.isnull().values.sum()))
```

We have 0 NaN/Null values in train  
We have 0 NaN/Null values in test

### 3. Check for data imbalance

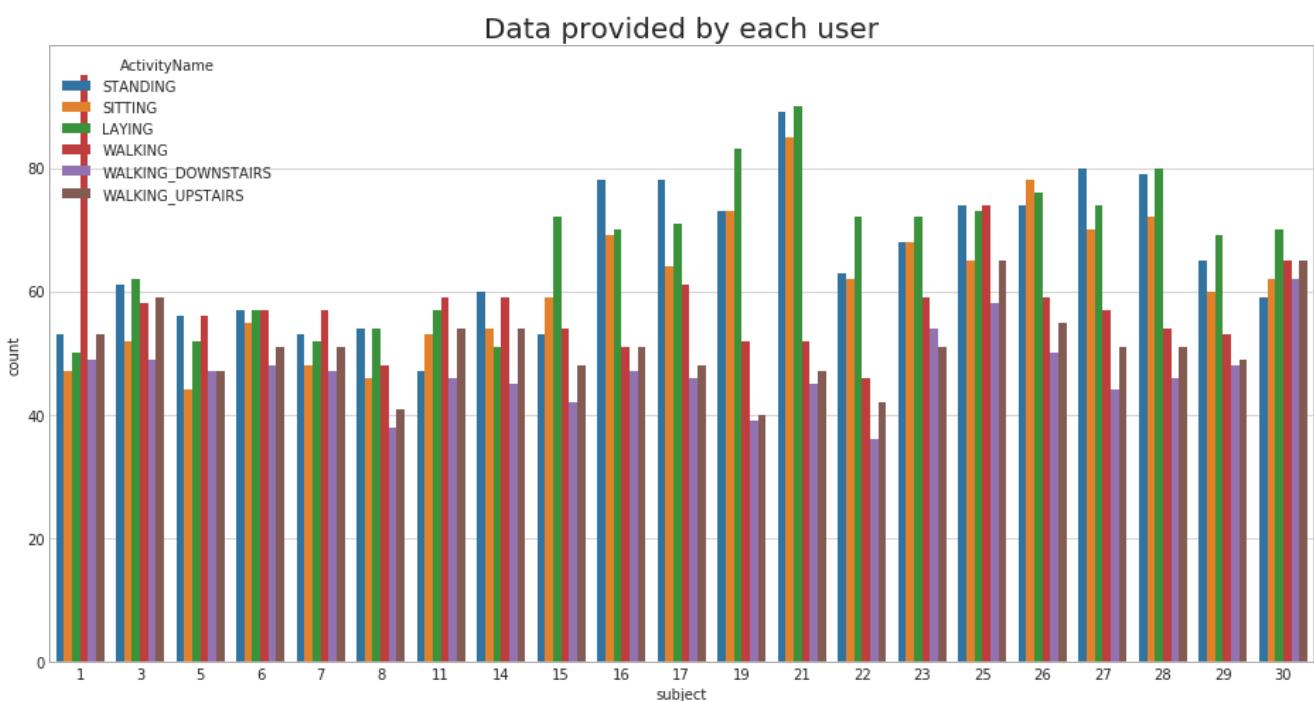
In [0]:

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.set_style('whitegrid')
plt.rcParams['font.family'] = 'Dejavu Sans'
```

In [0]:

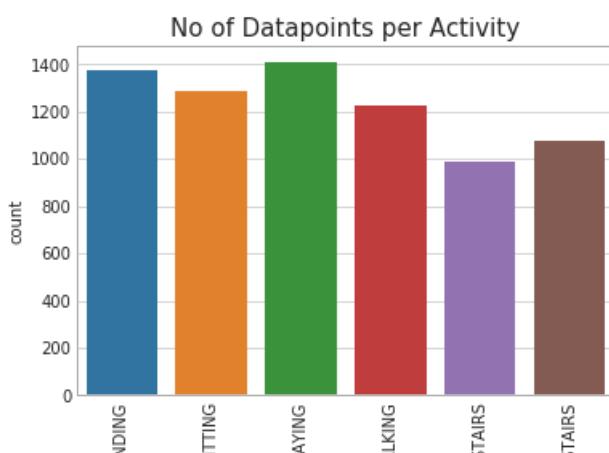
```
plt.figure(figsize=(16,8))
plt.title('Data provided by each user', fontsize=20)
sns.countplot(x='subject', hue='ActivityName', data = train)
plt.show()
```



We have got almost same number of reading from all the subjects

In [0]:

```
plt.title('No of Datapoints per Activity', fontsize=15)
sns.countplot(train.ActivityName)
plt.xticks(rotation=90)
plt.show()
```





## Observation

Our data is well balanced (almost)

## 4. Changing feature names

In [0]:

```
columns = train.columns

# Removing '()' from column names
columns = columns.str.replace('[( )]', '')
columns = columns.str.replace('[-]', '')
columns = columns.str.replace('[,]', '')

train.columns = columns
test.columns = columns

test.columns
```

Out [0]:

```
Index(['tBodyAccmeanX', 'tBodyAccmeanY', 'tBodyAccmeanZ', 'tBodyAccstdX',
       'tBodyAccstdY', 'tBodyAccstdZ', 'tBodyAccmadX', 'tBodyAccmadY',
       'tBodyAccmadZ', 'tBodyAccmaxX',
       ...
       'angletBodyAccMeangravity', 'angletBodyAccJerkMeangravityMean',
       'angletBodyGyroMeangravityMean', 'angletBodyGyroJerkMeangravityMean',
       'angleXgravityMean', 'angleYgravityMean', 'angleZgravityMean',
       'subject', 'Activity', 'ActivityName'],
      dtype='object', length=564)
```

## 5. Save this dataframe in a csv files

In [0]:

```
train.to_csv('UCI_HAR_Dataset/csv_files/train.csv', index=False)
test.to_csv('UCI_HAR_Dataset/csv_files/test.csv', index=False)
```

## Exploratory Data Analysis

*"Without domain knowledge EDA has no meaning, without EDA a problem has no soul."*

### 1. Featurizing Engineering from Domain Knowledge

- **Static and Dynamic Activities**

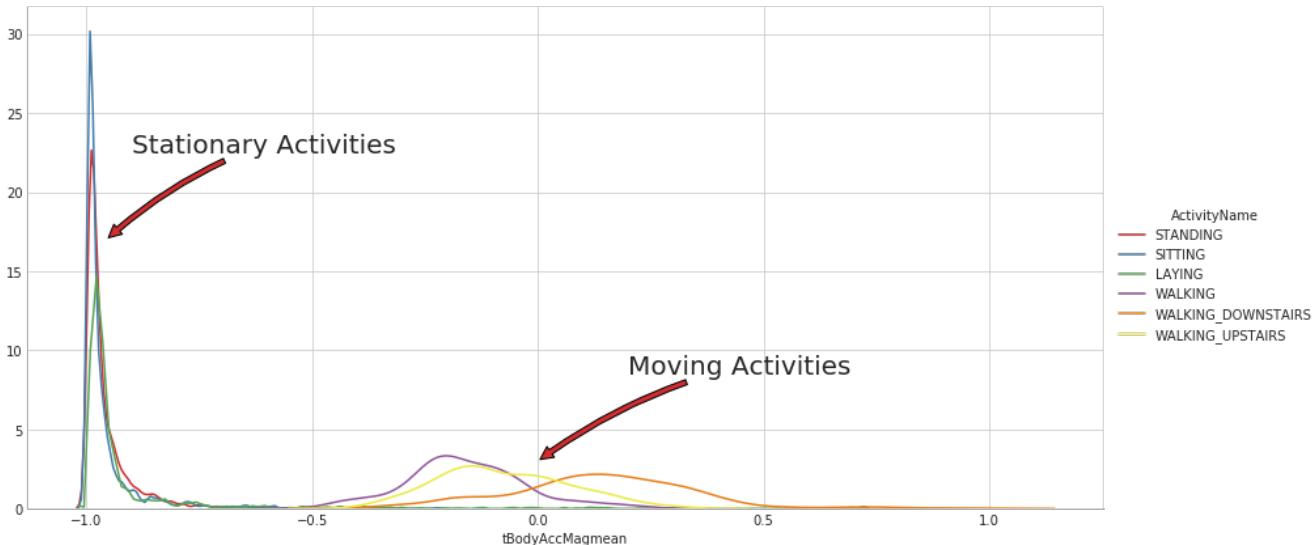
- In static activities (sit, stand, lie down) motion information will not be very useful.
- In the dynamic activities (Walking, WalkingUpstairs,WalkingDownstairs) motion info will be significant.

### 2. Stationary and Moving activities are completely different

In [0]:

```
sns.set_palette("Set1", desat=0.80)
facetgrid = sns.FacetGrid(train, hue='ActivityName', size=6, aspect=2)
facetgrid.map(sns.distplot, 'tBodyAccMagmean', hist=False) \
    .add_legend()
plt.annotate("Stationary Activities", xy=(-0.956, 17), xytext=(-0.9, 23), size=20, \
    va='center', ha='left', \
    arrowprops=dict(arrowstyle="simple", connectionstyle="arc3,rad=0.1"))

plt.annotate("Moving Activities", xy=(0, 3), xytext=(0.2, 9), size=20, \
    va='center', ha='left', \
    arrowprops=dict(arrowstyle="simple", connectionstyle="arc3,rad=0.1"))
plt.show()
```



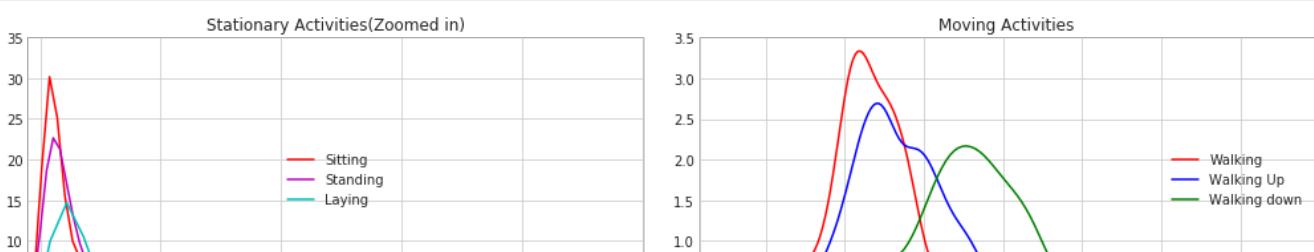
In [0]:

```
# for plotting purposes taking datapoints of each activity to a different dataframe
df1 = train[train['Activity']==1]
df2 = train[train['Activity']==2]
df3 = train[train['Activity']==3]
df4 = train[train['Activity']==4]
df5 = train[train['Activity']==5]
df6 = train[train['Activity']==6]

plt.figure(figsize=(14,7))
plt.subplot(2,2,1)
plt.title('Stationary Activities (Zoomed in)')
sns.distplot(df4['tBodyAccMagmean'], color = 'r', hist = False, label = 'Sitting')
sns.distplot(df5['tBodyAccMagmean'], color = 'm', hist = False, label = 'Standing')
sns.distplot(df6['tBodyAccMagmean'], color = 'c', hist = False, label = 'Laying')
plt.axis([-1.01, -0.5, 0, 35])
plt.legend(loc='center')

plt.subplot(2,2,2)
plt.title('Moving Activities')
sns.distplot(df1['tBodyAccMagmean'], color = 'red', hist = False, label = 'Walking')
sns.distplot(df2['tBodyAccMagmean'], color = 'blue', hist = False, label = 'Walking Up')
sns.distplot(df3['tBodyAccMagmean'], color = 'green', hist = False, label = 'Walking down')
plt.legend(loc='center right')

plt.tight_layout()
plt.show()
```

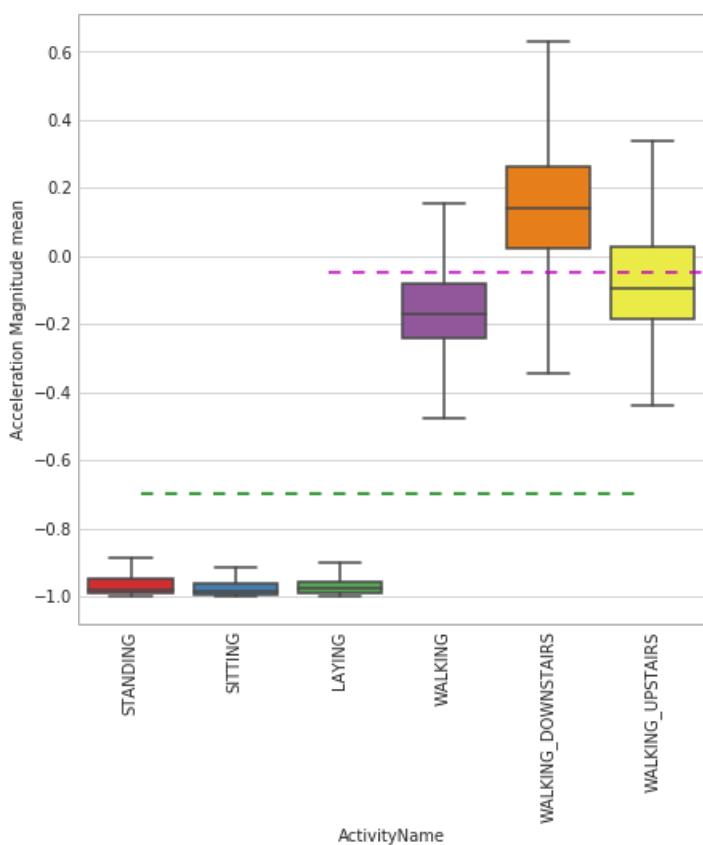




### 3. Magnitude of an acceleration can separate it well

In [0]:

```
plt.figure(figsize=(7,7))
sns.boxplot(x='ActivityName', y='tBodyAccMagmean', data=train, showfliers=False, saturation=1)
plt.ylabel('Acceleration Magnitude mean')
plt.axhline(y=-0.7, xmin=0.1, xmax=0.9, dashes=(5,5), c='g')
plt.axhline(y=-0.05, xmin=0.4, dashes=(5,5), c='m')
plt.xticks(rotation=90)
plt.show()
```



#### Observations:

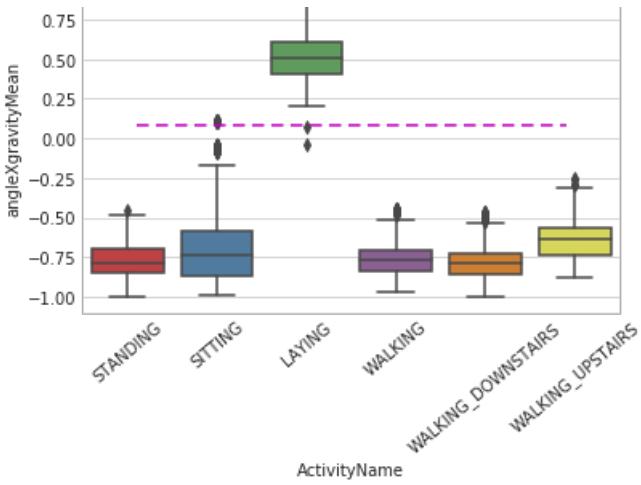
- If tAccMean is < -0.8 then the Activities are either Standing or Sitting or Laying.
- If tAccMean is > -0.6 then the Activities are either Walking or WalkingDownstairs or WalkingUpstairs.
- If tAccMean > 0.0 then the Activity is WalkingDownstairs.
- We can classify 75% the Acitivity labels with some errors.

### 4. Position of GravityAccelerationComponents also matters

In [0]:

```
sns.boxplot(x='ActivityName', y='angleXgravityMean', data=train)
plt.axhline(y=0.08, xmin=0.1, xmax=0.9, c='m', dashes=(5,3))
plt.title('Angle between X-axis and Gravity_mean', fontsize=15)
plt.xticks(rotation = 40)
plt.show()
```



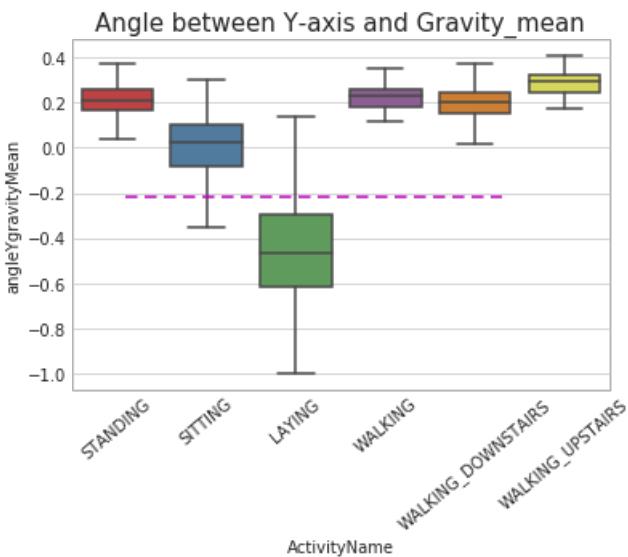


### Observations:

- If  $\text{angleX.gravityMean} > 0$  then Activity is Laying.
- We can classify all datapoints belonging to Laying activity with just a single if else statement.

In [0]:

```
sns.boxplot(x='ActivityName', y='angleYgravityMean', data = train, showfliers=False)
plt.title('Angle between Y-axis and Gravity_mean', fontsize=15)
plt.xticks(rotation = 40)
plt.axhline(y=-0.22, xmin=0.1, xmax=0.8, dashes=(5,3), c='m')
plt.show()
```



## Apply t-sne on the data

In [0]:

```
import numpy as np
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt
import seaborn as sns
```

In [0]:

```
# performs t-sne with different perplexity values and their repective plots..
```

```

def perform_tsne(X_data, y_data, perplexities, n_iter=1000, img_name_prefix='t-sne'):

    for index,perplexity in enumerate(perplexities):
        # perform t-sne
        print('\nperforming tsne with perplexity {} and with {} iterations at max'.format(perplexity, n_iter))
        X_reduced = TSNE(verbose=2, perplexity=perplexity).fit_transform(X_data)
        print('Done..')

        # prepare the data for seaborn
        print('Creating plot for this t-sne visualization..')
        df = pd.DataFrame({'x':X_reduced[:,0], 'y':X_reduced[:,1] , 'label':y_data})

        # draw the plot in appropriate place in the grid
        sns.lmplot(data=df, x='x', y='y', hue='label', fit_reg=False, size=8,\ 
                    palette="Set1", markers=['^','v','s','o', '1','2'])
        plt.title("perplexity : {} and max_iter : {}".format(perplexity, n_iter))
        img_name = img_name_prefix + '_perp_{}_iter_{}.png'.format(perplexity, n_iter)
        print('saving this plot as image in present working directory..')
        plt.savefig(img_name)
        plt.show()
        print('Done')

```

In [0]:

```

X_pre_tsne = train.drop(['subject', 'Activity','ActivityName'], axis=1)
y_pre_tsne = train['ActivityName']
perform_tsne(X_data = X_pre_tsne,y_data=y_pre_tsne, perplexities =[2,5,10,20,50])

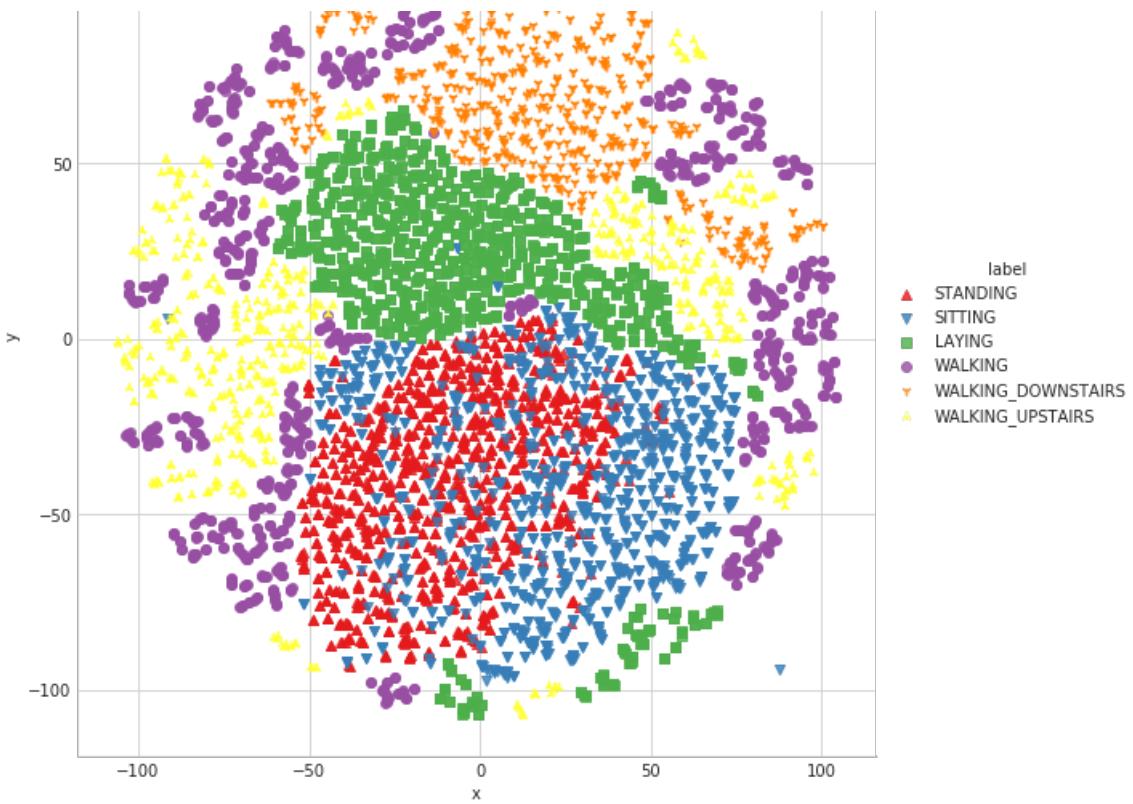
```

```

performing tsne with perplexity 2 and with 1000 iterations at max
[t-SNE] Computing 7 nearest neighbors...
[t-SNE] Indexed 7352 samples in 0.426s...
[t-SNE] Computed neighbors for 7352 samples in 72.001s...
[t-SNE] Computed conditional probabilities for sample 1000 / 7352
[t-SNE] Computed conditional probabilities for sample 2000 / 7352
[t-SNE] Computed conditional probabilities for sample 3000 / 7352
[t-SNE] Computed conditional probabilities for sample 4000 / 7352
[t-SNE] Computed conditional probabilities for sample 5000 / 7352
[t-SNE] Computed conditional probabilities for sample 6000 / 7352
[t-SNE] Computed conditional probabilities for sample 7000 / 7352
[t-SNE] Computed conditional probabilities for sample 7352 / 7352
[t-SNE] Mean sigma: 0.635855
[t-SNE] Computed conditional probabilities in 0.071s
[t-SNE] Iteration 50: error = 124.8017578, gradient norm = 0.0253939 (50 iterations in 16.625s)
[t-SNE] Iteration 100: error = 107.2019501, gradient norm = 0.0284782 (50 iterations in 9.735s)
[t-SNE] Iteration 150: error = 100.9872894, gradient norm = 0.0185151 (50 iterations in 5.346s)
[t-SNE] Iteration 200: error = 97.6054382, gradient norm = 0.0142084 (50 iterations in 7.013s)
[t-SNE] Iteration 250: error = 95.3084183, gradient norm = 0.0132592 (50 iterations in 5.703s)
[t-SNE] KL divergence after 250 iterations with early exaggeration: 95.308418
[t-SNE] Iteration 300: error = 4.1209540, gradient norm = 0.0015668 (50 iterations in 7.156s)
[t-SNE] Iteration 350: error = 3.2113254, gradient norm = 0.0009953 (50 iterations in 8.022s)
[t-SNE] Iteration 400: error = 2.7819963, gradient norm = 0.0007203 (50 iterations in 9.419s)
[t-SNE] Iteration 450: error = 2.5178111, gradient norm = 0.0005655 (50 iterations in 9.370s)
[t-SNE] Iteration 500: error = 2.3341548, gradient norm = 0.0004804 (50 iterations in 7.681s)
[t-SNE] Iteration 550: error = 2.1961622, gradient norm = 0.0004183 (50 iterations in 7.097s)
[t-SNE] Iteration 600: error = 2.0867445, gradient norm = 0.0003664 (50 iterations in 9.274s)
[t-SNE] Iteration 650: error = 1.9967778, gradient norm = 0.0003279 (50 iterations in 7.697s)
[t-SNE] Iteration 700: error = 1.9210005, gradient norm = 0.0002984 (50 iterations in 8.174s)
[t-SNE] Iteration 750: error = 1.8558111, gradient norm = 0.0002776 (50 iterations in 9.747s)
[t-SNE] Iteration 800: error = 1.7989457, gradient norm = 0.0002569 (50 iterations in 8.687s)
[t-SNE] Iteration 850: error = 1.7490212, gradient norm = 0.0002394 (50 iterations in 8.407s)
[t-SNE] Iteration 900: error = 1.7043383, gradient norm = 0.0002224 (50 iterations in 8.351s)
[t-SNE] Iteration 950: error = 1.6641431, gradient norm = 0.0002098 (50 iterations in 7.841s)
[t-SNE] Iteration 1000: error = 1.6279151, gradient norm = 0.0001989 (50 iterations in 5.623s)
[t-SNE] Error after 1000 iterations: 1.627915
Done..
Creating plot for this t-sne visualization..
saving this plot as image in present working directory..

```





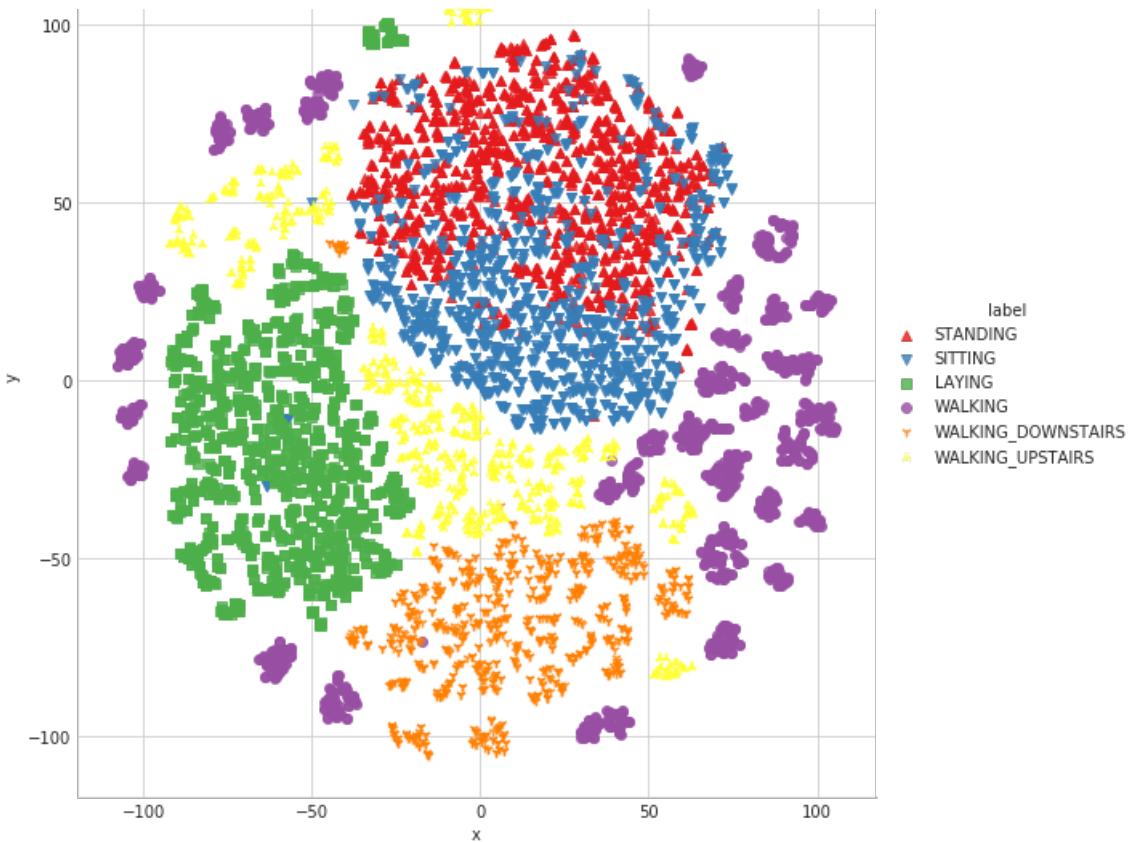
Done

```

performing tsne with perplexity 5 and with 1000 iterations at max
[t-SNE] Computing 16 nearest neighbors...
[t-SNE] Indexed 7352 samples in 0.263s...
[t-SNE] Computed neighbors for 7352 samples in 48.983s...
[t-SNE] Computed conditional probabilities for sample 1000 / 7352
[t-SNE] Computed conditional probabilities for sample 2000 / 7352
[t-SNE] Computed conditional probabilities for sample 3000 / 7352
[t-SNE] Computed conditional probabilities for sample 4000 / 7352
[t-SNE] Computed conditional probabilities for sample 5000 / 7352
[t-SNE] Computed conditional probabilities for sample 6000 / 7352
[t-SNE] Computed conditional probabilities for sample 7000 / 7352
[t-SNE] Computed conditional probabilities for sample 7352 / 7352
[t-SNE] Mean sigma: 0.961265
[t-SNE] Computed conditional probabilities in 0.122s
[t-SNE] Iteration 50: error = 114.1862640, gradient norm = 0.0184120 (50 iterations in 55.655s)
[t-SNE] Iteration 100: error = 97.6535568, gradient norm = 0.0174309 (50 iterations in 12.580s)
[t-SNE] Iteration 150: error = 93.1900101, gradient norm = 0.0101048 (50 iterations in 9.180s)
[t-SNE] Iteration 200: error = 91.2315445, gradient norm = 0.0074560 (50 iterations in 10.340s)
[t-SNE] Iteration 250: error = 90.0714417, gradient norm = 0.0057667 (50 iterations in 9.458s)
[t-SNE] KL divergence after 250 iterations with early exaggeration: 90.071442
[t-SNE] Iteration 300: error = 3.5796804, gradient norm = 0.0014691 (50 iterations in 8.718s)
[t-SNE] Iteration 350: error = 2.8173938, gradient norm = 0.0007508 (50 iterations in 10.180s)
[t-SNE] Iteration 400: error = 2.4344938, gradient norm = 0.0005251 (50 iterations in 10.506s)
[t-SNE] Iteration 450: error = 2.2156141, gradient norm = 0.0004069 (50 iterations in 10.072s)
[t-SNE] Iteration 500: error = 2.0703306, gradient norm = 0.0003340 (50 iterations in 10.511s)
[t-SNE] Iteration 550: error = 1.9646366, gradient norm = 0.0002816 (50 iterations in 9.792s)
[t-SNE] Iteration 600: error = 1.8835558, gradient norm = 0.0002471 (50 iterations in 9.098s)
[t-SNE] Iteration 650: error = 1.8184001, gradient norm = 0.0002184 (50 iterations in 8.656s)
[t-SNE] Iteration 700: error = 1.7647167, gradient norm = 0.0001961 (50 iterations in 9.063s)
[t-SNE] Iteration 750: error = 1.7193680, gradient norm = 0.0001796 (50 iterations in 9.754s)
[t-SNE] Iteration 800: error = 1.6803776, gradient norm = 0.0001655 (50 iterations in 9.540s)
[t-SNE] Iteration 850: error = 1.6465144, gradient norm = 0.0001538 (50 iterations in 9.953s)
[t-SNE] Iteration 900: error = 1.6166563, gradient norm = 0.0001421 (50 iterations in 10.270s)
[t-SNE] Iteration 950: error = 1.5901035, gradient norm = 0.0001335 (50 iterations in 6.609s)
[t-SNE] Iteration 1000: error = 1.5664237, gradient norm = 0.0001257 (50 iterations in 8.553s)
[t-SNE] Error after 1000 iterations: 1.566424
Done..
Creating plot for this t-sne visualization..
saving this plot as image in present working directory...

```

perplexity : 5 and max\_iter : 1000



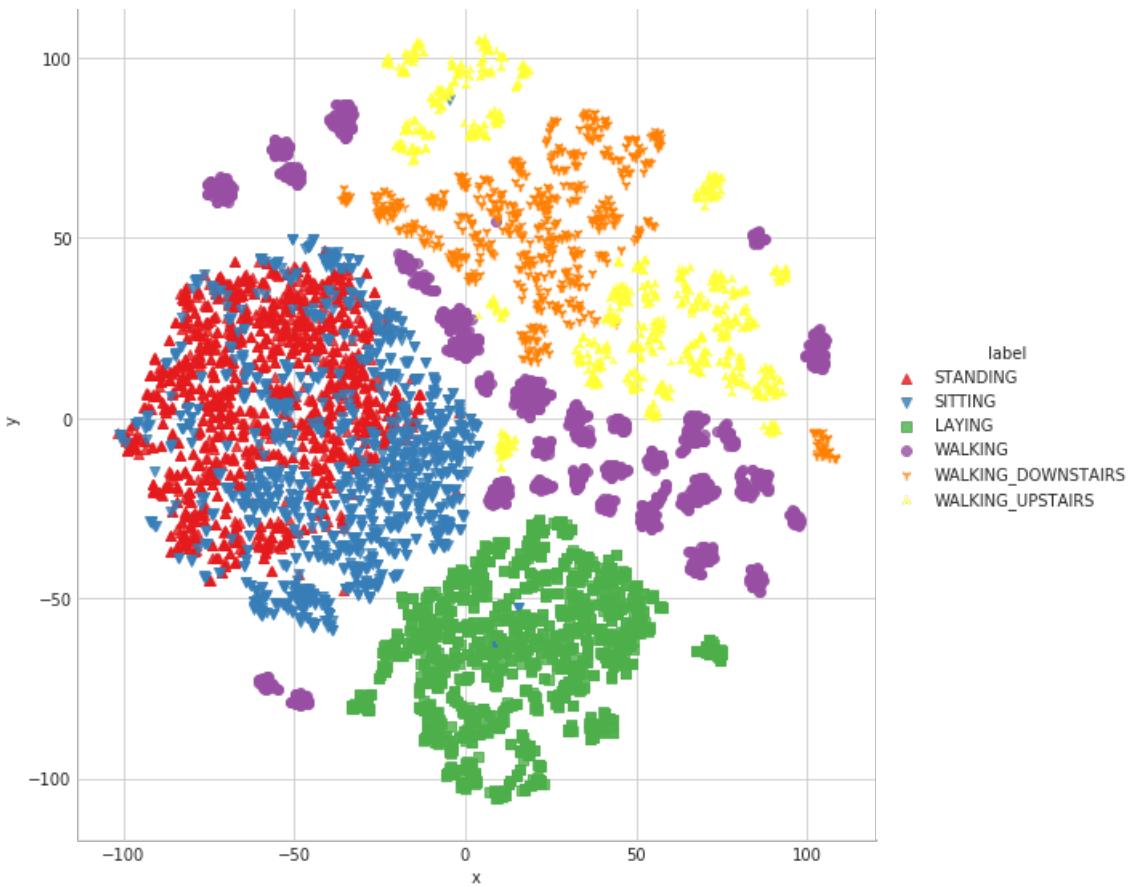
Done

```

performing tsne with perplexity 10 and with 1000 iterations at max
[t-SNE] Computing 31 nearest neighbors...
[t-SNE] Indexed 7352 samples in 0.410s...
[t-SNE] Computed neighbors for 7352 samples in 64.801s...
[t-SNE] Computed conditional probabilities for sample 1000 / 7352
[t-SNE] Computed conditional probabilities for sample 2000 / 7352
[t-SNE] Computed conditional probabilities for sample 3000 / 7352
[t-SNE] Computed conditional probabilities for sample 4000 / 7352
[t-SNE] Computed conditional probabilities for sample 5000 / 7352
[t-SNE] Computed conditional probabilities for sample 6000 / 7352
[t-SNE] Computed conditional probabilities for sample 7000 / 7352
[t-SNE] Computed conditional probabilities for sample 7352 / 7352
[t-SNE] Mean sigma: 1.133828
[t-SNE] Computed conditional probabilities in 0.214s
[t-SNE] Iteration 50: error = 106.0169220, gradient norm = 0.0194293 (50 iterations in 24.550s)
[t-SNE] Iteration 100: error = 90.3036194, gradient norm = 0.0097653 (50 iterations in 11.936s)
[t-SNE] Iteration 150: error = 87.3132935, gradient norm = 0.0053059 (50 iterations in 11.246s)
[t-SNE] Iteration 200: error = 86.1169128, gradient norm = 0.0035844 (50 iterations in 11.864s)
[t-SNE] Iteration 250: error = 85.4133606, gradient norm = 0.0029100 (50 iterations in 11.944s)
[t-SNE] KL divergence after 250 iterations with early exaggeration: 85.413361
[t-SNE] Iteration 300: error = 3.1394315, gradient norm = 0.0013976 (50 iterations in 11.742s)
[t-SNE] Iteration 350: error = 2.4929206, gradient norm = 0.0006466 (50 iterations in 11.627s)
[t-SNE] Iteration 400: error = 2.1733041, gradient norm = 0.0004230 (50 iterations in 11.846s)
[t-SNE] Iteration 450: error = 1.9884514, gradient norm = 0.0003124 (50 iterations in 11.405s)
[t-SNE] Iteration 500: error = 1.8702440, gradient norm = 0.0002514 (50 iterations in 11.320s)
[t-SNE] Iteration 550: error = 1.7870129, gradient norm = 0.0002107 (50 iterations in 12.009s)
[t-SNE] Iteration 600: error = 1.7246909, gradient norm = 0.0001824 (50 iterations in 10.632s)
[t-SNE] Iteration 650: error = 1.6758548, gradient norm = 0.0001590 (50 iterations in 11.270s)
[t-SNE] Iteration 700: error = 1.6361949, gradient norm = 0.0001451 (50 iterations in 12.072s)
[t-SNE] Iteration 750: error = 1.6034756, gradient norm = 0.0001305 (50 iterations in 11.607s)
[t-SNE] Iteration 800: error = 1.5761518, gradient norm = 0.0001188 (50 iterations in 9.409s)
[t-SNE] Iteration 850: error = 1.5527289, gradient norm = 0.0001113 (50 iterations in 8.309s)
[t-SNE] Iteration 900: error = 1.5328671, gradient norm = 0.0001021 (50 iterations in 9.433s)
[t-SNE] Iteration 950: error = 1.5152045, gradient norm = 0.0000974 (50 iterations in 11.488s)
[t-SNE] Iteration 1000: error = 1.4999681, gradient norm = 0.0000933 (50 iterations in 10.593s)
[t-SNE] Error after 1000 iterations: 1.499968
Done..
Creating plot for this t-sne visualization..
saving this plot as image in present working directory...

```

perplexity : 10 and max\_iter : 1000

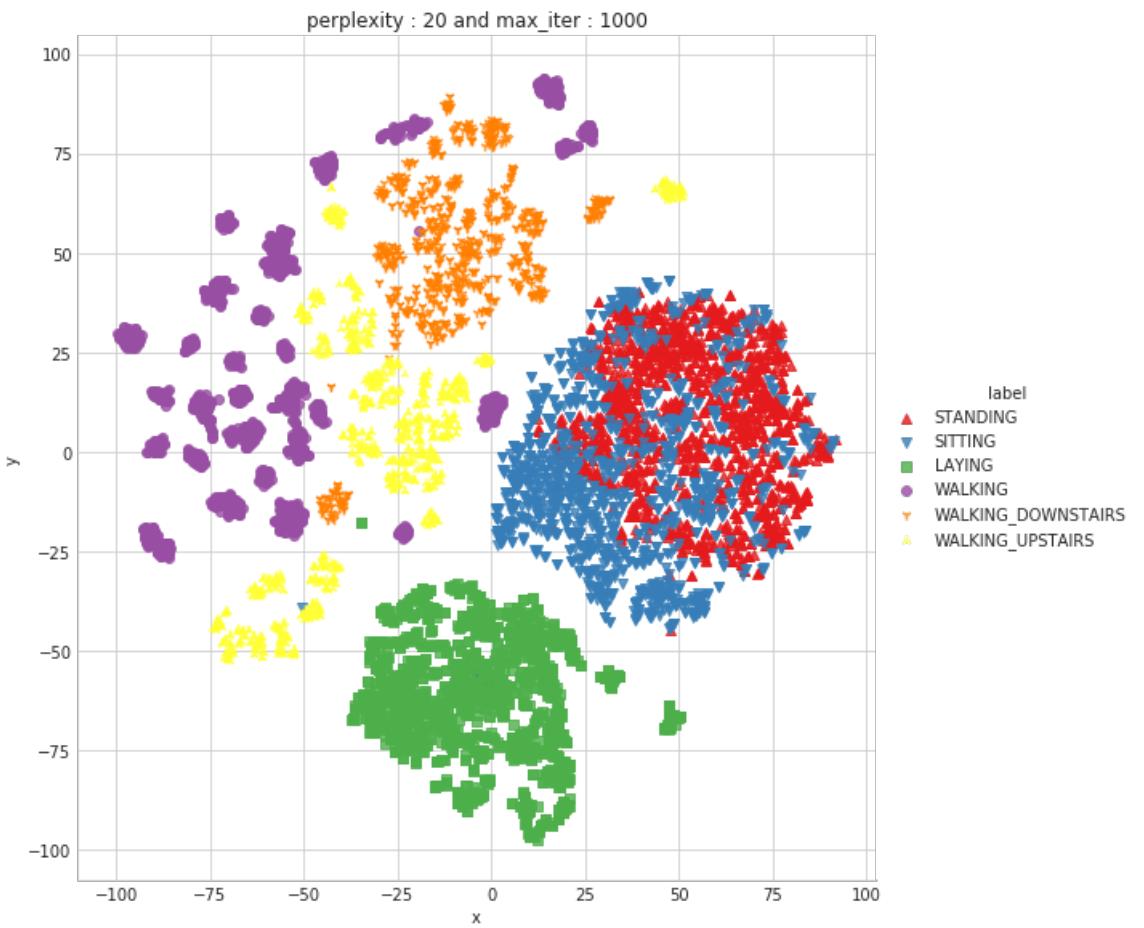


Done

```

performing tsne with perplexity 20 and with 1000 iterations at max
[t-SNE] Computing 61 nearest neighbors...
[t-SNE] Indexed 7352 samples in 0.425s...
[t-SNE] Computed neighbors for 7352 samples in 61.792s...
[t-SNE] Computed conditional probabilities for sample 1000 / 7352
[t-SNE] Computed conditional probabilities for sample 2000 / 7352
[t-SNE] Computed conditional probabilities for sample 3000 / 7352
[t-SNE] Computed conditional probabilities for sample 4000 / 7352
[t-SNE] Computed conditional probabilities for sample 5000 / 7352
[t-SNE] Computed conditional probabilities for sample 6000 / 7352
[t-SNE] Computed conditional probabilities for sample 7000 / 7352
[t-SNE] Computed conditional probabilities for sample 7352 / 7352
[t-SNE] Mean sigma: 1.274335
[t-SNE] Computed conditional probabilities in 0.355s
[t-SNE] Iteration 50: error = 97.5202179, gradient norm = 0.0223863 (50 iterations in 21.168s)
[t-SNE] Iteration 100: error = 83.9500732, gradient norm = 0.0059110 (50 iterations in 17.306s)
[t-SNE] Iteration 150: error = 81.8804779, gradient norm = 0.0035797 (50 iterations in 14.258s)
[t-SNE] Iteration 200: error = 81.1615143, gradient norm = 0.0022536 (50 iterations in 14.130s)
[t-SNE] Iteration 250: error = 80.7704086, gradient norm = 0.0018108 (50 iterations in 15.340s)
[t-SNE] KL divergence after 250 iterations with early exaggeration: 80.770409
[t-SNE] Iteration 300: error = 2.6957574, gradient norm = 0.0012993 (50 iterations in 13.605s)
[t-SNE] Iteration 350: error = 2.1637220, gradient norm = 0.0005765 (50 iterations in 13.248s)
[t-SNE] Iteration 400: error = 1.9143614, gradient norm = 0.0003474 (50 iterations in 14.774s)
[t-SNE] Iteration 450: error = 1.7684202, gradient norm = 0.0002458 (50 iterations in 15.502s)
[t-SNE] Iteration 500: error = 1.6744757, gradient norm = 0.0001923 (50 iterations in 14.808s)
[t-SNE] Iteration 550: error = 1.6101606, gradient norm = 0.0001575 (50 iterations in 14.043s)
[t-SNE] Iteration 600: error = 1.5641028, gradient norm = 0.0001344 (50 iterations in 15.769s)
[t-SNE] Iteration 650: error = 1.5291905, gradient norm = 0.0001182 (50 iterations in 15.834s)
[t-SNE] Iteration 700: error = 1.5024391, gradient norm = 0.0001055 (50 iterations in 15.398s)
[t-SNE] Iteration 750: error = 1.4809053, gradient norm = 0.0000965 (50 iterations in 14.594s)
[t-SNE] Iteration 800: error = 1.4631859, gradient norm = 0.0000884 (50 iterations in 15.025s)
[t-SNE] Iteration 850: error = 1.4486470, gradient norm = 0.0000832 (50 iterations in 14.060s)
[t-SNE] Iteration 900: error = 1.4367288, gradient norm = 0.0000804 (50 iterations in 12.389s)
[t-SNE] Iteration 950: error = 1.4270191, gradient norm = 0.0000761 (50 iterations in 10.392s)
[t-SNE] Iteration 1000: error = 1.4189968, gradient norm = 0.0000787 (50 iterations in 12.355s)
[t-SNE] Error after 1000 iterations: 1.418997
Done..
Creating plot for this t-sne visualization..
saving this plot as image in present working directory...

```



Done

```

performing tsne with perplexity 50 and with 1000 iterations at max
[t-SNE] Computing 151 nearest neighbors...
[t-SNE] Indexed 7352 samples in 0.376s...
[t-SNE] Computed neighbors for 7352 samples in 73.164s...
[t-SNE] Computed conditional probabilities for sample 1000 / 7352
[t-SNE] Computed conditional probabilities for sample 2000 / 7352
[t-SNE] Computed conditional probabilities for sample 3000 / 7352
[t-SNE] Computed conditional probabilities for sample 4000 / 7352
[t-SNE] Computed conditional probabilities for sample 5000 / 7352
[t-SNE] Computed conditional probabilities for sample 6000 / 7352
[t-SNE] Computed conditional probabilities for sample 7000 / 7352
[t-SNE] Computed conditional probabilities for sample 7352 / 7352
[t-SNE] Mean sigma: 1.437672
[t-SNE] Computed conditional probabilities in 0.844s
[t-SNE] Iteration 50: error = 86.1525574, gradient norm = 0.0242986 (50 iterations in 36.249s)
[t-SNE] Iteration 100: error = 75.9874649, gradient norm = 0.0061005 (50 iterations in 30.453s)
[t-SNE] Iteration 150: error = 74.7072296, gradient norm = 0.0024708 (50 iterations in 28.461s)
[t-SNE] Iteration 200: error = 74.2736282, gradient norm = 0.0018644 (50 iterations in 27.735s)
[t-SNE] Iteration 250: error = 74.0722427, gradient norm = 0.0014078 (50 iterations in 26.835s)
[t-SNE] KL divergence after 250 iterations with early exaggeration: 74.072243
[t-SNE] Iteration 300: error = 2.1539080, gradient norm = 0.0011796 (50 iterations in 25.445s)
[t-SNE] Iteration 350: error = 1.7567128, gradient norm = 0.0004845 (50 iterations in 21.282s)
[t-SNE] Iteration 400: error = 1.5888531, gradient norm = 0.0002798 (50 iterations in 21.015s)
[t-SNE] Iteration 450: error = 1.4956820, gradient norm = 0.0001894 (50 iterations in 23.332s)
[t-SNE] Iteration 500: error = 1.4359720, gradient norm = 0.0001420 (50 iterations in 23.083s)
[t-SNE] Iteration 550: error = 1.3947564, gradient norm = 0.0001117 (50 iterations in 19.626s)
[t-SNE] Iteration 600: error = 1.3653858, gradient norm = 0.0000949 (50 iterations in 22.752s)
[t-SNE] Iteration 650: error = 1.3441534, gradient norm = 0.0000814 (50 iterations in 23.972s)
[t-SNE] Iteration 700: error = 1.3284039, gradient norm = 0.0000742 (50 iterations in 20.636s)
[t-SNE] Iteration 750: error = 1.3171139, gradient norm = 0.0000700 (50 iterations in 20.407s)
[t-SNE] Iteration 800: error = 1.3085558, gradient norm = 0.0000657 (50 iterations in 24.951s)
[t-SNE] Iteration 850: error = 1.3017821, gradient norm = 0.0000603 (50 iterations in 24.719s)
[t-SNE] Iteration 900: error = 1.2962619, gradient norm = 0.0000586 (50 iterations in 24.500s)
[t-SNE] Iteration 950: error = 1.2914882, gradient norm = 0.0000573 (50 iterations in 24.132s)
[t-SNE] Iteration 1000: error = 1.2874244, gradient norm = 0.0000546 (50 iterations in 22.840s)
[t-SNE] Error after 1000 iterations: 1.287424
Done..
Creating plot for this t-sne visualization..
saving this plot as image in present working directory...

```



Done

## HAR Prediction on DomainExpert Features

### Obtain the train and test data

In [0]:

```
train = pd.read_csv('UCI_HAR_dataset/csv_files/train.csv')
test = pd.read_csv('UCI_HAR_dataset/csv_files/test.csv')
print(train.shape, test.shape)
```

(7352, 564) (2947, 564)

In [0]:

```
train.head(3)
```

Out[0]:

	tBodyAccmeanX	tBodyAccmeanY	tBodyAccmeanZ	tBodyAccstdX	tBodyAccstdY	tBodyAccstdZ	tBodyAccmadX	tBodyAccmadY	t
0	0.288585	-0.020294	-0.132905	-0.995279	-0.983111	-0.913526	-0.995112	-0.983185	
1	0.278419	-0.016411	-0.123520	-0.998245	-0.975300	-0.960322	-0.998807	-0.974914	
2	0.279653	-0.019467	-0.113462	-0.995380	-0.967187	-0.978944	-0.996520	-0.963668	

3 rows × 564 columns

In [0]:

```
# get X_train and y_train from csv files
```

```
# get X_train and y_train from csv files
X_train = train.drop(['subject', 'Activity', 'ActivityName'], axis=1)
y_train = train.ActivityName
```

In [0]:

```
# get X_test and y_test from test csv file
X_test = test.drop(['subject', 'Activity', 'ActivityName'], axis=1)
y_test = test.ActivityName
```

In [0]:

```
print('X_train and y_train : ({}, {})'.format(X_train.shape, y_train.shape))
print('X_test and y_test : ({}, {})'.format(X_test.shape, y_test.shape))
```

```
X_train and y_train : ((7352, 561), (7352,))
X_test and y_test : ((2947, 561), (2947,))
```

## Let's model with our data

### Labels that are useful in plotting confusion matrix

In [0]:

```
labels=['LAYING', 'SITTING', 'STANDING', 'WALKING', 'WALKING_DOWNSTAIRS', 'WALKING_UPSTAIRS']
```

### Function to plot the confusion matrix

In [0]:

```
import itertools
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
plt.rcParams["font.family"] = 'DejaVu Sans'

def plot_confusion_matrix(cm, classes,
                         normalize=False,
                         title='Confusion matrix',
                         cmap=plt.cm.Blues):
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=90)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```

### Generic function to run any model specified

In [0]:

```
from datetime import datetime
```

```

def perform_model(model, X_train, y_train, X_test, y_test, class_labels, cm_normalize=True, \
                  print_cm=True, cm_cmap=plt.cm.Greens):

    # to store results at various phases
    results = dict()

    # time at which model starts training
    train_start_time = datetime.now()
    print('training the model..')
    model.fit(X_train, y_train)
    print('Done \n \n')
    train_end_time = datetime.now()
    results['training_time'] = train_end_time - train_start_time
    print('training_time(HH:MM:SS.ms) - {} \n \n'.format(results['training_time']))

    # predict test data
    print('Predicting test data')
    test_start_time = datetime.now()
    y_pred = model.predict(X_test)
    test_end_time = datetime.now()
    print('Done \n \n')
    results['testing_time'] = test_end_time - test_start_time
    print('testing time(HH:MM:SS:ms) - {} \n \n'.format(results['testing_time']))
    results['predicted'] = y_pred

    # calculate overall accuracy of the model
    accuracy = metrics.accuracy_score(y_true=y_test, y_pred=y_pred)
    # store accuracy in results
    results['accuracy'] = accuracy
    print('-----')
    print('| Accuracy |')
    print('-----')
    print('\n {} \n \n'.format(accuracy))

    # confusion matrix
    cm = metrics.confusion_matrix(y_test, y_pred)
    results['confusion_matrix'] = cm
    if print_cm:
        print('-----')
        print('| Confusion Matrix |')
        print('-----')
        print('\n {} \n'.format(cm))

    # plot confusin matrix
    plt.figure(figsize=(8,8))
    plt.grid(b=False)
    plot_confusion_matrix(cm, classes=class_labels, normalize=True, title='Normalized confusion \
matrix', cmap = cm_cmap)
    plt.show()

    # get classification report
    print('-----')
    print('| Classification Report |')
    print('-----')
    classification_report = metrics.classification_report(y_test, y_pred)
    # store report in results
    results['classification_report'] = classification_report
    print(classification_report)

    # add the trained model to the results
    results['model'] = model

return results

```

## Method to print the gridsearch Attributes

In [0]:

```
def print_grid_search_attributes(model):
```

```

def print_grid_search_attributes(model):
    # Estimator that gave highest score among all the estimators formed in GridSearch
    print('-----')
    print('|      Best Estimator      |')
    print('-----')
    print('\n\t{}\\n'.format(model.best_estimator_))

    # parameters that gave best results while performing grid search
    print('-----')
    print('|      Best parameters      |')
    print('-----')
    print('\\n\\tParameters of best estimator : \\n\\n\\t{}\\n'.format(model.best_params_))

    # number of cross validation splits
    print('-----')
    print('|      No of CrossValidation sets      |')
    print('-----')
    print('\\n\\tTotal numbre of cross validation sets: {}\\n'.format(model.n_splits_))

    # Average cross validated score of the best estimator, from the Grid Search
    print('-----')
    print('|      Best Score      |')
    print('-----')
    print('\\n\\tAverage Cross Validate scores of best estimator : \\n\\n\\t{}\\n'.format(model.best_score_))

```

## 1. Logistic Regression with Grid Search

In [0]:

```

from sklearn import linear_model
from sklearn import metrics

from sklearn.model_selection import GridSearchCV

```

In [0]:

```

# start Grid search
parameters = {'C':[0.01, 0.1, 1, 10, 20, 30], 'penalty':['l2','l1']}
log_reg = linear_model.LogisticRegression()
log_reg_grid = GridSearchCV(log_reg, param_grid=parameters, cv=3, verbose=1, n_jobs=-1)
log_reg_grid_results = perform_model(log_reg_grid, X_train, y_train, X_test, y_test, class_labels=labels)

```

training the model..  
Fitting 3 folds for each of 12 candidates, totalling 36 fits

[Parallel(n\_jobs=-1)]: Done 36 out of 36 | elapsed: 1.2min finished

Done

training\_time(HH:MM:SS.ms) - 0:01:25.843810

Predicting test data  
Done

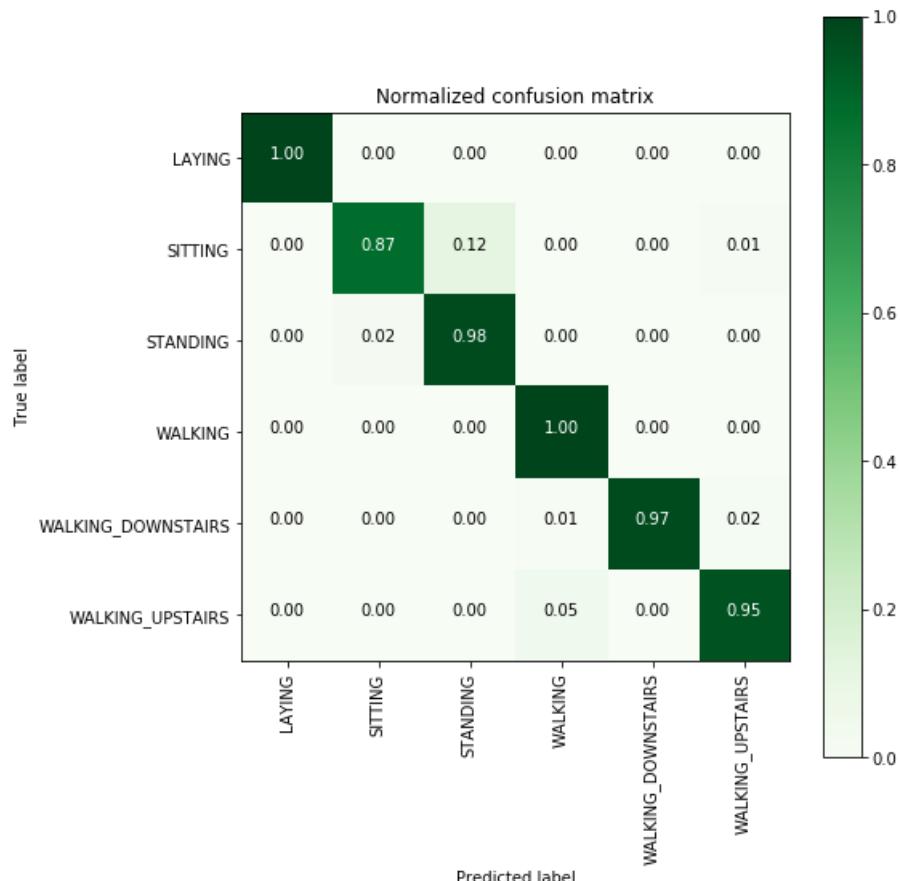
testing time(HH:MM:SS:ms) - 0:00:00.009192

-----  
Accuracy

0.9626739056667798

-----  
Confusion Matrix

```
[[537    0    0    0    0    0]
 [ 1 428   58    0    0    4]
 [ 0 12 519    1    0    0]
 [ 0    0    0 495    1    0]
 [ 0    0    0    3 409    8]
 [ 0    0    0   22    0 449]]
```



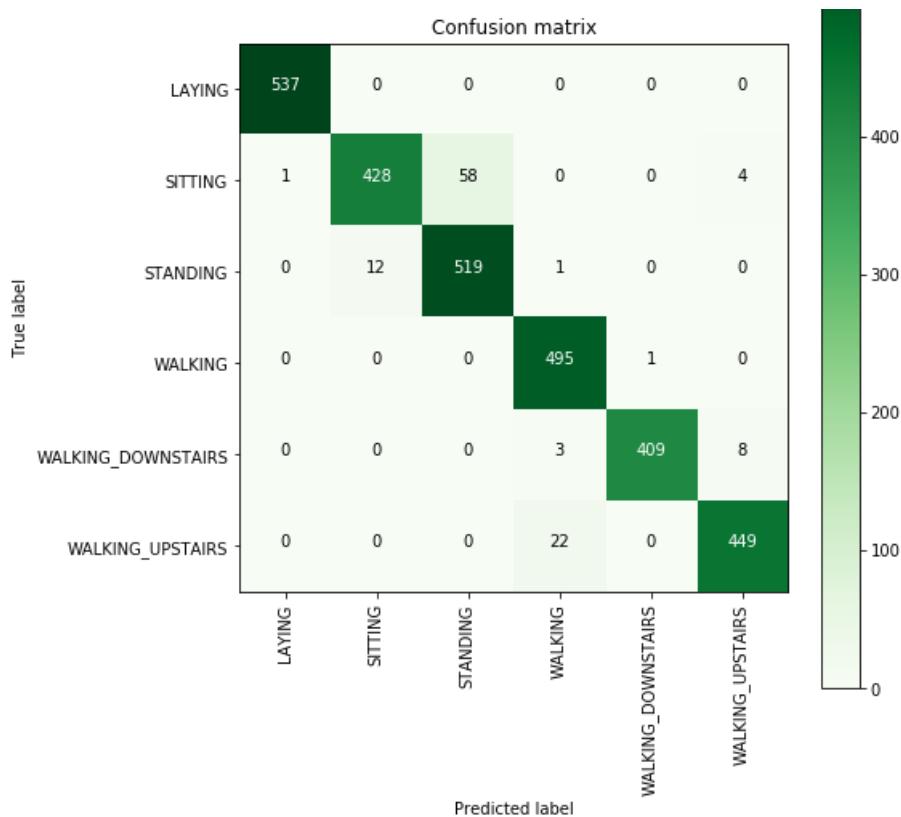
-----  
Classification Report

	precision	recall	f1-score	support
LAYING	1.00	1.00	1.00	537
SITTING	0.97	0.87	0.92	491
STANDING	0.90	0.98	0.94	532
WALKING	0.95	1.00	0.97	496
WALKING_DOWNSTAIRS	1.00	0.97	0.99	420
WALKING_UPSTAIRS	0.97	0.95	0.96	471
avg / total	0.96	0.96	0.96	2947

In [0]:

```
plt.figure(figsize=(8,8))
plt.grid(b=False)
plot_confusion_matrix(log_reg_grid_results['confusion_matrix'], classes=labels, cmap=plt.cm.Greens
, )
plt.show()
```





In [0]:

```
# observe the attributes of the model
print_grid_search_attributes(log_reg_grid_results['model'])
```

| Best Estimator |

```
LogisticRegression(C=30, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
verbose=0, warm_start=False)
```

| Best parameters |

Parameters of best estimator :

```
{'C': 30, 'penalty': 'l2'}
```

| No of CrossValidation sets |

Total number of cross validation sets: 3

| Best Score |

Average Cross Validate scores of best estimator :

```
0.9461371055495104
```

## 2. Linear SVC with GridSearch

In [0]:

```
parameters = {'C':[0.125, 0.5, 1, 2, 8, 16]}
for c in parameters:
    print(c)
```

```

lr_svc = LinearSVC(C=1.0)
lr_svc_grid = GridSearchCV(lr_svc, param_grid=parameters, n_jobs=-1, verbose=1)
lr_svc_grid_results = perform_model(lr_svc_grid, X_train, y_train, X_test, y_test, class_labels=labels)

```

training the model..  
Fitting 3 folds for each of 6 candidates, totalling 18 fits

```
[Parallel(n_jobs=-1)]: Done 18 out of 18 | elapsed: 24.9s finished
```

Done

training\_time(HH:MM:SS.ms) - 0:00:32.951942

Predicting test data

Done

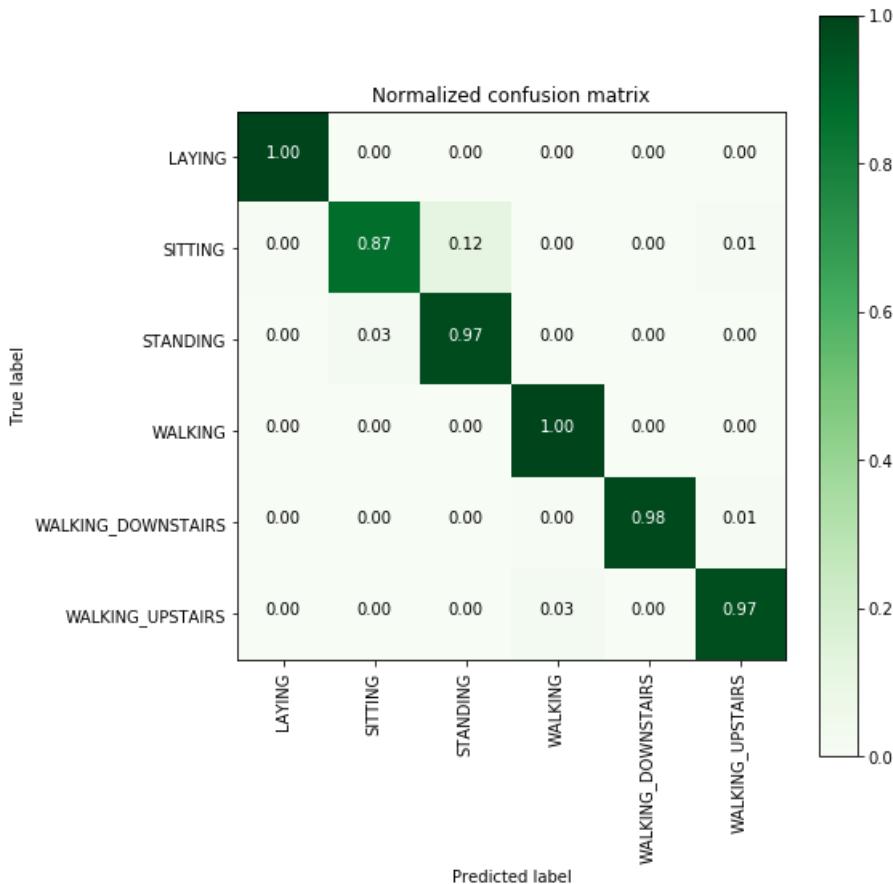
testing\_time(HH:MM:SS:ms) - 0:00:00.012182

```
-----| Accuracy |-----
```

0.9660671869697998

```
-----| Confusion Matrix |-----
```

```
[[537  0  0  0  0  0]
 [ 2 426 58  0  0  5]
 [ 0 14 518  0  0  0]
 [ 0  0  0 495  0  1]
 [ 0  0  0  2 413  5]
 [ 0  0  0 12  1 458]]
```



```
-----  
| Classification Report |  
-----  
precision    recall   f1-score   support  
  
    LAYING      1.00      1.00      1.00      537  
    SITTING     0.97      0.87      0.92      491  
    STANDING    0.90      0.97      0.94      532  
    WALKING     0.97      1.00      0.99      496  
WALKING_DOWNSTAIRS  1.00      0.98      0.99      420  
WALKING_UPSTAIRS   0.98      0.97      0.97      471  
  
avg / total       0.97      0.97      0.97      2947
```

In [0]:

```
print_grid_search_attributes(lr_svc_grid_results['model'])
```

```
-----  
|      Best Estimator      |  
-----
```

```
LinearSVC(C=8, class_weight=None, dual=True, fit_intercept=True,  
          intercept_scaling=1, loss='squared_hinge', max_iter=1000,  
          multi_class='ovr', penalty='l2', random_state=None, tol=5e-05,  
          verbose=0)
```

```
-----  
|      Best parameters      |  
-----
```

Parameters of best estimator :

```
{'C': 8}
```

```
-----  
|      No of CrossValidation sets      |  
-----
```

Total number of cross validation sets: 3

```
-----  
|      Best Score      |  
-----
```

Average Cross Validate scores of best estimator :

```
0.9465451577801959
```

### 3. Kernel SVM with GridSearch

In [0]:

```
from sklearn.svm import SVC  
parameters = {'C':[2,8,16],  
             'gamma': [ 0.0078125, 0.125, 2]}  
rbf_svm = SVC(kernel='rbf')  
rbf_svm_grid = GridSearchCV(rbf_svm,param_grid=parameters, n_jobs=-1)  
rbf_svm_grid_results = perform_model(rbf_svm_grid, X_train, y_train, X_test, y_test, class_labels=labels)
```

```
training the model..  
Done
```

```
training_time(HH:MM:SS.ms) - 0:05:46.182889
```

```
Predicting test data  
Done
```

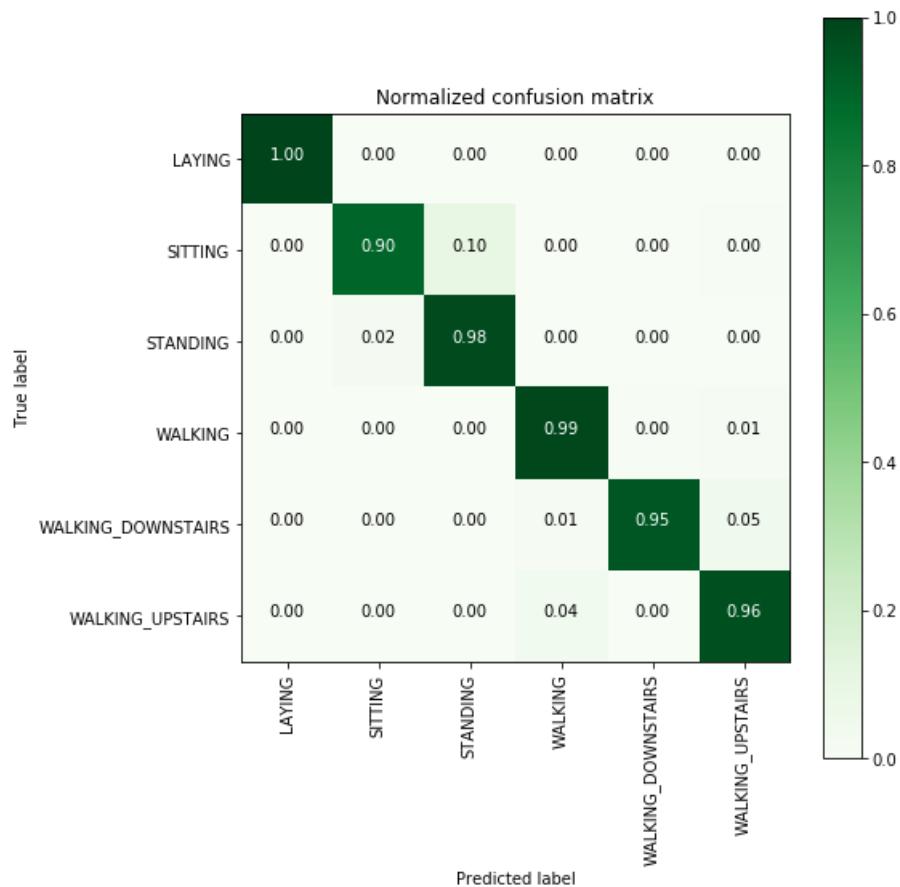
```
testing time(HH:MM:SS:ms) - 0:00:05.221285
```

```
| Accuracy |
```

```
0.9626739056667798
```

```
| Confusion Matrix |
```

```
[ [537 0 0 0 0 0]
[ 0 441 48 0 0 2]
[ 0 12 520 0 0 0]
[ 0 0 0 489 2 5]
[ 0 0 0 4 397 19]
[ 0 0 0 17 1 453]]
```



```
| Classification Report |
```

	precision	recall	f1-score	support
LAYING	1.00	1.00	1.00	537
SITTING	0.97	0.90	0.93	491
STANDING	0.92	0.98	0.95	532
WALKING	0.96	0.99	0.97	496
WALKING_DOWNSTAIRS	0.99	0.95	0.97	420
WALKING_UPSTAIRS	0.95	0.96	0.95	471
avg / total	0.96	0.96	0.96	2947

```
In [0]:
```

```
print grid_search.attributes(rbf_svm_grid_results['model'])
```

```

-----| Best Estimator |-----
SVC(C=16, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma=0.0078125, kernel='rbf',
max_iter=-1, probability=False, random_state=None, shrinking=True,
tol=0.001, verbose=False)

-----| Best parameters |-----
Parameters of best estimator :

{'C': 16, 'gamma': 0.0078125}

-----| No of CrossValidation sets |-----
Total numbre of cross validation sets: 3

-----| Best Score |-----
Average Cross Validate scores of best estimator :

0.9440968443960827

```

## 4. Decision Trees with GridSearchCV

In [0]:

```

from sklearn.tree import DecisionTreeClassifier
parameters = {'max_depth':np.arange(3,10,2)}
dt = DecisionTreeClassifier()
dt_grid = GridSearchCV(dt,param_grid=parameters, n_jobs=-1)
dt_grid_results = perform_model(dt_grid, X_train, y_train, X_test, y_test, class_labels=labels)
print_grid_search_attributes(dt_grid_results['model'])

```

training the model..  
Done

training\_time(HH:MM:SS.ms) - 0:00:19.476858

Predicting test data  
Done

testing time(HH:MM:SS:ms) - 0:00:00.012858

```

-----| Accuracy |-----

```

0.8642687478791992

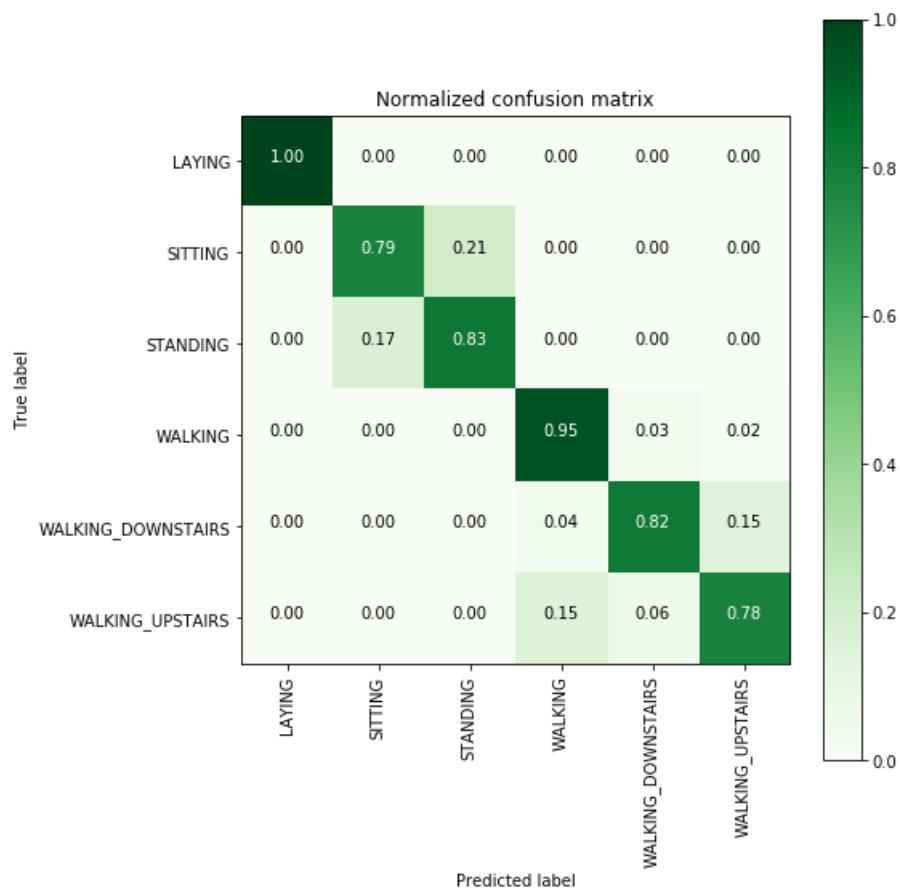
```

-----| Confusion Matrix |-----

```

[ 537 0 0 0 0]
[ 0 386 105 0 0 0]
[ 0 93 439 0 0 0]
[ 0 0 0 472 16 8]
[ 0 0 0 15 344 611]

```
[ 0  0  0  73  29 369]
```



---

| Classification Report |

---

	precision	recall	f1-score	support
LAYING	1.00	1.00	1.00	537
SITTING	0.81	0.79	0.80	491
STANDING	0.81	0.83	0.82	532
WALKING	0.84	0.95	0.89	496
WALKING_DOWNSTAIRS	0.88	0.82	0.85	420
WALKING_UPSTAIRS	0.84	0.78	0.81	471
avg / total	0.86	0.86	0.86	2947

---

| Best Estimator |

---

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=7,
                      max_features=None, max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                      splitter='best')
```

---

| Best parameters |

---

Parameters of best estimator :

```
{'max_depth': 7}
```

---

| No of CrossValidation sets |

---

Total number of cross validation sets: 3

---

| Best Score |

---

```
|      Best Score      |
-----
Average Cross Validate scores of best estimator :
0.8369151251360174
```

## 5. Random Forest Classifier with GridSearch

In [0]:

```
from sklearn.ensemble import RandomForestClassifier
params = {'n_estimators': np.arange(10,201,20), 'max_depth':np.arange(3,15,2)}
rfc = RandomForestClassifier()
rfc_grid = GridSearchCV(rfc, param_grid=params, n_jobs=-1)
rfc_grid_results = perform_model(rfc_grid, X_train, y_train, X_test, y_test, class_labels=labels)
print_grid_search_attributes(rfc_grid_results['model'])
```

training the model..
Done

training\_time(HH:MM:SS.ms) - 0:06:22.775270

Predicting test data
Done

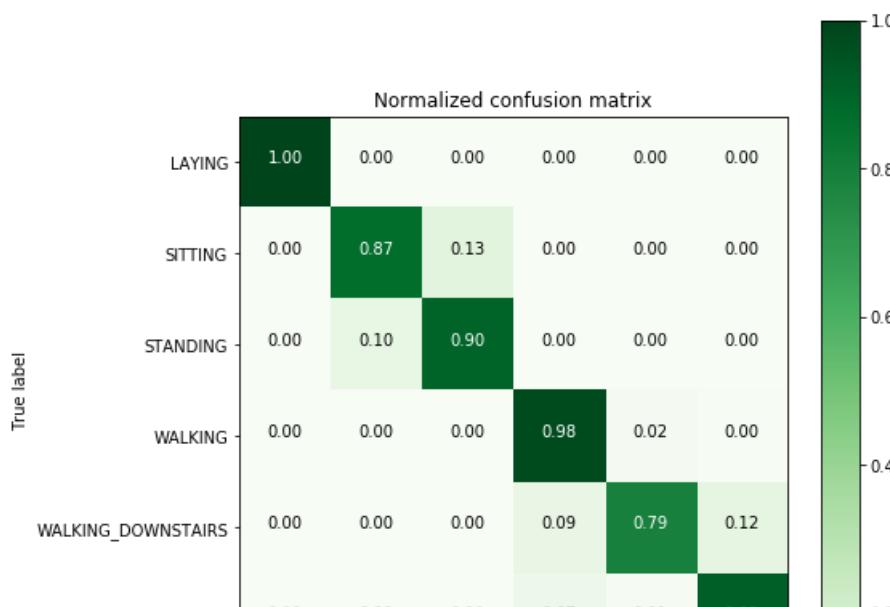
testing time(HH:MM:SS:ms) - 0:00:00.025937

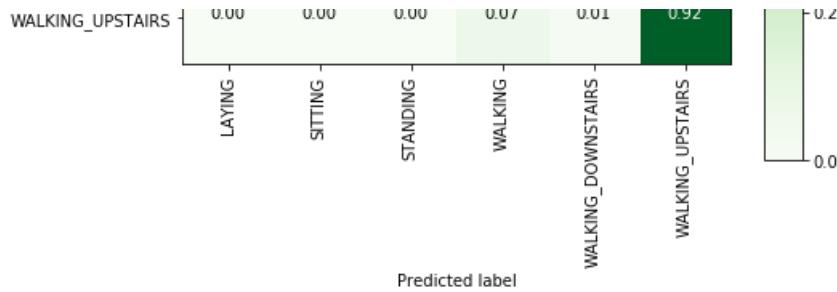
```
|      Accuracy      |
-----
```

0.9131319986426875

```
| Confusion Matrix |
-----
```

```
[[537  0  0  0  0  0]
 [ 0 427 64  0  0  0]
 [ 0  52 480  0  0  0]
 [ 0  0  0 484 10  2]
 [ 0  0  0  38 332 50]
 [ 0  0  0  34   6 431]]
```





| Classification Report |

	precision	recall	f1-score	support
LAYING	1.00	1.00	1.00	537
SITTING	0.89	0.87	0.88	491
STANDING	0.88	0.90	0.89	532
WALKING	0.87	0.98	0.92	496
WALKING_DOWNSTAIRS	0.95	0.79	0.86	420
WALKING_UPSTAIRS	0.89	0.92	0.90	471
avg / total	0.92	0.91	0.91	2947

| Best Estimator |

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                      max_depth=7, max_features='auto', max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, n_estimators=70, n_jobs=1,
                      oob_score=False, random_state=None, verbose=0,
                      warm_start=False)
```

| Best parameters |

Parameters of best estimator :

```
{'max_depth': 7, 'n_estimators': 70}
```

| No of CrossValidation sets |

Total number of cross validation sets: 3

| Best Score |

Average Cross Validate scores of best estimator :

0.9141730141458106

## 6. Gradient Boosted Decision Trees With GridSearch

In [0]:

```
from sklearn.ensemble import GradientBoostingClassifier
param_grid = {'max_depth': np.arange(5,8,1), \
              'n_estimators':np.arange(130,170,10)}
gbdt = GradientBoostingClassifier()
gbdt_grid = GridSearchCV(gbdt, param_grid=param_grid, n_jobs=-1)
gbdt_grid_results = perform_model(gbdt_grid, X_train, y_train, X_test, y_test, class_labels=labels)
print_grid_search_attributes(gbdt_grid_results['model'])
```

training the model..

Done

training\_time(HH:MM:SS.ms) - 0:28:03.653432

Predicting test data  
Done

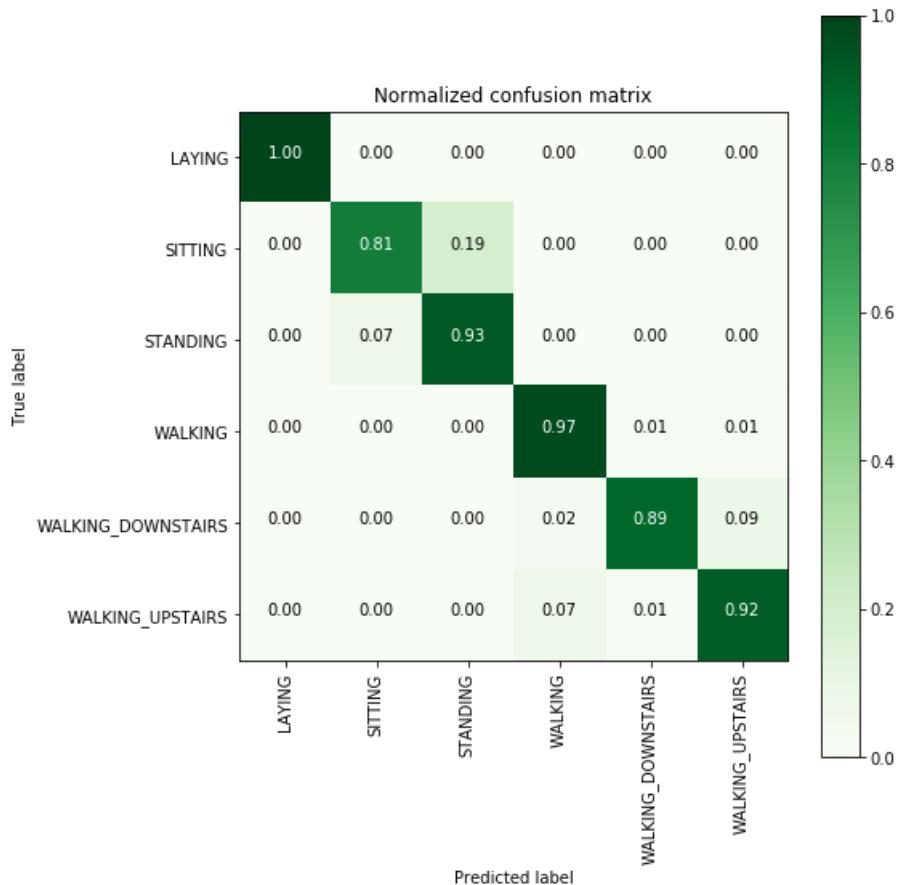
testing time(HH:MM:SS:ms) - 0:00:00.058843

-----  
Accuracy

0.9222938581608415

-----  
Confusion Matrix

```
[[537    0    0    0    0    0]
 [ 0 396   93    0    0    2]
 [ 0   37 495    0    0    0]
 [ 0     0 483    7    6]
 [ 0     0 10 374   36]
 [ 0     1  0 31   6 433]]
```



-----  
Classification Report

	precision	recall	f1-score	support
LAYING	1.00	1.00	1.00	537
SITTING	0.91	0.81	0.86	491
STANDING	0.84	0.93	0.88	532
WALKING	0.92	0.97	0.95	496
WALKING_DOWNSTAIRS	0.97	0.89	0.93	420
WALKING_UPSTAIRS	0.91	0.92	0.91	471

```

avg / total      0.92      0.92      0.92      2947

-----
|   Best Estimator   |
-----

GradientBoostingClassifier(criterion='friedman_mse', init=None,
                           learning_rate=0.1, loss='deviance', max_depth=5,
                           max_features=None, max_leaf_nodes=None,
                           min_impurity_decrease=0.0, min_impurity_split=None,
                           min_samples_leaf=1, min_samples_split=2,
                           min_weight_fraction_leaf=0.0, n_estimators=140,
                           presort='auto', random_state=None, subsample=1.0, verbose=0,
                           warm_start=False)

-----
|   Best parameters   |
-----
Parameters of best estimator :

{'max_depth': 5, 'n_estimators': 140}

-----
|   No of CrossValidation sets   |
-----

Total numbre of cross validation sets: 3

-----
|   Best Score   |
-----

Average Cross Validate scores of best estimator :

0.904379760609358

```

## 7. Comparing all models

In [0]:

```

print('\n          Accuracy      Error')
print('-----  -----')
print('Logistic Regression : {:.04}%    {:.04}%'.format(log_reg_grid_results['accuracy'] * 100, \
                                                       100-(log_reg_grid_results['accuracy'] * 100)))

print('Linear SVC        : {:.04}%    {:.04}%'.format(lr_svc_grid_results['accuracy'] * 100, \
                                                       100-(lr_svc_grid_results['accuracy'] * 100))

print('rbf SVM classifier : {:.04}%    {:.04}%'.format(rbf_svm_grid_results['accuracy'] * 100, \
                                                       100-(rbf_svm_grid_results['accuracy'] * 100)))
))

print('DecisionTree       : {:.04}%    {:.04}%'.format(dt_grid_results['accuracy'] * 100, \
                                                       100-(dt_grid_results['accuracy'] * 100)))

print('Random Forest      : {:.04}%    {:.04}%'.format(rfc_grid_results['accuracy'] * 100, \
                                                       100-(rfc_grid_results['accuracy'] * 100)))
print('GradientBoosting DT : {:.04}%    {:.04}%'.format(rfc_grid_results['accuracy'] * 100, \
                                                       100-(rfc_grid_results['accuracy'] * 100)))

```

	Accuracy	Error
	-----	-----
Logistic Regression :	96.27%	3.733%
Linear SVC :	96.61%	3.393%
rbf SVM classifier :	96.27%	3.733%
DecisionTree :	86.43%	13.57%
Random Forest :	91.31%	8.687%
GradientBoosting DT :	91.31%	8.687%

We can choose ***Logistic regression*** or ***Linear SVC*** or ***rbf SVM***.

## Conclusion :

In the real world, domain-knowledge, EDA and feature-engineering matter most.

## HAR by Just using LSTM (Without a domain expert)

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 [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 = data_path+f'Inertial Signals_{subset}/{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]:

```
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 = data_path + f'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 [0]:

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

In [0]:

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

In [0]:

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

In [0]:

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

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 [0]:

```
# 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 [0]:

```
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]:

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

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:66: The name tf.get_default_graph is deprecated. Plea-
se use tf.compat.v1.get_default_graph instead.
```

```
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`.
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 32)	5376
dropout_1 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 6)	198
=====		
Total params: 5,574		
Trainable params: 5,574		
Non-trainable params: 0		

```
In [0]:
```

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

```
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.
```

```
In [0]:
```

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

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/math_grad.py:1250: add_dispatch_support.<locals>.wrapper (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
Train on 7352 samples, validate on 2947 samples
Epoch 1/30
7352/7352 [=====] - 120s 16ms/step - loss: 1.3282 - acc: 0.4343 - val_los
s: 1.1742 - val_acc: 0.4774
Epoch 2/30
7352/7352 [=====] - 115s 16ms/step - loss: 0.9994 - acc: 0.5703 - val_los
s: 1.1734 - val_acc: 0.5036
Epoch 3/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.8066 - acc: 0.6489 - val_los
s: 0.7493 - val_acc: 0.6159
Epoch 4/30
7352/7352 [=====] - 115s 16ms/step - loss: 0.6889 - acc: 0.6683 - val_los
s: 0.7266 - val_acc: 0.6247
Epoch 5/30
7352/7352 [=====] - 115s 16ms/step - loss: 0.6455 - acc: 0.6831 - val_los
s: 0.7353 - val_acc: 0.6250
Epoch 6/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.6029 - acc: 0.7073 - val_los
s: 0.7757 - val_acc: 0.6834
Epoch 7/30
7352/7352 [=====] - 114s 15ms/step - loss: 0.5758 - acc: 0.7416 - val_los
s: 0.6909 - val_acc: 0.7044
Epoch 8/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.5318 - acc: 0.7647 - val_los
s: 0.6520 - val_acc: 0.7265
Epoch 9/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.4595 - acc: 0.7824 - val_los
s: 0.5209 - val_acc: 0.7472
Epoch 10/30
```

```

Epoch 10/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.4266 - acc: 0.7933 - val_loss: 0.4815 - val_acc: 0.7516
Epoch 11/30
7352/7352 [=====] - 117s 16ms/step - loss: 0.3865 - acc: 0.8002 - val_loss: 0.4803 - val_acc: 0.7441
Epoch 12/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.3530 - acc: 0.8090 - val_loss: 0.4857 - val_acc: 0.7543
Epoch 13/30
7352/7352 [=====] - 115s 16ms/step - loss: 0.3366 - acc: 0.8162 - val_loss: 0.4763 - val_acc: 0.7516
Epoch 14/30
7352/7352 [=====] - 117s 16ms/step - loss: 0.3216 - acc: 0.8384 - val_loss: 0.5007 - val_acc: 0.7659
Epoch 15/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.3143 - acc: 0.8561 - val_loss: 0.4441 - val_acc: 0.7866
Epoch 16/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.2966 - acc: 0.8690 - val_loss: 0.4284 - val_acc: 0.8700
Epoch 17/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.2814 - acc: 0.8943 - val_loss: 0.5250 - val_acc: 0.8605
Epoch 18/30
7352/7352 [=====] - 115s 16ms/step - loss: 0.2902 - acc: 0.9193 - val_loss: 0.3728 - val_acc: 0.8935
Epoch 19/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.2162 - acc: 0.9355 - val_loss: 0.4021 - val_acc: 0.8982
Epoch 20/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.2021 - acc: 0.9410 - val_loss: 0.3959 - val_acc: 0.8992
Epoch 21/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.1968 - acc: 0.9400 - val_loss: 0.4033 - val_acc: 0.8921
Epoch 22/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.1852 - acc: 0.9426 - val_loss: 0.4776 - val_acc: 0.8907
Epoch 23/30
7352/7352 [=====] - 115s 16ms/step - loss: 0.1781 - acc: 0.9406 - val_loss: 0.4668 - val_acc: 0.8941
Epoch 24/30
7352/7352 [=====] - 115s 16ms/step - loss: 0.1564 - acc: 0.9440 - val_loss: 0.4444 - val_acc: 0.9006
Epoch 25/30
7352/7352 [=====] - 116s 16ms/step - loss: 0.2013 - acc: 0.9400 - val_loss: 0.4416 - val_acc: 0.8948
Epoch 26/30
7352/7352 [=====] - 115s 16ms/step - loss: 0.1725 - acc: 0.9429 - val_loss: 0.4069 - val_acc: 0.8958
Epoch 27/30
7352/7352 [=====] - 115s 16ms/step - loss: 0.2030 - acc: 0.9350 - val_loss: 0.4044 - val_acc: 0.9023
Epoch 28/30
7352/7352 [=====] - 113s 15ms/step - loss: 0.1593 - acc: 0.9460 - val_loss: 0.5019 - val_acc: 0.8901
Epoch 29/30
7352/7352 [=====] - 113s 15ms/step - loss: 0.1805 - acc: 0.9407 - val_loss: 0.3900 - val_acc: 0.9080
Epoch 30/30
7352/7352 [=====] - 115s 16ms/step - loss: 0.1522 - acc: 0.9460 - val_loss: 0.5302 - val_acc: 0.8907

```

Out[0]:

```
<keras.callbacks.History at 0x7fa7103102e8>
```

In [0]:

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

Pred	LAYING	SITTING	...	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
True			...		
LAYING	510	0	...	0	27

```

SITTING      0    338 ...
STANDING     0    46 ...
WALKING      0     0 ...
WALKING_DOWNSTAIRS 0     0 ...
WALKING_UPSTAIRS 0     0 ...

```

[6 rows x 6 columns]

In [0]:

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

```
2947/2947 [=====] - 9s 3ms/step
```

In [0]:

```
score
```

Out[0]:

[0.5301554882347644, 0.8907363420427553]

- 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 [0]:

```
np.save(data_path + 'X_train.npy', X_train) # save X_train
np.save(data_path + 'Y_train.npy', Y_train) # save Y_train
np.save(data_path + 'X_test.npy', X_test) # save X_test
np.save(data_path + 'Y_test.npy', Y_test) # save Y_test
```

## Hyperparameter Tuning using hyperas

In [6]:

```
!pip install hyperas
```

```

Collecting hyperas
  Downloading
https://files.pythonhosted.org/packages/04/34/87ad6ffb42df9c1fa9c4c906f65813d42ad70d68c66af4ffff04&
cd4/hyperas-0.4.1-py3-none-any.whl
Requirement already satisfied: jupyter in /usr/local/lib/python3.6/dist-packages (from hyperas) (1.0.0)
Requirement already satisfied: hyperopt in /usr/local/lib/python3.6/dist-packages (from hyperas) (0.1.2)
Requirement already satisfied: keras in /usr/local/lib/python3.6/dist-packages (from hyperas) (2.2.5)
Requirement already satisfied: entrypoints in /usr/local/lib/python3.6/dist-packages (from hyperas) (0.3)
Requirement already satisfied: nbconvert in /usr/local/lib/python3.6/dist-packages (from hyperas) (5.6.0)
Requirement already satisfied: nbformat in /usr/local/lib/python3.6/dist-packages (from hyperas) (4.4.0)
Requirement already satisfied: qtconsole in /usr/local/lib/python3.6/dist-packages (from jupyter->hyperas) (4.5.5)
Requirement already satisfied: notebook in /usr/local/lib/python3.6/dist-packages (from jupyter->hyperas) (5.2.2)
Requirement already satisfied: ipykernel in /usr/local/lib/python3.6/dist-packages (from jupyter->hyperas) (4.6.1)
Requirement already satisfied: jupyter-console in /usr/local/lib/python3.6/dist-packages (from jupyter->hyperas) (5.2.0)
Requirement already satisfied: ipywidgets in /usr/local/lib/python3.6/dist-packages (from jupyter->hyperas) (7.5.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.6/dist-packages (from hyperopt->hyperas) (4.28.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.6/dist-packages (from hyperopt->hyperas) (2.3)

```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from hyperopt->hyperas) (1.16.5)
Requirement already satisfied: future in /usr/local/lib/python3.6/dist-packages (from hyperopt->hyperas) (0.16.0)
Requirement already satisfied: pymongo in /usr/local/lib/python3.6/dist-packages (from hyperopt->hyperas) (3.9.0)
Requirement already satisfied: scipy in /usr/local/lib/python3.6/dist-packages (from hyperopt->hyperas) (1.3.1)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from hyperopt->hyperas) (1.12.0)
Requirement already satisfied: h5py in /usr/local/lib/python3.6/dist-packages (from keras->hyperas) (2.8.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.6/dist-packages (from keras->hyperas) (3.13)
Requirement already satisfied: keras-preprocessing>=1.1.0 in /usr/local/lib/python3.6/dist-packages (from keras->hyperas) (1.1.0)
Requirement already satisfied: keras-applications>=1.0.8 in /usr/local/lib/python3.6/dist-packages (from keras->hyperas) (1.0.8)
Requirement already satisfied: testpath in /usr/local/lib/python3.6/dist-packages (from nbconvert->hyperas) (0.4.2)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.6/dist-packages (from nbconvert->hyperas) (0.6.0)
Requirement already satisfied: jupyter-core in /usr/local/lib/python3.6/dist-packages (from nbconvert->hyperas) (4.5.0)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.6/dist-packages (from nbconvert->hyperas) (1.4.2)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.6/dist-packages (from nbconvert->hyperas) (0.8.4)
Requirement already satisfied: bleach in /usr/local/lib/python3.6/dist-packages (from nbconvert->hyperas) (3.1.0)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.6/dist-packages (from nbconvert->hyperas) (4.3.2)
Requirement already satisfied: jinja2>=2.4 in /usr/local/lib/python3.6/dist-packages (from nbconvert->hyperas) (2.10.1)
Requirement already satisfied: pygments in /usr/local/lib/python3.6/dist-packages (from nbconvert->hyperas) (2.1.3)
Requirement already satisfied: ipython-genutils in /usr/local/lib/python3.6/dist-packages (from nbformat->hyperas) (0.2.0)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /usr/local/lib/python3.6/dist-packages (from nbformat->hyperas) (2.6.0)
Requirement already satisfied: jupyter-client>=4.1 in /usr/local/lib/python3.6/dist-packages (from qtconsole->jupyter->hyperas) (5.3.1)
Requirement already satisfied: tornado>=4 in /usr/local/lib/python3.6/dist-packages (from notebook->jupyter->hyperas) (4.5.3)
Requirement already satisfied: terminado>=0.3.3; sys_platform != "win32" in /usr/local/lib/python3.6/dist-packages (from notebook->jupyter->hyperas) (0.8.2)
Requirement already satisfied: ipython>=4.0.0 in /usr/local/lib/python3.6/dist-packages (from ipykernel->jupyter->hyperas) (5.5.0)
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.0 in /usr/local/lib/python3.6/dist-packages (from jupyter-console->jupyter->hyperas) (1.0.16)
Requirement already satisfied: widgetsnbextension~3.5.0 in /usr/local/lib/python3.6/dist-packages (from ipywidgets->jupyter->hyperas) (3.5.1)
Requirement already satisfied: decorator>=4.3.0 in /usr/local/lib/python3.6/dist-packages (from networkx->hyperopt->hyperas) (4.4.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.6/dist-packages (from bleach->nbconvert->hyperas) (0.5.1)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.6/dist-packages (from jinja2>=2.4->nbconvert->hyperas) (1.1.1)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.6/dist-packages (from jupyter-client>=4.1->qtconsole->jupyter->hyperas) (2.5.3)
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.6/dist-packages (from jupyter-client>=4.1->qtconsole->jupyter->hyperas) (17.0.0)
Requirement already satisfied: ptyprocess; os_name != "nt" in /usr/local/lib/python3.6/dist-packages (from terminado>=0.3.3; sys_platform != "win32">->notebook->jupyter->hyperas) (0.6.0)
Requirement already satisfied: simplegeneric>0.8 in /usr/local/lib/python3.6/dist-packages (from ipython>=4.0.0->ipykernel->jupyter->hyperas) (0.8.1)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.6/dist-packages (from ipython>=4.0.0->ipykernel->jupyter->hyperas) (41.2.0)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.6/dist-packages (from ipython>=4.0.0->ipykernel->jupyter->hyperas) (0.7.5)
Requirement already satisfied: pexpect; sys_platform != "win32" in /usr/local/lib/python3.6/dist-packages (from ipython>=4.0.0->ipykernel->jupyter->hyperas) (4.7.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.6/dist-packages (from prompt-toolkit<2.0.0,>=1.0.0->jupyter-console->jupyter->hyperas) (0.1.7)
Installing collected packages: hyperas
Successfully installed hyperas-0.4.1
```

In [7]:

```
#https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-i-hyper-parameter-8129009f131b

from hyperopt import Trials, STATUS_OK, tpe
from hyperas import optim
from hyperas.distributions import choice, uniform
```

Using TensorFlow backend.

In [0]:

```
import keras
```

In [0]:

```
def data():
    data_path = '/content/drive/My Drive/AAIC/Case Studies/Human Activity Recognition/'
    X_train = np.load(data_path + 'X_train.npy') # load X_train
    Y_train = np.load(data_path + 'Y_train.npy') # load Y_train
    X_val = np.load(data_path + 'X_test.npy') # load X_test
    Y_val = np.load(data_path + 'Y_test.npy') # load Y_test
    return X_train, Y_train, X_val, Y_val
```

In [0]:

```
def create_model(X_train, Y_train, X_val, Y_val):
    from keras.layers import Flatten
    model = Sequential()
    model.add(LSTM({{choice([4, 16, 32, 64, 128])}}, return_sequences=True, kernel_initializer="glorot_uniform", input_shape=(128,9), name='LSTMLayer1'))
    model.add(Dropout({{uniform(0, 1)}}))

    if {{choice(['one', 'two'])}} == 'two':
        model.add(LSTM({{choice([4, 16, 32, 64, 128])}}, return_sequences=True, kernel_initializer="glorot_uniform", name='LSTMLayer2'))

    model.add(Flatten())
    model.add(Dense(6, activation='softmax', name='DenseLayer1'))
    adam = keras.optimizers.Adam(lr={{choice([10**-3, 10**-2, 10**-1])}})
    rmsprop = keras.optimizers.RMSprop(lr={{choice([10**-3, 10**-2, 10**-1])}})
    sgd = keras.optimizers.SGD(lr={{choice([10**-3, 10**-2, 10**-1])}})

    choiceval = {{choice(['adam', 'sgd', 'rmsprop'])}}
    if choiceval == 'adam':
        optim = adam
    elif choiceval == 'rmsprop':
        optim = rmsprop
    else:
        optim = sgd

    model.compile(loss='categorical_crossentropy', metrics=['accuracy'], optimizer=optim)
    model.fit(X_train, Y_train,
              batch_size={{choice([64,128,256,512])}}},
              epochs=20,
              verbose=2,
              validation_data=(X_val, Y_val))
    score, acc = model.evaluate(X_val, Y_val, verbose=0)
    print('Test accuracy:', acc)
    return {'loss': -acc, 'status': STATUS_OK, 'model': model}
```

In [0]:

```
best_run, best_model = optim.minimize(model=create_model,
                                       data=data,
                                       algo=tpe.suggest,
                                       max_evals=30,
                                       trials=Trials(),
                                       notebook_name= 'drive/My Drive/AAIC/Case Studies/Human Activity Recognition/Human Activity Recognition')
```

```
>>> Imports:  
#coding=utf-8  
  
try:  
    from google.colab import drive  
except:  
    pass  
  
try:  
    import os  
except:  
    pass  
  
try:  
    import numpy as np  
except:  
    pass  
  
try:  
    import pandas as pd  
except:  
    pass  
  
try:  
    import matplotlib.pyplot as plt  
except:  
    pass  
  
try:  
    import seaborn as sns  
except:  
    pass  
  
try:  
    import numpy as np  
except:  
    pass  
  
try:  
    from sklearn.manifold import TSNE  
except:  
    pass  
  
try:  
    import matplotlib.pyplot as plt  
except:  
    pass  
  
try:  
    import seaborn as sns  
except:  
    pass  
  
try:  
    import itertools  
except:  
    pass  
  
try:  
    import numpy as np  
except:  
    pass  
  
try:  
    import matplotlib.pyplot as plt  
except:  
    pass  
  
try:  
    from sklearn.metrics import confusion_matrix  
except:  
    pass  
  
try:  
    from datetime import datetime  
except:
```

```
except:
    pass

try:
    from sklearn import linear_model
except:
    pass

try:
    from sklearn import metrics
except:
    pass

try:
    from sklearn.model_selection import GridSearchCV
except:
    pass

try:
    from sklearn.svm import SVC
except:
    pass

try:
    from sklearn.tree import DecisionTreeClassifier
except:
    pass

try:
    from sklearn.ensemble import RandomForestClassifier
except:
    pass

try:
    from sklearn.ensemble import GradientBoostingClassifier
except:
    pass

try:
    import tensorflow as tf
except:
    pass

try:
    from keras import backend as K
except:
    pass

try:
    from keras.models import Sequential
except:
    pass

try:
    from keras.layers import LSTM
except:
    pass

try:
    from keras.layers.core import Dense, Dropout
except:
    pass

try:
    from hyperopt import Trials, STATUS_OK, tpe
except:
    pass

try:
    from hyperas import optim
except:
    pass

try:
    from hyperas.distributions import choice, uniform
except:
    pass
```

```

try:
    import keras
except:
    pass

try:
    from keras.layers import Flatten
except:
    pass

>>> Hyperas search space:

def get_space():
    return {
        'LSTM': hp.choice('LSTM', [4, 16, 32, 64, 128]),
        'Dropout': hp.uniform('Dropout', 0, 1),
        'if': hp.choice('if', ['one', 'two']),
        'LSTM_1': hp.choice('LSTM_1', [4, 16, 32, 64, 128]),
        'lr': hp.choice('lr', [10**-3, 10**-2, 10**-1]),
        'lr_1': hp.choice('lr_1', [10**-3, 10**-2, 10**-1]),
        'lr_2': hp.choice('lr_2', [10**-3, 10**-2, 10**-1]),
        'choiceval': hp.choice('choiceval', ['adam', 'sgd', 'rmsprop']),
        'batch_size': hp.choice('batch_size', [64, 128, 256, 512]),
    }

>>> Data
1:
2: data_path = '/content/drive/My Drive/AAIC/Case Studies/Human Activity Recognition/'
3: X_train = np.load(data_path + 'X_train.npy') # load X_train
4: Y_train = np.load(data_path + 'Y_train.npy') # load Y_train
5: X_val = np.load(data_path + 'X_test.npy') # load X_test
6: Y_val = np.load(data_path + 'Y_test.npy') # load Y_test
7:
8:
9:
>>> Resulting replaced keras model:

1: def keras_fmin_fnct(space):
2:
3:     model = Sequential()
4:     model.add(LSTM(space['LSTM'], return_sequences=True, kernel_initializer="glorot_uniform",
input_shape=(128,9), name='LSTMLayer1'))
5:     model.add(Dropout(space['Dropout']))
6:
7:     if (space['if']) == 'two':
8:         model.add(LSTM(space['LSTM_1'], return_sequences=True,
kernel_initializer="glorot_uniform", name='LSTMLayer2'))
9:
10:    model.add(Flatten())
11:    model.add(Dense(6, activation='softmax', name='DenseLayer1'))
12:    adam = keras.optimizers.Adam(lr=space['lr'])
13:    rmsprop = keras.optimizers.RMSprop(lr=space['lr_1'])
14:    sgd = keras.optimizers.SGD(lr=space['lr_2'])
15:
16:    choiceval = space['choiceval']
17:    if choiceval == 'adam':
18:        optim = adam
19:    elif choiceval == 'rmsprop':
20:        optim = rmsprop
21:    else:
22:        optim = sgd
23:
24:    model.compile(loss='categorical_crossentropy', metrics=['accuracy'], optimizer=optim)
25:    model.fit(X_train, Y_train,
26:              batch_size=space['batch_size'],
27:              epochs=20,
28:              verbose=2,
29:              validation_data=(X_val, Y_val))
30:    score, acc = model.evaluate(X_val, Y_val, verbose=0)
31:    print('Test accuracy:', acc)
32:    return {'loss': -acc, 'status': STATUS_OK, 'model': model}
33:
0%|          | 0/30 [00:00<?, ?it/s, best loss: ?]WARNING:tensorflow:From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:66: The name
tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-

```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:541: The name tf.placeholder is deprecated. Please use 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. Please 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/python/ops/math_grad.py:1250: add_dispatch_support.<locals>.wrapper (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
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 21s - loss: 1.1919 - acc: 0.5037 - val_loss: 1.1392 - val_acc: 0.4917

Epoch 2/20
- 16s - loss: 0.9307 - acc: 0.6189 - val_loss: 0.9888 - val_acc: 0.5355

Epoch 3/20
- 16s - loss: 0.8304 - acc: 0.7025 - val_loss: 0.7789 - val_acc: 0.7095

Epoch 4/20
- 16s - loss: 0.5255 - acc: 0.7942 - val_loss: 0.6812 - val_acc: 0.7618

Epoch 5/20
- 16s - loss: 0.7681 - acc: 0.7511 - val_loss: 0.7051 - val_acc: 0.7414

Epoch 6/20
- 16s - loss: 0.4056 - acc: 0.8474 - val_loss: 0.6296 - val_acc: 0.7591

Epoch 7/20
- 16s - loss: 0.3345 - acc: 0.8709 - val_loss: 0.6150 - val_acc: 0.7733

Epoch 8/20
- 16s - loss: 0.2921 - acc: 0.8919 - val_loss: 0.4824 - val_acc: 0.8286

Epoch 9/20
- 16s - loss: 0.3170 - acc: 0.8825 - val_loss: 0.4769 - val_acc: 0.8303

Epoch 10/20
- 16s - loss: 0.2267 - acc: 0.9176 - val_loss: 0.4843 - val_acc: 0.8354

Epoch 11/20
- 16s - loss: 0.2177 - acc: 0.9188 - val_loss: 0.5370 - val_acc: 0.8229

Epoch 12/20
- 16s - loss: 0.1803 - acc: 0.9340 - val_loss: 0.3747 - val_acc: 0.8921

Epoch 13/20
- 16s - loss: 0.1809 - acc: 0.9308 - val_loss: 0.3602 - val_acc: 0.8918

Epoch 14/20
- 16s - loss: 0.1727 - acc: 0.9342 - val_loss: 0.3803 - val_acc: 0.8775

Epoch 15/20
- 16s - loss: 0.1468 - acc: 0.9410 - val_loss: 0.3108 - val_acc: 0.9026

Epoch 16/20
- 16s - loss: 0.1422 - acc: 0.9455 - val_loss: 0.2915 - val_acc: 0.9050
```

```
- 16s - loss: 0.1452 - acc: 0.9455 - val_loss: 0.5215 - val_acc: 0.0950

Epoch 17/20
- 16s - loss: 0.8106 - acc: 0.8507 - val_loss: 0.5480 - val_acc: 0.8059

Epoch 18/20
- 16s - loss: 0.2119 - acc: 0.9215 - val_loss: 0.4127 - val_acc: 0.8643

Epoch 19/20
- 16s - loss: 0.1725 - acc: 0.9354 - val_loss: 0.3617 - val_acc: 0.8755

Epoch 20/20
- 16s - loss: 0.1473 - acc: 0.9437 - val_loss: 0.3029 - val_acc: 0.8996

Test accuracy:
0.8995588734306074
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 17s - loss: 12.8421 - acc: 0.1904 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 2/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 3/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 4/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 5/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 6/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 7/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 8/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 9/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 10/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 11/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 12/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 13/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 14/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 15/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 16/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 17/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 18/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 19/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 20/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822
```

```
test accuracy:  
0.18221920597217509  
    7%|██████████| 2/30 [11:10<2:36:17, 334.93s/it, best loss: -  
0.8995588734306074]WARNING:tensorflow:Large dropout rate: 0.635788 (>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 1/20  
- 6s - loss: 1.6399 - acc: 0.3634 - val_loss: 1.5338 - val_acc: 0.3902  
  
Epoch 2/20  
- 4s - loss: 1.4476 - acc: 0.4342 - val_loss: 1.4174 - val_acc: 0.4408  
  
Epoch 3/20  
- 4s - loss: 1.3501 - acc: 0.4733 - val_loss: 1.3570 - val_acc: 0.4571  
  
Epoch 4/20  
- 5s - loss: 1.2904 - acc: 0.4988 - val_loss: 1.3177 - val_acc: 0.4764  
  
Epoch 5/20  
- 5s - loss: 1.2528 - acc: 0.5113 - val_loss: 1.2891 - val_acc: 0.4835  
  
Epoch 6/20  
- 5s - loss: 1.2201 - acc: 0.5200 - val_loss: 1.2660 - val_acc: 0.5049  
  
Epoch 7/20  
- 5s - loss: 1.1968 - acc: 0.5248 - val_loss: 1.2476 - val_acc: 0.4958  
  
Epoch 8/20  
- 5s - loss: 1.1756 - acc: 0.5273 - val_loss: 1.2312 - val_acc: 0.5005  
  
Epoch 9/20  
- 4s - loss: 1.1573 - acc: 0.5371 - val_loss: 1.2175 - val_acc: 0.5049  
  
Epoch 10/20  
- 5s - loss: 1.1406 - acc: 0.5416 - val_loss: 1.2076 - val_acc: 0.5070  
  
Epoch 11/20  
- 5s - loss: 1.1260 - acc: 0.5416 - val_loss: 1.1982 - val_acc: 0.5056  
  
Epoch 12/20  
- 4s - loss: 1.1080 - acc: 0.5524 - val_loss: 1.1904 - val_acc: 0.5002  
  
Epoch 13/20  
- 4s - loss: 1.0991 - acc: 0.5480 - val_loss: 1.1840 - val_acc: 0.5056  
  
Epoch 14/20  
- 5s - loss: 1.0862 - acc: 0.5555 - val_loss: 1.1796 - val_acc: 0.5066  
  
Epoch 15/20  
- 4s - loss: 1.0766 - acc: 0.5537 - val_loss: 1.1795 - val_acc: 0.5076  
  
Epoch 16/20  
- 4s - loss: 1.0649 - acc: 0.5544 - val_loss: 1.1776 - val_acc: 0.5226  
  
Epoch 17/20  
- 5s - loss: 1.0552 - acc: 0.5605 - val_loss: 1.1760 - val_acc: 0.5117  
  
Epoch 18/20  
- 5s - loss: 1.0438 - acc: 0.5623 - val_loss: 1.1775 - val_acc: 0.5137  
  
Epoch 19/20  
- 5s - loss: 1.0383 - acc: 0.5665 - val_loss: 1.1798 - val_acc: 0.5120  
  
Epoch 20/20  
- 5s - loss: 1.0267 - acc: 0.5686 - val_loss: 1.1837 - val_acc: 0.5161  
  
Test accuracy:  
0.5161180862095707  
Train on 7352 samples, validate on 2947 samples  
Epoch 1/20  
- 11s - loss: 9.7118 - acc: 0.3229 - val_loss: 10.6816 - val_acc: 0.3373  
  
Epoch 2/20  
- 9s - loss: 10.4531 - acc: 0.3515 - val_loss: 10.6816 - val_acc: 0.3373  
  
Epoch 3/20  
- 9s - loss: 10.4553 - acc: 0.3513 - val_loss: 10.6816 - val_acc: 0.3373
```

```
Epoch 4/20
- 9s - loss: 10.4487 - acc: 0.3517 - val_loss: 10.6816 - val_acc: 0.3373

Epoch 5/20
- 9s - loss: 10.4465 - acc: 0.3519 - val_loss: 10.6816 - val_acc: 0.3373

Epoch 6/20
- 9s - loss: 10.4509 - acc: 0.3516 - val_loss: 10.6816 - val_acc: 0.3373

Epoch 7/20
- 9s - loss: 10.4465 - acc: 0.3519 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 8/20
- 9s - loss: 10.4114 - acc: 0.3541 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 9/20
- 9s - loss: 10.4114 - acc: 0.3541 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 10/20
- 9s - loss: 10.4092 - acc: 0.3542 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 11/20
- 9s - loss: 10.4114 - acc: 0.3541 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 12/20
- 9s - loss: 10.4114 - acc: 0.3541 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 13/20
- 9s - loss: 10.4092 - acc: 0.3542 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 14/20
- 9s - loss: 10.4180 - acc: 0.3536 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 15/20
- 9s - loss: 10.4136 - acc: 0.3539 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 16/20
- 9s - loss: 10.4180 - acc: 0.3536 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 17/20
- 9s - loss: 10.4202 - acc: 0.3535 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 18/20
- 9s - loss: 10.4049 - acc: 0.3545 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 19/20
- 9s - loss: 10.4092 - acc: 0.3542 - val_loss: 10.6160 - val_acc: 0.3414

Epoch 20/20
- 9s - loss: 10.4005 - acc: 0.3547 - val_loss: 10.6160 - val_acc: 0.3414

Test accuracy:
0.34136409908381404
13%|██████████| 4/30 [16:10<1:46:05, 244.84s/it, best loss: -
0.8995588734306074]WARNING:tensorflow:Large dropout rate: 0.537372 (>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 1/20
- 11s - loss: 11.4109 - acc: 0.2380 - val_loss: 12.5302 - val_acc: 0.2226

Epoch 2/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226

Epoch 3/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226

Epoch 4/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226

Epoch 5/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226

Epoch 6/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226

Epoch 7/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
```

```
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 8/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 9/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 10/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 11/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 12/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 13/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 14/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 15/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 16/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 17/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 18/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 19/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226
Epoch 20/20
- 9s - loss: 12.1390 - acc: 0.2469 - val_loss: 12.5302 - val_acc: 0.2226

Test accuracy:
0.22259925347811332
17%|██████████| 5/30 [19:30<1:36:29, 231.58s/it, best loss: -
0.8995588734306074]WARNING:tensorflow:Large dropout rate: 0.626462 (>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 1/20
- 19s - loss: 12.7978 - acc: 0.1933 - val_loss: 13.1811 - val_acc: 0.1822
Epoch 2/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822
Epoch 3/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822
Epoch 4/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822
Epoch 5/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822
Epoch 6/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822
Epoch 7/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822
Epoch 8/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822
Epoch 9/20
- 16s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822
Epoch 10/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822
- - - - -
```

```
Epoch 11/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 12/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 13/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 14/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 15/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 16/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 17/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 18/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 19/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Epoch 20/20
- 17s - loss: 13.0335 - acc: 0.1914 - val_loss: 13.1811 - val_acc: 0.1822

Test accuracy:
0.18221920597217509
20%|██████| 6/30 [25:16<1:46:16, 265.70s/it, best loss: -
0.8995588734306074]WARNING:tensorflow:Large dropout rate: 0.769005 (>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 1/20
- 19s - loss: 12.3919 - acc: 0.2073 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 2/20
- 16s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 3/20
- 16s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 4/20
- 16s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 5/20
- 17s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 6/20
- 17s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 7/20
- 17s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 8/20
- 16s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 9/20
- 16s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 10/20
- 17s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 11/20
- 17s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 12/20
- 16s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 13/20
- 17s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 14/20
- 17s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805
```

```
Epoch 15/20
- 17s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 16/20
- 16s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 17/20
- 17s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 18/20
- 17s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 19/20
- 17s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Epoch 20/20
- 17s - loss: 13.1058 - acc: 0.1869 - val_loss: 13.2084 - val_acc: 0.1805

Test accuracy:
0.18052256532066507
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 35s - loss: 0.7212 - acc: 0.7576 - val_loss: 0.5872 - val_acc: 0.8015

Epoch 2/20
- 33s - loss: 0.2344 - acc: 0.9106 - val_loss: 0.3871 - val_acc: 0.8935

Epoch 3/20
- 33s - loss: 0.1708 - acc: 0.9327 - val_loss: 0.3717 - val_acc: 0.9063

Epoch 4/20
- 33s - loss: 0.1521 - acc: 0.9415 - val_loss: 0.3909 - val_acc: 0.8968

Epoch 5/20
- 32s - loss: 0.1365 - acc: 0.9416 - val_loss: 0.4940 - val_acc: 0.8928

Epoch 6/20
- 33s - loss: 0.1345 - acc: 0.9448 - val_loss: 0.5208 - val_acc: 0.8873

Epoch 7/20
- 33s - loss: 0.1228 - acc: 0.9480 - val_loss: 0.4535 - val_acc: 0.9108

Epoch 8/20
- 33s - loss: 0.1254 - acc: 0.9521 - val_loss: 0.4979 - val_acc: 0.9080

Epoch 9/20
- 33s - loss: 0.1216 - acc: 0.9513 - val_loss: 0.3559 - val_acc: 0.9169

Epoch 10/20
- 33s - loss: 0.1178 - acc: 0.9509 - val_loss: 0.4184 - val_acc: 0.9074

Epoch 11/20
- 33s - loss: 0.1151 - acc: 0.9502 - val_loss: 0.4604 - val_acc: 0.8968

Epoch 12/20
- 33s - loss: 0.1089 - acc: 0.9539 - val_loss: 0.4275 - val_acc: 0.9087

Epoch 13/20
- 32s - loss: 0.1066 - acc: 0.9542 - val_loss: 0.5732 - val_acc: 0.9043

Epoch 14/20
- 33s - loss: 0.1080 - acc: 0.9538 - val_loss: 0.4875 - val_acc: 0.9050

Epoch 15/20
- 34s - loss: 0.1089 - acc: 0.9569 - val_loss: 0.5383 - val_acc: 0.9030

Epoch 16/20
- 33s - loss: 0.1011 - acc: 0.9570 - val_loss: 0.5396 - val_acc: 0.9013

Epoch 17/20
- 33s - loss: 0.1001 - acc: 0.9589 - val_loss: 0.5264 - val_acc: 0.9087

Epoch 18/20
- 33s - loss: 0.0967 - acc: 0.9604 - val_loss: 0.6049 - val_acc: 0.9043

Epoch 19/20
```

```
- 33s - loss: 0.1034 - acc: 0.9597 - val_loss: 0.5696 - val_acc: 0.9046

Epoch 20/20
- 33s - loss: 0.0919 - acc: 0.9610 - val_loss: 0.6098 - val_acc: 0.9033

Test accuracy:
0.9032914828639295
27%|██████████| 8/30 [42:09<2:27:50, 403.22s/it, best loss: -
0.9032914828639295]WARNING:tensorflow:Large dropout rate: 0.815976 (>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 1/20
- 12s - loss: 1.7987 - acc: 0.1840 - val_loss: 1.7940 - val_acc: 0.2083

Epoch 2/20
- 9s - loss: 1.7937 - acc: 0.1993 - val_loss: 1.7902 - val_acc: 0.2162

Epoch 3/20
- 9s - loss: 1.7915 - acc: 0.2023 - val_loss: 1.7864 - val_acc: 0.2189

Epoch 4/20
- 9s - loss: 1.7869 - acc: 0.2172 - val_loss: 1.7828 - val_acc: 0.2202

Epoch 5/20
- 9s - loss: 1.7827 - acc: 0.2239 - val_loss: 1.7792 - val_acc: 0.2229

Epoch 6/20
- 9s - loss: 1.7779 - acc: 0.2456 - val_loss: 1.7757 - val_acc: 0.2314

Epoch 7/20
- 9s - loss: 1.7758 - acc: 0.2514 - val_loss: 1.7722 - val_acc: 0.2416

Epoch 8/20
- 9s - loss: 1.7730 - acc: 0.2607 - val_loss: 1.7688 - val_acc: 0.2562

Epoch 9/20
- 9s - loss: 1.7701 - acc: 0.2665 - val_loss: 1.7654 - val_acc: 0.2691

Epoch 10/20
- 9s - loss: 1.7672 - acc: 0.2794 - val_loss: 1.7620 - val_acc: 0.2786

Epoch 11/20
- 9s - loss: 1.7621 - acc: 0.2897 - val_loss: 1.7588 - val_acc: 0.2820

Epoch 12/20
- 9s - loss: 1.7597 - acc: 0.2941 - val_loss: 1.7554 - val_acc: 0.2911

Epoch 13/20
- 9s - loss: 1.7542 - acc: 0.3107 - val_loss: 1.7522 - val_acc: 0.2945

Epoch 14/20
- 9s - loss: 1.7516 - acc: 0.3220 - val_loss: 1.7489 - val_acc: 0.3017

Epoch 15/20
- 9s - loss: 1.7488 - acc: 0.3248 - val_loss: 1.7457 - val_acc: 0.3095

Epoch 16/20
- 9s - loss: 1.7460 - acc: 0.3271 - val_loss: 1.7425 - val_acc: 0.3163

Epoch 17/20
- 9s - loss: 1.7422 - acc: 0.3428 - val_loss: 1.7392 - val_acc: 0.3244

Epoch 18/20
- 9s - loss: 1.7391 - acc: 0.3490 - val_loss: 1.7360 - val_acc: 0.3346

Epoch 19/20
- 9s - loss: 1.7367 - acc: 0.3498 - val_loss: 1.7327 - val_acc: 0.3529

Epoch 20/20
- 9s - loss: 1.7329 - acc: 0.3496 - val_loss: 1.7295 - val_acc: 0.3845

Test accuracy:
0.3844587716321683
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 36s - loss: 0.7640 - acc: 0.6906 - val_loss: 0.7743 - val_acc: 0.7143
```

```
Epoch 2/20
- 33s - loss: 0.3398 - acc: 0.8753 - val_loss: 0.6216 - val_acc: 0.8056

Epoch 3/20
- 33s - loss: 0.2142 - acc: 0.9184 - val_loss: 0.4598 - val_acc: 0.8765

Epoch 4/20
- 33s - loss: 0.1587 - acc: 0.9351 - val_loss: 0.3656 - val_acc: 0.8907

Epoch 5/20
- 33s - loss: 0.1423 - acc: 0.9434 - val_loss: 0.3783 - val_acc: 0.8996

Epoch 6/20
- 33s - loss: 0.1303 - acc: 0.9464 - val_loss: 0.3912 - val_acc: 0.8941

Epoch 7/20
- 33s - loss: 0.1215 - acc: 0.9479 - val_loss: 0.4324 - val_acc: 0.8901

Epoch 8/20
- 33s - loss: 0.1258 - acc: 0.9471 - val_loss: 0.3249 - val_acc: 0.9152

Epoch 9/20
- 33s - loss: 0.1153 - acc: 0.9504 - val_loss: 0.3911 - val_acc: 0.9043

Epoch 10/20
- 33s - loss: 0.1194 - acc: 0.9494 - val_loss: 0.3265 - val_acc: 0.9046

Epoch 11/20
- 33s - loss: 0.1162 - acc: 0.9501 - val_loss: 0.3656 - val_acc: 0.9043

Epoch 12/20
- 33s - loss: 0.1140 - acc: 0.9499 - val_loss: 0.3442 - val_acc: 0.9070

Epoch 13/20
- 33s - loss: 0.1113 - acc: 0.9518 - val_loss: 0.3457 - val_acc: 0.9033

Epoch 14/20
- 33s - loss: 0.1159 - acc: 0.9508 - val_loss: 0.3537 - val_acc: 0.9108

Epoch 15/20
- 33s - loss: 0.1100 - acc: 0.9494 - val_loss: 0.3974 - val_acc: 0.9053

Epoch 16/20
- 33s - loss: 0.1071 - acc: 0.9508 - val_loss: 0.4000 - val_acc: 0.9128

Epoch 17/20
- 33s - loss: 0.1067 - acc: 0.9528 - val_loss: 0.3813 - val_acc: 0.9148

Epoch 18/20
- 33s - loss: 0.1010 - acc: 0.9558 - val_loss: 0.4151 - val_acc: 0.9104

Epoch 19/20
- 33s - loss: 0.0996 - acc: 0.9538 - val_loss: 0.3665 - val_acc: 0.9152

Epoch 20/20
- 33s - loss: 0.1034 - acc: 0.9532 - val_loss: 0.4693 - val_acc: 0.9060

Test accuracy:
0.9060061079063454
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 8s - loss: 1.7608 - acc: 0.2359 - val_loss: 1.6369 - val_acc: 0.4235

Epoch 2/20
- 4s - loss: 1.5968 - acc: 0.3687 - val_loss: 1.5178 - val_acc: 0.4649

Epoch 3/20
- 4s - loss: 1.4854 - acc: 0.4136 - val_loss: 1.4357 - val_acc: 0.4713

Epoch 4/20
- 4s - loss: 1.4153 - acc: 0.4173 - val_loss: 1.3849 - val_acc: 0.4798

Epoch 5/20
- 4s - loss: 1.3689 - acc: 0.4467 - val_loss: 1.3510 - val_acc: 0.4852

Epoch 6/20
- 4s - loss: 1.3339 - acc: 0.4601 - val_loss: 1.3252 - val_acc: 0.4978
```

```
Epoch 7/20
- 4s - loss: 1.3059 - acc: 0.4671 - val_loss: 1.3043 - val_acc: 0.5083

Epoch 8/20
- 4s - loss: 1.2824 - acc: 0.4720 - val_loss: 1.2861 - val_acc: 0.5171

Epoch 9/20
- 4s - loss: 1.2617 - acc: 0.4859 - val_loss: 1.2697 - val_acc: 0.5144

Epoch 10/20
- 4s - loss: 1.2414 - acc: 0.5004 - val_loss: 1.2552 - val_acc: 0.5110

Epoch 11/20
- 5s - loss: 1.2222 - acc: 0.5056 - val_loss: 1.2407 - val_acc: 0.5148

Epoch 12/20
- 5s - loss: 1.2068 - acc: 0.5087 - val_loss: 1.2272 - val_acc: 0.5212

Epoch 13/20
- 5s - loss: 1.1900 - acc: 0.5140 - val_loss: 1.2151 - val_acc: 0.5188

Epoch 14/20
- 4s - loss: 1.1778 - acc: 0.5197 - val_loss: 1.2043 - val_acc: 0.5205

Epoch 15/20
- 4s - loss: 1.1600 - acc: 0.5257 - val_loss: 1.1933 - val_acc: 0.5195

Epoch 16/20
- 4s - loss: 1.1458 - acc: 0.5316 - val_loss: 1.1831 - val_acc: 0.5243

Epoch 17/20
- 4s - loss: 1.1324 - acc: 0.5396 - val_loss: 1.1729 - val_acc: 0.5361

Epoch 18/20
- 4s - loss: 1.1205 - acc: 0.5408 - val_loss: 1.1644 - val_acc: 0.5331

Epoch 19/20
- 4s - loss: 1.1081 - acc: 0.5411 - val_loss: 1.1557 - val_acc: 0.5324

Epoch 20/20
- 5s - loss: 1.0942 - acc: 0.5562 - val_loss: 1.1474 - val_acc: 0.5348

Test accuracy:
0.534781133366068
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 12s - loss: 1.5649 - acc: 0.4181 - val_loss: 1.4406 - val_acc: 0.4408

Epoch 2/20
- 8s - loss: 1.3352 - acc: 0.4962 - val_loss: 1.3343 - val_acc: 0.4652

Epoch 3/20
- 8s - loss: 1.2495 - acc: 0.5291 - val_loss: 1.2825 - val_acc: 0.4723

Epoch 4/20
- 8s - loss: 1.1970 - acc: 0.5393 - val_loss: 1.2461 - val_acc: 0.4961

Epoch 5/20
- 8s - loss: 1.1578 - acc: 0.5472 - val_loss: 1.2194 - val_acc: 0.5073

Epoch 6/20
- 8s - loss: 1.1263 - acc: 0.5453 - val_loss: 1.1974 - val_acc: 0.5127

Epoch 7/20
- 9s - loss: 1.1006 - acc: 0.5573 - val_loss: 1.1818 - val_acc: 0.5192

Epoch 8/20
- 9s - loss: 1.0775 - acc: 0.5681 - val_loss: 1.1731 - val_acc: 0.5025

Epoch 9/20
- 9s - loss: 1.0570 - acc: 0.5683 - val_loss: 1.1661 - val_acc: 0.5195

Epoch 10/20
- 9s - loss: 1.0382 - acc: 0.5811 - val_loss: 1.1638 - val_acc: 0.5185

Epoch 11/20
```

```
- 9s - loss: 1.0191 - acc: 0.5808 - val_loss: 1.1633 - val_acc: 0.5239
Epoch 12/20
- 9s - loss: 1.0041 - acc: 0.5895 - val_loss: 1.1684 - val_acc: 0.5151
Epoch 13/20
- 8s - loss: 0.9886 - acc: 0.5894 - val_loss: 1.1688 - val_acc: 0.5256
Epoch 14/20
- 9s - loss: 0.9750 - acc: 0.6012 - val_loss: 1.1746 - val_acc: 0.5222
Epoch 15/20
- 9s - loss: 0.9642 - acc: 0.6032 - val_loss: 1.1729 - val_acc: 0.5450
Epoch 16/20
- 9s - loss: 0.9539 - acc: 0.6110 - val_loss: 1.1817 - val_acc: 0.5334
Epoch 17/20
- 9s - loss: 0.9434 - acc: 0.6151 - val_loss: 1.1748 - val_acc: 0.5426
Epoch 18/20
- 8s - loss: 0.9336 - acc: 0.6255 - val_loss: 1.1759 - val_acc: 0.5395
Epoch 19/20
- 8s - loss: 0.9208 - acc: 0.6273 - val_loss: 1.1672 - val_acc: 0.5545
Epoch 20/20
- 9s - loss: 0.9082 - acc: 0.6349 - val_loss: 1.1587 - val_acc: 0.5619
Test accuracy:
0.5619273837801154
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 21s - loss: 1.4988 - acc: 0.4085 - val_loss: 1.3680 - val_acc: 0.4740
Epoch 2/20
- 16s - loss: 1.2542 - acc: 0.5022 - val_loss: 1.2594 - val_acc: 0.4825
Epoch 3/20
- 17s - loss: 1.1535 - acc: 0.5205 - val_loss: 1.1968 - val_acc: 0.4998
Epoch 4/20
- 17s - loss: 1.0883 - acc: 0.5345 - val_loss: 1.1705 - val_acc: 0.5005
Epoch 5/20
- 17s - loss: 1.0474 - acc: 0.5482 - val_loss: 1.1508 - val_acc: 0.5185
Epoch 6/20
- 17s - loss: 1.0141 - acc: 0.5698 - val_loss: 1.1260 - val_acc: 0.5127
Epoch 7/20
- 17s - loss: 0.9773 - acc: 0.5881 - val_loss: 1.1033 - val_acc: 0.5331
Epoch 8/20
- 17s - loss: 0.9401 - acc: 0.6134 - val_loss: 1.0830 - val_acc: 0.5633
Epoch 9/20
- 17s - loss: 0.9045 - acc: 0.6362 - val_loss: 1.2196 - val_acc: 0.5185
Epoch 10/20
- 17s - loss: 0.8595 - acc: 0.6634 - val_loss: 0.9944 - val_acc: 0.5996
Epoch 11/20
- 17s - loss: 0.8041 - acc: 0.6854 - val_loss: 0.9412 - val_acc: 0.6244
Epoch 12/20
- 17s - loss: 0.7498 - acc: 0.7122 - val_loss: 0.9123 - val_acc: 0.6471
Epoch 13/20
- 17s - loss: 0.7202 - acc: 0.7224 - val_loss: 1.0370 - val_acc: 0.5809
Epoch 14/20
- 17s - loss: 0.6647 - acc: 0.7537 - val_loss: 0.9586 - val_acc: 0.6152
Epoch 15/20
- 17s - loss: 0.6292 - acc: 0.7697 - val_loss: 0.7461 - val_acc: 0.7075
```

```
Epoch 16/20
- 17s - loss: 0.5808 - acc: 0.7954 - val_loss: 0.7016 - val_acc: 0.7316

Epoch 17/20
- 17s - loss: 0.5431 - acc: 0.8143 - val_loss: 0.6910 - val_acc: 0.7306

Epoch 18/20
- 17s - loss: 0.5104 - acc: 0.8236 - val_loss: 0.6693 - val_acc: 0.7513

Epoch 19/20
- 17s - loss: 0.4825 - acc: 0.8364 - val_loss: 0.6233 - val_acc: 0.7506

Epoch 20/20
- 17s - loss: 0.4513 - acc: 0.8418 - val_loss: 0.6429 - val_acc: 0.7411

Test accuracy:
0.7410926365795725
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 22s - loss: 1.8368 - acc: 0.4547 - val_loss: 0.9987 - val_acc: 0.5874

Epoch 2/20
- 17s - loss: 0.9351 - acc: 0.6420 - val_loss: 0.7743 - val_acc: 0.6848

Epoch 3/20
- 17s - loss: 0.6556 - acc: 0.7535 - val_loss: 0.8230 - val_acc: 0.6485

Epoch 4/20
- 17s - loss: 0.4659 - acc: 0.8194 - val_loss: 0.6731 - val_acc: 0.7384

Epoch 5/20
- 17s - loss: 0.3379 - acc: 0.8721 - val_loss: 0.5055 - val_acc: 0.8588

Epoch 6/20
- 17s - loss: 0.2955 - acc: 0.8864 - val_loss: 0.6013 - val_acc: 0.8137

Epoch 7/20
- 17s - loss: 0.3142 - acc: 0.8883 - val_loss: 0.4540 - val_acc: 0.8806

Epoch 8/20
- 17s - loss: 0.1954 - acc: 0.9206 - val_loss: 0.4809 - val_acc: 0.8765

Epoch 9/20
- 17s - loss: 0.2519 - acc: 0.9070 - val_loss: 0.4455 - val_acc: 0.8880

Epoch 10/20
- 17s - loss: 0.2028 - acc: 0.9197 - val_loss: 0.4164 - val_acc: 0.8907

Epoch 11/20
- 17s - loss: 0.2132 - acc: 0.9238 - val_loss: 0.4232 - val_acc: 0.8914

Epoch 12/20
- 17s - loss: 0.1747 - acc: 0.9274 - val_loss: 0.4326 - val_acc: 0.8884

Epoch 13/20
- 17s - loss: 0.1735 - acc: 0.9329 - val_loss: 0.4708 - val_acc: 0.8921

Epoch 14/20
- 17s - loss: 0.1506 - acc: 0.9419 - val_loss: 0.5219 - val_acc: 0.8697

Epoch 15/20
- 17s - loss: 0.1715 - acc: 0.9338 - val_loss: 0.5266 - val_acc: 0.8758

Epoch 16/20
- 17s - loss: 0.1510 - acc: 0.9363 - val_loss: 0.4452 - val_acc: 0.8945

Epoch 17/20
- 17s - loss: 0.2154 - acc: 0.9354 - val_loss: 0.4394 - val_acc: 0.8921

Epoch 18/20
- 17s - loss: 0.1227 - acc: 0.9493 - val_loss: 0.9273 - val_acc: 0.8120

Epoch 19/20
- 17s - loss: 0.1589 - acc: 0.9423 - val_loss: 0.4891 - val_acc: 0.8904

Epoch 20/20
- 17s - loss: 0.1430 - acc: 0.9422 - val_loss: 0.5199 - val_acc: 0.8938
```

Test accuracy:  
0.8937902952154734  
Train on 7352 samples, validate on 2947 samples  
Epoch 1/20  
- 39s - loss: 1.6329 - acc: 0.4210 - val\_loss: 1.4499 - val\_acc: 0.4191

Epoch 2/20  
- 33s - loss: 1.3170 - acc: 0.4759 - val\_loss: 1.2974 - val\_acc: 0.4228

Epoch 3/20  
- 34s - loss: 1.1970 - acc: 0.5156 - val\_loss: 1.2148 - val\_acc: 0.4937

Epoch 4/20  
- 34s - loss: 1.1067 - acc: 0.5393 - val\_loss: 1.1691 - val\_acc: 0.5029

Epoch 5/20  
- 33s - loss: 1.0443 - acc: 0.5520 - val\_loss: 1.1584 - val\_acc: 0.5090

Epoch 6/20  
- 33s - loss: 0.9971 - acc: 0.5743 - val\_loss: 1.1539 - val\_acc: 0.4941

Epoch 7/20  
- 33s - loss: 0.9658 - acc: 0.5915 - val\_loss: 1.1647 - val\_acc: 0.5382

Epoch 8/20  
- 34s - loss: 0.9218 - acc: 0.6189 - val\_loss: 1.1152 - val\_acc: 0.5562

Epoch 9/20  
- 34s - loss: 0.8750 - acc: 0.6477 - val\_loss: 1.1457 - val\_acc: 0.5674

Epoch 10/20  
- 34s - loss: 0.8046 - acc: 0.6787 - val\_loss: 0.9540 - val\_acc: 0.6308

Epoch 11/20  
- 34s - loss: 0.7246 - acc: 0.7062 - val\_loss: 0.8471 - val\_acc: 0.6766

Epoch 12/20  
- 34s - loss: 0.6750 - acc: 0.7255 - val\_loss: 0.8517 - val\_acc: 0.6444

Epoch 13/20  
- 33s - loss: 0.6109 - acc: 0.7484 - val\_loss: 0.7975 - val\_acc: 0.6695

Epoch 14/20  
- 33s - loss: 0.5788 - acc: 0.7590 - val\_loss: 0.7815 - val\_acc: 0.6919

Epoch 15/20  
- 33s - loss: 0.5458 - acc: 0.7798 - val\_loss: 0.6969 - val\_acc: 0.7421

Epoch 16/20  
- 33s - loss: 0.5282 - acc: 0.7879 - val\_loss: 0.7095 - val\_acc: 0.7326

Epoch 17/20  
- 34s - loss: 0.4928 - acc: 0.8032 - val\_loss: 0.6868 - val\_acc: 0.7496

Epoch 18/20  
- 34s - loss: 0.5306 - acc: 0.8016 - val\_loss: 0.6882 - val\_acc: 0.7516

Epoch 19/20  
- 34s - loss: 0.4546 - acc: 0.8247 - val\_loss: 0.6945 - val\_acc: 0.7550

Epoch 20/20  
- 33s - loss: 0.4361 - acc: 0.8260 - val\_loss: 0.7333 - val\_acc: 0.7553

Test accuracy:  
0.7553444180522565  
Train on 7352 samples, validate on 2947 samples  
Epoch 1/20  
- 40s - loss: 0.9026 - acc: 0.6201 - val\_loss: 0.8218 - val\_acc: 0.7333

Epoch 2/20  
- 33s - loss: 0.4395 - acc: 0.8298 - val\_loss: 0.5881 - val\_acc: 0.8107

Epoch 3/20  
- 33s - loss: 0.2393 - acc: 0.9140 - val\_loss: 0.3804 - val\_acc: 0.8595

Epoch 4/20

```
- 33s - loss: 0.1598 - acc: 0.9351 - val_loss: 0.3429 - val_acc: 0.8904

Epoch 5/20
- 33s - loss: 0.1348 - acc: 0.9468 - val_loss: 0.2815 - val_acc: 0.8989

Epoch 6/20
- 33s - loss: 0.1266 - acc: 0.9459 - val_loss: 0.3448 - val_acc: 0.9019

Epoch 7/20
- 34s - loss: 0.1230 - acc: 0.9490 - val_loss: 0.2641 - val_acc: 0.9046

Epoch 8/20
- 33s - loss: 0.1155 - acc: 0.9518 - val_loss: 0.2519 - val_acc: 0.9084

Epoch 9/20
- 33s - loss: 0.1155 - acc: 0.9504 - val_loss: 0.2589 - val_acc: 0.9023

Epoch 10/20
- 34s - loss: 0.1209 - acc: 0.9486 - val_loss: 0.2749 - val_acc: 0.9046

Epoch 11/20
- 33s - loss: 0.1121 - acc: 0.9520 - val_loss: 0.2390 - val_acc: 0.9131

Epoch 12/20
- 33s - loss: 0.1073 - acc: 0.9521 - val_loss: 0.2595 - val_acc: 0.9138

Epoch 13/20
- 33s - loss: 0.1316 - acc: 0.9508 - val_loss: 0.2824 - val_acc: 0.9026

Epoch 14/20
- 33s - loss: 0.1271 - acc: 0.9450 - val_loss: 0.3312 - val_acc: 0.8928

Epoch 15/20
- 33s - loss: 0.1143 - acc: 0.9510 - val_loss: 0.2849 - val_acc: 0.9141

Epoch 16/20
- 34s - loss: 0.1152 - acc: 0.9498 - val_loss: 0.2846 - val_acc: 0.9155

Epoch 17/20
- 33s - loss: 0.1089 - acc: 0.9517 - val_loss: 0.2624 - val_acc: 0.9094

Epoch 18/20
- 33s - loss: 0.1085 - acc: 0.9535 - val_loss: 0.2663 - val_acc: 0.9165

Epoch 19/20
- 34s - loss: 0.1096 - acc: 0.9513 - val_loss: 0.2756 - val_acc: 0.9104

Epoch 20/20
- 34s - loss: 0.1099 - acc: 0.9517 - val_loss: 0.2759 - val_acc: 0.9206

Test accuracy:
0.9205972175093315
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 73s - loss: 1.4891 - acc: 0.3954 - val_loss: 1.3578 - val_acc: 0.4018

Epoch 2/20
- 66s - loss: 1.2320 - acc: 0.4993 - val_loss: 1.2097 - val_acc: 0.4883

Epoch 3/20
- 67s - loss: 1.0958 - acc: 0.5253 - val_loss: 1.1951 - val_acc: 0.4978

Epoch 4/20
- 67s - loss: 1.0472 - acc: 0.5488 - val_loss: 1.1701 - val_acc: 0.4964

Epoch 5/20
- 67s - loss: 1.0084 - acc: 0.5592 - val_loss: 1.1650 - val_acc: 0.4924

Epoch 6/20
- 66s - loss: 0.9711 - acc: 0.5793 - val_loss: 1.1446 - val_acc: 0.5168

Epoch 7/20
- 66s - loss: 0.9545 - acc: 0.5918 - val_loss: 1.1286 - val_acc: 0.5290

Epoch 8/20
- 67s - loss: 0.9198 - acc: 0.6079 - val_loss: 1.0786 - val_acc: 0.5541
```

```
Epoch 9/20
- 67s - loss: 0.8691 - acc: 0.6303 - val_loss: 1.0372 - val_acc: 0.5813

Epoch 10/20
- 67s - loss: 0.8162 - acc: 0.6606 - val_loss: 0.9318 - val_acc: 0.6166

Epoch 11/20
- 66s - loss: 0.7422 - acc: 0.6990 - val_loss: 0.8567 - val_acc: 0.6474

Epoch 12/20
- 66s - loss: 0.6651 - acc: 0.7410 - val_loss: 0.7907 - val_acc: 0.6820

Epoch 13/20
- 67s - loss: 0.6101 - acc: 0.7737 - val_loss: 0.7593 - val_acc: 0.6722

Epoch 14/20
- 66s - loss: 0.5592 - acc: 0.7953 - val_loss: 0.6276 - val_acc: 0.7482

Epoch 15/20
- 67s - loss: 0.4924 - acc: 0.8244 - val_loss: 0.6333 - val_acc: 0.7764

Epoch 16/20
- 66s - loss: 0.4483 - acc: 0.8345 - val_loss: 0.5861 - val_acc: 0.8008

Epoch 17/20
- 67s - loss: 0.4139 - acc: 0.8487 - val_loss: 0.5195 - val_acc: 0.8161

Epoch 18/20
- 67s - loss: 0.3602 - acc: 0.8739 - val_loss: 0.5427 - val_acc: 0.8215

Epoch 19/20
- 67s - loss: 0.3491 - acc: 0.8757 - val_loss: 0.5527 - val_acc: 0.8191

Epoch 20/20
- 66s - loss: 0.3481 - acc: 0.8736 - val_loss: 0.4949 - val_acc: 0.8354

Test accuracy:
0.835425856803529
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 40s - loss: 1.7270 - acc: 0.2952 - val_loss: 1.6144 - val_acc: 0.4187

Epoch 2/20
- 33s - loss: 1.5243 - acc: 0.4234 - val_loss: 1.4842 - val_acc: 0.4343

Epoch 3/20
- 33s - loss: 1.4145 - acc: 0.4498 - val_loss: 1.4118 - val_acc: 0.4493

Epoch 4/20
- 33s - loss: 1.3492 - acc: 0.4706 - val_loss: 1.3680 - val_acc: 0.4513

Epoch 5/20
- 33s - loss: 1.3108 - acc: 0.4788 - val_loss: 1.3369 - val_acc: 0.4628

Epoch 6/20
- 33s - loss: 1.2759 - acc: 0.4976 - val_loss: 1.3129 - val_acc: 0.4740

Epoch 7/20
- 33s - loss: 1.2511 - acc: 0.5084 - val_loss: 1.2931 - val_acc: 0.4846

Epoch 8/20
- 33s - loss: 1.2306 - acc: 0.5097 - val_loss: 1.2760 - val_acc: 0.4886

Epoch 9/20
- 33s - loss: 1.2102 - acc: 0.5184 - val_loss: 1.2606 - val_acc: 0.5036

Epoch 10/20
- 33s - loss: 1.1945 - acc: 0.5258 - val_loss: 1.2468 - val_acc: 0.5148

Epoch 11/20
- 33s - loss: 1.1789 - acc: 0.5341 - val_loss: 1.2348 - val_acc: 0.5117

Epoch 12/20
- 33s - loss: 1.1623 - acc: 0.5468 - val_loss: 1.2234 - val_acc: 0.5117

Epoch 13/20
- 33s - loss: 1.1508 - acc: 0.5480 - val_loss: 1.2131 - val_acc: 0.5188
```

```
... 1000. 1.1000 0.0100 val_1000. 1.2101 val_acc. 0.0100

Epoch 14/20
- 33s - loss: 1.1376 - acc: 0.5563 - val_loss: 1.2037 - val_acc: 0.5212

Epoch 15/20
- 33s - loss: 1.1254 - acc: 0.5537 - val_loss: 1.1947 - val_acc: 0.5171

Epoch 16/20
- 33s - loss: 1.1104 - acc: 0.5673 - val_loss: 1.1863 - val_acc: 0.5188

Epoch 17/20
- 33s - loss: 1.1031 - acc: 0.5709 - val_loss: 1.1787 - val_acc: 0.5158

Epoch 18/20
- 33s - loss: 1.0910 - acc: 0.5700 - val_loss: 1.1710 - val_acc: 0.5226

Epoch 19/20
- 33s - loss: 1.0806 - acc: 0.5808 - val_loss: 1.1643 - val_acc: 0.5290

Epoch 20/20
- 33s - loss: 1.0728 - acc: 0.5807 - val_loss: 1.1575 - val_acc: 0.5280

Test accuracy:
0.5279945707499152
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 42s - loss: 0.9209 - acc: 0.6167 - val_loss: 0.8914 - val_acc: 0.6603

Epoch 2/20
- 34s - loss: 0.4717 - acc: 0.8211 - val_loss: 0.5336 - val_acc: 0.8229

Epoch 3/20
- 34s - loss: 0.2448 - acc: 0.9115 - val_loss: 0.3663 - val_acc: 0.8823

Epoch 4/20
- 34s - loss: 0.1841 - acc: 0.9267 - val_loss: 0.2614 - val_acc: 0.9030

Epoch 5/20
- 34s - loss: 0.1480 - acc: 0.9377 - val_loss: 0.2942 - val_acc: 0.9141

Epoch 6/20
- 34s - loss: 0.1545 - acc: 0.9426 - val_loss: 0.2705 - val_acc: 0.9104

Epoch 7/20
- 34s - loss: 0.1389 - acc: 0.9442 - val_loss: 0.3912 - val_acc: 0.8680

Epoch 8/20
- 34s - loss: 0.1330 - acc: 0.9476 - val_loss: 0.2870 - val_acc: 0.9070

Epoch 9/20
- 34s - loss: 0.1234 - acc: 0.9484 - val_loss: 0.3509 - val_acc: 0.8914

Epoch 10/20
- 34s - loss: 0.1285 - acc: 0.9459 - val_loss: 0.3714 - val_acc: 0.8877

Epoch 11/20
- 34s - loss: 0.1133 - acc: 0.9513 - val_loss: 0.3367 - val_acc: 0.9074

Epoch 12/20
- 34s - loss: 0.1189 - acc: 0.9472 - val_loss: 0.3496 - val_acc: 0.8823

Epoch 13/20
- 34s - loss: 0.1293 - acc: 0.9474 - val_loss: 0.3639 - val_acc: 0.8989

Epoch 14/20
- 34s - loss: 0.1145 - acc: 0.9533 - val_loss: 0.3064 - val_acc: 0.8901

Epoch 15/20
- 34s - loss: 0.1333 - acc: 0.9459 - val_loss: 0.3111 - val_acc: 0.9094

Epoch 16/20
- 34s - loss: 0.1081 - acc: 0.9531 - val_loss: 0.3937 - val_acc: 0.9074

Epoch 17/20
- 34s - loss: 0.1095 - acc: 0.9523 - val_loss: 0.6955 - val_acc: 0.8785

Epoch 18/20
```

```
Epoch 10/20
- 34s - loss: 0.1134 - acc: 0.9505 - val_loss: 0.3008 - val_acc: 0.9026

Epoch 19/20
- 34s - loss: 0.1273 - acc: 0.9501 - val_loss: 0.3449 - val_acc: 0.8992

Epoch 20/20
- 34s - loss: 0.1066 - acc: 0.9536 - val_loss: 0.3388 - val_acc: 0.9128

Test accuracy:
0.9127926705123854
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 16s - loss: 1.6285 - acc: 0.3413 - val_loss: 1.5079 - val_acc: 0.4255

Epoch 2/20
- 9s - loss: 1.4168 - acc: 0.4535 - val_loss: 1.3871 - val_acc: 0.4557

Epoch 3/20
- 9s - loss: 1.3168 - acc: 0.4826 - val_loss: 1.3257 - val_acc: 0.4751

Epoch 4/20
- 9s - loss: 1.2624 - acc: 0.5035 - val_loss: 1.2864 - val_acc: 0.4876

Epoch 5/20
- 9s - loss: 1.2241 - acc: 0.5208 - val_loss: 1.2556 - val_acc: 0.5083

Epoch 6/20
- 9s - loss: 1.1889 - acc: 0.5311 - val_loss: 1.2303 - val_acc: 0.5070

Epoch 7/20
- 9s - loss: 1.1566 - acc: 0.5462 - val_loss: 1.2082 - val_acc: 0.5202

Epoch 8/20
- 9s - loss: 1.1338 - acc: 0.5539 - val_loss: 1.1900 - val_acc: 0.5188

Epoch 9/20
- 9s - loss: 1.1122 - acc: 0.5502 - val_loss: 1.1730 - val_acc: 0.5226

Epoch 10/20
- 9s - loss: 1.0838 - acc: 0.5665 - val_loss: 1.1589 - val_acc: 0.5327

Epoch 11/20
- 9s - loss: 1.0629 - acc: 0.5728 - val_loss: 1.1475 - val_acc: 0.5287

Epoch 12/20
- 9s - loss: 1.0442 - acc: 0.5763 - val_loss: 1.1381 - val_acc: 0.5361

Epoch 13/20
- 9s - loss: 1.0234 - acc: 0.5813 - val_loss: 1.1319 - val_acc: 0.5334

Epoch 14/20
- 9s - loss: 1.0062 - acc: 0.5903 - val_loss: 1.1283 - val_acc: 0.5385

Epoch 15/20
- 9s - loss: 0.9873 - acc: 0.5998 - val_loss: 1.1249 - val_acc: 0.5422

Epoch 16/20
- 9s - loss: 0.9699 - acc: 0.6053 - val_loss: 1.1238 - val_acc: 0.5480

Epoch 17/20
- 9s - loss: 0.9542 - acc: 0.6164 - val_loss: 1.1221 - val_acc: 0.5456

Epoch 18/20
- 9s - loss: 0.9402 - acc: 0.6181 - val_loss: 1.1199 - val_acc: 0.5551

Epoch 19/20
- 9s - loss: 0.9263 - acc: 0.6202 - val_loss: 1.1189 - val_acc: 0.5551

Epoch 20/20
- 9s - loss: 0.9127 - acc: 0.6348 - val_loss: 1.1142 - val_acc: 0.5589

Test accuracy:
0.5588734306073974
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 42s - loss: 0.9504 - acc: 0.5906 - val_loss: 1.1543 - val_acc: 0.5789
```

```
Epoch 2/20
- 34s - loss: 0.4812 - acc: 0.8058 - val_loss: 3.1267 - val_acc: 0.5399

Epoch 3/20
- 34s - loss: 0.2713 - acc: 0.8946 - val_loss: 2.3001 - val_acc: 0.6722

Epoch 4/20
- 34s - loss: 0.1647 - acc: 0.9336 - val_loss: 2.0348 - val_acc: 0.7238

Epoch 5/20
- 34s - loss: 0.1495 - acc: 0.9384 - val_loss: 1.4753 - val_acc: 0.7506

Epoch 6/20
- 34s - loss: 0.1399 - acc: 0.9423 - val_loss: 1.9244 - val_acc: 0.7526

Epoch 7/20
- 34s - loss: 0.1288 - acc: 0.9453 - val_loss: 1.9862 - val_acc: 0.7329

Epoch 8/20
- 34s - loss: 0.1304 - acc: 0.9460 - val_loss: 1.7273 - val_acc: 0.7486

Epoch 9/20
- 34s - loss: 0.1225 - acc: 0.9493 - val_loss: 2.2291 - val_acc: 0.7360

Epoch 10/20
- 34s - loss: 0.1334 - acc: 0.9456 - val_loss: 2.2179 - val_acc: 0.7591

Epoch 11/20
- 34s - loss: 0.1303 - acc: 0.9468 - val_loss: 1.7246 - val_acc: 0.7520

Epoch 12/20
- 34s - loss: 0.1201 - acc: 0.9499 - val_loss: 1.6897 - val_acc: 0.7482

Epoch 13/20
- 34s - loss: 0.1206 - acc: 0.9478 - val_loss: 1.8947 - val_acc: 0.7608

Epoch 14/20
- 34s - loss: 0.1171 - acc: 0.9509 - val_loss: 1.9454 - val_acc: 0.7560

Epoch 15/20
- 34s - loss: 0.1203 - acc: 0.9495 - val_loss: 1.5592 - val_acc: 0.7679

Epoch 16/20
- 34s - loss: 0.1167 - acc: 0.9487 - val_loss: 1.4220 - val_acc: 0.7679

Epoch 17/20
- 34s - loss: 0.1179 - acc: 0.9471 - val_loss: 1.7276 - val_acc: 0.7384

Epoch 18/20
- 34s - loss: 0.1405 - acc: 0.9478 - val_loss: 2.3373 - val_acc: 0.7112

Epoch 19/20
- 34s - loss: 0.1201 - acc: 0.9474 - val_loss: 1.2926 - val_acc: 0.7727

Epoch 20/20
- 34s - loss: 0.1371 - acc: 0.9431 - val_loss: 1.8672 - val_acc: 0.6987

Test accuracy:
0.6986766203120477
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 42s - loss: 0.9687 - acc: 0.6049 - val_loss: 0.7625 - val_acc: 0.7445

Epoch 2/20
- 34s - loss: 0.4051 - acc: 0.8463 - val_loss: 0.6492 - val_acc: 0.8269

Epoch 3/20
- 34s - loss: 0.1927 - acc: 0.9272 - val_loss: 0.4200 - val_acc: 0.8772

Epoch 4/20
- 34s - loss: 0.1384 - acc: 0.9422 - val_loss: 0.3572 - val_acc: 0.9050

Epoch 5/20
- 34s - loss: 0.1260 - acc: 0.9484 - val_loss: 0.3355 - val_acc: 0.9104

Epoch 6/20
- 34s - loss: 0.1376 - acc: 0.9425 - val_loss: 0.3272 - val_acc: 0.9067
```

```
- 34s - loss: 0.1510 - acc: 0.9420 - val_loss: 0.3212 - val_acc: 0.9081
Epoch 7/20
- 34s - loss: 0.1250 - acc: 0.9479 - val_loss: 0.3292 - val_acc: 0.9087
Epoch 8/20
- 34s - loss: 0.1163 - acc: 0.9510 - val_loss: 0.4345 - val_acc: 0.8901
Epoch 9/20
- 34s - loss: 0.1396 - acc: 0.9452 - val_loss: 0.3740 - val_acc: 0.8968
Epoch 10/20
- 34s - loss: 0.1306 - acc: 0.9476 - val_loss: 0.3345 - val_acc: 0.9046
Epoch 11/20
- 34s - loss: 0.1169 - acc: 0.9499 - val_loss: 0.2762 - val_acc: 0.9213
Epoch 12/20
- 34s - loss: 0.1205 - acc: 0.9464 - val_loss: 0.3923 - val_acc: 0.9057
Epoch 13/20
- 34s - loss: 0.1209 - acc: 0.9465 - val_loss: 0.3049 - val_acc: 0.9125
Epoch 14/20
- 34s - loss: 0.1141 - acc: 0.9504 - val_loss: 0.2867 - val_acc: 0.9189
Epoch 15/20
- 34s - loss: 0.1105 - acc: 0.9502 - val_loss: 0.2419 - val_acc: 0.9264
Epoch 16/20
- 34s - loss: 0.1115 - acc: 0.9509 - val_loss: 0.2738 - val_acc: 0.9277
Epoch 17/20
- 34s - loss: 0.1113 - acc: 0.9504 - val_loss: 0.2709 - val_acc: 0.9213
Epoch 18/20
- 34s - loss: 0.1087 - acc: 0.9542 - val_loss: 0.2693 - val_acc: 0.9230
Epoch 19/20
- 34s - loss: 0.1112 - acc: 0.9510 - val_loss: 0.2657 - val_acc: 0.9240
Epoch 20/20
- 34s - loss: 0.1074 - acc: 0.9525 - val_loss: 0.2601 - val_acc: 0.9186

Test accuracy:
0.9185612487275195
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 43s - loss: 1.3618 - acc: 0.4172 - val_loss: 5.1270 - val_acc: 0.3705
Epoch 2/20
- 34s - loss: 0.8472 - acc: 0.6069 - val_loss: 8.0132 - val_acc: 0.3885
Epoch 3/20
- 34s - loss: 0.7410 - acc: 0.6549 - val_loss: 7.5071 - val_acc: 0.4052
Epoch 4/20
- 34s - loss: 0.7393 - acc: 0.6595 - val_loss: 7.9044 - val_acc: 0.4092
Epoch 5/20
- 35s - loss: 0.6734 - acc: 0.6922 - val_loss: 7.8628 - val_acc: 0.4167
Epoch 6/20
- 34s - loss: 0.6910 - acc: 0.6904 - val_loss: 7.8476 - val_acc: 0.4204
Epoch 7/20
- 34s - loss: 0.6369 - acc: 0.7168 - val_loss: 7.7756 - val_acc: 0.4550
Epoch 8/20
- 34s - loss: 0.5654 - acc: 0.7507 - val_loss: 7.7974 - val_acc: 0.4591
Epoch 9/20
- 34s - loss: 0.4981 - acc: 0.7881 - val_loss: 7.5595 - val_acc: 0.4398
Epoch 10/20
- 34s - loss: 0.5446 - acc: 0.7799 - val_loss: 7.6699 - val_acc: 0.4194
```

```
Epoch 11/20
- 34s - loss: 0.4341 - acc: 0.8168 - val_loss: 7.6746 - val_acc: 0.4659

Epoch 12/20
- 34s - loss: 0.5060 - acc: 0.7953 - val_loss: 7.2810 - val_acc: 0.4347

Epoch 13/20
- 34s - loss: 0.5445 - acc: 0.7677 - val_loss: 7.8176 - val_acc: 0.4561

Epoch 14/20
- 35s - loss: 0.4350 - acc: 0.8202 - val_loss: 7.8636 - val_acc: 0.4557

Epoch 15/20
- 34s - loss: 0.3943 - acc: 0.8400 - val_loss: 7.8194 - val_acc: 0.4574

Epoch 16/20
- 34s - loss: 0.3528 - acc: 0.8589 - val_loss: 7.9041 - val_acc: 0.4364

Epoch 17/20
- 34s - loss: 0.3963 - acc: 0.8365 - val_loss: 7.7922 - val_acc: 0.4554

Epoch 18/20
- 34s - loss: 0.3769 - acc: 0.8426 - val_loss: 7.8341 - val_acc: 0.4578

Epoch 19/20
- 34s - loss: 0.3321 - acc: 0.8630 - val_loss: 7.9015 - val_acc: 0.4343

Epoch 20/20
- 34s - loss: 0.3171 - acc: 0.8800 - val_loss: 7.8713 - val_acc: 0.4432

Test accuracy:
0.44316253817441464
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 44s - loss: 0.8914 - acc: 0.6223 - val_loss: 0.7283 - val_acc: 0.7428

Epoch 2/20
- 34s - loss: 0.3889 - acc: 0.8546 - val_loss: 0.5547 - val_acc: 0.8375

Epoch 3/20
- 34s - loss: 0.2194 - acc: 0.9163 - val_loss: 0.4076 - val_acc: 0.8809

Epoch 4/20
- 34s - loss: 0.1516 - acc: 0.9393 - val_loss: 0.3621 - val_acc: 0.8938

Epoch 5/20
- 34s - loss: 0.1298 - acc: 0.9472 - val_loss: 0.3250 - val_acc: 0.9084

Epoch 6/20
- 34s - loss: 0.1314 - acc: 0.9460 - val_loss: 0.3542 - val_acc: 0.9040

Epoch 7/20
- 34s - loss: 0.1201 - acc: 0.9474 - val_loss: 0.3086 - val_acc: 0.9209

Epoch 8/20
- 34s - loss: 0.1141 - acc: 0.9505 - val_loss: 0.2915 - val_acc: 0.9230

Epoch 9/20
- 34s - loss: 0.1170 - acc: 0.9484 - val_loss: 0.3350 - val_acc: 0.9135

Epoch 10/20
- 34s - loss: 0.1238 - acc: 0.9484 - val_loss: 0.3433 - val_acc: 0.9138

Epoch 11/20
- 34s - loss: 0.1444 - acc: 0.9430 - val_loss: 0.2968 - val_acc: 0.9257

Epoch 12/20
- 34s - loss: 0.1170 - acc: 0.9528 - val_loss: 0.3148 - val_acc: 0.9131

Epoch 13/20
- 34s - loss: 0.1116 - acc: 0.9523 - val_loss: 0.3092 - val_acc: 0.9206

Epoch 14/20
- 34s - loss: 0.1130 - acc: 0.9516 - val_loss: 0.3073 - val_acc: 0.9223

Epoch 15/20
- 34s - loss: 0.1102 - acc: 0.9510 - val_loss: 0.3321 - val_acc: 0.9165
```

```
Epoch 16/20
- 34s - loss: 0.1139 - acc: 0.9487 - val_loss: 0.3543 - val_acc: 0.9128

Epoch 17/20
- 34s - loss: 0.1093 - acc: 0.9489 - val_loss: 0.3252 - val_acc: 0.9182

Epoch 18/20
- 34s - loss: 0.1135 - acc: 0.9525 - val_loss: 0.4385 - val_acc: 0.9026

Epoch 19/20
- 34s - loss: 0.1118 - acc: 0.9513 - val_loss: 0.3504 - val_acc: 0.9223

Epoch 20/20
- 35s - loss: 0.1076 - acc: 0.9521 - val_loss: 0.4067 - val_acc: 0.9192

Test accuracy:
0.9192399049881235
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 44s - loss: 0.9185 - acc: 0.6158 - val_loss: 0.8008 - val_acc: 0.7170

Epoch 2/20
- 34s - loss: 0.4326 - acc: 0.8330 - val_loss: 0.7086 - val_acc: 0.7594

Epoch 3/20
- 34s - loss: 0.2448 - acc: 0.9072 - val_loss: 0.5266 - val_acc: 0.8609

Epoch 4/20
- 34s - loss: 0.1821 - acc: 0.9276 - val_loss: 0.4757 - val_acc: 0.8354

Epoch 5/20
- 33s - loss: 0.1542 - acc: 0.9357 - val_loss: 0.3650 - val_acc: 0.8945

Epoch 6/20
- 34s - loss: 0.1334 - acc: 0.9436 - val_loss: 0.5379 - val_acc: 0.8626

Epoch 7/20
- 34s - loss: 0.1323 - acc: 0.9434 - val_loss: 0.4888 - val_acc: 0.8744

Epoch 8/20
- 34s - loss: 0.1254 - acc: 0.9491 - val_loss: 0.3819 - val_acc: 0.9023

Epoch 9/20
- 34s - loss: 0.1270 - acc: 0.9475 - val_loss: 0.3396 - val_acc: 0.9128

Epoch 10/20
- 34s - loss: 0.1167 - acc: 0.9505 - val_loss: 0.3421 - val_acc: 0.9080

Epoch 11/20
- 33s - loss: 0.1217 - acc: 0.9501 - val_loss: 0.3871 - val_acc: 0.8999

Epoch 12/20
- 34s - loss: 0.1168 - acc: 0.9513 - val_loss: 0.4035 - val_acc: 0.8962

Epoch 13/20
- 34s - loss: 0.1299 - acc: 0.9438 - val_loss: 0.4200 - val_acc: 0.8867

Epoch 14/20
- 34s - loss: 0.1171 - acc: 0.9489 - val_loss: 0.3357 - val_acc: 0.9023

Epoch 15/20
- 34s - loss: 0.1220 - acc: 0.9497 - val_loss: 0.3109 - val_acc: 0.9080

Epoch 16/20
- 34s - loss: 0.1134 - acc: 0.9518 - val_loss: 0.3213 - val_acc: 0.9199

Epoch 17/20
- 34s - loss: 0.1110 - acc: 0.9518 - val_loss: 0.3251 - val_acc: 0.9182

Epoch 18/20
- 34s - loss: 0.1151 - acc: 0.9527 - val_loss: 0.3469 - val_acc: 0.9172

Epoch 19/20
- 34s - loss: 0.1102 - acc: 0.9514 - val_loss: 0.3153 - val_acc: 0.9209

Epoch 20/20
- 34s - loss: 0.1134 - acc: 0.9518 - val_loss: 0.3213 - val_acc: 0.9199
```

```
- 34s - loss: 0.1097 - acc: 0.9506 - val_loss: 0.3423 - val_acc: 0.9070

Test accuracy:
0.9070240922972514
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 28s - loss: 1.2291 - acc: 0.5106 - val_loss: 1.2771 - val_acc: 0.5236

Epoch 2/20
- 17s - loss: 0.8698 - acc: 0.6492 - val_loss: 0.9413 - val_acc: 0.6077

Epoch 3/20
- 17s - loss: 0.6771 - acc: 0.7135 - val_loss: 0.7829 - val_acc: 0.6417

Epoch 4/20
- 17s - loss: 0.5566 - acc: 0.7681 - val_loss: 0.6762 - val_acc: 0.7503

Epoch 5/20
- 17s - loss: 0.4355 - acc: 0.8255 - val_loss: 0.6226 - val_acc: 0.7859

Epoch 6/20
- 17s - loss: 0.3565 - acc: 0.8592 - val_loss: 0.4178 - val_acc: 0.8802

Epoch 7/20
- 17s - loss: 0.2797 - acc: 0.8935 - val_loss: 0.3800 - val_acc: 0.8690

Epoch 8/20
- 17s - loss: 0.2577 - acc: 0.9006 - val_loss: 0.3341 - val_acc: 0.8728

Epoch 9/20
- 17s - loss: 0.2524 - acc: 0.8998 - val_loss: 0.3205 - val_acc: 0.8989

Epoch 10/20
- 17s - loss: 0.2174 - acc: 0.9120 - val_loss: 0.2650 - val_acc: 0.9019

Epoch 11/20
- 17s - loss: 0.2306 - acc: 0.9093 - val_loss: 0.2741 - val_acc: 0.9080

Epoch 12/20
- 17s - loss: 0.2119 - acc: 0.9129 - val_loss: 0.2676 - val_acc: 0.8867

Epoch 13/20
- 17s - loss: 0.1998 - acc: 0.9187 - val_loss: 0.2617 - val_acc: 0.9030

Epoch 14/20
- 17s - loss: 0.1834 - acc: 0.9268 - val_loss: 0.2318 - val_acc: 0.9141

Epoch 15/20
- 17s - loss: 0.1795 - acc: 0.9272 - val_loss: 0.2513 - val_acc: 0.9118

Epoch 16/20
- 17s - loss: 0.1775 - acc: 0.9249 - val_loss: 0.2406 - val_acc: 0.9057

Epoch 17/20
- 17s - loss: 0.1787 - acc: 0.9276 - val_loss: 0.2270 - val_acc: 0.9216

Epoch 18/20
- 17s - loss: 0.1672 - acc: 0.9290 - val_loss: 0.2746 - val_acc: 0.8938

Epoch 19/20
- 17s - loss: 0.1707 - acc: 0.9306 - val_loss: 0.2124 - val_acc: 0.9182

Epoch 20/20
- 17s - loss: 0.1821 - acc: 0.9255 - val_loss: 0.2398 - val_acc: 0.9077

Test accuracy:
0.9077027485578555
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 45s - loss: 0.9007 - acc: 0.6247 - val_loss: 0.7809 - val_acc: 0.7353

Epoch 2/20
- 33s - loss: 0.4099 - acc: 0.8497 - val_loss: 0.6094 - val_acc: 0.8202

Epoch 3/20
- 34s - loss: 0.2065 - acc: 0.9241 - val_loss: 0.3273 - val_acc: 0.9060
- - - - -
```

```
Epoch 4/20
- 34s - loss: 0.1501 - acc: 0.9408 - val_loss: 0.2937 - val_acc: 0.9043

Epoch 5/20
- 34s - loss: 0.1225 - acc: 0.9476 - val_loss: 0.3677 - val_acc: 0.8938

Epoch 6/20
- 34s - loss: 0.1308 - acc: 0.9444 - val_loss: 0.3281 - val_acc: 0.9101

Epoch 7/20
- 34s - loss: 0.1284 - acc: 0.9464 - val_loss: 0.3534 - val_acc: 0.9108

Epoch 8/20
- 34s - loss: 0.1203 - acc: 0.9509 - val_loss: 0.3050 - val_acc: 0.9152

Epoch 9/20
- 34s - loss: 0.1219 - acc: 0.9495 - val_loss: 0.3958 - val_acc: 0.9002

Epoch 10/20
- 34s - loss: 0.1434 - acc: 0.9402 - val_loss: 0.2944 - val_acc: 0.9077

Epoch 11/20
- 34s - loss: 0.1197 - acc: 0.9484 - val_loss: 0.2914 - val_acc: 0.9108

Epoch 12/20
- 34s - loss: 0.1105 - acc: 0.9524 - val_loss: 0.2627 - val_acc: 0.9199

Epoch 13/20
- 34s - loss: 0.1105 - acc: 0.9520 - val_loss: 0.3932 - val_acc: 0.8880

Epoch 14/20
- 34s - loss: 0.1239 - acc: 0.9467 - val_loss: 0.3839 - val_acc: 0.8955

Epoch 15/20
- 34s - loss: 0.1147 - acc: 0.9501 - val_loss: 0.3068 - val_acc: 0.9182

Epoch 16/20
- 34s - loss: 0.1142 - acc: 0.9495 - val_loss: 0.3154 - val_acc: 0.9121

Epoch 17/20
- 34s - loss: 0.1085 - acc: 0.9513 - val_loss: 0.2846 - val_acc: 0.9247

Epoch 18/20
- 34s - loss: 0.1085 - acc: 0.9502 - val_loss: 0.2767 - val_acc: 0.9237

Epoch 19/20
- 34s - loss: 0.1076 - acc: 0.9512 - val_loss: 0.2840 - val_acc: 0.9148

Epoch 20/20
- 33s - loss: 0.1143 - acc: 0.9494 - val_loss: 0.2822 - val_acc: 0.9175

Test accuracy:
0.9175432643366135
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 46s - loss: 0.9827 - acc: 0.5875 - val_loss: 0.9463 - val_acc: 0.6295

Epoch 2/20
- 34s - loss: 0.4611 - acc: 0.8292 - val_loss: 0.7127 - val_acc: 0.7777

Epoch 3/20
- 34s - loss: 0.2670 - acc: 0.9053 - val_loss: 0.4739 - val_acc: 0.8748

Epoch 4/20
- 34s - loss: 0.1723 - acc: 0.9321 - val_loss: 0.3714 - val_acc: 0.9006

Epoch 5/20
- 34s - loss: 0.1362 - acc: 0.9441 - val_loss: 0.3360 - val_acc: 0.9152

Epoch 6/20
- 34s - loss: 0.1314 - acc: 0.9486 - val_loss: 0.2886 - val_acc: 0.9091

Epoch 7/20
- 34s - loss: 0.1287 - acc: 0.9480 - val_loss: 0.2807 - val_acc: 0.9220

Epoch 8/20
- 34s - loss: 0.1248 - acc: 0.9486 - val_loss: 0.3238 - val_acc: 0.9091
```

```
Epoch 9/20
- 34s - loss: 0.1202 - acc: 0.9518 - val_loss: 0.2667 - val_acc: 0.9125

Epoch 10/20
- 34s - loss: 0.1185 - acc: 0.9495 - val_loss: 0.2768 - val_acc: 0.9230

Epoch 11/20
- 34s - loss: 0.1233 - acc: 0.9475 - val_loss: 0.2878 - val_acc: 0.9104

Epoch 12/20
- 34s - loss: 0.1154 - acc: 0.9501 - val_loss: 0.2916 - val_acc: 0.9267

Epoch 13/20
- 34s - loss: 0.1093 - acc: 0.9540 - val_loss: 0.2945 - val_acc: 0.9148

Epoch 14/20
- 34s - loss: 0.1123 - acc: 0.9547 - val_loss: 0.2800 - val_acc: 0.9260

Epoch 15/20
- 34s - loss: 0.1132 - acc: 0.9493 - val_loss: 0.2866 - val_acc: 0.9240

Epoch 16/20
- 34s - loss: 0.1182 - acc: 0.9517 - val_loss: 0.2978 - val_acc: 0.9220

Epoch 17/20
- 34s - loss: 0.1175 - acc: 0.9497 - val_loss: 0.3178 - val_acc: 0.9067

Epoch 18/20
- 34s - loss: 0.1212 - acc: 0.9484 - val_loss: 0.2941 - val_acc: 0.9186

Epoch 19/20
- 34s - loss: 0.1154 - acc: 0.9489 - val_loss: 0.2920 - val_acc: 0.9145

Epoch 20/20
- 34s - loss: 0.1098 - acc: 0.9502 - val_loss: 0.2843 - val_acc: 0.9264

Test accuracy:
0.9263657957244655
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 46s - loss: 0.9414 - acc: 0.6117 - val_loss: 0.8269 - val_acc: 0.7370

Epoch 2/20
- 34s - loss: 0.4297 - acc: 0.8398 - val_loss: 0.5404 - val_acc: 0.8205

Epoch 3/20
- 34s - loss: 0.2388 - acc: 0.9085 - val_loss: 0.3659 - val_acc: 0.8714

Epoch 4/20
- 34s - loss: 0.1650 - acc: 0.9328 - val_loss: 0.3896 - val_acc: 0.8918

Epoch 5/20
- 34s - loss: 0.1425 - acc: 0.9438 - val_loss: 0.3447 - val_acc: 0.8989

Epoch 6/20
- 34s - loss: 0.1297 - acc: 0.9474 - val_loss: 0.3596 - val_acc: 0.9114

Epoch 7/20
- 35s - loss: 0.1217 - acc: 0.9490 - val_loss: 0.3205 - val_acc: 0.9108

Epoch 8/20
- 35s - loss: 0.1187 - acc: 0.9489 - val_loss: 0.3466 - val_acc: 0.9111

Epoch 9/20
- 34s - loss: 0.1203 - acc: 0.9497 - val_loss: 0.3413 - val_acc: 0.9063

Epoch 10/20
- 34s - loss: 0.1195 - acc: 0.9472 - val_loss: 0.4260 - val_acc: 0.9030

Epoch 11/20
- 34s - loss: 0.1249 - acc: 0.9476 - val_loss: 0.3287 - val_acc: 0.9135

Epoch 12/20
- 34s - loss: 0.1163 - acc: 0.9489 - val_loss: 0.3861 - val_acc: 0.9070

Epoch 13/20
```

```
- 34s - loss: 0.1173 - acc: 0.9499 - val_loss: 0.3632 - val_acc: 0.9087

Epoch 14/20
- 35s - loss: 0.1190 - acc: 0.9490 - val_loss: 0.4549 - val_acc: 0.9057

Epoch 15/20
- 35s - loss: 0.1192 - acc: 0.9476 - val_loss: 0.4678 - val_acc: 0.8924

Epoch 16/20
- 34s - loss: 0.1100 - acc: 0.9521 - val_loss: 0.4642 - val_acc: 0.8989

Epoch 17/20
- 34s - loss: 0.1105 - acc: 0.9510 - val_loss: 0.4309 - val_acc: 0.9050

Epoch 18/20
- 34s - loss: 0.1288 - acc: 0.9478 - val_loss: 0.3818 - val_acc: 0.9050

Epoch 19/20
- 34s - loss: 0.1157 - acc: 0.9510 - val_loss: 0.3939 - val_acc: 0.9087

Epoch 20/20
- 34s - loss: 0.1103 - acc: 0.9506 - val_loss: 0.3021 - val_acc: 0.9209

Test accuracy:
0.9209365456396336
Train on 7352 samples, validate on 2947 samples
Epoch 1/20
- 47s - loss: 0.9237 - acc: 0.6117 - val_loss: 0.6924 - val_acc: 0.7506

Epoch 2/20
- 34s - loss: 0.3691 - acc: 0.8579 - val_loss: 0.5148 - val_acc: 0.8334

Epoch 3/20
- 34s - loss: 0.2219 - acc: 0.9139 - val_loss: 0.3647 - val_acc: 0.8633

Epoch 4/20
- 34s - loss: 0.1599 - acc: 0.9335 - val_loss: 0.3112 - val_acc: 0.9057

Epoch 5/20
- 35s - loss: 0.1415 - acc: 0.9431 - val_loss: 0.3214 - val_acc: 0.9101

Epoch 6/20
- 34s - loss: 0.1238 - acc: 0.9493 - val_loss: 0.3466 - val_acc: 0.9121

Epoch 7/20
- 34s - loss: 0.1231 - acc: 0.9494 - val_loss: 0.3627 - val_acc: 0.9019

Epoch 8/20
- 34s - loss: 0.1227 - acc: 0.9465 - val_loss: 0.3780 - val_acc: 0.8979

Epoch 9/20
- 34s - loss: 0.1225 - acc: 0.9464 - val_loss: 0.3625 - val_acc: 0.9067

Epoch 10/20
- 34s - loss: 0.1121 - acc: 0.9516 - val_loss: 0.3693 - val_acc: 0.8958

Epoch 11/20
- 34s - loss: 0.1185 - acc: 0.9502 - val_loss: 0.4276 - val_acc: 0.9013

Epoch 12/20
- 35s - loss: 0.1305 - acc: 0.9450 - val_loss: 0.2877 - val_acc: 0.9165

Epoch 13/20
- 34s - loss: 0.1129 - acc: 0.9504 - val_loss: 0.3615 - val_acc: 0.9074

Epoch 14/20
- 35s - loss: 0.1130 - acc: 0.9512 - val_loss: 0.3484 - val_acc: 0.9087

Epoch 15/20
- 34s - loss: 0.1116 - acc: 0.9487 - val_loss: 0.3455 - val_acc: 0.8985

Epoch 16/20
- 34s - loss: 0.1119 - acc: 0.9486 - val_loss: 0.3229 - val_acc: 0.9040

Epoch 17/20
- 34s - loss: 0.1195 - acc: 0.9483 - val_loss: 0.3544 - val_acc: 0.9145
```

```
Epoch 18/20
- 34s - loss: 0.1124 - acc: 0.9494 - val_loss: 0.3826 - val_acc: 0.9094
```

```
Epoch 19/20
- 34s - loss: 0.1144 - acc: 0.9497 - val_loss: 0.3696 - val_acc: 0.9067
```

```
Epoch 20/20
- 34s - loss: 0.1104 - acc: 0.9527 - val_loss: 0.3537 - val_acc: 0.9026
```

Test accuracy:

0.9026128266033254

100%|██████████| 30/30 [4:18:14<00:00, 685.69s/it, best loss: -0.9263657957244655]

In [0]:

```
print("Best Performing Model Chosen After Hyperparameter tuning is :-")
print(best_run)
```

```
Best Performing Model Chosen After Hyperparameter tuning is :-
{'Dropout': 0.8108563255378749, 'LSTM': 3, 'LSTM_1': 4, 'batch_size': 1, 'choiceval': 0, 'if': 1,
'lr': 0, 'lr_1': 1, 'lr_2': 2}
```

In [0]:

```
from keras.layers import Conv1D
from keras.layers import BatchNormalization
from keras.layers import MaxPooling1D
```

In [0]:

```
X_train, Y_train , X_val, Y_val = data()
```

In [0]:

```
model = Sequential()
model.add(LSTM(64, return_sequences=True, kernel_initializer="glorot_uniform", input_shape=(128,9),
name='LSTMLayer1'))
model.add(Dropout(0.81))

model.add(LSTM(128 , return_sequences=True, kernel_initializer="glorot_uniform", name='LSTMLayer2'))
model.add(Flatten())
model.add(Dense(6, activation='softmax', name='DenseLayer1'))
adam = keras.optimizers.Adam(lr=10**-3)

model.compile(loss='categorical_crossentropy', metrics=['accuracy'],optimizer=adam)
model.fit(X_train, Y_train,
          batch_size=128,
          epochs=30,
          validation_data=(X_val, Y_val))
```

Train on 7352 samples, validate on 2947 samples

Epoch 1/30

```
7352/7352 [=====] - 50s 7ms/step - loss: 0.9165 - acc: 0.6089 - val_loss:
0.7430 - val_acc: 0.7272
```

Epoch 2/30

```
7352/7352 [=====] - 35s 5ms/step - loss: 0.4391 - acc: 0.8376 - val_loss:
0.6172 - val_acc: 0.7981
```

Epoch 3/30

```
7352/7352 [=====] - 35s 5ms/step - loss: 0.2793 - acc: 0.8961 - val_loss:
0.5558 - val_acc: 0.8561
```

Epoch 4/30

```
7352/7352 [=====] - 35s 5ms/step - loss: 0.1733 - acc: 0.9321 - val_loss:
0.4496 - val_acc: 0.8802
```

Epoch 5/30

```
7352/7352 [=====] - 35s 5ms/step - loss: 0.1424 - acc: 0.9429 - val_loss:
0.4174 - val_acc: 0.8901
```

Epoch 6/30

```
7352/7352 [=====] - 35s 5ms/step - loss: 0.1309 - acc: 0.9461 - val_loss:
0.4229 - val_acc: 0.8816
```

Epoch 7/30

```
7352/7352 [=====] - 35s 5ms/step - loss: 0.1233 - acc: 0.9499 - val_loss:
```

```
0.3550 - val_acc: 0.9053
Epoch 8/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1194 - acc: 0.9499 - val_loss: 0.4106 - val_acc: 0.9023
Epoch 9/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1455 - acc: 0.9455 - val_loss: 0.3610 - val_acc: 0.8951
Epoch 10/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1268 - acc: 0.9480 - val_loss: 0.3214 - val_acc: 0.9101
Epoch 11/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1203 - acc: 0.9487 - val_loss: 0.3564 - val_acc: 0.9070
Epoch 12/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1139 - acc: 0.9521 - val_loss: 0.3087 - val_acc: 0.9145
Epoch 13/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1150 - acc: 0.9480 - val_loss: 0.3127 - val_acc: 0.9213
Epoch 14/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1132 - acc: 0.9514 - val_loss: 0.3210 - val_acc: 0.9135
Epoch 15/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1101 - acc: 0.9512 - val_loss: 0.4750 - val_acc: 0.8846
Epoch 16/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1138 - acc: 0.9510 - val_loss: 0.3394 - val_acc: 0.9155
Epoch 17/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1121 - acc: 0.9502 - val_loss: 0.3475 - val_acc: 0.9128
Epoch 18/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1089 - acc: 0.9524 - val_loss: 0.3542 - val_acc: 0.9040
Epoch 19/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1119 - acc: 0.9513 - val_loss: 0.3261 - val_acc: 0.9131
Epoch 20/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1157 - acc: 0.9478 - val_loss: 0.3296 - val_acc: 0.9158
Epoch 21/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1203 - acc: 0.9483 - val_loss: 0.5307 - val_acc: 0.8819
Epoch 22/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1104 - acc: 0.9517 - val_loss: 0.3794 - val_acc: 0.9030
Epoch 23/30
7352/7352 [=====] - 34s 5ms/step - loss: 0.1113 - acc: 0.9517 - val_loss: 0.3543 - val_acc: 0.9135
Epoch 24/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1085 - acc: 0.9521 - val_loss: 0.3546 - val_acc: 0.9131
Epoch 25/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1074 - acc: 0.9523 - val_loss: 0.3308 - val_acc: 0.9206
Epoch 26/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1086 - acc: 0.9527 - val_loss: 0.3404 - val_acc: 0.9199
Epoch 27/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1121 - acc: 0.9527 - val_loss: 0.3186 - val_acc: 0.9206
Epoch 28/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1083 - acc: 0.9538 - val_loss: 0.3212 - val_acc: 0.9206
Epoch 29/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1065 - acc: 0.9548 - val_loss: 0.3428 - val_acc: 0.9186
Epoch 30/30
7352/7352 [=====] - 35s 5ms/step - loss: 0.1048 - acc: 0.9535 - val_loss: 0.3404 - val_acc: 0.9169
```

Out[0]:

```
<keras.callbacks.History at 0x7f93f9d56940>
```

```
In [0]:
```

```
score, acc = model.evaluate(X_val, Y_val, verbose=0)
```

```
In [0]:
```

```
print(score, acc)
```

```
0.3404184462123013 0.9168646080760094
```

## Divide and Conquer Approach

- In this we would first classify the activity as static or a dynamic one.
- Then we would create 2 classifiers, one for static and the other for dynamic, to classify the sub activities in the data.

### Sources for the approach

- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5949027/>
- <https://github.com/heeryoncho/sensors2018cnnhar>
- <https://machinelearningmastery.com/deep-learning-models-for-human-activity-recognition/>

## 2 Class (Static and Dynamic Activity) Classification

```
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)

def load_y_2c(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 = data_path + f'y_{subset}.txt'
    y = _read_csv(filename)[0]
    y[y<=3] = 0
    y[y>3] = 1
    return pd.get_dummies(y).as_matrix()

def load_data_2c():
    """
    Obtain the dataset from multiple files.
    Returns: y_train, y_test
    """
    y_train, y_test = load_y_2c('train'), load_y_2c('test')

    return y_train, y_test
```

```
In [0]:
```

```
# Loading the train and test data
Y_train_2c, Y_test_2c = load_data_2c()
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:15: FutureWarning: Method .as_matrix
will be removed in a future version. Use .values instead.
from ipykernel import kernelapp as app
```

```
In [0]:
```

```
np.save(data_path + 'Y_train_2c.npy', Y_train_2c) # save Y_train_2c
np.save(data_path + 'Y_test_2c.npy', Y_test_2c) # save Y_test_2c
```

```
In [0]:
```

```
def data_2c():
    data_path = '/content/drive/My Drive/AAIC/Case Studies/Human Activity Recognition/'
    X_train = np.load(data_path + 'X_train.npy') # load X_train
    Y_train = np.load(data_path + 'Y_train_2c.npy') # load Y_train_2c
    X_val = np.load(data_path + 'X_test.npy') # load X_test
    Y_val = np.load(data_path + 'Y_test_2c.npy') # load Y_test_2c
    return X_train, Y_train , X_val, Y_val
```

```
In [0]:
```

```
X_train_2c, Y_train_2c, X_val_2c, Y_val_2c = data_2c()
```

```
In [0]:
```

```
print(Y_train_2c.shape)
print(Y_val_2c.shape)
```

```
(7352, 2)
(2947, 2)
```

```
In [0]:
```

```
model = Sequential()
model.add(Conv1D(filters=64, kernel_size=4, activation='relu',kernel_initializer='glorot_uniform',input_shape=(128,9)))
model.add(Conv1D(filters=32, kernel_size=3, activation='relu',kernel_initializer='glorot_uniform'))
model.add(Dropout(0.5))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(32, activation='relu'))
model.add(Dense(2, activation='softmax'))
model.summary()
adam = keras.optimizers.Adam(lr=10**-3)
model.compile(loss='categorical_crossentropy', optimizer=adam, metrics=['accuracy'])
model.fit(X_train_2c,Y_train_2c, epochs=20, batch_size=16,validation_data=(X_val_2c, Y_val_2c), verbose=1)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:4267: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.
```

```
Model: "sequential_63"
```

Layer (type)	Output Shape	Param #
conv1d_203 (Conv1D)	(None, 125, 64)	2368
conv1d_204 (Conv1D)	(None, 123, 32)	6176
dropout_156 (Dropout)	(None, 123, 32)	0
max_pooling1d_1 (MaxPooling1D)	(None, 61, 32)	0
flatten_62 (Flatten)	(None, 1952)	0
dense_1 (Dense)	(None, 32)	62496
dense_2 (Dense)	(None, 2)	66

```
Total params: 71,106
```

```
Trainable params: 71,106
```

```
Non-trainable params: 0
```

```
Train on 7352 samples, validate on 2947 samples
```

```
Epoch 1/20
```

```
7352/7352 [=====] - 14s 2ms/step - loss: 0.0304 - acc: 0.9874 - val_loss: 0.0394 - val_acc: 0.9878
```

```
Epoch 2/20
```

```
7352/7352 [=====] - 5s 625us/step - loss: 0.0033 - acc: 0.9989 - val_loss: 0.0050 - val_acc: 0.9986
```

```
Epoch 3/20
```

```
7352/7352 [=====] - 5s 637us/step - loss: 1.4912e-04 - acc: 1.0000 - val_loss: 0.0049 - val_acc: 0.9986
Epoch 4/20
7352/7352 [=====] - 4s 607us/step - loss: 2.3796e-05 - acc: 1.0000 - val_loss: 0.0053 - val_acc: 0.9986
Epoch 5/20
7352/7352 [=====] - 4s 594us/step - loss: 1.2655e-05 - acc: 1.0000 - val_loss: 0.0049 - val_acc: 0.9986
Epoch 6/20
7352/7352 [=====] - 4s 602us/step - loss: 7.6909e-06 - acc: 1.0000 - val_loss: 0.0051 - val_acc: 0.9986
Epoch 7/20
7352/7352 [=====] - 4s 604us/step - loss: 4.9631e-06 - acc: 1.0000 - val_loss: 0.0049 - val_acc: 0.9986
Epoch 8/20
7352/7352 [=====] - 5s 620us/step - loss: 4.4795e-06 - acc: 1.0000 - val_loss: 0.0051 - val_acc: 0.9990
Epoch 9/20
7352/7352 [=====] - 5s 626us/step - loss: 4.0134e-06 - acc: 1.0000 - val_loss: 0.0048 - val_acc: 0.9990
Epoch 10/20
7352/7352 [=====] - 4s 594us/step - loss: 5.8950e-06 - acc: 1.0000 - val_loss: 0.0048 - val_acc: 0.9990
Epoch 11/20
7352/7352 [=====] - 4s 612us/step - loss: 2.4478e-06 - acc: 1.0000 - val_loss: 0.0062 - val_acc: 0.9986
Epoch 12/20
7352/7352 [=====] - 4s 600us/step - loss: 3.6349e-06 - acc: 1.0000 - val_loss: 0.0056 - val_acc: 0.9986
Epoch 13/20
7352/7352 [=====] - 4s 601us/step - loss: 7.9572e-07 - acc: 1.0000 - val_loss: 0.0050 - val_acc: 0.9986
Epoch 14/20
7352/7352 [=====] - 4s 604us/step - loss: 4.4019e-07 - acc: 1.0000 - val_loss: 0.0058 - val_acc: 0.9986
Epoch 15/20
7352/7352 [=====] - 4s 607us/step - loss: 5.2595e-07 - acc: 1.0000 - val_loss: 0.0064 - val_acc: 0.9986
Epoch 16/20
7352/7352 [=====] - 5s 620us/step - loss: 5.8375e-07 - acc: 1.0000 - val_loss: 0.0070 - val_acc: 0.9986
Epoch 17/20
7352/7352 [=====] - 5s 620us/step - loss: 2.7060e-07 - acc: 1.0000 - val_loss: 0.0065 - val_acc: 0.9986
Epoch 18/20
7352/7352 [=====] - 5s 617us/step - loss: 2.3422e-07 - acc: 1.0000 - val_loss: 0.0066 - val_acc: 0.9986
Epoch 19/20
7352/7352 [=====] - 4s 611us/step - loss: 2.7951e-07 - acc: 1.0000 - val_loss: 0.0067 - val_acc: 0.9986
Epoch 20/20
7352/7352 [=====] - 4s 599us/step - loss: 2.2202e-07 - acc: 1.0000 - val_loss: 0.0067 - val_acc: 0.9986
```

Out[0]:

```
<keras.callbacks.History at 0x7f09485ac668>
```

In [0]:

```
score, acc = model.evaluate(X_val_2c, Y_val_2c)
print("Train Accuracy:", acc)
```

```
2947/2947 [=====] - 0s 132us/step
Train Accuracy: 0.998642687478792
```

Now the model can categorize the activity as static or dynamic with 99.86% accuracy

In [0]:

```
##saving model
model.save(data_path + '2class_classification.h5')
```

### 3 class Classification of static activities

In [0]:

```
from sklearn.preprocessing import StandardScaler
import pickle
```

In [0]:

```
def _read_csv(filename):
    return pd.read_csv(filename, delim_whitespace=True, header=None)

def load_y_static(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 = data_path + f'y_{subset}.txt'
    y = _read_csv(filename)[0]
    y_subset = y>3                      #boolean list of trues and false
    y = y[y_subset]
    return pd.get_dummies(y).as_matrix(), y_subset

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

    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"
    ]

    for signal in SIGNALS:
        filename = data_path+f'Inertial Signals {subset}/{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))

def load_data_static():
    """
    Obtain the dataset from multiple files.
    Returns: y_train, y_test
    """
    X_train, X_test = load_signals('train'), load_signals('test')
    y_train,y_train_subset = load_y_static('train')
    y_test,y_test_subset = load_y_static('test')
    X_train_s = X_train[y_train_subset]
    X_test_s = X_test[y_test_subset]

    ###Scaling data
    remove = int(X_train_s.shape[1] / 2)           #for eliminating the overlap
    temp_X = X_train_s[:, -remove:, :]
    # flatten data
    temp_X = temp_X.reshape((temp_X.shape[0] * temp_X.shape[1], temp_X.shape[2]))
    scale = StandardScaler()
    scale.fit(temp_X)
    #for furter use at prediction pipeline
    pickle.dump(scale,open(data_path + 'Scale_static.p','wb'))

    temp_X1 = X_train_s.reshape((X_train_s.shape[0] * X_train_s.shape[1], X_train_s.shape[2]))
```

```

temp_X1 = scale.transform(temp_X1)
X_train_s = temp_X1.reshape(X_train_s.shape)

temp_X1 = X_test_s.reshape((X_test_s.shape[0] * X_test_s.shape[1], X_test_s.shape[2]))
temp_X1 = scale.transform(temp_X1)
X_test_s = temp_X1.reshape(X_test_s.shape)

return X_train_s, y_train, X_test_s, y_test

```

In [30]:

```
# Loading the train and test data
X_train_static, Y_train_static, X_test_static, Y_test_static = load_data_static()

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:36: FutureWarning: Method .as_matrix
will be removed in a future version. Use .values instead.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:15: FutureWarning: Method .as_matrix
will be removed in a future version. Use .values instead.

    from ipykernel import kernelapp as app
```

In [0]:

```
np.save(data_path + 'X_train_static.npy', X_train_static) # save X_train_static
np.save(data_path + 'X_test_static.npy', X_test_static) # save X_test_static
np.save(data_path + 'Y_train_static.npy', Y_train_static) # save Y_train_static
np.save(data_path + 'Y_test_static.npy', Y_test_static) # save Y_test_static
```

In [0]:

```
def data_static():
    data_path = '/content/drive/My Drive/AAIC/CASE STUDIES/Human Activity Recognition/'
    X_train = np.load(data_path + 'X_train_static.npy') # load X_train_static
    Y_train = np.load(data_path + 'Y_train_static.npy') # load Y_train_static
    X_val = np.load(data_path + 'X_test_static.npy') # load X_test_static
    Y_val = np.load(data_path + 'Y_test_static.npy') # load Y_test_static
    return X_train, Y_train, X_val, Y_val
```

In [0]:

```
score, acc = model.evaluate(X_val, Y_val, verbose=0)
print('Test accuracy:', acc)
return {'loss': -acc, 'status': STATUS_OK, 'model': model}
```

In [36]:

```
best_run_static, best_model_static = optim.minimize(model=create_model_static,
                                                    data=data_static,
                                                    algo=tpe.suggest,
                                                    max_evals=40,
                                                    trials=Trials(),
                                                    notebook_name= 'drive/My Drive/AAIC/Case Studies/Human Activity Recognition/Human Activity Recognition')
```

```
>>> Imports:  
#coding=utf-8
```

```
try:  
    from google.colab import drive  
except:  
    pass
```

```
try:  
    import os  
except:  
    pass
```

```
try:  
    import numpy as np  
except:  
    pass
```

```
try:  
    import pandas as pd  
except:  
    pass
```

```
try:  
    import matplotlib.pyplot as plt  
except:  
    pass
```

```
try:  
    import seaborn as sns  
except:  
    pass
```

```
try:  
    import numpy as np  
except:  
    pass
```

```
try:  
    from sklearn.manifold import TSNE  
except:  
    pass
```

```
try:  
    import matplotlib.pyplot as plt  
except:  
    pass
```

```
try:  
    import seaborn as sns  
except:  
    pass
```

```
try:  
    import itertools  
except:  
    pass
```

```
try:  
    import numpy as np
```

```
except:
    pass

try:
    import matplotlib.pyplot as plt
except:
    pass

try:
    from sklearn.metrics import confusion_matrix
except:
    pass

try:
    from datetime import datetime
except:
    pass

try:
    from sklearn import linear_model
except:
    pass

try:
    from sklearn import metrics
except:
    pass

try:
    from sklearn.model_selection import GridSearchCV
except:
    pass

try:
    from sklearn.svm import SVC
except:
    pass

try:
    from sklearn.tree import DecisionTreeClassifier
except:
    pass

try:
    from sklearn.ensemble import RandomForestClassifier
except:
    pass

try:
    from sklearn.ensemble import GradientBoostingClassifier
except:
    pass

try:
    import tensorflow as tf
except:
    pass

try:
    from keras import backend as K
except:
    pass

try:
    from keras.models import Sequential
except:
    pass

try:
    from keras.layers import LSTM
except:
    pass

try:
    from keras.layers.core import Dense, Dropout
except:
    pass
```

```

try:
    from keras.layers import Flatten
except:
    pass

try:
    from hyperopt import Trials, STATUS_OK, tpe
except:
    pass

try:
    from hyperas import optim
except:
    pass

try:
    from hyperas.distributions import choice, uniform
except:
    pass

try:
    import keras
except:
    pass

try:
    from keras.layers import Conv1D
except:
    pass

try:
    from keras.layers import BatchNormalization
except:
    pass

try:
    from keras.layers import MaxPooling1D
except:
    pass

try:
    from sklearn.preprocessing import StandardScaler
except:
    pass

try:
    import pickle
except:
    pass

>>> Hyperas search space:

def get_space():
    return {
        'filters': hp.choice('filters', [16,32,64]),
        'kernel_size': hp.choice('kernel_size', [2,3,4,5]),
        'filters_1': hp.choice('filters_1', [16,32,64]),
        'kernel_size_1': hp.choice('kernel_size_1', [2,3,4,5]),
        'Dropout': hp.uniform('Dropout', 0,1),
        'pool_size': hp.choice('pool_size', [2,3,5]),
        'filters_2': hp.choice('filters_2', [16,32,64]),
        'lr': hp.choice('lr', [10**-3, 10**-2, 10**-1]),
        'lr_1': hp.choice('lr_1', [10**-3, 10**-2, 10**-1]),
        'choiceval': hp.choice('choiceval', ['adam', 'rmsprop']),
        'filters_3': hp.choice('filters_3', [16,32,64]),
        'epochs': hp.choice('epochs', [25,30,35]),
    }

>>> Data
1:
2: data_path = '/content/drive/My Drive/AAIC/Case Studies/Human Activity Recognition/'
3: X_train = np.load(data_path + 'X_train_static.npy') # load X_train_static
4: Y_train = np.load(data_path + 'Y_train_static.npy') # load Y_train_static
5: X_val = np.load(data_path + 'X_test_static.npy') # load X_test_static
6: Y_val = np.load(data_path + 'Y_test_static.npy') # load Y_test_static
7:

```

```

8:
9:
>>> Resulting replaced keras model:

1: def keras_fmin_fnct(space):
2:
3:     # Initialazing the sequential model
4:     model = Sequential()
5:
6:     model.add(Conv1D(filters=space['filters'],
kernel_size=space['kernel_size'],activation='relu',kernel_initializer='he_uniform',input_shape=(12
8,9)))
7:
8:     model.add(Conv1D(filters=space['filters_1'],
kernel_size=space['kernel_size_1'],activation='relu',kernel_initializer='he_uniform'))
9:     model.add(Dropout(space['Dropout']))
10:    model.add(MaxPooling1D(pool_size=space['pool_size']))
11:    model.add(Flatten())
12:    model.add(Dense(space['filters_2'], activation='relu'))
13:    model.add(Dense(3, activation='softmax'))
14:
15:    adam = keras.optimizers.Adam(lr=space['lr'])
16:    rmsprop = keras.optimizers.RMSprop(lr=space['lr_1'])
17:
18:    choiceval = space['choiceval']
19:
20:    if choiceval == 'adam':
21:        optim = adam
22:    else:
23:        optim = rmsprop
24:
25:    model.compile(loss='categorical_crossentropy', metrics=['accuracy'],optimizer=optim)
26:    model.summary()
27:
28:    result = model.fit(X_train, Y_train,
29:                        batch_size=space['filters_3'],
30:                        epochs=space['epochs'],
31:                        verbose=2,
32:                        validation_data=(X_val, Y_val))
33:
34:    score, acc = model.evaluate(X_val, Y_val, verbose=0)
35:    print('Test accuracy:', acc)
36:    return {'loss': -acc, 'status': STATUS_OK, 'model': model}
37:
Model: "sequential_4"

```

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_7 (Conv1D)	(None, 124, 64)	2944
conv1d_8 (Conv1D)	(None, 121, 32)	8224
dropout_4 (Dropout)	(None, 121, 32)	0
max_pooling1d_4 (MaxPooling1	(None, 24, 32)	0
flatten_4 (Flatten)	(None, 768)	0
dense_7 (Dense)	(None, 32)	24608
dense_8 (Dense)	(None, 3)	99
<hr/>		

Total params: 35,875  
Trainable params: 35,875  
Non-trainable params: 0

---

Train on 4067 samples, validate on 1560 samples  
Epoch 1/35  
- 1s - loss: 0.3175 - acc: 0.8778 - val\_loss: 0.3146 - val\_acc: 0.8821

Epoch 2/35  
- 1s - loss: 0.1810 - acc: 0.9253 - val\_loss: 0.3036 - val\_acc: 0.8833

Epoch 3/35  
- 1s - loss: 0.1576 - acc: 0.9343 - val\_loss: 0.2707 - val\_acc: 0.9038

Epoch 4/35

```
- 1s - loss: 0.1416 - acc: 0.9447 - val_loss: 0.2484 - val_acc: 0.9199
Epoch 5/35
- 1s - loss: 0.1294 - acc: 0.9523 - val_loss: 0.2705 - val_acc: 0.9058
Epoch 6/35
- 1s - loss: 0.1110 - acc: 0.9570 - val_loss: 0.2396 - val_acc: 0.9218
Epoch 7/35
- 1s - loss: 0.0842 - acc: 0.9631 - val_loss: 0.2560 - val_acc: 0.9231
Epoch 8/35
- 1s - loss: 0.0725 - acc: 0.9737 - val_loss: 0.2283 - val_acc: 0.9385
Epoch 9/35
- 1s - loss: 0.0665 - acc: 0.9722 - val_loss: 0.2866 - val_acc: 0.9231
Epoch 10/35
- 1s - loss: 0.0604 - acc: 0.9766 - val_loss: 0.2343 - val_acc: 0.9442
Epoch 11/35
- 1s - loss: 0.0536 - acc: 0.9776 - val_loss: 0.2214 - val_acc: 0.9474
Epoch 12/35
- 1s - loss: 0.0408 - acc: 0.9833 - val_loss: 0.2431 - val_acc: 0.9442
Epoch 13/35
- 1s - loss: 0.0839 - acc: 0.9730 - val_loss: 0.4692 - val_acc: 0.9109
Epoch 14/35
- 1s - loss: 0.0953 - acc: 0.9744 - val_loss: 0.2882 - val_acc: 0.9397
Epoch 15/35
- 1s - loss: 0.0415 - acc: 0.9865 - val_loss: 0.2363 - val_acc: 0.9442
Epoch 16/35
- 1s - loss: 0.0285 - acc: 0.9902 - val_loss: 0.2916 - val_acc: 0.9308
Epoch 17/35
- 1s - loss: 0.0261 - acc: 0.9911 - val_loss: 0.2604 - val_acc: 0.9404
Epoch 18/35
- 1s - loss: 0.0283 - acc: 0.9894 - val_loss: 0.2428 - val_acc: 0.9449
Epoch 19/35
- 1s - loss: 0.0176 - acc: 0.9946 - val_loss: 0.2578 - val_acc: 0.9494
Epoch 20/35
- 1s - loss: 0.0149 - acc: 0.9951 - val_loss: 0.2736 - val_acc: 0.9513
Epoch 21/35
- 1s - loss: 0.0119 - acc: 0.9970 - val_loss: 0.2835 - val_acc: 0.9494
Epoch 22/35
- 1s - loss: 0.0219 - acc: 0.9919 - val_loss: 0.3495 - val_acc: 0.9263
Epoch 23/35
- 1s - loss: 0.0235 - acc: 0.9919 - val_loss: 0.3771 - val_acc: 0.9372
Epoch 24/35
- 1s - loss: 0.0173 - acc: 0.9966 - val_loss: 0.3644 - val_acc: 0.9455
Epoch 25/35
- 1s - loss: 0.0145 - acc: 0.9963 - val_loss: 0.3513 - val_acc: 0.9442
Epoch 26/35
- 1s - loss: 0.0078 - acc: 0.9980 - val_loss: 0.4327 - val_acc: 0.9308
Epoch 27/35
- 1s - loss: 0.0260 - acc: 0.9909 - val_loss: 0.3238 - val_acc: 0.9417
Epoch 28/35
- 1s - loss: 0.0331 - acc: 0.9887 - val_loss: 0.4197 - val_acc: 0.9391
Epoch 29/35
- 1s - loss: 0.0163 - acc: 0.9958 - val_loss: 0.4669 - val_acc: 0.9321
```

```

Epoch 30/35
- 1s - loss: 0.0168 - acc: 0.9951 - val_loss: 0.5306 - val_acc: 0.9186

Epoch 31/35
- 1s - loss: 0.0531 - acc: 0.9865 - val_loss: 0.5349 - val_acc: 0.9135

Epoch 32/35
- 1s - loss: 0.0693 - acc: 0.9813 - val_loss: 0.3636 - val_acc: 0.9481

Epoch 33/35
- 1s - loss: 0.0146 - acc: 0.9973 - val_loss: 0.3822 - val_acc: 0.9513

Epoch 34/35
- 1s - loss: 0.0140 - acc: 0.9973 - val_loss: 0.4201 - val_acc: 0.9410

Epoch 35/35
- 1s - loss: 0.0075 - acc: 0.9998 - val_loss: 0.3775 - val_acc: 0.9481

Test accuracy:
0.948076923076923
Model: "sequential_5"

Layer (type)          Output Shape         Param #
=====
conv1d_9 (Conv1D)     (None, 125, 64)      2368
conv1d_10 (Conv1D)    (None, 124, 64)       8256
dropout_5 (Dropout)   (None, 124, 64)       0
max_pooling1d_5 (MaxPooling1) (None, 62, 64)  0
flatten_5 (Flatten)   (None, 3968)        0
dense_9 (Dense)       (None, 64)           254016
dense_10 (Dense)      (None, 3)            195
=====
Total params: 264,835
Trainable params: 264,835
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/30
- 1s - loss: 10.6057 - acc: 0.3359 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 2/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 3/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 4/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 5/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 6/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 7/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 8/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 9/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 10/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 11/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 12/30

```

```

- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 13/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 14/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 15/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 16/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 17/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 18/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 19/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 20/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 21/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 22/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 23/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 24/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 25/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 26/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 27/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 28/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 29/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 30/30
- 1s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Test accuracy:
0.34102564102564104
 5%|██████████| 2/40 [01:06<21:43, 34.29s/it, best loss: -
0.948076923076923]WARNING:tensorflow:Large dropout rate: 0.762558 (>0.5). In TensorFlow 2.x,
dropout() uses dropout rate instead of keep_prob. Please ensure that this is intended.
Model: "sequential_6"


```

Layer (type)	Output Shape	Param #
conv1d_11 (Conv1D)	(None, 124, 32)	1472
conv1d_12 (Conv1D)	(None, 123, 16)	1040
dropout_6 (Dropout)	(None, 123, 16)	0
max_pooling1d_6 (MaxPooling1)	(None, 41, 16)	0
flatten_6 (Flatten)	(None, 656)	0
dense_11 (Dense)	(None, 64)	42048

```
dense_12 (Dense)           (None, 3)          195
=====
Total params: 44,755
Trainable params: 44,755
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/35
- 2s - loss: 10.4591 - acc: 0.3469 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 2/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 3/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 4/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 5/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 6/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 7/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 8/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 9/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 10/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 11/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 12/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 13/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 14/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 15/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 16/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 17/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 18/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 19/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 20/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 21/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 22/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 23/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 24/35
```

```

- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 25/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 26/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 27/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 28/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 29/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 30/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 31/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 32/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 33/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 34/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 35/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Test accuracy:
0.34423076923076923
 8%|██████████| 3/40 [01:40<21:03, 34.14s/it, best loss: -
0.948076923076923]WARNING:tensorflow:Large dropout rate: 0.81458 (>0.5). In TensorFlow 2.x,
dropout() uses dropout rate instead of keep_prob. Please ensure that this is intended.
Model: "sequential_7"

Layer (type)          Output Shape         Param #
=================================================================
conv1d_13 (Conv1D)    (None, 125, 16)      592
conv1d_14 (Conv1D)    (None, 123, 16)      784
dropout_7 (Dropout)   (None, 123, 16)      0
max_pooling1d_7 (MaxPooling1) (None, 41, 16)  0
flatten_7 (Flatten)   (None, 656)          0
dense_13 (Dense)     (None, 64)           42048
dense_14 (Dense)     (None, 3)            195
=================================================================
Total params: 43,619
Trainable params: 43,619
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/35
- 2s - loss: 0.5989 - acc: 0.8608 - val_loss: 0.2951 - val_acc: 0.9000

Epoch 2/35
- 2s - loss: 0.4405 - acc: 0.8982 - val_loss: 0.2559 - val_acc: 0.9231

Epoch 3/35
- 2s - loss: 0.5278 - acc: 0.9019 - val_loss: 0.4898 - val_acc: 0.9000

Epoch 4/35
- 2s - loss: 0.4663 - acc: 0.9056 - val_loss: 0.3354 - val_acc: 0.9026

Epoch 5/35
- 2s - loss: 0.4966 - acc: 0.9058 - val_loss: 0.3130 - val_acc: 0.9199

```

Epoch 6/35  
- 2s - loss: 0.5083 - acc: 0.9115 - val\_loss: 0.8116 - val\_acc: 0.8808

Epoch 7/35  
- 2s - loss: 0.4902 - acc: 0.9046 - val\_loss: 0.4763 - val\_acc: 0.9167

Epoch 8/35  
- 2s - loss: 0.6780 - acc: 0.8955 - val\_loss: 1.7180 - val\_acc: 0.8308

Epoch 9/35  
- 2s - loss: 0.6350 - acc: 0.8832 - val\_loss: 0.7481 - val\_acc: 0.8865

Epoch 10/35  
- 2s - loss: 0.4275 - acc: 0.8923 - val\_loss: 0.3269 - val\_acc: 0.8795

Epoch 11/35  
- 2s - loss: 0.4394 - acc: 0.8921 - val\_loss: 0.7290 - val\_acc: 0.8904

Epoch 12/35  
- 2s - loss: 1.0724 - acc: 0.8653 - val\_loss: 1.4004 - val\_acc: 0.8455

Epoch 13/35  
- 2s - loss: 1.1187 - acc: 0.8581 - val\_loss: 0.8765 - val\_acc: 0.8782

Epoch 14/35  
- 2s - loss: 0.6255 - acc: 0.8827 - val\_loss: 1.2507 - val\_acc: 0.6994

Epoch 15/35  
- 2s - loss: 0.6180 - acc: 0.8908 - val\_loss: 0.9006 - val\_acc: 0.8635

Epoch 16/35  
- 2s - loss: 0.6634 - acc: 0.8803 - val\_loss: 1.1109 - val\_acc: 0.8622

Epoch 17/35  
- 2s - loss: 0.7150 - acc: 0.8662 - val\_loss: 0.9004 - val\_acc: 0.8417

Epoch 18/35  
- 2s - loss: 1.2935 - acc: 0.8421 - val\_loss: 1.7860 - val\_acc: 0.8077

Epoch 19/35  
- 2s - loss: 1.0226 - acc: 0.8633 - val\_loss: 0.8869 - val\_acc: 0.8538

Epoch 20/35  
- 2s - loss: 0.7466 - acc: 0.8716 - val\_loss: 1.1184 - val\_acc: 0.8494

Epoch 21/35  
- 2s - loss: 0.9944 - acc: 0.8589 - val\_loss: 1.3348 - val\_acc: 0.8244

Epoch 22/35  
- 2s - loss: 0.9549 - acc: 0.8537 - val\_loss: 1.2021 - val\_acc: 0.7955

Epoch 23/35  
- 2s - loss: 1.0325 - acc: 0.8505 - val\_loss: 1.6279 - val\_acc: 0.7949

Epoch 24/35  
- 2s - loss: 1.1309 - acc: 0.8535 - val\_loss: 1.3877 - val\_acc: 0.8654

Epoch 25/35  
- 2s - loss: 0.9160 - acc: 0.8655 - val\_loss: 0.6192 - val\_acc: 0.8949

Epoch 26/35  
- 2s - loss: 0.6657 - acc: 0.8822 - val\_loss: 0.7487 - val\_acc: 0.8699

Epoch 27/35  
- 2s - loss: 0.6605 - acc: 0.8771 - val\_loss: 0.6006 - val\_acc: 0.8538

Epoch 28/35  
- 2s - loss: 0.7214 - acc: 0.8795 - val\_loss: 0.5331 - val\_acc: 0.8199

Epoch 29/35  
- 2s - loss: 0.5030 - acc: 0.8736 - val\_loss: 0.7473 - val\_acc: 0.8250

Epoch 30/35  
- 2s - loss: 0.8632 - acc: 0.8660 - val\_loss: 1.0499 - val\_acc: 0.8500

Epoch 31/35

```

- 2s - loss: 0.5855 - acc: 0.8825 - val_loss: 1.1523 - val_acc: 0.8532

Epoch 32/35
- 2s - loss: 0.4778 - acc: 0.8842 - val_loss: 0.6493 - val_acc: 0.8763

Epoch 33/35
- 2s - loss: 0.5420 - acc: 0.8778 - val_loss: 1.0483 - val_acc: 0.8795

Epoch 34/35
- 2s - loss: 0.7796 - acc: 0.8724 - val_loss: 1.1705 - val_acc: 0.8385

Epoch 35/35
- 2s - loss: 0.7506 - acc: 0.8680 - val_loss: 1.1025 - val_acc: 0.8596

Test accuracy:
0.8596153846153847
10%|██████████| 4/40 [02:38<24:42, 41.18s/it, best loss: -
0.948076923076923]WARNING:tensorflow:Large dropout rate: 0.814239 (>0.5). In TensorFlow 2.x,
dropout() uses dropout rate instead of keep_prob. Please ensure that this is intended.
Model: "sequential_8"

Layer (type)          Output Shape         Param #
=================================================================
conv1d_15 (Conv1D)    (None, 124, 32)      1472
conv1d_16 (Conv1D)    (None, 122, 16)       1552
dropout_8 (Dropout)   (None, 122, 16)       0
max_pooling1d_8 (MaxPooling1) (None, 61, 16)  0
flatten_8 (Flatten)   (None, 976)           0
dense_15 (Dense)     (None, 64)            62528
dense_16 (Dense)     (None, 3)             195
=====
Total params: 65,747
Trainable params: 65,747
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/30
- 1s - loss: 10.8810 - acc: 0.3167 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 2/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 3/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 4/30
- 1s - loss: 11.0255 - acc: 0.3160 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 5/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 6/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 7/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 8/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 9/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 10/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 11/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 12/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

```

```

-- 1000. 11.0215  acc: 0.3162  val_loss: 11.0450  val_acc: 0.3147

Epoch 13/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 14/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 15/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 16/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 17/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 18/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 19/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 20/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 21/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 22/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 23/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 24/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 25/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 26/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 27/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 28/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 29/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 30/30
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Test accuracy:
0.31474358974358974
12%|██████████| 5/40 [02:54<19:45, 33.86s/it, best loss: -
0.948076923076923]WARNING:tensorflow:Large dropout rate: 0.927916 (>0.5). In TensorFlow 2.x,
dropout() uses dropout rate instead of keep_prob. Please ensure that this is intended.
Model: "sequential_9"


```

Layer (type)	Output Shape	Param #
conv1d_17 (Conv1D)	(None, 124, 64)	2944
conv1d_18 (Conv1D)	(None, 121, 16)	4112
dropout_9 (Dropout)	(None, 121, 16)	0
max_pooling1d_9 (MaxPooling1)	(None, 24, 16)	0
flatten_9 (Flatten)	(None, 384)	0
dense_17 (Dense)	(None, 32)	12320
dense_18 (Dense)	(None, 3)	99

```
accuracy_percent, loss_percent, val_accuracy_percent, val_loss_percent
=====
Total params: 19,475
Trainable params: 19,475
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/30
- 3s - loss: 9.5165 - acc: 0.4087 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 2/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 3/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 4/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 5/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 6/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 7/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 8/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 9/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 10/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 11/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 12/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 13/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 14/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 15/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 16/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 17/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 18/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 19/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 20/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 21/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 22/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 23/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 24/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410
```

```

2s    loss: 10.6727    acc: 0.3378    val_loss: 10.6214    val_acc: 0.3410
Epoch 25/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 26/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 27/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 28/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 29/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Epoch 30/30
- 2s - loss: 10.6727 - acc: 0.3378 - val_loss: 10.6214 - val_acc: 0.3410

Test accuracy:
0.34102564102564104
Model: "sequential_10"

Layer (type)          Output Shape         Param #
=====             ======           =====
conv1d_19 (Conv1D)      (None, 124, 16)       736
conv1d_20 (Conv1D)      (None, 121, 32)      2080
dropout_10 (Dropout)    (None, 121, 32)        0
max_pooling1d_10 (MaxPooling) (None, 60, 32)      0
flatten_10 (Flatten)    (None, 1920)          0
dense_19 (Dense)        (None, 32)          61472
dense_20 (Dense)        (None, 3)            99
=====
Total params: 64,387
Trainable params: 64,387
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/35
- 3s - loss: 0.3292 - acc: 0.8793 - val_loss: 0.2585 - val_acc: 0.8942

Epoch 2/35
- 2s - loss: 0.1950 - acc: 0.9272 - val_loss: 0.2433 - val_acc: 0.8917

Epoch 3/35
- 2s - loss: 0.1684 - acc: 0.9368 - val_loss: 0.2507 - val_acc: 0.9071

Epoch 4/35
- 2s - loss: 0.1387 - acc: 0.9457 - val_loss: 0.2843 - val_acc: 0.9083

Epoch 5/35
- 2s - loss: 0.1216 - acc: 0.9565 - val_loss: 0.3034 - val_acc: 0.9071

Epoch 6/35
- 2s - loss: 0.1067 - acc: 0.9597 - val_loss: 0.2362 - val_acc: 0.9173

Epoch 7/35
- 2s - loss: 0.0892 - acc: 0.9685 - val_loss: 0.2960 - val_acc: 0.9199

Epoch 8/35
- 2s - loss: 0.0790 - acc: 0.9742 - val_loss: 0.3120 - val_acc: 0.9237

Epoch 9/35
- 2s - loss: 0.0801 - acc: 0.9747 - val_loss: 0.2596 - val_acc: 0.9346

Epoch 10/35
- 2s - loss: 0.0580 - acc: 0.9806 - val_loss: 0.3241 - val_acc: 0.9333

Epoch 11/35
- 2s - loss: 0.0734 - acc: 0.9791 - val_loss: 0.2579 - val_acc: 0.9288

```

```
Epoch 12/35
- 2s - loss: 0.0513 - acc: 0.9830 - val_loss: 0.3239 - val_acc: 0.9372

Epoch 13/35
- 2s - loss: 0.0658 - acc: 0.9845 - val_loss: 0.3677 - val_acc: 0.9314

Epoch 14/35
- 2s - loss: 0.0599 - acc: 0.9843 - val_loss: 0.3526 - val_acc: 0.9321

Epoch 15/35
- 2s - loss: 0.0571 - acc: 0.9867 - val_loss: 0.4633 - val_acc: 0.9314

Epoch 16/35
- 2s - loss: 0.0559 - acc: 0.9875 - val_loss: 0.3636 - val_acc: 0.9333

Epoch 17/35
- 2s - loss: 0.0459 - acc: 0.9875 - val_loss: 0.4266 - val_acc: 0.9372

Epoch 18/35
- 2s - loss: 0.0454 - acc: 0.9882 - val_loss: 0.4612 - val_acc: 0.9391

Epoch 19/35
- 2s - loss: 0.0411 - acc: 0.9892 - val_loss: 0.4633 - val_acc: 0.9346

Epoch 20/35
- 2s - loss: 0.0446 - acc: 0.9892 - val_loss: 0.4323 - val_acc: 0.9353

Epoch 21/35
- 2s - loss: 0.0450 - acc: 0.9887 - val_loss: 0.5337 - val_acc: 0.9327

Epoch 22/35
- 2s - loss: 0.0488 - acc: 0.9894 - val_loss: 0.4608 - val_acc: 0.9372

Epoch 23/35
- 2s - loss: 0.0454 - acc: 0.9916 - val_loss: 0.4639 - val_acc: 0.9410

Epoch 24/35
- 2s - loss: 0.0375 - acc: 0.9929 - val_loss: 0.5880 - val_acc: 0.9186

Epoch 25/35
- 2s - loss: 0.0499 - acc: 0.9916 - val_loss: 0.4947 - val_acc: 0.9135

Epoch 26/35
- 2s - loss: 0.0502 - acc: 0.9914 - val_loss: 0.5112 - val_acc: 0.9340

Epoch 27/35
- 2s - loss: 0.0409 - acc: 0.9916 - val_loss: 0.4545 - val_acc: 0.9404

Epoch 28/35
- 2s - loss: 0.0375 - acc: 0.9929 - val_loss: 0.5348 - val_acc: 0.9385

Epoch 29/35
- 2s - loss: 0.0407 - acc: 0.9924 - val_loss: 0.4579 - val_acc: 0.9442

Epoch 30/35
- 2s - loss: 0.0260 - acc: 0.9939 - val_loss: 0.5012 - val_acc: 0.9410

Epoch 31/35
- 2s - loss: 0.0602 - acc: 0.9907 - val_loss: 0.4812 - val_acc: 0.9462

Epoch 32/35
- 2s - loss: 0.0377 - acc: 0.9953 - val_loss: 0.5492 - val_acc: 0.9385

Epoch 33/35
- 2s - loss: 0.0476 - acc: 0.9936 - val_loss: 0.7020 - val_acc: 0.9308

Epoch 34/35
- 2s - loss: 0.0378 - acc: 0.9939 - val_loss: 0.5369 - val_acc: 0.9372

Epoch 35/35
- 2s - loss: 0.0388 - acc: 0.9941 - val_loss: 0.5143 - val_acc: 0.9385

Test accuracy:
0.9384615384615385
Model: "sequential_11"
```

layer (type)	output shape	params #
conv1d_21 (Conv1D)	(None, 125, 16)	592
conv1d_22 (Conv1D)	(None, 122, 64)	4160
dropout_11 (Dropout)	(None, 122, 64)	0
max_pooling1d_11 (MaxPooling)	(None, 24, 64)	0
flatten_11 (Flatten)	(None, 1536)	0
dense_21 (Dense)	(None, 32)	49184
dense_22 (Dense)	(None, 3)	99

Total params: 54,035

Trainable params: 54,035

Non-trainable params: 0

---

Train on 4067 samples, validate on 1560 samples

Epoch 1/30

- 3s - loss: 1.1440 - acc: 0.8367 - val\_loss: 0.5189 - val\_acc: 0.8814

Epoch 2/30

- 2s - loss: 0.5538 - acc: 0.8997 - val\_loss: 0.4325 - val\_acc: 0.9173

Epoch 3/30

- 2s - loss: 1.1855 - acc: 0.8709 - val\_loss: 0.2738 - val\_acc: 0.9096

Epoch 4/30

- 2s - loss: 0.2618 - acc: 0.9380 - val\_loss: 0.3165 - val\_acc: 0.9115

Epoch 5/30

- 2s - loss: 0.3264 - acc: 0.9343 - val\_loss: 0.4510 - val\_acc: 0.8942

Epoch 6/30

- 2s - loss: 0.3335 - acc: 0.9326 - val\_loss: 0.2787 - val\_acc: 0.9256

Epoch 7/30

- 2s - loss: 0.2446 - acc: 0.9452 - val\_loss: 0.2314 - val\_acc: 0.9423

Epoch 8/30

- 2s - loss: 0.2958 - acc: 0.9415 - val\_loss: 0.3408 - val\_acc: 0.9071

Epoch 9/30

- 2s - loss: 0.1972 - acc: 0.9469 - val\_loss: 0.4117 - val\_acc: 0.9506

Epoch 10/30

- 2s - loss: 0.2825 - acc: 0.9358 - val\_loss: 0.4735 - val\_acc: 0.9359

Epoch 11/30

- 2s - loss: 0.3452 - acc: 0.9427 - val\_loss: 0.3780 - val\_acc: 0.9391

Epoch 12/30

- 2s - loss: 0.3836 - acc: 0.9432 - val\_loss: 0.8486 - val\_acc: 0.8962

Epoch 13/30

- 2s - loss: 0.3760 - acc: 0.9366 - val\_loss: 0.8323 - val\_acc: 0.9064

Epoch 14/30

- 2s - loss: 0.3904 - acc: 0.9412 - val\_loss: 0.3144 - val\_acc: 0.9378

Epoch 15/30

- 2s - loss: 0.2842 - acc: 0.9474 - val\_loss: 0.5352 - val\_acc: 0.7904

Epoch 16/30

- 2s - loss: 0.2824 - acc: 0.9457 - val\_loss: 0.9041 - val\_acc: 0.9135

Epoch 17/30

- 2s - loss: 0.2680 - acc: 0.9457 - val\_loss: 0.7463 - val\_acc: 0.9179

Epoch 18/30

- 2s - loss: 0.3937 - acc: 0.9427 - val\_loss: 0.8177 - val\_acc: 0.9218

Epoch 19/30

- 2s - loss: 0.5373 - acc: 0.9319 - val\_loss: 1.2553 - val\_acc: 0.8904

```
Epoch 20/30
- 2s - loss: 0.4187 - acc: 0.9358 - val_loss: 0.7752 - val_acc: 0.9276

Epoch 21/30
- 2s - loss: 0.5449 - acc: 0.9248 - val_loss: 1.0281 - val_acc: 0.9013

Epoch 22/30
- 2s - loss: 0.5693 - acc: 0.9282 - val_loss: 0.8317 - val_acc: 0.9038

Epoch 23/30
- 2s - loss: 0.5238 - acc: 0.9343 - val_loss: 1.3341 - val_acc: 0.8827

Epoch 24/30
- 2s - loss: 0.7590 - acc: 0.9157 - val_loss: 1.3478 - val_acc: 0.8288

Epoch 25/30
- 2s - loss: 1.0165 - acc: 0.8820 - val_loss: 2.4102 - val_acc: 0.8237

Epoch 26/30
- 2s - loss: 0.7106 - acc: 0.9075 - val_loss: 0.5670 - val_acc: 0.9295

Epoch 27/30
- 2s - loss: 0.4127 - acc: 0.9412 - val_loss: 0.6187 - val_acc: 0.9321

Epoch 28/30
- 2s - loss: 1.0785 - acc: 0.8945 - val_loss: 1.5786 - val_acc: 0.8731

Epoch 29/30
- 2s - loss: 0.7501 - acc: 0.9221 - val_loss: 0.8895 - val_acc: 0.9115

Epoch 30/30
- 2s - loss: 0.5315 - acc: 0.9336 - val_loss: 0.8533 - val_acc: 0.9141
```

Test accuracy:

0.9141025641025641

20%|██████████| 8/40 [05:43<25:48, 48.39s/it, best loss: -

0.948076923076923]WARNING:tensorflow:Large dropout rate: 0.841409 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate instead of keep\_prob. Please ensure that this is intended.

Model: "sequential\_12"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_23 (Conv1D)	(None, 126, 32)	896
conv1d_24 (Conv1D)	(None, 122, 16)	2576
dropout_12 (Dropout)	(None, 122, 16)	0
max_pooling1d_12 (MaxPooling)	(None, 61, 16)	0
flatten_12 (Flatten)	(None, 976)	0
dense_23 (Dense)	(None, 32)	31264
dense_24 (Dense)	(None, 3)	99
<hr/>		

Total params: 34,835

Trainable params: 34,835

Non-trainable params: 0

---

Train on 4067 samples, validate on 1560 samples

Epoch 1/30

- 2s - loss: 10.6803 - acc: 0.3314 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 2/30

- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 3/30

- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 4/30

- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 5/30

- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Test accuracy:  
0.34423076923076923

Model: "sequential\_13"

Layer (type)	Output Shape	Param #
conv1d_25 (Conv1D)	(None, 125, 32)	1184
conv1d_26 (Conv1D)	(None, 124, 64)	4160
dropout_13 (Dropout)	(None, 124, 64)	0
max_pooling1d_13 (MaxPooling)	(None, 41, 64)	0
flatten_13 (Flatten)	(None, 2624)	0
dense_25 (Dense)	(None, 64)	168000
dense_26 (Dense)	(None, 3)	195

Total params: 173,539  
Trainable params: 173,539  
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples

Epoch 1/25

- 3s - loss: 0.4596 - acc: 0.8635 - val\_loss: 0.2949 - val\_acc: 0.8814

Epoch 2/25

- 2s - loss: 0.2238 - acc: 0.9262 - val\_loss: 0.5016 - val\_acc: 0.8712

Epoch 3/25

- 2s - loss: 0.1645 - acc: 0.9447 - val\_loss: 0.4010 - val\_acc: 0.8545

Epoch 4/25

- 2s - loss: 0.1268 - acc: 0.9572 - val\_loss: 0.3058 - val\_acc: 0.9263

Epoch 5/25

- 2s - loss: 0.1195 - acc: 0.9629 - val\_loss: 0.3741 - val\_acc: 0.9045

Epoch 6/25

- 2s - loss: 0.0959 - acc: 0.9707 - val\_loss: 0.4732 - val\_acc: 0.9154

Epoch 7/25

- 2s - loss: 0.0978 - acc: 0.9744 - val\_loss: 0.3604 - val\_acc: 0.9167

Epoch 8/25

- 2s - loss: 0.0881 - acc: 0.9771 - val\_loss: 0.3000 - val\_acc: 0.9462

Epoch 9/25

- 2s - loss: 0.0684 - acc: 0.9798 - val\_loss: 0.3935 - val\_acc: 0.9308

Epoch 10/25

- 2s - loss: 0.0647 - acc: 0.9855 - val\_loss: 0.3190 - val\_acc: 0.9455

Epoch 11/25

- 2s - loss: 0.0603 - acc: 0.9850 - val\_loss: 0.3765 - val\_acc: 0.9365

Epoch 12/25

- 2s - loss: 0.0624 - acc: 0.9880 - val\_loss: 0.2376 - val\_acc: 0.9481

Epoch 13/25

- 2s - loss: 0.0654 - acc: 0.9862 - val\_loss: 0.4401 - val\_acc: 0.9282

Epoch 14/25

- 2s - loss: 0.0767 - acc: 0.9884 - val\_loss: 0.4971 - val\_acc: 0.9385

Epoch 15/25

- 2s - loss: 0.0573 - acc: 0.9875 - val\_loss: 0.3387 - val\_acc: 0.9410

Epoch 16/25

- 2s - loss: 0.0421 - acc: 0.9911 - val\_loss: 0.4231 - val\_acc: 0.9410

Epoch 17/25

- 2s - loss: 0.0641 - acc: 0.9899 - val\_loss: 0.4668 - val\_acc: 0.9455

Epoch 18/25

- 2s - loss: 0.0454 - acc: 0.9924 - val\_loss: 0.4838 - val\_acc: 0.9391

-----

```
Epoch 19/25
- 2s - loss: 0.0731 - acc: 0.9909 - val_loss: 0.5859 - val_acc: 0.9224

Epoch 20/25
- 2s - loss: 0.0611 - acc: 0.9916 - val_loss: 0.4754 - val_acc: 0.9417

Epoch 21/25
- 2s - loss: 0.0312 - acc: 0.9951 - val_loss: 0.4699 - val_acc: 0.9397

Epoch 22/25
- 2s - loss: 0.0407 - acc: 0.9936 - val_loss: 0.5994 - val_acc: 0.9346

Epoch 23/25
- 2s - loss: 0.0285 - acc: 0.9961 - val_loss: 0.4288 - val_acc: 0.9513

Epoch 24/25
- 2s - loss: 0.0427 - acc: 0.9931 - val_loss: 0.5865 - val_acc: 0.9282

Epoch 25/25
- 2s - loss: 0.0648 - acc: 0.9931 - val_loss: 0.4198 - val_acc: 0.9532
```

Test accuracy:

0.9532051282051283

Model: "sequential\_14"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_27 (Conv1D)	(None, 126, 32)	896
conv1d_28 (Conv1D)	(None, 125, 32)	2080
dropout_14 (Dropout)	(None, 125, 32)	0
max_pooling1d_14 (MaxPooling)	(None, 41, 32)	0
flatten_14 (Flatten)	(None, 1312)	0
dense_27 (Dense)	(None, 64)	84032
dense_28 (Dense)	(None, 3)	195
<hr/>		

Total params: 87,203

Trainable params: 87,203

Non-trainable params: 0

---

Train on 4067 samples, validate on 1560 samples

Epoch 1/25

- 2s - loss: 10.8839 - acc: 0.3204 - val\_loss: 11.0450 - val\_acc: 0.3147

Epoch 2/25

- 0s - loss: 11.0215 - acc: 0.3162 - val\_loss: 11.0450 - val\_acc: 0.3147

Epoch 3/25

- 1s - loss: 11.0215 - acc: 0.3162 - val\_loss: 11.0450 - val\_acc: 0.3147

Epoch 4/25

- 1s - loss: 11.0215 - acc: 0.3162 - val\_loss: 11.0450 - val\_acc: 0.3147

Epoch 5/25

- 0s - loss: 11.0215 - acc: 0.3162 - val\_loss: 11.0450 - val\_acc: 0.3147

Epoch 6/25

- 0s - loss: 11.0215 - acc: 0.3162 - val\_loss: 11.0450 - val\_acc: 0.3147

Epoch 7/25

- 1s - loss: 11.0215 - acc: 0.3162 - val\_loss: 11.0450 - val\_acc: 0.3147

Epoch 8/25

- 1s - loss: 11.0215 - acc: 0.3162 - val\_loss: 11.0450 - val\_acc: 0.3147

Epoch 9/25

- 0s - loss: 11.0215 - acc: 0.3162 - val\_loss: 11.0450 - val\_acc: 0.3147

Epoch 10/25

- 0s - loss: 11.0215 - acc: 0.3162 - val\_loss: 11.0450 - val\_acc: 0.3147

Epoch 11/25

```

- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 12/25
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 13/25
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 14/25
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 15/25
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 16/25
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 17/25
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 18/25
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 19/25
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 20/25
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 21/25
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 22/25
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 23/25
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 24/25
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 25/25
- 0s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Test accuracy:
0.31474358974358974
Model: "sequential_15"

Layer (type)          Output Shape         Param #
=====
conv1d_29 (Conv1D)    (None, 124, 64)      2944
conv1d_30 (Conv1D)    (None, 121, 32)       8224
dropout_15 (Dropout) (None, 121, 32)       0
max_pooling1d_15 (MaxPooling) (None, 40, 32) 0
flatten_15 (Flatten)  (None, 1280)        0
dense_29 (Dense)     (None, 64)           81984
dense_30 (Dense)     (None, 3)            195
=====
Total params: 93,347
Trainable params: 93,347
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/35
- 2s - loss: 0.3503 - acc: 0.8692 - val_loss: 0.3155 - val_acc: 0.8897

Epoch 2/35
- 1s - loss: 0.1906 - acc: 0.9245 - val_loss: 0.2726 - val_acc: 0.9179

Epoch 3/35
- 1s - loss: 0.1512 - acc: 0.9422 - val_loss: 0.3105 - val_acc: 0.9173

```

```
Epoch 4/35
- 1s - loss: 0.1245 - acc: 0.9523 - val_loss: 0.6799 - val_acc: 0.8449

Epoch 5/35
- 1s - loss: 0.1135 - acc: 0.9629 - val_loss: 0.3123 - val_acc: 0.9455

Epoch 6/35
- 1s - loss: 0.0836 - acc: 0.9732 - val_loss: 0.4172 - val_acc: 0.9231

Epoch 7/35
- 1s - loss: 0.0603 - acc: 0.9781 - val_loss: 0.3932 - val_acc: 0.9442

Epoch 8/35
- 1s - loss: 0.0661 - acc: 0.9781 - val_loss: 0.4312 - val_acc: 0.9237

Epoch 9/35
- 1s - loss: 0.0419 - acc: 0.9857 - val_loss: 0.3983 - val_acc: 0.9353

Epoch 10/35
- 1s - loss: 0.0424 - acc: 0.9867 - val_loss: 0.4575 - val_acc: 0.9404

Epoch 11/35
- 1s - loss: 0.0437 - acc: 0.9904 - val_loss: 0.4325 - val_acc: 0.9404

Epoch 12/35
- 1s - loss: 0.0429 - acc: 0.9897 - val_loss: 0.4492 - val_acc: 0.9449

Epoch 13/35
- 1s - loss: 0.0265 - acc: 0.9916 - val_loss: 0.5578 - val_acc: 0.9199

Epoch 14/35
- 1s - loss: 0.0272 - acc: 0.9921 - val_loss: 0.4750 - val_acc: 0.9288

Epoch 15/35
- 1s - loss: 0.0260 - acc: 0.9946 - val_loss: 0.4225 - val_acc: 0.9442

Epoch 16/35
- 1s - loss: 0.0166 - acc: 0.9943 - val_loss: 0.4615 - val_acc: 0.9410

Epoch 17/35
- 1s - loss: 0.0092 - acc: 0.9978 - val_loss: 0.5513 - val_acc: 0.9346

Epoch 18/35
- 1s - loss: 0.0291 - acc: 0.9951 - val_loss: 0.4619 - val_acc: 0.9429

Epoch 19/35
- 1s - loss: 0.0177 - acc: 0.9956 - val_loss: 0.5000 - val_acc: 0.9423

Epoch 20/35
- 1s - loss: 0.0170 - acc: 0.9958 - val_loss: 0.4600 - val_acc: 0.9436

Epoch 21/35
- 1s - loss: 0.0082 - acc: 0.9980 - val_loss: 1.0408 - val_acc: 0.8622

Epoch 22/35
- 1s - loss: 0.0408 - acc: 0.9943 - val_loss: 0.5639 - val_acc: 0.9359

Epoch 23/35
- 1s - loss: 0.0255 - acc: 0.9961 - val_loss: 0.6051 - val_acc: 0.9321

Epoch 24/35
- 1s - loss: 0.0118 - acc: 0.9970 - val_loss: 0.5664 - val_acc: 0.9372

Epoch 25/35
- 1s - loss: 0.0418 - acc: 0.9946 - val_loss: 0.6734 - val_acc: 0.9250

Epoch 26/35
- 1s - loss: 0.0133 - acc: 0.9973 - val_loss: 0.5321 - val_acc: 0.9436

Epoch 27/35
- 1s - loss: 0.0082 - acc: 0.9990 - val_loss: 0.7560 - val_acc: 0.9218

Epoch 28/35
- 1s - loss: 0.0188 - acc: 0.9970 - val_loss: 0.6284 - val_acc: 0.9295

Epoch 29/35
```

```

- 1s - loss: 0.0156 - acc: 0.9973 - val_loss: 0.6175 - val_acc: 0.9378

Epoch 30/35
- 1s - loss: 0.0195 - acc: 0.9968 - val_loss: 0.8695 - val_acc: 0.9109

Epoch 31/35
- 1s - loss: 0.0116 - acc: 0.9983 - val_loss: 0.6425 - val_acc: 0.9333

Epoch 32/35
- 1s - loss: 0.0200 - acc: 0.9973 - val_loss: 0.5696 - val_acc: 0.9365

Epoch 33/35
- 1s - loss: 0.0158 - acc: 0.9970 - val_loss: 0.6497 - val_acc: 0.9340

Epoch 34/35
- 1s - loss: 0.0121 - acc: 0.9978 - val_loss: 0.7280 - val_acc: 0.9321

Epoch 35/35
- 1s - loss: 0.0204 - acc: 0.9966 - val_loss: 0.7522 - val_acc: 0.9321

Test accuracy:
0.9320512820512821
Model: "sequential_16"

Layer (type)          Output Shape         Param #
=====
conv1d_31 (Conv1D)    (None, 126, 64)      1792
conv1d_32 (Conv1D)    (None, 125, 32)       4128
dropout_16 (Dropout)  (None, 125, 32)       0
max_pooling1d_16 (MaxPooling) (None, 41, 32)  0
flatten_16 (Flatten)  (None, 1312)          0
dense_31 (Dense)     (None, 64)            84032
dense_32 (Dense)     (None, 3)              195
=====
Total params: 90,147
Trainable params: 90,147
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/35
- 2s - loss: 10.4890 - acc: 0.3450 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 2/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 3/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 4/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 5/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 6/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 7/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 8/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 9/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 10/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 11/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

```

```
Epoch 12/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 13/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 14/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 15/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 16/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 17/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 18/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 19/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 20/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 21/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 22/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 23/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 24/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 25/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 26/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 27/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 28/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 29/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 30/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 31/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 32/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 33/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 34/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 35/35
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Test accuracy:
0.34423076923076923
Model: "sequential_17"
```

Layer (type)	Output Shape	Param #
conv1d_33 (Conv1D)	(None, 124, 16)	736
conv1d_34 (Conv1D)	(None, 122, 16)	784
dropout_17 (Dropout)	(None, 122, 16)	0
max_pooling1d_17 (MaxPooling)	(None, 61, 16)	0
flatten_17 (Flatten)	(None, 976)	0
dense_33 (Dense)	(None, 64)	62528
dense_34 (Dense)	(None, 3)	195

Total params: 64,243  
Trainable params: 64,243  
Non-trainable params: 0

---

Train on 4067 samples, validate on 1560 samples

Epoch 1/25  
- 2s - loss: 0.8681 - acc: 0.8249 - val\_loss: 0.2865 - val\_acc: 0.8827

Epoch 2/25  
- 0s - loss: 0.3156 - acc: 0.9021 - val\_loss: 0.3418 - val\_acc: 0.8724

Epoch 3/25  
- 0s - loss: 0.2364 - acc: 0.9061 - val\_loss: 0.3385 - val\_acc: 0.8987

Epoch 4/25  
- 0s - loss: 0.2600 - acc: 0.9171 - val\_loss: 0.2631 - val\_acc: 0.9058

Epoch 5/25  
- 0s - loss: 0.2240 - acc: 0.9265 - val\_loss: 0.4877 - val\_acc: 0.8545

Epoch 6/25  
- 1s - loss: 0.2281 - acc: 0.9260 - val\_loss: 0.2962 - val\_acc: 0.8981

Epoch 7/25  
- 0s - loss: 0.2503 - acc: 0.9341 - val\_loss: 0.3255 - val\_acc: 0.8859

Epoch 8/25  
- 0s - loss: 0.2109 - acc: 0.9316 - val\_loss: 0.2342 - val\_acc: 0.8788

Epoch 9/25  
- 1s - loss: 0.1908 - acc: 0.9339 - val\_loss: 0.2668 - val\_acc: 0.9295

Epoch 10/25  
- 0s - loss: 0.3937 - acc: 0.9223 - val\_loss: 0.3222 - val\_acc: 0.9032

Epoch 11/25  
- 0s - loss: 0.1925 - acc: 0.9363 - val\_loss: 0.2455 - val\_acc: 0.9314

Epoch 12/25  
- 0s - loss: 0.2660 - acc: 0.9319 - val\_loss: 0.3303 - val\_acc: 0.8949

Epoch 13/25  
- 0s - loss: 0.2553 - acc: 0.9339 - val\_loss: 0.3538 - val\_acc: 0.9250

Epoch 14/25  
- 0s - loss: 0.2701 - acc: 0.9319 - val\_loss: 0.4187 - val\_acc: 0.9051

Epoch 15/25  
- 0s - loss: 0.2278 - acc: 0.9398 - val\_loss: 0.5426 - val\_acc: 0.8615

Epoch 16/25  
- 0s - loss: 1.0267 - acc: 0.8940 - val\_loss: 0.4475 - val\_acc: 0.9000

Epoch 17/25  
- 0s - loss: 0.3353 - acc: 0.9385 - val\_loss: 0.3960 - val\_acc: 0.9263

Epoch 18/25  
- 0s - loss: 0.2310 - acc: 0.9430 - val\_loss: 0.3627 - val\_acc: 0.9353

Epoch 19/25  
- 0s - loss: 0.2231 - acc: 0.9430 - val\_loss: 0.3466 - val\_acc: 0.9321

```

Epoch 20/25
- 0s - loss: 0.2649 - acc: 0.9388 - val_loss: 0.3703 - val_acc: 0.9314

Epoch 21/25
- 0s - loss: 0.5889 - acc: 0.9238 - val_loss: 0.2655 - val_acc: 0.9346

Epoch 22/25
- 0s - loss: 0.2054 - acc: 0.9410 - val_loss: 0.4794 - val_acc: 0.7724

Epoch 23/25
- 0s - loss: 0.2683 - acc: 0.9432 - val_loss: 0.3402 - val_acc: 0.9205

Epoch 24/25
- 1s - loss: 0.2404 - acc: 0.9427 - val_loss: 0.3415 - val_acc: 0.9468

Epoch 25/25
- 0s - loss: 0.2400 - acc: 0.9430 - val_loss: 0.2673 - val_acc: 0.9353

Test accuracy:
0.9352564102564103
Model: "sequential_18"

Layer (type)          Output Shape         Param #
=====
conv1d_35 (Conv1D)    (None, 126, 64)      1792
conv1d_36 (Conv1D)    (None, 124, 64)       12352
dropout_18 (Dropout)  (None, 124, 64)       0
max_pooling1d_18 (MaxPooling) (None, 41, 64)  0
flatten_18 (Flatten)  (None, 2624)        0
dense_35 (Dense)     (None, 64)           168000
dense_36 (Dense)     (None, 3)            195
=====
Total params: 182,339
Trainable params: 182,339
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/30
- 3s - loss: 0.7499 - acc: 0.8387 - val_loss: 0.3601 - val_acc: 0.8327

Epoch 2/30
- 1s - loss: 0.3368 - acc: 0.9036 - val_loss: 0.2887 - val_acc: 0.8987

Epoch 3/30
- 1s - loss: 0.2839 - acc: 0.9144 - val_loss: 0.3267 - val_acc: 0.9077

Epoch 4/30
- 1s - loss: 0.2467 - acc: 0.9154 - val_loss: 0.3330 - val_acc: 0.8660

Epoch 5/30
- 1s - loss: 0.2190 - acc: 0.9243 - val_loss: 0.2761 - val_acc: 0.9160

Epoch 6/30
- 1s - loss: 0.2252 - acc: 0.9319 - val_loss: 0.2478 - val_acc: 0.9256

Epoch 7/30
- 1s - loss: 0.1878 - acc: 0.9383 - val_loss: 0.2555 - val_acc: 0.9218

Epoch 8/30
- 1s - loss: 0.1920 - acc: 0.9400 - val_loss: 0.2465 - val_acc: 0.9308

Epoch 9/30
- 1s - loss: 0.1734 - acc: 0.9388 - val_loss: 0.2300 - val_acc: 0.9308

Epoch 10/30
- 1s - loss: 0.1753 - acc: 0.9425 - val_loss: 0.2409 - val_acc: 0.9173

Epoch 11/30
- 1s - loss: 0.1532 - acc: 0.9501 - val_loss: 0.2265 - val_acc: 0.9391

```

```

Epoch 12/30
- 1s - loss: 0.1879 - acc: 0.9525 - val_loss: 0.2889 - val_acc: 0.9244

Epoch 13/30
- 1s - loss: 0.1370 - acc: 0.9540 - val_loss: 0.2231 - val_acc: 0.9455

Epoch 14/30
- 1s - loss: 0.1354 - acc: 0.9582 - val_loss: 0.2461 - val_acc: 0.9385

Epoch 15/30
- 1s - loss: 0.1442 - acc: 0.9614 - val_loss: 0.2332 - val_acc: 0.9276

Epoch 16/30
- 1s - loss: 0.1318 - acc: 0.9626 - val_loss: 0.2690 - val_acc: 0.9231

Epoch 17/30
- 1s - loss: 0.1440 - acc: 0.9594 - val_loss: 0.3948 - val_acc: 0.8878

Epoch 18/30
- 1s - loss: 0.1259 - acc: 0.9641 - val_loss: 0.2648 - val_acc: 0.9071

Epoch 19/30
- 1s - loss: 0.0964 - acc: 0.9680 - val_loss: 0.2409 - val_acc: 0.9365

Epoch 20/30
- 1s - loss: 0.1213 - acc: 0.9643 - val_loss: 0.3157 - val_acc: 0.9109

Epoch 21/30
- 1s - loss: 0.1128 - acc: 0.9717 - val_loss: 0.2839 - val_acc: 0.9372

Epoch 22/30
- 1s - loss: 0.1109 - acc: 0.9702 - val_loss: 0.2777 - val_acc: 0.9199

Epoch 23/30
- 1s - loss: 0.1206 - acc: 0.9678 - val_loss: 0.2433 - val_acc: 0.9295

Epoch 24/30
- 1s - loss: 0.1145 - acc: 0.9693 - val_loss: 0.2129 - val_acc: 0.9397

Epoch 25/30
- 1s - loss: 0.1292 - acc: 0.9690 - val_loss: 0.2069 - val_acc: 0.9474

Epoch 26/30
- 1s - loss: 0.1030 - acc: 0.9717 - val_loss: 0.1999 - val_acc: 0.9506

Epoch 27/30
- 1s - loss: 0.1099 - acc: 0.9695 - val_loss: 0.1884 - val_acc: 0.9532

Epoch 28/30
- 1s - loss: 0.1542 - acc: 0.9671 - val_loss: 0.2764 - val_acc: 0.9378

Epoch 29/30
- 1s - loss: 0.0973 - acc: 0.9737 - val_loss: 0.2323 - val_acc: 0.9321

Epoch 30/30
- 1s - loss: 0.0845 - acc: 0.9744 - val_loss: 0.2920 - val_acc: 0.9237

Test accuracy:
0.9237179487179488
Model: "sequential_19"


```

Layer (type)	Output Shape	Param #
conv1d_37 (Conv1D)	(None, 127, 32)	608
conv1d_38 (Conv1D)	(None, 125, 64)	6208
dropout_19 (Dropout)	(None, 125, 64)	0
max_pooling1d_19 (MaxPooling)	(None, 41, 64)	0
flatten_19 (Flatten)	(None, 2624)	0
dense_37 (Dense)	(None, 16)	42000
dense_38 (Dense)	(None, 3)	51

```
Total params: 48,867
Trainable params: 48,867
Non-trainable params: 0
```

```
Train on 4067 samples, validate on 1560 samples
Epoch 1/35
- 3s - loss: 0.6909 - acc: 0.8689 - val_loss: 0.4313 - val_acc: 0.8833

Epoch 2/35
- 1s - loss: 0.3100 - acc: 0.9203 - val_loss: 0.5559 - val_acc: 0.8833

Epoch 3/35
- 1s - loss: 0.3207 - acc: 0.9307 - val_loss: 0.3585 - val_acc: 0.9058

Epoch 4/35
- 1s - loss: 0.3620 - acc: 0.9218 - val_loss: 0.3077 - val_acc: 0.9269

Epoch 5/35
- 1s - loss: 0.3559 - acc: 0.9390 - val_loss: 0.3177 - val_acc: 0.9276

Epoch 6/35
- 1s - loss: 0.3174 - acc: 0.9363 - val_loss: 0.3938 - val_acc: 0.8833

Epoch 7/35
- 1s - loss: 0.2585 - acc: 0.9422 - val_loss: 0.3882 - val_acc: 0.9436

Epoch 8/35
- 1s - loss: 0.2530 - acc: 0.9471 - val_loss: 0.4421 - val_acc: 0.9340

Epoch 9/35
- 1s - loss: 0.1998 - acc: 0.9572 - val_loss: 0.4045 - val_acc: 0.9385

Epoch 10/35
- 1s - loss: 0.2886 - acc: 0.9506 - val_loss: 0.4843 - val_acc: 0.9199

Epoch 11/35
- 1s - loss: 0.2869 - acc: 0.9493 - val_loss: 0.5152 - val_acc: 0.9205

Epoch 12/35
- 1s - loss: 0.3635 - acc: 0.9481 - val_loss: 0.6224 - val_acc: 0.9205

Epoch 13/35
- 1s - loss: 0.3483 - acc: 0.9496 - val_loss: 0.7714 - val_acc: 0.9167

Epoch 14/35
- 1s - loss: 0.2744 - acc: 0.9604 - val_loss: 0.6152 - val_acc: 0.9115

Epoch 15/35
- 1s - loss: 0.3465 - acc: 0.9486 - val_loss: 1.6327 - val_acc: 0.8513

Epoch 16/35
- 1s - loss: 0.2887 - acc: 0.9575 - val_loss: 0.4198 - val_acc: 0.9359

Epoch 17/35
- 1s - loss: 0.2234 - acc: 0.9614 - val_loss: 0.5247 - val_acc: 0.9244

Epoch 18/35
- 1s - loss: 0.2430 - acc: 0.9582 - val_loss: 0.6356 - val_acc: 0.9353

Epoch 19/35
- 1s - loss: 0.2391 - acc: 0.9661 - val_loss: 0.5424 - val_acc: 0.9494

Epoch 20/35
- 1s - loss: 0.2221 - acc: 0.9653 - val_loss: 0.7560 - val_acc: 0.9269

Epoch 21/35
- 1s - loss: 0.2876 - acc: 0.9580 - val_loss: 0.9069 - val_acc: 0.9045

Epoch 22/35
- 1s - loss: 0.3630 - acc: 0.9533 - val_loss: 0.8221 - val_acc: 0.9224

Epoch 23/35
- 1s - loss: 0.3697 - acc: 0.9639 - val_loss: 0.8206 - val_acc: 0.9071

Epoch 24/35
- 1s - loss: 0.5582 - acc: 0.9343 - val_loss: 0.7031 - val_acc: 0.9173
```

```
Epoch 25/35
- 1s - loss: 0.3064 - acc: 0.9577 - val_loss: 0.9089 - val_acc: 0.9256

Epoch 26/35
- 1s - loss: 0.3556 - acc: 0.9506 - val_loss: 1.0713 - val_acc: 0.9109

Epoch 27/35
- 1s - loss: 0.5380 - acc: 0.9447 - val_loss: 0.9607 - val_acc: 0.9115

Epoch 28/35
- 1s - loss: 0.3308 - acc: 0.9565 - val_loss: 0.4846 - val_acc: 0.9321

Epoch 29/35
- 1s - loss: 0.2813 - acc: 0.9533 - val_loss: 0.5264 - val_acc: 0.9346

Epoch 30/35
- 1s - loss: 0.4183 - acc: 0.9491 - val_loss: 0.6432 - val_acc: 0.9378

Epoch 31/35
- 1s - loss: 0.2542 - acc: 0.9552 - val_loss: 0.9930 - val_acc: 0.9141

Epoch 32/35
- 1s - loss: 0.4148 - acc: 0.9557 - val_loss: 0.6038 - val_acc: 0.9365

Epoch 33/35
- 1s - loss: 0.3859 - acc: 0.9540 - val_loss: 0.9054 - val_acc: 0.8878

Epoch 34/35
- 1s - loss: 0.3130 - acc: 0.9557 - val_loss: 0.7941 - val_acc: 0.9141

Epoch 35/35
- 1s - loss: 0.2514 - acc: 0.9663 - val_loss: 1.0381 - val_acc: 0.8654
```

Test accuracy:  
0.8653846153846154  
Model: "sequential\_20"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_39 (Conv1D)	(None, 126, 16)	448
conv1d_40 (Conv1D)	(None, 122, 32)	2592
dropout_20 (Dropout)	(None, 122, 32)	0
max_pooling1d_20 (MaxPooling)	(None, 40, 32)	0
flatten_20 (Flatten)	(None, 1280)	0
dense_39 (Dense)	(None, 16)	20496
dense_40 (Dense)	(None, 3)	51
<hr/>		

Total params: 23,587  
Trainable params: 23,587  
Non-trainable params: 0

---

Train on 4067 samples, validate on 1560 samples  
Epoch 1/25  
- 3s - loss: 10.5112 - acc: 0.3460 - val\_loss: 10.5438 - val\_acc: 0.3442

Epoch 2/25  
- 2s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5438 - val\_acc: 0.3442

Epoch 3/25  
- 2s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5438 - val\_acc: 0.3442

Epoch 4/25  
- 2s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5438 - val\_acc: 0.3442

Epoch 5/25  
- 2s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5438 - val\_acc: 0.3442

Epoch 6/25  
- 2s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5438 - val\_acc: 0.3442

Epoch 7/25

```
Epoch 7/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 8/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 9/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 10/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 11/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 12/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 13/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 14/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 15/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 16/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 17/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 18/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 19/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 20/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 21/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 22/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 23/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 24/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Epoch 25/25
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5438 - val_acc: 0.3442
```

```
Test accuracy:
```

```
0.34423076923076923
```

```
Model: "sequential_21"
```

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_41 (Conv1D)	(None, 126, 64)	1792
conv1d_42 (Conv1D)	(None, 123, 16)	4112
dropout_21 (Dropout)	(None, 123, 16)	0
max_pooling1d_21 (MaxPooling)	(None, 61, 16)	0
flatten_21 (Flatten)	(None, 976)	0
dense_41 (Dense)	(None, 64)	62528
dense_42 (Dense)	(None, 3)	195
<hr/>		
Total params.	68 627	

```
local params: 00,02,
Trainable params: 68,627
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/25
- 4s - loss: 0.3260 - acc: 0.8800 - val_loss: 0.3642 - val_acc: 0.8840

Epoch 2/25
- 2s - loss: 0.1866 - acc: 0.9326 - val_loss: 0.2887 - val_acc: 0.9077

Epoch 3/25
- 2s - loss: 0.1594 - acc: 0.9464 - val_loss: 0.2083 - val_acc: 0.9327

Epoch 4/25
- 2s - loss: 0.1503 - acc: 0.9506 - val_loss: 0.4979 - val_acc: 0.9090

Epoch 5/25
- 2s - loss: 0.1372 - acc: 0.9565 - val_loss: 0.3171 - val_acc: 0.9263

Epoch 6/25
- 2s - loss: 0.1240 - acc: 0.9616 - val_loss: 0.2528 - val_acc: 0.9500

Epoch 7/25
- 3s - loss: 0.1424 - acc: 0.9621 - val_loss: 0.2841 - val_acc: 0.9404

Epoch 8/25
- 2s - loss: 0.1233 - acc: 0.9651 - val_loss: 0.4191 - val_acc: 0.9167

Epoch 9/25
- 2s - loss: 0.1133 - acc: 0.9730 - val_loss: 0.3991 - val_acc: 0.9295

Epoch 10/25
- 2s - loss: 0.1172 - acc: 0.9715 - val_loss: 0.3981 - val_acc: 0.9410

Epoch 11/25
- 2s - loss: 0.0833 - acc: 0.9744 - val_loss: 0.6404 - val_acc: 0.9045

Epoch 12/25
- 2s - loss: 0.0775 - acc: 0.9803 - val_loss: 0.3865 - val_acc: 0.9385

Epoch 13/25
- 2s - loss: 0.0913 - acc: 0.9798 - val_loss: 0.3902 - val_acc: 0.9410

Epoch 14/25
- 2s - loss: 0.0589 - acc: 0.9835 - val_loss: 0.4297 - val_acc: 0.9385

Epoch 15/25
- 2s - loss: 0.0861 - acc: 0.9825 - val_loss: 0.4652 - val_acc: 0.9295

Epoch 16/25
- 2s - loss: 0.0975 - acc: 0.9833 - val_loss: 0.4550 - val_acc: 0.9359

Epoch 17/25
- 2s - loss: 0.0597 - acc: 0.9867 - val_loss: 0.4259 - val_acc: 0.9410

Epoch 18/25
- 2s - loss: 0.0850 - acc: 0.9852 - val_loss: 0.4188 - val_acc: 0.9494

Epoch 19/25
- 2s - loss: 0.0593 - acc: 0.9877 - val_loss: 0.4822 - val_acc: 0.9372

Epoch 20/25
- 2s - loss: 0.0787 - acc: 0.9852 - val_loss: 0.5031 - val_acc: 0.9378

Epoch 21/25
- 2s - loss: 0.0546 - acc: 0.9870 - val_loss: 0.5342 - val_acc: 0.9353

Epoch 22/25
- 2s - loss: 0.0745 - acc: 0.9882 - val_loss: 0.5763 - val_acc: 0.9321

Epoch 23/25
- 2s - loss: 0.0685 - acc: 0.9884 - val_loss: 0.4014 - val_acc: 0.9481

Epoch 24/25
- 2s - loss: 0.0500 - acc: 0.9894 - val_loss: 0.4903 - val_acc: 0.9372

Epoch 25/25
```

Epoch 25/25

- 2s - loss: 0.0606 - acc: 0.9911 - val\_loss: 0.5388 - val\_acc: 0.9417

Test accuracy:

0.9416666666666667

Model: "sequential\_22"

Layer (type)	Output Shape	Param #
conv1d_43 (Conv1D)	(None, 125, 64)	2368
conv1d_44 (Conv1D)	(None, 123, 64)	12352
dropout_22 (Dropout)	(None, 123, 64)	0
max_pooling1d_22 (MaxPooling)	(None, 61, 64)	0
flatten_22 (Flatten)	(None, 3904)	0
dense_43 (Dense)	(None, 32)	124960
dense_44 (Dense)	(None, 3)	99

Total params: 139,779

Trainable params: 139,779

Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples

Epoch 1/35

- 3s - loss: 0.3787 - acc: 0.8670 - val\_loss: 0.2857 - val\_acc: 0.8801

Epoch 2/35

- 1s - loss: 0.1990 - acc: 0.9250 - val\_loss: 0.5096 - val\_acc: 0.8821

Epoch 3/35

- 1s - loss: 0.1703 - acc: 0.9405 - val\_loss: 0.5228 - val\_acc: 0.8718

Epoch 4/35

- 1s - loss: 0.1231 - acc: 0.9552 - val\_loss: 0.2875 - val\_acc: 0.9224

Epoch 5/35

- 1s - loss: 0.1087 - acc: 0.9602 - val\_loss: 0.3224 - val\_acc: 0.9218

Epoch 6/35

- 1s - loss: 0.0817 - acc: 0.9700 - val\_loss: 0.2588 - val\_acc: 0.9205

Epoch 7/35

- 1s - loss: 0.0714 - acc: 0.9776 - val\_loss: 0.4053 - val\_acc: 0.9179

Epoch 8/35

- 1s - loss: 0.0684 - acc: 0.9761 - val\_loss: 0.2929 - val\_acc: 0.9372

Epoch 9/35

- 1s - loss: 0.0520 - acc: 0.9840 - val\_loss: 0.6445 - val\_acc: 0.9179

Epoch 10/35

- 1s - loss: 0.0595 - acc: 0.9838 - val\_loss: 0.3168 - val\_acc: 0.9205

Epoch 11/35

- 1s - loss: 0.0496 - acc: 0.9855 - val\_loss: 0.3443 - val\_acc: 0.9423

Epoch 12/35

- 1s - loss: 0.0399 - acc: 0.9884 - val\_loss: 0.6546 - val\_acc: 0.9218

Epoch 13/35

- 1s - loss: 0.0486 - acc: 0.9894 - val\_loss: 0.4271 - val\_acc: 0.9397

Epoch 14/35

- 1s - loss: 0.0308 - acc: 0.9924 - val\_loss: 0.4446 - val\_acc: 0.9327

Epoch 15/35

- 1s - loss: 0.0397 - acc: 0.9892 - val\_loss: 0.4393 - val\_acc: 0.9359

Epoch 16/35

- 1s - loss: 0.0418 - acc: 0.9911 - val\_loss: 0.5752 - val\_acc: 0.9218

Epoch 17/35

1s 1000. 0.0262 200. 0.0024 1000. 0.5720 1000. 0.0246

```

- 1s - loss: 0.0302 - acc: 0.9924 - val_loss: 0.5129 - val_acc: 0.9340
Epoch 18/35
- 1s - loss: 0.0319 - acc: 0.9936 - val_loss: 0.6289 - val_acc: 0.9269
Epoch 19/35
- 1s - loss: 0.0471 - acc: 0.9909 - val_loss: 0.4968 - val_acc: 0.9321
Epoch 20/35
- 1s - loss: 0.0265 - acc: 0.9953 - val_loss: 0.5140 - val_acc: 0.9282
Epoch 21/35
- 1s - loss: 0.0254 - acc: 0.9956 - val_loss: 0.5285 - val_acc: 0.9346
Epoch 22/35
- 1s - loss: 0.0161 - acc: 0.9963 - val_loss: 0.5530 - val_acc: 0.9365
Epoch 23/35
- 1s - loss: 0.0274 - acc: 0.9948 - val_loss: 0.4125 - val_acc: 0.9423
Epoch 24/35
- 1s - loss: 0.0369 - acc: 0.9961 - val_loss: 0.5018 - val_acc: 0.9410
Epoch 25/35
- 1s - loss: 0.0317 - acc: 0.9953 - val_loss: 0.4727 - val_acc: 0.9442
Epoch 26/35
- 1s - loss: 0.0210 - acc: 0.9963 - val_loss: 0.4768 - val_acc: 0.9417
Epoch 27/35
- 1s - loss: 0.0282 - acc: 0.9956 - val_loss: 0.5765 - val_acc: 0.9346
Epoch 28/35
- 1s - loss: 0.0360 - acc: 0.9953 - val_loss: 0.5310 - val_acc: 0.9340
Epoch 29/35
- 1s - loss: 0.0253 - acc: 0.9963 - val_loss: 0.4307 - val_acc: 0.9333
Epoch 30/35
- 1s - loss: 0.0228 - acc: 0.9958 - val_loss: 0.5407 - val_acc: 0.9423
Epoch 31/35
- 1s - loss: 0.0339 - acc: 0.9943 - val_loss: 0.5749 - val_acc: 0.9391
Epoch 32/35
- 1s - loss: 0.0242 - acc: 0.9963 - val_loss: 0.6696 - val_acc: 0.9224
Epoch 33/35
- 1s - loss: 0.0394 - acc: 0.9948 - val_loss: 0.5501 - val_acc: 0.9378
Epoch 34/35
- 1s - loss: 0.0143 - acc: 0.9978 - val_loss: 0.4966 - val_acc: 0.9391
Epoch 35/35
- 1s - loss: 0.0346 - acc: 0.9951 - val_loss: 0.5308 - val_acc: 0.9314

Test accuracy:
0.9314102564102564
Model: "sequential_23"

Layer (type)          Output Shape         Param #
=================================================================
conv1d_45 (Conv1D)    (None, 126, 16)      448
conv1d_46 (Conv1D)    (None, 125, 32)      1056
dropout_23 (Dropout) (None, 125, 32)      0
max_pooling1d_23 (MaxPooling) (None, 41, 32) 0
flatten_23 (Flatten)  (None, 1312)        0
dense_45 (Dense)     (None, 32)           42016
dense_46 (Dense)     (None, 3)            99
=====
Total params: 43,619

```

```
trainable params: 43,619
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/35
- 3s - loss: 1.3172 - acc: 0.6971 - val_loss: 0.5252 - val_acc: 0.8372

Epoch 2/35
- 1s - loss: 0.6986 - acc: 0.7947 - val_loss: 0.4455 - val_acc: 0.8679

Epoch 3/35
- 1s - loss: 0.4845 - acc: 0.8453 - val_loss: 0.4226 - val_acc: 0.8737

Epoch 4/35
- 1s - loss: 0.3425 - acc: 0.8756 - val_loss: 0.3969 - val_acc: 0.8833

Epoch 5/35
- 1s - loss: 0.3139 - acc: 0.8773 - val_loss: 0.3975 - val_acc: 0.8891

Epoch 6/35
- 1s - loss: 0.3024 - acc: 0.8866 - val_loss: 0.3528 - val_acc: 0.8853

Epoch 7/35
- 1s - loss: 0.2673 - acc: 0.8970 - val_loss: 0.3511 - val_acc: 0.9019

Epoch 8/35
- 1s - loss: 0.2555 - acc: 0.8997 - val_loss: 0.3398 - val_acc: 0.9038

Epoch 9/35
- 1s - loss: 0.2487 - acc: 0.9004 - val_loss: 0.3318 - val_acc: 0.9032

Epoch 10/35
- 1s - loss: 0.2295 - acc: 0.9058 - val_loss: 0.3239 - val_acc: 0.9051

Epoch 11/35
- 1s - loss: 0.2128 - acc: 0.9115 - val_loss: 0.3201 - val_acc: 0.9051

Epoch 12/35
- 1s - loss: 0.2159 - acc: 0.9120 - val_loss: 0.3112 - val_acc: 0.8923

Epoch 13/35
- 1s - loss: 0.1977 - acc: 0.9142 - val_loss: 0.2985 - val_acc: 0.9013

Epoch 14/35
- 1s - loss: 0.1982 - acc: 0.9216 - val_loss: 0.2930 - val_acc: 0.9090

Epoch 15/35
- 1s - loss: 0.1997 - acc: 0.9181 - val_loss: 0.2882 - val_acc: 0.9071

Epoch 16/35
- 1s - loss: 0.1950 - acc: 0.9206 - val_loss: 0.2791 - val_acc: 0.9006

Epoch 17/35
- 1s - loss: 0.1926 - acc: 0.9245 - val_loss: 0.2898 - val_acc: 0.9051

Epoch 18/35
- 1s - loss: 0.1838 - acc: 0.9245 - val_loss: 0.2777 - val_acc: 0.9096

Epoch 19/35
- 1s - loss: 0.1837 - acc: 0.9225 - val_loss: 0.2676 - val_acc: 0.9071

Epoch 20/35
- 1s - loss: 0.1942 - acc: 0.9216 - val_loss: 0.2732 - val_acc: 0.9077

Epoch 21/35
- 1s - loss: 0.1818 - acc: 0.9270 - val_loss: 0.2602 - val_acc: 0.9045

Epoch 22/35
- 1s - loss: 0.1779 - acc: 0.9280 - val_loss: 0.2548 - val_acc: 0.9000

Epoch 23/35
- 1s - loss: 0.1826 - acc: 0.9243 - val_loss: 0.2549 - val_acc: 0.9032

Epoch 24/35
- 1s - loss: 0.1751 - acc: 0.9299 - val_loss: 0.2539 - val_acc: 0.9032

Epoch 25/35
- 1s - loss: 0.1699 - acc: 0.9307 - val_loss: 0.2550 - val_acc: 0.9041
```

```

- 1s - loss: 0.1688 - acc: 0.9291 - val_loss: 0.2652 - val_acc: 0.9141

Epoch 26/35
- 1s - loss: 0.1776 - acc: 0.9289 - val_loss: 0.2529 - val_acc: 0.9051

Epoch 27/35
- 1s - loss: 0.1734 - acc: 0.9302 - val_loss: 0.2460 - val_acc: 0.9026

Epoch 28/35
- 1s - loss: 0.1728 - acc: 0.9304 - val_loss: 0.2525 - val_acc: 0.9083

Epoch 29/35
- 1s - loss: 0.1764 - acc: 0.9307 - val_loss: 0.2525 - val_acc: 0.9154

Epoch 30/35
- 1s - loss: 0.1676 - acc: 0.9346 - val_loss: 0.2497 - val_acc: 0.9064

Epoch 31/35
- 1s - loss: 0.1677 - acc: 0.9329 - val_loss: 0.2439 - val_acc: 0.9109

Epoch 32/35
- 1s - loss: 0.1785 - acc: 0.9339 - val_loss: 0.2484 - val_acc: 0.9058

Epoch 33/35
- 1s - loss: 0.1563 - acc: 0.9356 - val_loss: 0.2458 - val_acc: 0.9167

Epoch 34/35
- 1s - loss: 0.1570 - acc: 0.9371 - val_loss: 0.2377 - val_acc: 0.9135

Epoch 35/35
- 1s - loss: 0.1582 - acc: 0.9361 - val_loss: 0.2392 - val_acc: 0.9109

Test accuracy:
0.9108974358974359
Model: "sequential_24"

Layer (type)          Output Shape         Param #
=====
conv1d_47 (Conv1D)    (None, 127, 32)      608
conv1d_48 (Conv1D)    (None, 123, 64)       10304
dropout_24 (Dropout)  (None, 123, 64)       0
max_pooling1d_24 (MaxPooling) (None, 24, 64)  0
flatten_24 (Flatten)  (None, 1536)          0
dense_47 (Dense)     (None, 16)            24592
dense_48 (Dense)     (None, 3)              51
=====
Total params: 35,555
Trainable params: 35,555
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/25
- 4s - loss: 0.3342 - acc: 0.8778 - val_loss: 0.3835 - val_acc: 0.8692

Epoch 2/25
- 2s - loss: 0.1766 - acc: 0.9294 - val_loss: 0.2237 - val_acc: 0.9103

Epoch 3/25
- 2s - loss: 0.1462 - acc: 0.9410 - val_loss: 0.2384 - val_acc: 0.9205

Epoch 4/25
- 2s - loss: 0.1159 - acc: 0.9550 - val_loss: 0.1654 - val_acc: 0.9410

Epoch 5/25
- 2s - loss: 0.1074 - acc: 0.9575 - val_loss: 0.1853 - val_acc: 0.9385

Epoch 6/25
- 2s - loss: 0.0889 - acc: 0.9658 - val_loss: 0.1463 - val_acc: 0.9538

Epoch 7/25
- 2s - loss: 0.0823 - acc: 0.9695 - val_loss: 0.1810 - val_acc: 0.9429

```

```

Epoch 8/25
- 2s - loss: 0.0718 - acc: 0.9730 - val_loss: 0.2224 - val_acc: 0.9218

Epoch 9/25
- 2s - loss: 0.0603 - acc: 0.9774 - val_loss: 0.2303 - val_acc: 0.9314

Epoch 10/25
- 2s - loss: 0.0506 - acc: 0.9823 - val_loss: 0.1919 - val_acc: 0.9468

Epoch 11/25
- 2s - loss: 0.0518 - acc: 0.9803 - val_loss: 0.2587 - val_acc: 0.9237

Epoch 12/25
- 2s - loss: 0.0778 - acc: 0.9752 - val_loss: 0.2282 - val_acc: 0.9423

Epoch 13/25
- 2s - loss: 0.0568 - acc: 0.9808 - val_loss: 0.1499 - val_acc: 0.9571

Epoch 14/25
- 2s - loss: 0.0432 - acc: 0.9848 - val_loss: 0.2321 - val_acc: 0.9474

Epoch 15/25
- 2s - loss: 0.0472 - acc: 0.9852 - val_loss: 0.2555 - val_acc: 0.9481

Epoch 16/25
- 2s - loss: 0.0242 - acc: 0.9924 - val_loss: 0.2699 - val_acc: 0.9468

Epoch 17/25
- 2s - loss: 0.0391 - acc: 0.9892 - val_loss: 0.2066 - val_acc: 0.9519

Epoch 18/25
- 2s - loss: 0.0270 - acc: 0.9894 - val_loss: 0.2707 - val_acc: 0.9410

Epoch 19/25
- 2s - loss: 0.0270 - acc: 0.9907 - val_loss: 0.2203 - val_acc: 0.9429

Epoch 20/25
- 2s - loss: 0.0273 - acc: 0.9929 - val_loss: 0.2874 - val_acc: 0.9423

Epoch 21/25
- 2s - loss: 0.0146 - acc: 0.9963 - val_loss: 0.2335 - val_acc: 0.9532

Epoch 22/25
- 2s - loss: 0.0104 - acc: 0.9970 - val_loss: 0.2581 - val_acc: 0.9487

Epoch 23/25
- 2s - loss: 0.0049 - acc: 0.9995 - val_loss: 0.2462 - val_acc: 0.9558

Epoch 24/25
- 2s - loss: 0.0037 - acc: 0.9995 - val_loss: 0.3143 - val_acc: 0.9494

Epoch 25/25
- 2s - loss: 0.0029 - acc: 1.0000 - val_loss: 0.3054 - val_acc: 0.9506

Test accuracy:
0.9506410256410256
Model: "sequential_25"


```

Layer (type)	Output Shape	Param #
conv1d_49 (Conv1D)	(None, 127, 32)	608
conv1d_50 (Conv1D)	(None, 123, 64)	10304
dropout_25 (Dropout)	(None, 123, 64)	0
max_pooling1d_25 (MaxPooling)	(None, 24, 64)	0
flatten_25 (Flatten)	(None, 1536)	0
dense_49 (Dense)	(None, 16)	24592
dense_50 (Dense)	(None, 3)	51

```

Total params: 35,555
Trainable params: 35,555

```

Non-trainable params: 0

---

Train on 4067 samples, validate on 1560 samples

Epoch 1/25

- 4s - loss: 0.3193 - acc: 0.8729 - val\_loss: 0.2478 - val\_acc: 0.8878

Epoch 2/25

- 2s - loss: 0.1762 - acc: 0.9238 - val\_loss: 0.2238 - val\_acc: 0.9058

Epoch 3/25

- 2s - loss: 0.1511 - acc: 0.9390 - val\_loss: 0.2149 - val\_acc: 0.9199

Epoch 4/25

- 2s - loss: 0.1321 - acc: 0.9444 - val\_loss: 0.1871 - val\_acc: 0.9244

Epoch 5/25

- 2s - loss: 0.1251 - acc: 0.9528 - val\_loss: 0.1751 - val\_acc: 0.9256

Epoch 6/25

- 2s - loss: 0.1054 - acc: 0.9582 - val\_loss: 0.1757 - val\_acc: 0.9314

Epoch 7/25

- 2s - loss: 0.1369 - acc: 0.9506 - val\_loss: 0.2091 - val\_acc: 0.9205

Epoch 8/25

- 2s - loss: 0.0879 - acc: 0.9671 - val\_loss: 0.1849 - val\_acc: 0.9288

Epoch 9/25

- 2s - loss: 0.0716 - acc: 0.9717 - val\_loss: 0.1668 - val\_acc: 0.9359

Epoch 10/25

- 2s - loss: 0.0598 - acc: 0.9739 - val\_loss: 0.2024 - val\_acc: 0.9340

Epoch 11/25

- 2s - loss: 0.0580 - acc: 0.9779 - val\_loss: 0.2104 - val\_acc: 0.9288

Epoch 12/25

- 2s - loss: 0.0484 - acc: 0.9806 - val\_loss: 0.1661 - val\_acc: 0.9474

Epoch 13/25

- 2s - loss: 0.0341 - acc: 0.9880 - val\_loss: 0.1707 - val\_acc: 0.9500

Epoch 14/25

- 2s - loss: 0.0310 - acc: 0.9899 - val\_loss: 0.1785 - val\_acc: 0.9481

Epoch 15/25

- 2s - loss: 0.0231 - acc: 0.9921 - val\_loss: 0.2197 - val\_acc: 0.9359

Epoch 16/25

- 2s - loss: 0.0285 - acc: 0.9892 - val\_loss: 0.2113 - val\_acc: 0.9410

Epoch 17/25

- 2s - loss: 0.0878 - acc: 0.9781 - val\_loss: 0.3318 - val\_acc: 0.9282

Epoch 18/25

- 2s - loss: 0.0632 - acc: 0.9816 - val\_loss: 0.1830 - val\_acc: 0.9545

Epoch 19/25

- 2s - loss: 0.0370 - acc: 0.9902 - val\_loss: 0.2262 - val\_acc: 0.9391

Epoch 20/25

- 2s - loss: 0.0260 - acc: 0.9921 - val\_loss: 0.2711 - val\_acc: 0.9378

Epoch 21/25

- 2s - loss: 0.0245 - acc: 0.9931 - val\_loss: 0.2064 - val\_acc: 0.9410

Epoch 22/25

- 2s - loss: 0.0135 - acc: 0.9956 - val\_loss: 0.2107 - val\_acc: 0.9429

Epoch 23/25

- 2s - loss: 0.0140 - acc: 0.9948 - val\_loss: 0.2078 - val\_acc: 0.9410

Epoch 24/25

- 2s - loss: 0.0090 - acc: 0.9978 - val\_loss: 0.1984 - val\_acc: 0.9532

Epoch 25/25

- 2s - loss: 0.0085 - acc: 0.9975 - val\_loss: 0.2026 - val\_acc: 0.9487

Test accuracy:  
0.9487179487179487  
Model: "sequential\_26"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_51 (Conv1D)	(None, 127, 32)	608
conv1d_52 (Conv1D)	(None, 123, 64)	10304
dropout_26 (Dropout)	(None, 123, 64)	0
max_pooling1d_26 (MaxPooling)	(None, 24, 64)	0
flatten_26 (Flatten)	(None, 1536)	0
dense_51 (Dense)	(None, 16)	24592
dense_52 (Dense)	(None, 3)	51
<hr/>		
Total params:	35,555	
Trainable params:	35,555	
Non-trainable params:	0	

Train on 4067 samples, validate on 1560 samples

Epoch 1/25  
- 4s - loss: 0.4571 - acc: 0.8471 - val\_loss: 0.2939 - val\_acc: 0.8891

Epoch 2/25  
- 2s - loss: 0.2481 - acc: 0.9068 - val\_loss: 0.2923 - val\_acc: 0.9013

Epoch 3/25  
- 2s - loss: 0.2200 - acc: 0.9233 - val\_loss: 0.3363 - val\_acc: 0.8910

Epoch 4/25  
- 2s - loss: 0.2758 - acc: 0.9137 - val\_loss: 0.2433 - val\_acc: 0.9038

Epoch 5/25  
- 2s - loss: 0.1780 - acc: 0.9361 - val\_loss: 0.3010 - val\_acc: 0.8776

Epoch 6/25  
- 2s - loss: 0.3036 - acc: 0.9260 - val\_loss: 0.4007 - val\_acc: 0.8429

Epoch 7/25  
- 2s - loss: 0.3086 - acc: 0.9302 - val\_loss: 0.2564 - val\_acc: 0.9026

Epoch 8/25  
- 2s - loss: 0.1570 - acc: 0.9484 - val\_loss: 0.2751 - val\_acc: 0.8897

Epoch 9/25  
- 2s - loss: 2.6675 - acc: 0.7853 - val\_loss: 5.8084 - val\_acc: 0.5500

Epoch 10/25  
- 2s - loss: 5.6327 - acc: 0.5943 - val\_loss: 5.8479 - val\_acc: 0.5186

Epoch 11/25  
- 2s - loss: 2.1776 - acc: 0.6245 - val\_loss: 0.6697 - val\_acc: 0.6891

Epoch 12/25  
- 2s - loss: 0.8326 - acc: 0.7413 - val\_loss: 0.7085 - val\_acc: 0.7410

Epoch 13/25  
- 2s - loss: 0.6810 - acc: 0.7531 - val\_loss: 0.5685 - val\_acc: 0.7885

Epoch 14/25  
- 2s - loss: 0.6212 - acc: 0.8018 - val\_loss: 0.5583 - val\_acc: 0.7949

Epoch 15/25  
- 2s - loss: 0.4794 - acc: 0.8358 - val\_loss: 0.5088 - val\_acc: 0.8199

Epoch 16/25  
- 2s - loss: 0.4072 - acc: 0.8643 - val\_loss: 0.5312 - val\_acc: 0.8481

Epoch 17/25  
- 2s - loss: 0.3927 - acc: 0.8645 - val\_loss: 0.5722 - val\_acc: 0.8571

```

Epoch 18/25
- 2s - loss: 0.6810 - acc: 0.8035 - val_loss: 0.6896 - val_acc: 0.7929

Epoch 19/25
- 2s - loss: 0.5502 - acc: 0.8188 - val_loss: 0.7174 - val_acc: 0.8109

Epoch 20/25
- 2s - loss: 0.4700 - acc: 0.8377 - val_loss: 0.6818 - val_acc: 0.8269

Epoch 21/25
- 2s - loss: 0.4508 - acc: 0.8468 - val_loss: 0.7337 - val_acc: 0.8199

Epoch 22/25
- 2s - loss: 0.4395 - acc: 0.8515 - val_loss: 0.6438 - val_acc: 0.8327

Epoch 23/25
- 2s - loss: 0.6737 - acc: 0.8109 - val_loss: 0.7196 - val_acc: 0.8487

Epoch 24/25
- 2s - loss: 0.4964 - acc: 0.8768 - val_loss: 0.7297 - val_acc: 0.8551

Epoch 25/25
- 2s - loss: 0.4207 - acc: 0.8699 - val_loss: 0.4762 - val_acc: 0.8564

Test accuracy:
0.8564102564102564
Model: "sequential_27"

Layer (type)          Output Shape         Param #
=====
conv1d_53 (Conv1D)    (None, 127, 32)      608
conv1d_54 (Conv1D)    (None, 123, 64)       10304
dropout_27 (Dropout)  (None, 123, 64)       0
max_pooling1d_27 (MaxPooling) (None, 24, 64)  0
flatten_27 (Flatten)  (None, 1536)        0
dense_53 (Dense)     (None, 16)           24592
dense_54 (Dense)     (None, 3)            51
=====
Total params: 35,555
Trainable params: 35,555
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/25
- 5s - loss: 0.3500 - acc: 0.8633 - val_loss: 0.2951 - val_acc: 0.8891

Epoch 2/25
- 2s - loss: 0.1791 - acc: 0.9262 - val_loss: 0.2852 - val_acc: 0.8904

Epoch 3/25
- 2s - loss: 0.1745 - acc: 0.9356 - val_loss: 0.3100 - val_acc: 0.8782

Epoch 4/25
- 2s - loss: 0.1619 - acc: 0.9427 - val_loss: 0.2705 - val_acc: 0.9179

Epoch 5/25
- 2s - loss: 0.1301 - acc: 0.9530 - val_loss: 0.2027 - val_acc: 0.9295

Epoch 6/25
- 2s - loss: 0.1193 - acc: 0.9540 - val_loss: 0.2378 - val_acc: 0.9192

Epoch 7/25
- 2s - loss: 0.0962 - acc: 0.9631 - val_loss: 0.2410 - val_acc: 0.9244

Epoch 8/25
- 2s - loss: 0.0861 - acc: 0.9675 - val_loss: 0.2268 - val_acc: 0.9301

Epoch 9/25
- 2s - loss: 0.0813 - acc: 0.9715 - val_loss: 0.2474 - val_acc: 0.9301

Epoch 10/25

```

```

- 2s - loss: 0.0868 - acc: 0.9693 - val_loss: 0.2318 - val_acc: 0.9276

Epoch 11/25
- 2s - loss: 0.0667 - acc: 0.9747 - val_loss: 0.2477 - val_acc: 0.9231

Epoch 12/25
- 2s - loss: 0.0754 - acc: 0.9752 - val_loss: 0.1943 - val_acc: 0.9365

Epoch 13/25
- 2s - loss: 0.0466 - acc: 0.9813 - val_loss: 0.2061 - val_acc: 0.9474

Epoch 14/25
- 2s - loss: 0.0435 - acc: 0.9855 - val_loss: 0.3657 - val_acc: 0.9071

Epoch 15/25
- 2s - loss: 0.0419 - acc: 0.9884 - val_loss: 0.2353 - val_acc: 0.9423

Epoch 16/25
- 2s - loss: 0.0278 - acc: 0.9902 - val_loss: 0.2967 - val_acc: 0.9288

Epoch 17/25
- 2s - loss: 0.0277 - acc: 0.9897 - val_loss: 0.3225 - val_acc: 0.9192

Epoch 18/25
- 2s - loss: 0.0444 - acc: 0.9860 - val_loss: 0.2528 - val_acc: 0.9429

Epoch 19/25
- 2s - loss: 0.0320 - acc: 0.9897 - val_loss: 0.3312 - val_acc: 0.9186

Epoch 20/25
- 2s - loss: 0.0195 - acc: 0.9941 - val_loss: 0.2939 - val_acc: 0.9346

Epoch 21/25
- 2s - loss: 0.0332 - acc: 0.9884 - val_loss: 0.3596 - val_acc: 0.9199

Epoch 22/25
- 2s - loss: 0.0550 - acc: 0.9877 - val_loss: 0.2493 - val_acc: 0.9506

Epoch 23/25
- 2s - loss: 0.0150 - acc: 0.9939 - val_loss: 0.2820 - val_acc: 0.9397

Epoch 24/25
- 2s - loss: 0.0104 - acc: 0.9958 - val_loss: 0.2516 - val_acc: 0.9506

Epoch 25/25
- 2s - loss: 0.0071 - acc: 0.9983 - val_loss: 0.3362 - val_acc: 0.9263

Test accuracy:
0.9262820512820513
Model: "sequential_28"

Layer (type)          Output Shape         Param #
=====
conv1d_55 (Conv1D)    (None, 125, 32)      1184
=====
conv1d_56 (Conv1D)    (None, 121, 64)       10304
=====
dropout_28 (Dropout)  (None, 121, 64)       0
=====
max_pooling1d_28 (MaxPooling) (None, 24, 64)  0
=====
flatten_28 (Flatten)  (None, 1536)          0
=====
dense_55 (Dense)     (None, 16)             24592
=====
dense_56 (Dense)     (None, 3)              51
=====

Total params: 36,131
Trainable params: 36,131
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/25
- 5s - loss: 0.3291 - acc: 0.8702 - val_loss: 0.3198 - val_acc: 0.8667

Epoch 2/25
- 2s - loss: 0.2007 - acc: 0.9179 - val_loss: 0.2512 - val_acc: 0.9013

```

Epoch 3/25  
 - 2s - loss: 0.1731 - acc: 0.9270 - val\_loss: 0.2195 - val\_acc: 0.9096  
  
 Epoch 4/25  
 - 2s - loss: 0.1484 - acc: 0.9415 - val\_loss: 0.2409 - val\_acc: 0.9231  
  
 Epoch 5/25  
 - 2s - loss: 0.1308 - acc: 0.9506 - val\_loss: 0.2410 - val\_acc: 0.9045  
  
 Epoch 6/25  
 - 2s - loss: 0.1199 - acc: 0.9538 - val\_loss: 0.2219 - val\_acc: 0.9346  
  
 Epoch 7/25  
 - 2s - loss: 0.1042 - acc: 0.9612 - val\_loss: 0.2254 - val\_acc: 0.9263  
  
 Epoch 8/25  
 - 2s - loss: 0.1166 - acc: 0.9562 - val\_loss: 0.2362 - val\_acc: 0.9160  
  
 Epoch 9/25  
 - 2s - loss: 0.1204 - acc: 0.9538 - val\_loss: 0.1825 - val\_acc: 0.9321  
  
 Epoch 10/25  
 - 2s - loss: 0.0792 - acc: 0.9690 - val\_loss: 0.2098 - val\_acc: 0.9372  
  
 Epoch 11/25  
 - 2s - loss: 0.0676 - acc: 0.9759 - val\_loss: 0.2579 - val\_acc: 0.9147  
  
 Epoch 12/25  
 - 2s - loss: 0.0812 - acc: 0.9695 - val\_loss: 0.2358 - val\_acc: 0.9404  
  
 Epoch 13/25  
 - 2s - loss: 0.0756 - acc: 0.9747 - val\_loss: 0.2467 - val\_acc: 0.9346  
  
 Epoch 14/25  
 - 2s - loss: 0.0636 - acc: 0.9779 - val\_loss: 0.3400 - val\_acc: 0.9282  
  
 Epoch 15/25  
 - 2s - loss: 0.0512 - acc: 0.9843 - val\_loss: 0.2565 - val\_acc: 0.9385  
  
 Epoch 16/25  
 - 2s - loss: 0.0450 - acc: 0.9857 - val\_loss: 0.3122 - val\_acc: 0.9321  
  
 Epoch 17/25  
 - 2s - loss: 0.0417 - acc: 0.9830 - val\_loss: 0.2706 - val\_acc: 0.9397  
  
 Epoch 18/25  
 - 2s - loss: 0.0341 - acc: 0.9889 - val\_loss: 0.2967 - val\_acc: 0.9346  
  
 Epoch 19/25  
 - 2s - loss: 0.0291 - acc: 0.9867 - val\_loss: 0.2691 - val\_acc: 0.9410  
  
 Epoch 20/25  
 - 2s - loss: 0.0206 - acc: 0.9931 - val\_loss: 0.3605 - val\_acc: 0.9218  
  
 Epoch 21/25  
 - 2s - loss: 0.0344 - acc: 0.9877 - val\_loss: 0.3096 - val\_acc: 0.9417  
  
 Epoch 22/25  
 - 2s - loss: 0.0132 - acc: 0.9975 - val\_loss: 0.3446 - val\_acc: 0.9404  
  
 Epoch 23/25  
 - 2s - loss: 0.0140 - acc: 0.9961 - val\_loss: 0.3842 - val\_acc: 0.9333  
  
 Epoch 24/25  
 - 2s - loss: 0.0088 - acc: 0.9980 - val\_loss: 0.3691 - val\_acc: 0.9417  
  
 Epoch 25/25  
 - 2s - loss: 0.0129 - acc: 0.9961 - val\_loss: 0.3883 - val\_acc: 0.9321

Test accuracy:  
 0.9320512820512821  
 Model: "sequential\_29"

Layer (type)	Output Shape	Param #
conv1d_57 (Conv1D)	(None, 127, 32)	608

```
-  
conv1d_58 (Conv1D)           (None, 126, 64)      4160  
dropout_29 (Dropout)         (None, 126, 64)      0  
max_pooling1d_29 (MaxPooling) (None, 25, 64)      0  
flatten_29 (Flatten)         (None, 1600)        0  
dense_57 (Dense)            (None, 16)          25616  
dense_58 (Dense)            (None, 3)           51  
=====  
Total params: 30,435  
Trainable params: 30,435  
Non-trainable params: 0  
  
Train on 4067 samples, validate on 1560 samples  
Epoch 1/25  
- 5s - loss: 0.4758 - acc: 0.8685 - val_loss: 0.2347 - val_acc: 0.8968  
  
Epoch 2/25  
- 2s - loss: 0.2932 - acc: 0.8857 - val_loss: 0.3313 - val_acc: 0.8641  
  
Epoch 3/25  
- 2s - loss: 0.3348 - acc: 0.8925 - val_loss: 0.5066 - val_acc: 0.8705  
  
Epoch 4/25  
- 2s - loss: 0.3845 - acc: 0.9044 - val_loss: 0.3162 - val_acc: 0.8917  
  
Epoch 5/25  
- 2s - loss: 0.2691 - acc: 0.9277 - val_loss: 0.3439 - val_acc: 0.9058  
  
Epoch 6/25  
- 2s - loss: 0.2432 - acc: 0.9331 - val_loss: 0.2299 - val_acc: 0.8968  
  
Epoch 7/25  
- 2s - loss: 0.2227 - acc: 0.9299 - val_loss: 0.2090 - val_acc: 0.9231  
  
Epoch 8/25  
- 2s - loss: 0.2126 - acc: 0.9398 - val_loss: 0.2363 - val_acc: 0.9173  
  
Epoch 9/25  
- 2s - loss: 0.2817 - acc: 0.9289 - val_loss: 0.2554 - val_acc: 0.9263  
  
Epoch 10/25  
- 2s - loss: 0.1927 - acc: 0.9420 - val_loss: 0.3347 - val_acc: 0.8846  
  
Epoch 11/25  
- 2s - loss: 0.2563 - acc: 0.9034 - val_loss: 0.2948 - val_acc: 0.9122  
  
Epoch 12/25  
- 2s - loss: 0.2117 - acc: 0.9270 - val_loss: 0.2841 - val_acc: 0.8994  
  
Epoch 13/25  
- 2s - loss: 0.5807 - acc: 0.9144 - val_loss: 5.9916 - val_acc: 0.5532  
  
Epoch 14/25  
- 2s - loss: 5.9087 - acc: 0.5788 - val_loss: 6.0143 - val_acc: 0.5551  
  
Epoch 15/25  
- 2s - loss: 5.8144 - acc: 0.5751 - val_loss: 5.6369 - val_acc: 0.5712  
  
Epoch 16/25  
- 2s - loss: 5.7761 - acc: 0.5798 - val_loss: 5.6717 - val_acc: 0.5365  
  
Epoch 17/25  
- 2s - loss: 5.7341 - acc: 0.5980 - val_loss: 5.6427 - val_acc: 0.5628  
  
Epoch 18/25  
- 2s - loss: 3.2149 - acc: 0.7492 - val_loss: 5.6064 - val_acc: 0.5019  
  
Epoch 19/25  
- 2s - loss: 5.6190 - acc: 0.5960 - val_loss: 5.3038 - val_acc: 0.5346  
  
Epoch 20/25  
- 2s - loss: 5.5098 - acc: 0.5712 - val_loss: 5.2970 - val_acc: 0.5558
```

```
Epoch 21/25
- 2s - loss: 5.4469 - acc: 0.5975 - val_loss: 5.2323 - val_acc: 0.5551

Epoch 22/25
- 2s - loss: 5.4050 - acc: 0.6061 - val_loss: 5.2909 - val_acc: 0.5756

Epoch 23/25
- 2s - loss: 5.3557 - acc: 0.6019 - val_loss: 5.2280 - val_acc: 0.5564

Epoch 24/25
- 2s - loss: 3.4262 - acc: 0.5754 - val_loss: 0.9061 - val_acc: 0.5397

Epoch 25/25
- 2s - loss: 0.8414 - acc: 0.5530 - val_loss: 0.8672 - val_acc: 0.5333
```

Test accuracy:  
0.533333333333333  
Model: "sequential\_30"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_59 (Conv1D)	(None, 127, 32)	608
conv1d_60 (Conv1D)	(None, 126, 64)	4160
dropout_30 (Dropout)	(None, 126, 64)	0
max_pooling1d_30 (MaxPooling)	(None, 25, 64)	0
flatten_30 (Flatten)	(None, 1600)	0
dense_59 (Dense)	(None, 16)	25616
dense_60 (Dense)	(None, 3)	51
<hr/>		

Total params: 30,435  
Trainable params: 30,435  
Non-trainable params: 0

---

Train on 4067 samples, validate on 1560 samples

Epoch 1/25  
- 5s - loss: 0.4804 - acc: 0.8544 - val\_loss: 0.5246 - val\_acc: 0.8442

Epoch 2/25  
- 2s - loss: 0.2521 - acc: 0.9132 - val\_loss: 0.3037 - val\_acc: 0.8782

Epoch 3/25  
- 2s - loss: 0.2646 - acc: 0.9198 - val\_loss: 0.2489 - val\_acc: 0.9006

Epoch 4/25  
- 2s - loss: 0.2670 - acc: 0.9098 - val\_loss: 0.2759 - val\_acc: 0.9071

Epoch 5/25  
- 2s - loss: 0.1982 - acc: 0.9343 - val\_loss: 0.2748 - val\_acc: 0.9205

Epoch 6/25  
- 2s - loss: 0.2580 - acc: 0.9213 - val\_loss: 0.3153 - val\_acc: 0.8724

Epoch 7/25  
- 2s - loss: 0.4748 - acc: 0.8532 - val\_loss: 0.5596 - val\_acc: 0.8353

Epoch 8/25  
- 2s - loss: 0.3754 - acc: 0.8803 - val\_loss: 0.3228 - val\_acc: 0.9288

Epoch 9/25  
- 2s - loss: 0.3270 - acc: 0.8999 - val\_loss: 0.3267 - val\_acc: 0.9192

Epoch 10/25  
- 2s - loss: 0.2756 - acc: 0.9179 - val\_loss: 0.3487 - val\_acc: 0.8885

Epoch 11/25  
- 2s - loss: 0.9085 - acc: 0.8626 - val\_loss: 0.4494 - val\_acc: 0.8071

Epoch 12/25  
- 2s - loss: 2.3451 - acc: 0.7915 - val\_loss: 5.8044 - val\_acc: 0.5615

```

Epoch 13/25
- 2s - loss: 5.8264 - acc: 0.5739 - val_loss: 5.7453 - val_acc: 0.5808

Epoch 14/25
- 2s - loss: 5.7212 - acc: 0.5985 - val_loss: 5.7465 - val_acc: 0.5814

Epoch 15/25
- 2s - loss: 5.7412 - acc: 0.5928 - val_loss: 5.7875 - val_acc: 0.5724

Epoch 16/25
- 2s - loss: 5.7222 - acc: 0.5987 - val_loss: 5.7671 - val_acc: 0.5840

Epoch 17/25
- 2s - loss: 5.7054 - acc: 0.6076 - val_loss: 5.8408 - val_acc: 0.4865

Epoch 18/25
- 2s - loss: 5.7070 - acc: 0.6059 - val_loss: 5.7475 - val_acc: 0.5994

Epoch 19/25
- 2s - loss: 5.6978 - acc: 0.6078 - val_loss: 5.7467 - val_acc: 0.5782

Epoch 20/25
- 2s - loss: 5.7442 - acc: 0.5909 - val_loss: 5.8804 - val_acc: 0.4853

Epoch 21/25
- 2s - loss: 5.7076 - acc: 0.6012 - val_loss: 5.7541 - val_acc: 0.5968

Epoch 22/25
- 2s - loss: 5.6721 - acc: 0.6152 - val_loss: 5.8302 - val_acc: 0.5718

Epoch 23/25
- 2s - loss: 5.6803 - acc: 0.6132 - val_loss: 5.7349 - val_acc: 0.5846

Epoch 24/25
- 2s - loss: 5.6847 - acc: 0.6115 - val_loss: 5.7591 - val_acc: 0.5686

Epoch 25/25
- 2s - loss: 5.6949 - acc: 0.6095 - val_loss: 5.7431 - val_acc: 0.5936

Test accuracy:
0.5935897435897436
Model: "sequential_31"

Layer (type)          Output Shape         Param #
=================================================================
conv1d_61 (Conv1D)    (None, 125, 32)      1184
conv1d_62 (Conv1D)    (None, 121, 64)       10304
dropout_31 (Dropout) (None, 121, 64)       0
max_pooling1d_31 (MaxPooling) (None, 24, 64) 0
flatten_31 (Flatten)  (None, 1536)        0
dense_61 (Dense)     (None, 16)           24592
dense_62 (Dense)     (None, 3)            51
=================================================================
Total params: 36,131
Trainable params: 36,131
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/25
- 5s - loss: 0.3781 - acc: 0.8596 - val_loss: 0.3092 - val_acc: 0.8814

Epoch 2/25
- 2s - loss: 0.1975 - acc: 0.9198 - val_loss: 0.3020 - val_acc: 0.8833

Epoch 3/25
- 2s - loss: 0.1779 - acc: 0.9275 - val_loss: 0.2876 - val_acc: 0.8859

Epoch 4/25
- 2s - loss: 0.1673 - acc: 0.9302 - val_loss: 0.2629 - val_acc: 0.9013

Epoch 5/25

```

```

- 2s - loss: 0.1393 - acc: 0.9415 - val_loss: 0.2835 - val_acc: 0.9167

Epoch 6/25
- 2s - loss: 0.1260 - acc: 0.9489 - val_loss: 0.2571 - val_acc: 0.9250

Epoch 7/25
- 2s - loss: 0.1075 - acc: 0.9557 - val_loss: 0.2351 - val_acc: 0.9231

Epoch 8/25
- 2s - loss: 0.0920 - acc: 0.9624 - val_loss: 0.2343 - val_acc: 0.9417

Epoch 9/25
- 2s - loss: 0.1078 - acc: 0.9570 - val_loss: 0.2510 - val_acc: 0.9115

Epoch 10/25
- 2s - loss: 0.1540 - acc: 0.9594 - val_loss: 0.2999 - val_acc: 0.9205

Epoch 11/25
- 2s - loss: 0.0969 - acc: 0.9680 - val_loss: 0.2658 - val_acc: 0.9224

Epoch 12/25
- 2s - loss: 0.0596 - acc: 0.9754 - val_loss: 0.2503 - val_acc: 0.9276

Epoch 13/25
- 2s - loss: 0.0523 - acc: 0.9801 - val_loss: 0.2641 - val_acc: 0.9327

Epoch 14/25
- 2s - loss: 0.0472 - acc: 0.9806 - val_loss: 0.2771 - val_acc: 0.9224

Epoch 15/25
- 2s - loss: 0.0437 - acc: 0.9830 - val_loss: 0.2672 - val_acc: 0.9276

Epoch 16/25
- 2s - loss: 0.0388 - acc: 0.9880 - val_loss: 0.2529 - val_acc: 0.9468

Epoch 17/25
- 2s - loss: 0.0288 - acc: 0.9899 - val_loss: 0.2535 - val_acc: 0.9340

Epoch 18/25
- 2s - loss: 0.0431 - acc: 0.9852 - val_loss: 0.3140 - val_acc: 0.9321

Epoch 19/25
- 2s - loss: 0.0343 - acc: 0.9889 - val_loss: 0.2950 - val_acc: 0.9308

Epoch 20/25
- 2s - loss: 0.0320 - acc: 0.9875 - val_loss: 0.3519 - val_acc: 0.9327

Epoch 21/25
- 2s - loss: 0.0244 - acc: 0.9914 - val_loss: 0.3061 - val_acc: 0.9410

Epoch 22/25
- 2s - loss: 0.0276 - acc: 0.9904 - val_loss: 0.4064 - val_acc: 0.9333

Epoch 23/25
- 2s - loss: 0.0344 - acc: 0.9916 - val_loss: 0.3926 - val_acc: 0.9115

Epoch 24/25
- 2s - loss: 0.0436 - acc: 0.9850 - val_loss: 0.4471 - val_acc: 0.9288

Epoch 25/25
- 2s - loss: 0.0237 - acc: 0.9921 - val_loss: 0.4810 - val_acc: 0.9186

Test accuracy:
0.9185897435897435
Model: "sequential_32"

Layer (type)          Output Shape         Param #
=====
conv1d_63 (Conv1D)    (None, 125, 32)      1184
conv1d_64 (Conv1D)    (None, 124, 64)       4160
dropout_32 (Dropout)  (None, 124, 64)       0
max_pooling1d_32 (MaxPooling) (None, 41, 64)   0
flatten_32 (Flatten)  (None, 2624)        0

```

```
layer_63 -> dense_63, (None, 16), 42000
dense_64 -> dense_64, (None, 3), 51
=====
Total params: 47,395
Trainable params: 47,395
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/25
- 5s - loss: 0.3313 - acc: 0.8704 - val_loss: 0.3268 - val_acc: 0.8936

Epoch 2/25
- 2s - loss: 0.2038 - acc: 0.9098 - val_loss: 0.3083 - val_acc: 0.8917

Epoch 3/25
- 2s - loss: 0.1684 - acc: 0.9262 - val_loss: 0.2748 - val_acc: 0.9051

Epoch 4/25
- 2s - loss: 0.1532 - acc: 0.9356 - val_loss: 0.2900 - val_acc: 0.8987

Epoch 5/25
- 2s - loss: 0.1383 - acc: 0.9442 - val_loss: 0.2802 - val_acc: 0.9051

Epoch 6/25
- 2s - loss: 0.1186 - acc: 0.9523 - val_loss: 0.2472 - val_acc: 0.9141

Epoch 7/25
- 2s - loss: 0.1093 - acc: 0.9570 - val_loss: 0.2586 - val_acc: 0.9250

Epoch 8/25
- 2s - loss: 0.0998 - acc: 0.9631 - val_loss: 0.2232 - val_acc: 0.9141

Epoch 9/25
- 2s - loss: 0.0787 - acc: 0.9712 - val_loss: 0.2286 - val_acc: 0.9333

Epoch 10/25
- 2s - loss: 0.0852 - acc: 0.9707 - val_loss: 0.2483 - val_acc: 0.9314

Epoch 11/25
- 2s - loss: 0.0705 - acc: 0.9781 - val_loss: 0.2476 - val_acc: 0.9179

Epoch 12/25
- 2s - loss: 0.0529 - acc: 0.9786 - val_loss: 0.2240 - val_acc: 0.9327

Epoch 13/25
- 2s - loss: 0.0507 - acc: 0.9806 - val_loss: 0.2426 - val_acc: 0.9308

Epoch 14/25
- 2s - loss: 0.0332 - acc: 0.9887 - val_loss: 0.2477 - val_acc: 0.9378

Epoch 15/25
- 2s - loss: 0.0293 - acc: 0.9897 - val_loss: 0.2637 - val_acc: 0.9365

Epoch 16/25
- 2s - loss: 0.0263 - acc: 0.9911 - val_loss: 0.2772 - val_acc: 0.9385

Epoch 17/25
- 2s - loss: 0.0231 - acc: 0.9914 - val_loss: 0.4147 - val_acc: 0.9167

Epoch 18/25
- 2s - loss: 0.0243 - acc: 0.9931 - val_loss: 0.4383 - val_acc: 0.9301

Epoch 19/25
- 2s - loss: 0.0841 - acc: 0.9816 - val_loss: 0.3115 - val_acc: 0.9282

Epoch 20/25
- 2s - loss: 0.0274 - acc: 0.9926 - val_loss: 0.4259 - val_acc: 0.9199

Epoch 21/25
- 2s - loss: 0.0203 - acc: 0.9934 - val_loss: 0.3608 - val_acc: 0.9212

Epoch 22/25
- 2s - loss: 0.0193 - acc: 0.9941 - val_loss: 0.3708 - val_acc: 0.9346

Epoch 23/25
```

```
Epoch 23/25
- 2s - loss: 0.0066 - acc: 0.9988 - val_loss: 0.3444 - val_acc: 0.9385
```

```
Epoch 24/25
- 2s - loss: 0.0058 - acc: 0.9983 - val_loss: 0.3971 - val_acc: 0.9282
```

```
Epoch 25/25
- 2s - loss: 0.0058 - acc: 0.9975 - val_loss: 0.4931 - val_acc: 0.9154
```

Test accuracy:

0.9153846153846154

Model: "sequential\_33"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_65 (Conv1D)	(None, 127, 32)	608
conv1d_66 (Conv1D)	(None, 123, 64)	10304
dropout_33 (Dropout)	(None, 123, 64)	0
max_pooling1d_33 (MaxPooling)	(None, 24, 64)	0
flatten_33 (Flatten)	(None, 1536)	0
dense_65 (Dense)	(None, 64)	98368
dense_66 (Dense)	(None, 3)	195
<hr/>		
Total params: 109,475		
Trainable params: 109,475		
Non-trainable params: 0		

---

Train on 4067 samples, validate on 1560 samples

Epoch 1/25

```
- 5s - loss: 10.5132 - acc: 0.3455 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 2/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 3/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 4/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 5/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 6/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 7/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 8/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 9/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 10/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 11/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 12/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 13/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 14/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 15/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

```
2s    loss: 10.5420    acc: 0.3460    val_loss: 10.5698    val_acc: 0.3442
```

Epoch 16/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 17/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 18/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 19/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 20/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 21/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 22/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 23/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 24/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Epoch 25/25

```
- 2s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442
```

Test accuracy:

0.34423076923076923

Model: "sequential\_34"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_67 (Conv1D)	(None, 125, 32)	1184
conv1d_68 (Conv1D)	(None, 124, 64)	4160
dropout_34 (Dropout)	(None, 124, 64)	0
max_pooling1d_34 (MaxPooling)	(None, 41, 64)	0
flatten_34 (Flatten)	(None, 2624)	0
dense_67 (Dense)	(None, 16)	42000
dense_68 (Dense)	(None, 3)	51
<hr/>		

Total params: 47,395

Trainable params: 47,395

Non-trainable params: 0

---

Train on 4067 samples, validate on 1560 samples

Epoch 1/25

```
- 5s - loss: 0.3617 - acc: 0.8719 - val_loss: 0.3268 - val_acc: 0.8750
```

Epoch 2/25

```
- 2s - loss: 0.2065 - acc: 0.9171 - val_loss: 0.2910 - val_acc: 0.8718
```

Epoch 3/25

```
- 2s - loss: 0.1879 - acc: 0.9255 - val_loss: 0.2600 - val_acc: 0.8814
```

Epoch 4/25

```
- 2s - loss: 0.1503 - acc: 0.9334 - val_loss: 0.2492 - val_acc: 0.9058
```

Epoch 5/25

```
- 2s - loss: 0.1360 - acc: 0.9452 - val_loss: 0.2575 - val_acc: 0.8904
```

Epoch 6/25

```
- 2s - loss: 0.1166 - acc: 0.9525 - val_loss: 0.2459 - val_acc: 0.9109
```

Epoch 7/25

```
- 2s - loss: 0.1095 - acc: 0.9552 - val_loss: 0.2104 - val_acc: 0.9109
```

```

Epoch 8/25
- 2s - loss: 0.1057 - acc: 0.9575 - val_loss: 0.2172 - val_acc: 0.9212

Epoch 9/25
- 2s - loss: 0.0841 - acc: 0.9688 - val_loss: 0.1921 - val_acc: 0.9321

Epoch 10/25
- 2s - loss: 0.0791 - acc: 0.9702 - val_loss: 0.1987 - val_acc: 0.9212

Epoch 11/25
- 2s - loss: 0.0873 - acc: 0.9698 - val_loss: 0.2468 - val_acc: 0.9192

Epoch 12/25
- 2s - loss: 0.0615 - acc: 0.9786 - val_loss: 0.2133 - val_acc: 0.9199

Epoch 13/25
- 2s - loss: 0.0812 - acc: 0.9774 - val_loss: 0.2573 - val_acc: 0.9301

Epoch 14/25
- 2s - loss: 0.0813 - acc: 0.9761 - val_loss: 0.2735 - val_acc: 0.9077

Epoch 15/25
- 2s - loss: 0.0561 - acc: 0.9806 - val_loss: 0.2550 - val_acc: 0.9321

Epoch 16/25
- 2s - loss: 0.0441 - acc: 0.9840 - val_loss: 0.2771 - val_acc: 0.9224

Epoch 17/25
- 2s - loss: 0.0323 - acc: 0.9884 - val_loss: 0.2471 - val_acc: 0.9276

Epoch 18/25
- 2s - loss: 0.0267 - acc: 0.9904 - val_loss: 0.2998 - val_acc: 0.9218

Epoch 19/25
- 2s - loss: 0.0319 - acc: 0.9880 - val_loss: 0.2742 - val_acc: 0.9301

Epoch 20/25
- 2s - loss: 0.0343 - acc: 0.9894 - val_loss: 0.2634 - val_acc: 0.9237

Epoch 21/25
- 2s - loss: 0.0700 - acc: 0.9828 - val_loss: 0.3328 - val_acc: 0.9192

Epoch 22/25
- 2s - loss: 0.0533 - acc: 0.9867 - val_loss: 0.3365 - val_acc: 0.9199

Epoch 23/25
- 2s - loss: 0.0223 - acc: 0.9931 - val_loss: 0.2887 - val_acc: 0.9314

Epoch 24/25
- 2s - loss: 0.0187 - acc: 0.9943 - val_loss: 0.2715 - val_acc: 0.9276

Epoch 25/25
- 2s - loss: 0.0627 - acc: 0.9848 - val_loss: 0.3523 - val_acc: 0.9141

Test accuracy:
0.9141025641025641
Model: "sequential_35"


```

Layer (type)	Output Shape	Param #
conv1d_69 (Conv1D)	(None, 127, 32)	608
conv1d_70 (Conv1D)	(None, 123, 64)	10304
dropout_35 (Dropout)	(None, 123, 64)	0
max_pooling1d_35 (MaxPooling)	(None, 24, 64)	0
flatten_35 (Flatten)	(None, 1536)	0
dense_69 (Dense)	(None, 64)	98368
dense_70 (Dense)	(None, 3)	195

```

Total params: 109,475
Trainable params: 109,475
Non-trainable params: 0

```

non-trainable params: 0

Train on 4067 samples, validate on 1560 samples

---

Epoch 1/25  
- 6s - loss: 0.3771 - acc: 0.8729 - val\_loss: 0.2376 - val\_acc: 0.9167

Epoch 2/25  
- 2s - loss: 0.1860 - acc: 0.9329 - val\_loss: 0.2209 - val\_acc: 0.9167

Epoch 3/25  
- 2s - loss: 0.1644 - acc: 0.9447 - val\_loss: 0.2033 - val\_acc: 0.9365

Epoch 4/25  
- 2s - loss: 0.1303 - acc: 0.9570 - val\_loss: 0.2295 - val\_acc: 0.9404

Epoch 5/25  
- 2s - loss: 0.1139 - acc: 0.9616 - val\_loss: 0.2208 - val\_acc: 0.9526

Epoch 6/25  
- 2s - loss: 0.1037 - acc: 0.9680 - val\_loss: 0.2116 - val\_acc: 0.9481

Epoch 7/25  
- 2s - loss: 0.0886 - acc: 0.9754 - val\_loss: 0.3153 - val\_acc: 0.9365

Epoch 8/25  
- 2s - loss: 0.0746 - acc: 0.9791 - val\_loss: 0.2767 - val\_acc: 0.9487

Epoch 9/25  
- 2s - loss: 0.0671 - acc: 0.9806 - val\_loss: 0.2970 - val\_acc: 0.9436

Epoch 10/25  
- 2s - loss: 0.0699 - acc: 0.9840 - val\_loss: 0.3029 - val\_acc: 0.9538

Epoch 11/25  
- 2s - loss: 0.0635 - acc: 0.9840 - val\_loss: 0.3658 - val\_acc: 0.9474

Epoch 12/25  
- 2s - loss: 0.0970 - acc: 0.9813 - val\_loss: 0.3188 - val\_acc: 0.9513

Epoch 13/25  
- 2s - loss: 0.0480 - acc: 0.9884 - val\_loss: 0.4347 - val\_acc: 0.9372

Epoch 14/25  
- 2s - loss: 0.0594 - acc: 0.9884 - val\_loss: 0.3441 - val\_acc: 0.9526

Epoch 15/25  
- 2s - loss: 0.0462 - acc: 0.9889 - val\_loss: 0.3475 - val\_acc: 0.9500

Epoch 16/25  
- 2s - loss: 0.0265 - acc: 0.9936 - val\_loss: 0.4100 - val\_acc: 0.9519

Epoch 17/25  
- 2s - loss: 0.0333 - acc: 0.9924 - val\_loss: 0.3159 - val\_acc: 0.9462

Epoch 18/25  
- 2s - loss: 0.0415 - acc: 0.9916 - val\_loss: 0.4728 - val\_acc: 0.9455

Epoch 19/25  
- 2s - loss: 0.0267 - acc: 0.9931 - val\_loss: 0.4020 - val\_acc: 0.9442

Epoch 20/25  
- 2s - loss: 0.0562 - acc: 0.9909 - val\_loss: 0.4670 - val\_acc: 0.9455

Epoch 21/25  
- 2s - loss: 0.0350 - acc: 0.9951 - val\_loss: 0.5393 - val\_acc: 0.9423

Epoch 22/25  
- 2s - loss: 0.0341 - acc: 0.9956 - val\_loss: 0.3950 - val\_acc: 0.9577

Epoch 23/25  
- 2s - loss: 0.0225 - acc: 0.9956 - val\_loss: 0.4987 - val\_acc: 0.9494

Epoch 24/25  
- 2s - loss: 0.0457 - acc: 0.9939 - val\_loss: 0.4097 - val\_acc: 0.9506

Epoch 25/25  
- 2s - loss: 0.0355 - acc: 0.9939 - val\_loss: 0.4792 - val\_acc: 0.9506

Test accuracy:  
0.9506410256410256  
Model: "sequential\_36"

Layer (type)	Output Shape	Param #
conv1d_71 (Conv1D)	(None, 125, 32)	1184
conv1d_72 (Conv1D)	(None, 124, 64)	4160
dropout_36 (Dropout)	(None, 124, 64)	0
max_pooling1d_36 (MaxPooling)	(None, 41, 64)	0
flatten_36 (Flatten)	(None, 2624)	0
dense_71 (Dense)	(None, 64)	168000
dense_72 (Dense)	(None, 3)	195

Total params: 173,539  
Trainable params: 173,539  
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples  
Epoch 1/25  
- 6s - loss: 0.3552 - acc: 0.8746 - val\_loss: 0.4499 - val\_acc: 0.8359  
  
Epoch 2/25  
- 2s - loss: 0.1947 - acc: 0.9282 - val\_loss: 0.2854 - val\_acc: 0.9032  
  
Epoch 3/25  
- 2s - loss: 0.1409 - acc: 0.9481 - val\_loss: 0.2473 - val\_acc: 0.9269  
  
Epoch 4/25  
- 2s - loss: 0.1245 - acc: 0.9612 - val\_loss: 0.2959 - val\_acc: 0.9288  
  
Epoch 5/25  
- 2s - loss: 0.1131 - acc: 0.9675 - val\_loss: 0.3405 - val\_acc: 0.9359  
  
Epoch 6/25  
- 2s - loss: 0.0958 - acc: 0.9759 - val\_loss: 0.4019 - val\_acc: 0.9192  
  
Epoch 7/25  
- 2s - loss: 0.0695 - acc: 0.9808 - val\_loss: 0.3006 - val\_acc: 0.9353  
  
Epoch 8/25  
- 2s - loss: 0.0561 - acc: 0.9848 - val\_loss: 0.3649 - val\_acc: 0.9263  
  
Epoch 9/25  
- 2s - loss: 0.0555 - acc: 0.9852 - val\_loss: 0.3558 - val\_acc: 0.9417  
  
Epoch 10/25  
- 2s - loss: 0.0403 - acc: 0.9907 - val\_loss: 0.4777 - val\_acc: 0.9327  
  
Epoch 11/25  
- 2s - loss: 0.0401 - acc: 0.9911 - val\_loss: 0.4309 - val\_acc: 0.9462  
  
Epoch 12/25  
- 2s - loss: 0.0511 - acc: 0.9911 - val\_loss: 0.4504 - val\_acc: 0.9365  
  
Epoch 13/25  
- 2s - loss: 0.0365 - acc: 0.9931 - val\_loss: 0.4659 - val\_acc: 0.9397  
  
Epoch 14/25  
- 2s - loss: 0.0316 - acc: 0.9929 - val\_loss: 0.5569 - val\_acc: 0.9404  
  
Epoch 15/25  
- 2s - loss: 0.0421 - acc: 0.9924 - val\_loss: 0.4776 - val\_acc: 0.9462  
  
Epoch 16/25  
- 2s - loss: 0.0246 - acc: 0.9966 - val\_loss: 0.4497 - val\_acc: 0.9513  
  
Epoch 17/25  
- 2s - loss: 0.0268 - acc: 0.9951 - val\_loss: 0.6025 - val\_acc: 0.9269

```

Epoch 18/25
- 2s - loss: 0.0283 - acc: 0.9958 - val_loss: 0.5849 - val_acc: 0.9359

Epoch 19/25
- 2s - loss: 0.0330 - acc: 0.9961 - val_loss: 0.5285 - val_acc: 0.9391

Epoch 20/25
- 2s - loss: 0.0196 - acc: 0.9963 - val_loss: 0.5536 - val_acc: 0.9359

Epoch 21/25
- 2s - loss: 0.0155 - acc: 0.9975 - val_loss: 0.5375 - val_acc: 0.9404

Epoch 22/25
- 2s - loss: 0.0218 - acc: 0.9963 - val_loss: 0.6292 - val_acc: 0.9353

Epoch 23/25
- 2s - loss: 0.0292 - acc: 0.9968 - val_loss: 0.7481 - val_acc: 0.9256

Epoch 24/25
- 2s - loss: 0.0230 - acc: 0.9975 - val_loss: 0.5805 - val_acc: 0.9365

Epoch 25/25
- 2s - loss: 0.0213 - acc: 0.9970 - val_loss: 0.5644 - val_acc: 0.9429

Test accuracy:
0.9429487179487179
Model: "sequential_37"

Layer (type)          Output Shape         Param #
=====
conv1d_73 (Conv1D)    (None, 125, 32)      1184
conv1d_74 (Conv1D)    (None, 124, 64)       4160
dropout_37 (Dropout)  (None, 124, 64)       0
max_pooling1d_37 (MaxPooling) (None, 41, 64)  0
flatten_37 (Flatten)  (None, 2624)        0
dense_73 (Dense)     (None, 16)           42000
dense_74 (Dense)     (None, 3)            51
=====
Total params: 47,395
Trainable params: 47,395
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/25
- 5s - loss: 0.3676 - acc: 0.8554 - val_loss: 0.3373 - val_acc: 0.8962

Epoch 2/25
- 1s - loss: 0.1859 - acc: 0.9233 - val_loss: 0.2568 - val_acc: 0.8987

Epoch 3/25
- 1s - loss: 0.1518 - acc: 0.9346 - val_loss: 0.2489 - val_acc: 0.9141

Epoch 4/25
- 1s - loss: 0.1494 - acc: 0.9371 - val_loss: 0.2778 - val_acc: 0.9064

Epoch 5/25
- 1s - loss: 0.1337 - acc: 0.9437 - val_loss: 0.2635 - val_acc: 0.9026

Epoch 6/25
- 1s - loss: 0.1116 - acc: 0.9528 - val_loss: 0.2463 - val_acc: 0.9147

Epoch 7/25
- 1s - loss: 0.1061 - acc: 0.9533 - val_loss: 0.2573 - val_acc: 0.9173

Epoch 8/25
- 1s - loss: 0.0989 - acc: 0.9577 - val_loss: 0.2705 - val_acc: 0.9154

Epoch 9/25
- 1s - loss: 0.0855 - acc: 0.9639 - val_loss: 0.2860 - val_acc: 0.9244

Epoch 10/25
- 1s - loss: 0.0855 - acc: 0.9639 - val_loss: 0.2860 - val_acc: 0.9244

```

```
- ls - loss: 0.0972 - acc: 0.9636 - val_loss: 0.2716 - val_acc: 0.9263

Epoch 11/25
- ls - loss: 0.0879 - acc: 0.9668 - val_loss: 0.2776 - val_acc: 0.9199

Epoch 12/25
- ls - loss: 0.0648 - acc: 0.9754 - val_loss: 0.2680 - val_acc: 0.9224

Epoch 13/25
- ls - loss: 0.0628 - acc: 0.9744 - val_loss: 0.3023 - val_acc: 0.9256

Epoch 14/25
- ls - loss: 0.0627 - acc: 0.9793 - val_loss: 0.2423 - val_acc: 0.9365

Epoch 15/25
- ls - loss: 0.0490 - acc: 0.9813 - val_loss: 0.2914 - val_acc: 0.9276

Epoch 16/25
- ls - loss: 0.0478 - acc: 0.9816 - val_loss: 0.2979 - val_acc: 0.9256

Epoch 17/25
- ls - loss: 0.0351 - acc: 0.9880 - val_loss: 0.2813 - val_acc: 0.9353

Epoch 18/25
- ls - loss: 0.0292 - acc: 0.9909 - val_loss: 0.2928 - val_acc: 0.9340

Epoch 19/25
- ls - loss: 0.0258 - acc: 0.9926 - val_loss: 0.2957 - val_acc: 0.9442

Epoch 20/25
- ls - loss: 0.0264 - acc: 0.9921 - val_loss: 0.3326 - val_acc: 0.9372

Epoch 21/25
- ls - loss: 0.0354 - acc: 0.9867 - val_loss: 0.4880 - val_acc: 0.9045

Epoch 22/25
- ls - loss: 0.0319 - acc: 0.9889 - val_loss: 0.3528 - val_acc: 0.9244

Epoch 23/25
- ls - loss: 0.0316 - acc: 0.9894 - val_loss: 0.3578 - val_acc: 0.9353

Epoch 24/25
- ls - loss: 0.0259 - acc: 0.9914 - val_loss: 0.3441 - val_acc: 0.9346

Epoch 25/25
- ls - loss: 0.0199 - acc: 0.9939 - val_loss: 0.3200 - val_acc: 0.9449

Test accuracy:
0.9448717948717948
Model: "sequential_38"

Layer (type)          Output Shape         Param #
=====
```

Layer (type)	Output Shape	Param #
conv1d_75 (Conv1D)	(None, 127, 32)	608
conv1d_76 (Conv1D)	(None, 123, 64)	10304
dropout_38 (Dropout)	(None, 123, 64)	0
max_pooling1d_38 (MaxPooling)	(None, 24, 64)	0
flatten_38 (Flatten)	(None, 1536)	0
dense_75 (Dense)	(None, 64)	98368
dense_76 (Dense)	(None, 3)	195

```
Total params: 109,475
```

```
Trainable params: 109,475
```

```
Non-trainable params: 0
```

---

```
Train on 4067 samples, validate on 1560 samples
```

```
Epoch 1/30
```

```
- 6s - loss: 0.4004 - acc: 0.8677 - val_loss: 0.3196 - val_acc: 0.8660
```

```
Epoch 2/30
```

```
- 2s - loss: 0.2005 - acc: 0.9299 - val_loss: 0.2138 - val_acc: 0.9154
```

```
Epoch 3/30
- 2s - loss: 0.1432 - acc: 0.9498 - val_loss: 0.2072 - val_acc: 0.9256

Epoch 4/30
- 2s - loss: 0.1038 - acc: 0.9639 - val_loss: 0.1952 - val_acc: 0.9462

Epoch 5/30
- 2s - loss: 0.0919 - acc: 0.9715 - val_loss: 0.2458 - val_acc: 0.9269

Epoch 6/30
- 2s - loss: 0.0768 - acc: 0.9774 - val_loss: 0.2062 - val_acc: 0.9506

Epoch 7/30
- 2s - loss: 0.0767 - acc: 0.9796 - val_loss: 0.2317 - val_acc: 0.9506

Epoch 8/30
- 2s - loss: 0.0522 - acc: 0.9845 - val_loss: 0.2236 - val_acc: 0.9538

Epoch 9/30
- 2s - loss: 0.0649 - acc: 0.9865 - val_loss: 0.3116 - val_acc: 0.9327

Epoch 10/30
- 2s - loss: 0.0577 - acc: 0.9884 - val_loss: 0.3302 - val_acc: 0.9327

Epoch 11/30
- 2s - loss: 0.0511 - acc: 0.9880 - val_loss: 0.2624 - val_acc: 0.9519

Epoch 12/30
- 2s - loss: 0.0534 - acc: 0.9902 - val_loss: 0.2614 - val_acc: 0.9590

Epoch 13/30
- 2s - loss: 0.0335 - acc: 0.9907 - val_loss: 0.3536 - val_acc: 0.9372

Epoch 14/30
- 2s - loss: 0.0256 - acc: 0.9948 - val_loss: 0.3420 - val_acc: 0.9474

Epoch 15/30
- 2s - loss: 0.0347 - acc: 0.9941 - val_loss: 0.4206 - val_acc: 0.9301

Epoch 16/30
- 2s - loss: 0.0181 - acc: 0.9966 - val_loss: 0.4293 - val_acc: 0.9378

Epoch 17/30
- 2s - loss: 0.0407 - acc: 0.9941 - val_loss: 0.4229 - val_acc: 0.9417

Epoch 18/30
- 2s - loss: 0.0451 - acc: 0.9946 - val_loss: 0.5269 - val_acc: 0.9327

Epoch 19/30
- 2s - loss: 0.0414 - acc: 0.9941 - val_loss: 0.3480 - val_acc: 0.9609

Epoch 20/30
- 2s - loss: 0.0291 - acc: 0.9943 - val_loss: 0.5185 - val_acc: 0.9314

Epoch 21/30
- 2s - loss: 0.0247 - acc: 0.9951 - val_loss: 0.4095 - val_acc: 0.9365

Epoch 22/30
- 2s - loss: 0.0199 - acc: 0.9968 - val_loss: 0.3379 - val_acc: 0.9558

Epoch 23/30
- 2s - loss: 0.0233 - acc: 0.9951 - val_loss: 0.4191 - val_acc: 0.9404

Epoch 24/30
- 2s - loss: 0.0291 - acc: 0.9956 - val_loss: 0.4687 - val_acc: 0.9378

Epoch 25/30
- 2s - loss: 0.0301 - acc: 0.9963 - val_loss: 0.3634 - val_acc: 0.9494

Epoch 26/30
- 2s - loss: 0.0286 - acc: 0.9961 - val_loss: 0.3978 - val_acc: 0.9468

Epoch 27/30
- 2s - loss: 0.0241 - acc: 0.9978 - val_loss: 0.5497 - val_acc: 0.9321

Epoch 28/30
```

```

- 2s - loss: 0.0275 - acc: 0.9968 - val_loss: 0.3838 - val_acc: 0.9577

Epoch 29/30
- 2s - loss: 0.0300 - acc: 0.9956 - val_loss: 0.4368 - val_acc: 0.9487

Epoch 30/30
- 2s - loss: 0.0161 - acc: 0.9973 - val_loss: 0.5484 - val_acc: 0.9365

Test accuracy:
0.9365384615384615
Model: "sequential_39"

Layer (type)          Output Shape         Param #
=================================================================
conv1d_77 (Conv1D)    (None, 125, 32)      1184
conv1d_78 (Conv1D)    (None, 124, 16)       1040
dropout_39 (Dropout) (None, 124, 16)       0
max_pooling1d_39 (MaxPooling) (None, 24, 16) 0
flatten_39 (Flatten) (None, 384)           0
dense_77 (Dense)     (None, 64)            24640
dense_78 (Dense)     (None, 3)              195
=====
Total params: 27,059
Trainable params: 27,059
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/25
- 6s - loss: 0.3210 - acc: 0.8699 - val_loss: 0.3137 - val_acc: 0.8769

Epoch 2/25
- 2s - loss: 0.1933 - acc: 0.9250 - val_loss: 0.2982 - val_acc: 0.9058

Epoch 3/25
- 2s - loss: 0.1723 - acc: 0.9385 - val_loss: 0.3384 - val_acc: 0.9019

Epoch 4/25
- 2s - loss: 0.1462 - acc: 0.9454 - val_loss: 0.2764 - val_acc: 0.9173

Epoch 5/25
- 2s - loss: 0.1374 - acc: 0.9525 - val_loss: 0.2344 - val_acc: 0.9288

Epoch 6/25
- 2s - loss: 0.1172 - acc: 0.9609 - val_loss: 0.2544 - val_acc: 0.9154

Epoch 7/25
- 2s - loss: 0.1029 - acc: 0.9641 - val_loss: 0.2379 - val_acc: 0.9391

Epoch 8/25
- 2s - loss: 0.0987 - acc: 0.9698 - val_loss: 0.2354 - val_acc: 0.9487

Epoch 9/25
- 2s - loss: 0.0798 - acc: 0.9710 - val_loss: 0.2275 - val_acc: 0.9519

Epoch 10/25
- 2s - loss: 0.0775 - acc: 0.9734 - val_loss: 0.2279 - val_acc: 0.9462

Epoch 11/25
- 2s - loss: 0.0686 - acc: 0.9776 - val_loss: 0.2051 - val_acc: 0.9590

Epoch 12/25
- 2s - loss: 0.0580 - acc: 0.9786 - val_loss: 0.2847 - val_acc: 0.9429

Epoch 13/25
- 2s - loss: 0.0508 - acc: 0.9840 - val_loss: 0.2585 - val_acc: 0.9474

Epoch 14/25
- 2s - loss: 0.0686 - acc: 0.9821 - val_loss: 0.2862 - val_acc: 0.9391

Epoch 15/25
- 2s - loss: 0.0496 - acc: 0.9855 - val_loss: 0.3476 - val_acc: 0.9487

```

```
Epoch 16/25
- 2s - loss: 0.0447 - acc: 0.9867 - val_loss: 0.4362 - val_acc: 0.9250

Epoch 17/25
- 2s - loss: 0.0427 - acc: 0.9877 - val_loss: 0.3165 - val_acc: 0.9551

Epoch 18/25
- 2s - loss: 0.0344 - acc: 0.9889 - val_loss: 0.3581 - val_acc: 0.9545

Epoch 19/25
- 2s - loss: 0.0407 - acc: 0.9867 - val_loss: 0.3221 - val_acc: 0.9545

Epoch 20/25
- 2s - loss: 0.0441 - acc: 0.9907 - val_loss: 0.4478 - val_acc: 0.9353

Epoch 21/25
- 2s - loss: 0.0348 - acc: 0.9899 - val_loss: 0.6148 - val_acc: 0.9154

Epoch 22/25
- 2s - loss: 0.0291 - acc: 0.9921 - val_loss: 0.4584 - val_acc: 0.9353

Epoch 23/25
- 2s - loss: 0.0335 - acc: 0.9943 - val_loss: 0.5202 - val_acc: 0.9295

Epoch 24/25
- 2s - loss: 0.0285 - acc: 0.9939 - val_loss: 0.4747 - val_acc: 0.9474

Epoch 25/25
- 2s - loss: 0.0268 - acc: 0.9934 - val_loss: 0.4795 - val_acc: 0.9455
```

Test accuracy:  
0.9455128205128205  
Model: "sequential\_40"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_79 (Conv1D)	(None, 127, 16)	304
conv1d_80 (Conv1D)	(None, 124, 64)	4160
dropout_40 (Dropout)	(None, 124, 64)	0
max_pooling1d_40 (MaxPooling)	(None, 41, 64)	0
flatten_40 (Flatten)	(None, 2624)	0
dense_79 (Dense)	(None, 64)	168000
dense_80 (Dense)	(None, 3)	195
<hr/>		
Total params: 172,659		
Trainable params: 172,659		
Non-trainable params: 0		

---

Train on 4067 samples, validate on 1560 samples

Epoch 1/30

```
- 5s - loss: 10.5343 - acc: 0.3393 - val_loss: 11.0450 - val_acc: 0.3147
```

Epoch 2/30

```
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147
```

Epoch 3/30

```
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147
```

Epoch 4/30

```
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147
```

Epoch 5/30

```
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147
```

Epoch 6/30

```
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147
```

Epoch 7/30

```
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147
```

```
Epoch 8/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 9/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 10/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 11/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 12/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 13/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 14/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 15/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 16/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 17/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 18/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 19/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 20/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 21/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 22/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 23/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 24/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 25/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 26/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 27/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 28/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 29/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147

Epoch 30/30
- 1s - loss: 11.0215 - acc: 0.3162 - val_loss: 11.0450 - val_acc: 0.3147
```

Test accuracy:  
0.31474358974358974  
Model: "sequential\_41"

Layer (type)	Output Shape	Param #
conv1d_81 (Conv1D)	(None, 124, 32)	1472

conv1d_82 (Conv1D)	(None, 120, 16)	2576
dropout_41 (Dropout)	(None, 120, 16)	0
max_pooling1d_41 (MaxPooling1D)	(None, 60, 16)	0
flatten_41 (Flatten)	(None, 960)	0
dense_81 (Dense)	(None, 64)	61504
dense_82 (Dense)	(None, 3)	195
=====		
Total params:	65,747	
Trainable params:	65,747	
Non-trainable params:	0	

Train on 4067 samples, validate on 1560 samples  
Epoch 1/25  
- 5s - loss: 10.8879 - acc: 0.3162 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 2/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 3/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 4/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 5/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 6/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 7/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 8/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 9/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 10/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 11/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 12/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 13/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 14/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 15/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 16/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 17/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 18/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 19/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

Epoch 20/25  
- 1s - loss: 10.5420 - acc: 0.3460 - val\_loss: 10.5698 - val\_acc: 0.3442

```

Epoch 21/25
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 22/25
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 23/25
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 24/25
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Epoch 25/25
- 1s - loss: 10.5420 - acc: 0.3460 - val_loss: 10.5698 - val_acc: 0.3442

Test accuracy:
0.34423076923076923
Model: "sequential_42"

Layer (type)          Output Shape         Param #
=====
conv1d_83 (Conv1D)    (None, 125, 16)      592
conv1d_84 (Conv1D)    (None, 124, 32)       1056
dropout_42 (Dropout) (None, 124, 32)       0
max_pooling1d_42 (MaxPooling) (None, 24, 32) 0
flatten_42 (Flatten)  (None, 768)          0
dense_83 (Dense)     (None, 64)           49216
dense_84 (Dense)     (None, 3)            195
=====
Total params: 51,059
Trainable params: 51,059
Non-trainable params: 0

Train on 4067 samples, validate on 1560 samples
Epoch 1/30
- 6s - loss: 0.4582 - acc: 0.8547 - val_loss: 0.4028 - val_acc: 0.8526

Epoch 2/30
- 2s - loss: 0.2477 - acc: 0.9026 - val_loss: 0.4086 - val_acc: 0.8699

Epoch 3/30
- 2s - loss: 0.1990 - acc: 0.9257 - val_loss: 0.2891 - val_acc: 0.8949

Epoch 4/30
- 2s - loss: 0.2050 - acc: 0.9309 - val_loss: 0.2824 - val_acc: 0.9013

Epoch 5/30
- 2s - loss: 0.1806 - acc: 0.9422 - val_loss: 0.2500 - val_acc: 0.9167

Epoch 6/30
- 2s - loss: 0.1415 - acc: 0.9503 - val_loss: 0.2810 - val_acc: 0.8974

Epoch 7/30
- 2s - loss: 0.1590 - acc: 0.9538 - val_loss: 0.2188 - val_acc: 0.8974

Epoch 8/30
- 2s - loss: 0.1389 - acc: 0.9577 - val_loss: 0.1966 - val_acc: 0.9346

Epoch 9/30
- 2s - loss: 0.1210 - acc: 0.9614 - val_loss: 0.3629 - val_acc: 0.8609

Epoch 10/30
- 2s - loss: 0.1250 - acc: 0.9621 - val_loss: 0.2272 - val_acc: 0.9244

Epoch 11/30
- 2s - loss: 0.1145 - acc: 0.9663 - val_loss: 0.1876 - val_acc: 0.9500

Epoch 12/30
- 2s - loss: 0.1135 - acc: 0.9688 - val_loss: 0.2660 - val_acc: 0.8987

Epoch 13/30

```

```

- 2s - loss: 0.1116 - acc: 0.9715 - val_loss: 0.3926 - val_acc: 0.8949

Epoch 14/30
- 2s - loss: 0.0929 - acc: 0.9690 - val_loss: 0.2858 - val_acc: 0.9026

Epoch 15/30
- 2s - loss: 0.0955 - acc: 0.9727 - val_loss: 0.3468 - val_acc: 0.9199

Epoch 16/30
- 2s - loss: 0.1160 - acc: 0.9739 - val_loss: 0.2269 - val_acc: 0.9359

Epoch 17/30
- 2s - loss: 0.0906 - acc: 0.9757 - val_loss: 0.1793 - val_acc: 0.9583

Epoch 18/30
- 2s - loss: 0.1073 - acc: 0.9761 - val_loss: 0.1882 - val_acc: 0.9532

Epoch 19/30
- 2s - loss: 0.0656 - acc: 0.9776 - val_loss: 0.3183 - val_acc: 0.9237

Epoch 20/30
- 2s - loss: 0.0879 - acc: 0.9793 - val_loss: 0.2846 - val_acc: 0.9417

Epoch 21/30
- 2s - loss: 0.0763 - acc: 0.9781 - val_loss: 0.2388 - val_acc: 0.9333

Epoch 22/30
- 2s - loss: 0.0604 - acc: 0.9833 - val_loss: 0.2639 - val_acc: 0.9372

Epoch 23/30
- 2s - loss: 0.0950 - acc: 0.9798 - val_loss: 0.3902 - val_acc: 0.9269

Epoch 24/30
- 2s - loss: 0.0718 - acc: 0.9833 - val_loss: 0.2942 - val_acc: 0.9442

Epoch 25/30
- 2s - loss: 0.0911 - acc: 0.9838 - val_loss: 0.2685 - val_acc: 0.9519

Epoch 26/30
- 2s - loss: 0.0593 - acc: 0.9840 - val_loss: 0.2608 - val_acc: 0.9314

Epoch 27/30
- 2s - loss: 0.0867 - acc: 0.9840 - val_loss: 0.3129 - val_acc: 0.9301

Epoch 28/30
- 2s - loss: 0.0968 - acc: 0.9825 - val_loss: 0.3200 - val_acc: 0.9462

Epoch 29/30
- 2s - loss: 0.0830 - acc: 0.9845 - val_loss: 0.2658 - val_acc: 0.9500

Epoch 30/30
- 2s - loss: 0.0843 - acc: 0.9833 - val_loss: 0.3785 - val_acc: 0.9218

Test accuracy:
0.9217948717948717
Model: "sequential_43"

Layer (type)          Output Shape         Param #
=================================================================
conv1d_85 (Conv1D)    (None, 127, 64)      1216
=================================================================
conv1d_86 (Conv1D)    (None, 124, 64)       16448
=================================================================
dropout_43 (Dropout) (None, 124, 64)       0
=================================================================
max_pooling1d_43 (MaxPooling) (None, 41, 64)  0
=================================================================
flatten_43 (Flatten)  (None, 2624)        0
=================================================================
dense_85 (Dense)     (None, 32)           84000
=================================================================
dense_86 (Dense)     (None, 3)            99
=================================================================
Total params: 101,763
Trainable params: 101,763
Non-trainable params: 0

```

Train on 4067 samples, validate on 1560 samples

Epoch 1/25  
- 7s - loss: 0.3301 - acc: 0.8716 - val\_loss: 0.2819 - val\_acc: 0.8795

Epoch 2/25  
- 2s - loss: 0.1782 - acc: 0.9262 - val\_loss: 0.2483 - val\_acc: 0.9058

Epoch 3/25  
- 2s - loss: 0.1784 - acc: 0.9334 - val\_loss: 0.2418 - val\_acc: 0.8949

Epoch 4/25  
- 2s - loss: 0.1482 - acc: 0.9462 - val\_loss: 0.1935 - val\_acc: 0.9250

Epoch 5/25  
- 2s - loss: 0.1160 - acc: 0.9565 - val\_loss: 0.2323 - val\_acc: 0.8949

Epoch 6/25  
- 2s - loss: 0.1021 - acc: 0.9609 - val\_loss: 0.1814 - val\_acc: 0.9359

Epoch 7/25  
- 2s - loss: 0.0759 - acc: 0.9707 - val\_loss: 0.1952 - val\_acc: 0.9147

Epoch 8/25  
- 2s - loss: 0.0726 - acc: 0.9705 - val\_loss: 0.2272 - val\_acc: 0.9199

Epoch 9/25  
- 2s - loss: 0.0871 - acc: 0.9693 - val\_loss: 0.1908 - val\_acc: 0.9468

Epoch 10/25  
- 2s - loss: 0.0621 - acc: 0.9776 - val\_loss: 0.1812 - val\_acc: 0.9481

Epoch 11/25  
- 2s - loss: 0.0474 - acc: 0.9840 - val\_loss: 0.2082 - val\_acc: 0.9449

Epoch 12/25  
- 2s - loss: 0.0479 - acc: 0.9835 - val\_loss: 0.3598 - val\_acc: 0.9045

Epoch 13/25  
- 2s - loss: 0.0406 - acc: 0.9880 - val\_loss: 0.2543 - val\_acc: 0.9410

Epoch 14/25  
- 2s - loss: 0.0271 - acc: 0.9911 - val\_loss: 0.2227 - val\_acc: 0.9423

Epoch 15/25  
- 2s - loss: 0.0332 - acc: 0.9880 - val\_loss: 0.2556 - val\_acc: 0.9353

Epoch 16/25  
- 2s - loss: 0.0393 - acc: 0.9884 - val\_loss: 0.2661 - val\_acc: 0.9506

Epoch 17/25  
- 2s - loss: 0.0604 - acc: 0.9825 - val\_loss: 0.5138 - val\_acc: 0.9250

Epoch 18/25  
- 2s - loss: 0.1400 - acc: 0.9786 - val\_loss: 0.2488 - val\_acc: 0.9410

Epoch 19/25  
- 2s - loss: 0.0547 - acc: 0.9862 - val\_loss: 0.2391 - val\_acc: 0.9410

Epoch 20/25  
- 2s - loss: 0.0253 - acc: 0.9919 - val\_loss: 0.2691 - val\_acc: 0.9410

Epoch 21/25  
- 2s - loss: 0.0123 - acc: 0.9963 - val\_loss: 0.3142 - val\_acc: 0.9378

Epoch 22/25  
- 2s - loss: 0.0141 - acc: 0.9956 - val\_loss: 0.3008 - val\_acc: 0.9417

Epoch 23/25  
- 2s - loss: 0.0062 - acc: 0.9990 - val\_loss: 0.2694 - val\_acc: 0.9468

Epoch 24/25  
- 2s - loss: 0.0075 - acc: 0.9990 - val\_loss: 0.3238 - val\_acc: 0.9404

Epoch 25/25  
- 2s - loss: 0.0121 - acc: 0.9961 - val\_loss: 0.3154 - val\_acc: 0.9538

Test accuracy:

```
0.9538461538461539
100%|██████████| 40/40 [29:06<00:00, 50.14s/it, best loss: -0.9538461538461539]
```

In [40]:

```
X_train_static, Y_train_static , X_val_static, Y_val_static = data_static()

score, acc =best_model_static.evaluate(X_train_static,Y_train_static)
print("Train Accuracy:",acc)

score, acc =best_model_static.evaluate(X_val_static,Y_val_static)
print("Test Accuracy:",acc)
```

```
4067/4067 [=====] - 0s 106us/step
Train Accuracy: 0.9987705925743792
1560/1560 [=====] - 0s 100us/step
Test Accuracy: 0.9538461538461539
```

In [0]:

```
##saving static model
best_model_static.save(data_path + 'static_classification.h5')
```

### 3 class Classification of Dynamic activities

In [0]:

```
def _read_csv(filename):
    return pd.read_csv(filename, delim_whitespace=True, header=None)

def load_y_static(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 = data_path + f'y_{subset}.txt'
    y = _read_csv(filename)[0]
    y_subset = y<=3                         #boolean list of trues and false for dynamic
classification
    y = y[y_subset]
    return pd.get_dummies(y).as_matrix(),y_subset

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

    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"
    ]

    for signal in SIGNALS:
        filename = data_path+f'Inertial Signals {subset}/{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))

def load_data_dimmin():
```

```

def load_data_dynamic():
    """
    Obtain the dataset from multiple files.
    Returns: y_train, y_test
    """
    X_train, X_test = load_signals('train'), load_signals('test')
    y_train,y_train_subset = load_y_static('train')
    y_test,y_test_subset = load_y_static('test')
    X_train_d = X_train[y_train_subset]
    X_test_d = X_test[y_test_subset]

    ###Scaling data
    remove = int(X_train_d.shape[1] / 2)           #for eliminating the overlap
    temp_X = X_train_d[:, -remove:, :]
    # flatten data
    temp_X = temp_X.reshape((temp_X.shape[0] * temp_X.shape[1], temp_X.shape[2]))
    scale = StandardScaler()
    scale.fit(temp_X)
    #saving scale for future use
    pickle.dump(scale,open(data_path + 'Scale_dynamic.p','wb'))

    temp_X1 = X_train_d.reshape((X_train_d.shape[0] * X_train_d.shape[1], X_train_d.shape[2]))
    temp_X1 = scale.transform(temp_X1)
    X_train_d = temp_X1.reshape(X_train_d.shape)

    temp_X1 = X_test_d.reshape((X_test_d.shape[0] * X_test_d.shape[1], X_test_d.shape[2]))
    temp_X1 = scale.transform(temp_X1)
    X_test_d = temp_X1.reshape(X_test_d.shape)

    return X_train_d, y_train, X_test_d, y_test

```

In [44]:

```

# Loading the train and test data
X_train_dynamic, Y_train_dynamic, X_test_dynamic, Y_test_dynamic = load_data_dynamic()

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:36: FutureWarning: Method .as_matrix
will be removed in a future version. Use .values instead.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:15: FutureWarning: Method .as_matrix
will be removed in a future version. Use .values instead.
from ipykernel import kernelapp as app

```

In [0]:

```

np.save(data_path + 'X_train_dynamic.npy', X_train_dynamic) # save X_train_dynamic
np.save(data_path + 'X_test_dynamic.npy', X_test_dynamic) # save X_test_dynamic
np.save(data_path + 'Y_train_dynamic.npy', Y_train_dynamic) # save Y_train_dynamic
np.save(data_path + 'Y_test_dynamic.npy', Y_test_dynamic) # save Y_test_dynamic

```

In [0]:

```

def data_dynamic():
    data_path = '/content/drive/My Drive/AAIC/Case Studies/Human Activity Recognition/'
    X_train = np.load(data_path + 'X_train_dynamic.npy') # load X_train_dynamic
    Y_train = np.load(data_path + 'Y_train_dynamic.npy') # load Y_train_dynamic
    X_val = np.load(data_path + 'X_test_dynamic.npy') # load X_test_dynamic
    Y_val = np.load(data_path + 'Y_test_dynamic.npy') # load Y_test_dynamic
    return X_train, Y_train , X_val, Y_val

```

In [0]:

```

def create_model_dynamic(X_train, Y_train, X_val, Y_val):
    # Initialiazing the sequential model
    model = Sequential()

    model.add(Conv1D(filters={{choice([16,32,64])}}, kernel_size={{choice([2,3,4,5])}}),activation='relu',kernel_initializer='he_uniform',input_shape=(128,9))

    model.add(Conv1D(filters={{choice([16,32,64])}}, kernel_size={{choice([2,3,4,5])}}),activation='relu',kernel_initializer='he_uniform'))
    model.add(Dropout({{uniform(0,1)}}))
    model.add(MaxPooling1D(pool_size={{choice([2,3,5])}}))
    model.add(Flatten())

```

```

model.add(Dense({{choice([16,32,64])}}), activation='relu'))
model.add(Dense(3, activation='softmax'))

choiceval = {{choice(['adam', 'rmsprop'])}}

if choiceval == 'adam':
    optim = adam
else:
    optim = rmsprop

model.compile(loss='categorical_crossentropy', metrics=['accuracy'], optimizer=optim)
model.summary()

result = model.fit(X_train, Y_train,
                    batch_size={{choice([16,32,64])}},
                    epochs={{choice([25,30,35])}},
                    verbose=2,
                    validation_data=(X_val, Y_val))

score, acc = model.evaluate(X_val, Y_val, verbose=0)
print('Test accuracy:', acc)
return {'loss': -acc, 'status': STATUS_OK, 'model': model}

```

In [51]:

```

best_run_dynamic, best_model_dynamic = optim.minimize(model=create_model_dynamic,
                                                       data=data_dynamic,
                                                       algo=tpe.suggest,
                                                       max_evals=40,
                                                       trials=Trials(),
                                                       notebook_name= 'drive/My Drive/AAIC/Case Studies/Human Activity Recognition/Human Activity Recognition')

```

```

>>> Imports:
#coding=utf-8

try:
    from google.colab import drive
except:
    pass

try:
    import os
except:
    pass

try:
    import numpy as np
except:
    pass

try:
    import pandas as pd
except:
    pass

try:
    import matplotlib.pyplot as plt
except:
    pass

try:
    import seaborn as sns
except:
    pass

try:
    import numpy as np
except:
    pass

try:

```

```
    from sklearn.manifold import TSNE
except:
    pass

try:
    import matplotlib.pyplot as plt
except:
    pass

try:
    import seaborn as sns
except:
    pass

try:
    import itertools
except:
    pass

try:
    import numpy as np
except:
    pass

try:
    import matplotlib.pyplot as plt
except:
    pass

try:
    from sklearn.metrics import confusion_matrix
except:
    pass

try:
    from datetime import datetime
except:
    pass

try:
    from sklearn import linear_model
except:
    pass

try:
    from sklearn import metrics
except:
    pass

try:
    from sklearn.model_selection import GridSearchCV
except:
    pass

try:
    from sklearn.svm import SVC
except:
    pass

try:
    from sklearn.tree import DecisionTreeClassifier
except:
    pass

try:
    from sklearn.ensemble import RandomForestClassifier
except:
    pass

try:
    from sklearn.ensemble import GradientBoostingClassifier
except:
    pass

try:
    import tensorflow as tf
except:
```

```
pass

try:
    from keras import backend as K
except:
    pass

try:
    from keras.models import Sequential
except:
    pass

try:
    from keras.layers import LSTM
except:
    pass

try:
    from keras.layers.core import Dense, Dropout
except:
    pass

try:
    from keras.layers import Flatten
except:
    pass

try:
    from hyperopt import Trials, STATUS_OK, tpe
except:
    pass

try:
    from hyperas import optim
except:
    pass

try:
    from hyperas.distributions import choice, uniform
except:
    pass

try:
    import keras
except:
    pass

try:
    from keras.layers import Conv1D
except:
    pass

try:
    from keras.layers import BatchNormalization
except:
    pass

try:
    from keras.layers import MaxPooling1D
except:
    pass

try:
    from sklearn.preprocessing import StandardScaler
except:
    pass

try:
    import pickle
except:
    pass

>>> Hyperas search space:

def get_space():
    return {
        'filters': hp.choice('filters', [16,32,64]),

```

```

'kernel_size': hp.choice('kernel_size', [2,3,4,5]),
'filters_1': hp.choice('filters_1', [16,32,64]),
'kernel_size_1': hp.choice('kernel_size_1', [2,3,4,5]),
'Dropout': hp.uniform('Dropout', 0,1),
'pool_size': hp.choice('pool_size', [2,3,5]),
'filters_2': hp.choice('filters_2', [16,32,64]),
'lr': hp.choice('lr', [10**-3, 10**-2, 10**-1]),
'lr_1': hp.choice('lr_1', [10**-3, 10**-2, 10**-1]),
'choiceval': hp.choice('choiceval', ['adam', 'rmsprop']),
'filters_3': hp.choice('filters_3', [16,32,64]),
'epochs': hp.choice('epochs', [25,30,35]),
}

>>> Data
1:
2: data_path = '/content/drive/My Drive/AAIC/Case Studies/Human Activity Recognition/'
3: X_train = np.load(data_path + 'X_train_dynamic.npy') # load X_train_dynamic
4: Y_train = np.load(data_path + 'Y_train_dynamic.npy') # load Y_train_dynamic
5: X_val = np.load(data_path + 'X_test_dynamic.npy') # load X_test_dynamic
6: Y_val = np.load(data_path + 'Y_test_dynamic.npy') # load Y_test_dynamic
7:
8:
9:
>>> Resulting replaced keras model:

1: def keras_fmin_fnct(space):
2:
3:     # Initilazing the sequential model
4:     model = Sequential()
5:
6:     model.add(Conv1D(filters=space['filters'],
kernel_size=space['kernel_size'],activation='relu',kernel_initializer='he_uniform',input_shape=(12
8,9)))
7:
8:     model.add(Conv1D(filters=space['filters_1'],
kernel_size=space['kernel_size_1'],activation='relu',kernel_initializer='he_uniform'))
9:     model.add(Dropout(space['Dropout']))
10:    model.add(MaxPooling1D(pool_size=space['pool_size']))
11:    model.add(Flatten())
12:    model.add(Dense(space['filters_2'], activation='relu'))
13:    model.add(Dense(3, activation='softmax'))
14:
15:    adam = keras.optimizers.Adam(lr=space['lr'])
16:    rmsprop = keras.optimizers.RMSprop(lr=space['lr_1'])
17:
18:    choiceval = space['choiceval']
19:
20:    if choiceval == 'adam':
21:        optim = adam
22:    else:
23:        optim = rmsprop
24:
25:    model.compile(loss='categorical_crossentropy', metrics=['accuracy'],optimizer=optim)
26:    model.summary()
27:
28:    result = model.fit(X_train, Y_train,
29:                        batch_size=space['filters_3'],
30:                        epochs=space['epochs'],
31:                        verbose=2,
32:                        validation_data=(X_val, Y_val))
33:
34:    score, acc = model.evaluate(X_val, Y_val, verbose=0)
35:    print('Test accuracy:', acc)
36:    return {'loss': -acc, 'status': STATUS_OK, 'model': model}
37:
Model: "sequential_44"

Layer (type)                  Output Shape                 Param #
=================================================================
conv1d_87 (Conv1D)           (None, 124, 64)            2944
conv1d_88 (Conv1D)           (None, 121, 32)             8224
dropout_44 (Dropout)          (None, 121, 32)              0
max_pooling1d_44 (MaxPooling (None, 24, 32)                0

```

flatten_44 (Flatten)	(None, 768)	0
dense_87 (Dense)	(None, 32)	24608
dense_88 (Dense)	(None, 3)	99
=====		
Total params: 35,875		
Trainable params: 35,875		
Non-trainable params: 0		

Train on 3285 samples, validate on 1387 samples  
Epoch 1/35  
- 6s - loss: 0.4978 - acc: 0.8061 - val\_loss: 0.2727 - val\_acc: 0.9178

Epoch 2/35  
- 1s - loss: 0.0235 - acc: 0.9948 - val\_loss: 0.1643 - val\_acc: 0.9611

Epoch 3/35  
- 1s - loss: 0.0040 - acc: 1.0000 - val\_loss: 0.1460 - val\_acc: 0.9683

Epoch 4/35  
- 1s - loss: 0.0018 - acc: 1.0000 - val\_loss: 0.1417 - val\_acc: 0.9690

Epoch 5/35  
- 1s - loss: 8.6119e-04 - acc: 1.0000 - val\_loss: 0.1479 - val\_acc: 0.9683

Epoch 6/35  
- 1s - loss: 6.1076e-04 - acc: 1.0000 - val\_loss: 0.1373 - val\_acc: 0.9704

Epoch 7/35  
- 1s - loss: 4.3168e-04 - acc: 1.0000 - val\_loss: 0.1477 - val\_acc: 0.9697

Epoch 8/35  
- 1s - loss: 3.3482e-04 - acc: 1.0000 - val\_loss: 0.1576 - val\_acc: 0.9697

Epoch 9/35  
- 1s - loss: 2.7188e-04 - acc: 1.0000 - val\_loss: 0.1433 - val\_acc: 0.9712

Epoch 10/35  
- 1s - loss: 2.3529e-04 - acc: 1.0000 - val\_loss: 0.1516 - val\_acc: 0.9712

Epoch 11/35  
- 1s - loss: 1.8107e-04 - acc: 1.0000 - val\_loss: 0.1551 - val\_acc: 0.9704

Epoch 12/35  
- 1s - loss: 1.4789e-04 - acc: 1.0000 - val\_loss: 0.1573 - val\_acc: 0.9704

Epoch 13/35  
- 1s - loss: 1.2996e-04 - acc: 1.0000 - val\_loss: 0.1554 - val\_acc: 0.9712

Epoch 14/35  
- 1s - loss: 1.0752e-04 - acc: 1.0000 - val\_loss: 0.1664 - val\_acc: 0.9697

Epoch 15/35  
- 1s - loss: 9.5217e-05 - acc: 1.0000 - val\_loss: 0.1732 - val\_acc: 0.9697

Epoch 16/35  
- 1s - loss: 8.3861e-05 - acc: 1.0000 - val\_loss: 0.1609 - val\_acc: 0.9719

Epoch 17/35  
- 1s - loss: 7.1138e-05 - acc: 1.0000 - val\_loss: 0.1629 - val\_acc: 0.9719

Epoch 18/35  
- 1s - loss: 6.3519e-05 - acc: 1.0000 - val\_loss: 0.1632 - val\_acc: 0.9719

Epoch 19/35  
- 1s - loss: 5.4995e-05 - acc: 1.0000 - val\_loss: 0.1746 - val\_acc: 0.9704

Epoch 20/35  
- 1s - loss: 5.1423e-05 - acc: 1.0000 - val\_loss: 0.1685 - val\_acc: 0.9719

Epoch 21/35  
- 1s - loss: 4.3550e-05 - acc: 1.0000 - val\_loss: 0.1712 - val\_acc: 0.9719

Epoch 22/35  
- 1s - loss: 4.2293e-05 - acc: 1.0000 - val\_loss: 0.1727 - val\_acc: 0.9719

```

Epoch 23/35
- 1s - loss: 3.6338e-05 - acc: 1.0000 - val_loss: 0.1763 - val_acc: 0.9712

Epoch 24/35
- 1s - loss: 3.2257e-05 - acc: 1.0000 - val_loss: 0.1774 - val_acc: 0.9719

Epoch 25/35
- 1s - loss: 3.1545e-05 - acc: 1.0000 - val_loss: 0.1764 - val_acc: 0.9726

Epoch 26/35
- 1s - loss: 2.7265e-05 - acc: 1.0000 - val_loss: 0.1829 - val_acc: 0.9712

Epoch 27/35
- 1s - loss: 2.5085e-05 - acc: 1.0000 - val_loss: 0.1790 - val_acc: 0.9719

Epoch 28/35
- 1s - loss: 2.3366e-05 - acc: 1.0000 - val_loss: 0.1775 - val_acc: 0.9726

Epoch 29/35
- 1s - loss: 2.1463e-05 - acc: 1.0000 - val_loss: 0.1866 - val_acc: 0.9719

Epoch 30/35
- 1s - loss: 1.9544e-05 - acc: 1.0000 - val_loss: 0.1856 - val_acc: 0.9719

Epoch 31/35
- 1s - loss: 1.7654e-05 - acc: 1.0000 - val_loss: 0.1925 - val_acc: 0.9719

Epoch 32/35
- 1s - loss: 1.6982e-05 - acc: 1.0000 - val_loss: 0.1889 - val_acc: 0.9719

Epoch 33/35
- 1s - loss: 1.5070e-05 - acc: 1.0000 - val_loss: 0.1932 - val_acc: 0.9726

Epoch 34/35
- 1s - loss: 1.4318e-05 - acc: 1.0000 - val_loss: 0.1913 - val_acc: 0.9726

Epoch 35/35
- 1s - loss: 1.3502e-05 - acc: 1.0000 - val_loss: 0.1972 - val_acc: 0.9719

Test accuracy:
0.9718817591925017
Model: "sequential_45"

Layer (type)          Output Shape         Param #
=====
conv1d_89 (Conv1D)    (None, 125, 64)      2368
conv1d_90 (Conv1D)    (None, 124, 64)      8256
dropout_45 (Dropout) (None, 124, 64)      0
max_pooling1d_45 (MaxPooling) (None, 62, 64) 0
flatten_45 (Flatten)  (None, 3968)        0
dense_89 (Dense)     (None, 64)           254016
dense_90 (Dense)     (None, 3)            195
=====
Total params: 264,835
Trainable params: 264,835
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples
Epoch 1/30
- 6s - loss: 10.0517 - acc: 0.3705 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 2/30
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 3/30
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 4/30
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 5/30

```



Test accuracy:  
0.357606344628695  
Model: "sequential\_46"

Layer (type)	Output Shape	Param #
conv1d_91 (Conv1D)	(None, 124, 32)	1472
conv1d_92 (Conv1D)	(None, 123, 16)	1040
dropout_46 (Dropout)	(None, 123, 16)	0
max_pooling1d_46 (MaxPooling)	(None, 41, 16)	0
flatten_46 (Flatten)	(None, 656)	0
dense_91 (Dense)	(None, 64)	42048
dense_92 (Dense)	(None, 3)	195

Total params: 44,755  
Trainable params: 44,755  
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples  
Epoch 1/35  
- 6s - loss: 10.0639 - acc: 0.3717 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 2/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 3/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 4/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 5/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 6/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 7/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 8/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 9/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 10/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 11/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 12/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 13/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 14/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 15/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 16/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 17/35  
- 1s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 18/35

```
--Epoch 10/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 19/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 20/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 21/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 22/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 23/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 24/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 25/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 26/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 27/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 28/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 29/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 30/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 31/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 32/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 33/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 34/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

```
Epoch 35/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Test accuracy:

0.357606344628695

Model: "sequential\_47"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_93 (Conv1D)	(None, 125, 16)	592
conv1d_94 (Conv1D)	(None, 123, 16)	784
dropout_47 (Dropout)	(None, 123, 16)	0
max_pooling1d_47 (MaxPooling)	(None, 41, 16)	0
flatten_47 (Flatten)	(None, 656)	0
dense_93 (Dense)	(None, 64)	42048
dense_94 (Dense)	(None, 3)	195
<hr/>		

Total params: 43,619

Trainable params: 43,619

Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples

Epoch 1/35  
- 7s - loss: 0.8113 - acc: 0.7425 - val\_loss: 0.2245 - val\_acc: 0.9438

Epoch 2/35  
- 2s - loss: 0.2677 - acc: 0.9272 - val\_loss: 0.3168 - val\_acc: 0.9229

Epoch 3/35  
- 1s - loss: 0.2233 - acc: 0.9458 - val\_loss: 0.1181 - val\_acc: 0.9647

Epoch 4/35  
- 2s - loss: 0.2604 - acc: 0.9592 - val\_loss: 0.2623 - val\_acc: 0.9503

Epoch 5/35  
- 2s - loss: 0.2218 - acc: 0.9635 - val\_loss: 0.5858 - val\_acc: 0.9019

Epoch 6/35  
- 2s - loss: 0.2328 - acc: 0.9613 - val\_loss: 0.3216 - val\_acc: 0.9466

Epoch 7/35  
- 2s - loss: 0.2808 - acc: 0.9683 - val\_loss: 0.3809 - val\_acc: 0.9539

Epoch 8/35  
- 2s - loss: 0.3279 - acc: 0.9693 - val\_loss: 1.0124 - val\_acc: 0.9142

Epoch 9/35  
- 2s - loss: 0.3379 - acc: 0.9717 - val\_loss: 0.4822 - val\_acc: 0.9531

Epoch 10/35  
- 2s - loss: 0.3947 - acc: 0.9659 - val\_loss: 0.6618 - val\_acc: 0.9430

Epoch 11/35  
- 2s - loss: 0.4371 - acc: 0.9662 - val\_loss: 1.5789 - val\_acc: 0.8919

Epoch 12/35  
- 2s - loss: 0.5026 - acc: 0.9632 - val\_loss: 1.1171 - val\_acc: 0.9193

Epoch 13/35  
- 1s - loss: 0.3734 - acc: 0.9735 - val\_loss: 0.9494 - val\_acc: 0.9250

Epoch 14/35  
- 2s - loss: 0.4191 - acc: 0.9705 - val\_loss: 0.7368 - val\_acc: 0.9409

Epoch 15/35  
- 1s - loss: 0.5129 - acc: 0.9632 - val\_loss: 1.2764 - val\_acc: 0.9156

Epoch 16/35  
- 2s - loss: 0.5178 - acc: 0.9619 - val\_loss: 0.8054 - val\_acc: 0.9459

Epoch 17/35  
- 2s - loss: 0.3410 - acc: 0.9750 - val\_loss: 0.6424 - val\_acc: 0.9524

Epoch 18/35  
- 2s - loss: 0.4055 - acc: 0.9729 - val\_loss: 1.2129 - val\_acc: 0.9178

Epoch 19/35  
- 1s - loss: 0.4331 - acc: 0.9708 - val\_loss: 1.1350 - val\_acc: 0.9243

Epoch 20/35  
- 2s - loss: 0.4415 - acc: 0.9705 - val\_loss: 1.9048 - val\_acc: 0.8702

Epoch 21/35  
- 2s - loss: 0.3801 - acc: 0.9753 - val\_loss: 1.1412 - val\_acc: 0.9265

Epoch 22/35  
- 1s - loss: 0.8533 - acc: 0.9458 - val\_loss: 1.2243 - val\_acc: 0.9193

Epoch 23/35  
- 2s - loss: 0.6226 - acc: 0.9598 - val\_loss: 1.4897 - val\_acc: 0.9034

Epoch 24/35  
- 2s - loss: 0.5567 - acc: 0.9644 - val\_loss: 0.9158 - val\_acc: 0.9402

Epoch 25/35  
- 1s - loss: 0.6689 - acc: 0.9571 - val\_loss: 1.5351 - val\_acc: 0.9019

Epoch 26/35

```

Epoch 26/35
- 2s - loss: 0.8961 - acc: 0.9428 - val_loss: 1.8314 - val_acc: 0.8810

Epoch 27/35
- 2s - loss: 0.6738 - acc: 0.9556 - val_loss: 1.0384 - val_acc: 0.9329

Epoch 28/35
- 1s - loss: 0.4376 - acc: 0.9720 - val_loss: 1.5009 - val_acc: 0.9041

Epoch 29/35
- 2s - loss: 0.6702 - acc: 0.9580 - val_loss: 0.5616 - val_acc: 0.9632

Epoch 30/35
- 2s - loss: 0.4808 - acc: 0.9693 - val_loss: 1.1587 - val_acc: 0.9229

Epoch 31/35
- 2s - loss: 0.7225 - acc: 0.9540 - val_loss: 0.8179 - val_acc: 0.9459

Epoch 32/35
- 2s - loss: 0.6766 - acc: 0.9568 - val_loss: 1.3108 - val_acc: 0.9156

Epoch 33/35
- 2s - loss: 0.8622 - acc: 0.9455 - val_loss: 1.2632 - val_acc: 0.9178

Epoch 34/35
- 2s - loss: 0.4833 - acc: 0.9696 - val_loss: 1.2563 - val_acc: 0.9185

Epoch 35/35
- 1s - loss: 0.7607 - acc: 0.9525 - val_loss: 1.8380 - val_acc: 0.8825

```

Test accuracy:  
0.882480173035328  
Model: "sequential\_48"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_95 (Conv1D)	(None, 124, 32)	1472
conv1d_96 (Conv1D)	(None, 122, 16)	1552
dropout_48 (Dropout)	(None, 122, 16)	0
max_pooling1d_48 (MaxPooling)	(None, 61, 16)	0
flatten_48 (Flatten)	(None, 976)	0
dense_95 (Dense)	(None, 64)	62528
dense_96 (Dense)	(None, 3)	195
<hr/>		

Total params: 65,747  
Trainable params: 65,747  
Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/30

```

- 6s - loss: 11.1930 - acc: 0.2959 - val_loss: 11.2373 - val_acc: 0.3028

```

Epoch 2/30

```

- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

```

Epoch 3/30

```

- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

```

Epoch 4/30

```

- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

```

Epoch 5/30

```

- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

```

Epoch 6/30

```

- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

```

Epoch 7/30

```

- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

```

Epoch 8/30

```

- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

```

0s 1000. 11 2000 0 2000 1000. 11 2000 0 2000

```

- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 9/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 10/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 11/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 12/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 13/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 14/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 15/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 16/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 17/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 18/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 19/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 20/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 21/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 22/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 23/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 24/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 25/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 26/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 27/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 28/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 29/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
Epoch 30/30
- 0s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Test accuracy:
0.3028118241022367
Model: "sequential_49"

```

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_97 (Conv1D)	(None, 124, 64)	2944
<hr/>		
conv1d_98 (Conv1D)	(None, 121, 16)	4112

dropout_49 (Dropout)	(None, 121, 16)	0
max_pooling1d_49 (MaxPooling)	(None, 24, 16)	0
flatten_49 (Flatten)	(None, 384)	0
dense_97 (Dense)	(None, 32)	12320
dense_98 (Dense)	(None, 3)	99
=====		
Total params:	19,475	
Trainable params:	19,475	
Non-trainable params:	0	

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/30  
- 7s - loss: 10.6614 - acc: 0.3373 - val\_loss: 10.6679 - val\_acc: 0.3381

Epoch 2/30  
- 2s - loss: 10.7699 - acc: 0.3318 - val\_loss: 10.6679 - val\_acc: 0.3381

Epoch 3/30  
- 2s - loss: 10.7356 - acc: 0.3339 - val\_loss: 10.6679 - val\_acc: 0.3381

Epoch 4/30  
- 2s - loss: 10.7748 - acc: 0.3315 - val\_loss: 10.6679 - val\_acc: 0.3381

Epoch 5/30  
- 2s - loss: 10.6816 - acc: 0.3373 - val\_loss: 10.6679 - val\_acc: 0.3381

Epoch 6/30  
- 2s - loss: 10.7994 - acc: 0.3300 - val\_loss: 10.6679 - val\_acc: 0.3381

Epoch 7/30  
- 2s - loss: 10.5589 - acc: 0.3449 - val\_loss: 10.6679 - val\_acc: 0.3381

Epoch 8/30  
- 2s - loss: 10.8582 - acc: 0.3263 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 9/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 10/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 11/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 12/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 13/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 14/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 15/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 16/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 17/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 18/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 19/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 20/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

Epoch 21/30  
- 2s - loss: 10.8533 - acc: 0.3266 - val\_loss: 10.6447 - val\_acc: 0.3396

---

^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^

```

- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 22/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 23/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 24/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 25/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 26/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 27/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 28/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 29/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 30/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Test accuracy:
0.3395818315054241
Model: "sequential_50"

Layer (type)          Output Shape         Param #
=================================================================
conv1d_99 (Conv1D)    (None, 124, 16)      736
conv1d_100 (Conv1D)   (None, 121, 32)       2080
dropout_50 (Dropout)  (None, 121, 32)       0
max_pooling1d_50 (MaxPooling) (None, 60, 32)  0
flatten_50 (Flatten)  (None, 1920)        0
dense_99 (Dense)     (None, 32)           61472
dense_100 (Dense)    (None, 3)            99
=====
Total params: 64,387
Trainable params: 64,387
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples
Epoch 1/35
- 7s - loss: 0.4883 - acc: 0.8064 - val_loss: 0.3565 - val_acc: 0.8695

Epoch 2/35
- 2s - loss: 0.0493 - acc: 0.9839 - val_loss: 0.2268 - val_acc: 0.9416

Epoch 3/35
- 2s - loss: 0.0248 - acc: 0.9927 - val_loss: 0.2851 - val_acc: 0.9423

Epoch 4/35
- 2s - loss: 0.0089 - acc: 0.9967 - val_loss: 0.5397 - val_acc: 0.8998

Epoch 5/35
- 2s - loss: 0.0104 - acc: 0.9973 - val_loss: 0.4336 - val_acc: 0.9387

Epoch 6/35
- 2s - loss: 0.0030 - acc: 0.9991 - val_loss: 0.3702 - val_acc: 0.9503

Epoch 7/35
- 2s - loss: 0.0048 - acc: 0.9982 - val_loss: 0.3721 - val_acc: 0.9582

Epoch 8/35
- 2s - loss: 0.0034 - acc: 0.9985 - val_loss: 0.4221 - val_acc: 0.9394

```

```
Epoch 9/35
- 2s - loss: 0.0018 - acc: 0.9997 - val_loss: 0.4074 - val_acc: 0.9452

Epoch 10/35
- 2s - loss: 0.0033 - acc: 0.9991 - val_loss: 0.4920 - val_acc: 0.9510

Epoch 11/35
- 2s - loss: 0.0027 - acc: 0.9991 - val_loss: 0.4445 - val_acc: 0.9466

Epoch 12/35
- 2s - loss: 0.0047 - acc: 0.9988 - val_loss: 0.4414 - val_acc: 0.9452

Epoch 13/35
- 2s - loss: 0.0023 - acc: 0.9991 - val_loss: 0.3991 - val_acc: 0.9625

Epoch 14/35
- 2s - loss: 0.0019 - acc: 0.9997 - val_loss: 0.4740 - val_acc: 0.9481

Epoch 15/35
- 2s - loss: 0.0048 - acc: 0.9988 - val_loss: 0.5276 - val_acc: 0.9510

Epoch 16/35
- 2s - loss: 0.0025 - acc: 0.9991 - val_loss: 0.5167 - val_acc: 0.9380

Epoch 17/35
- 2s - loss: 0.0029 - acc: 0.9997 - val_loss: 0.5331 - val_acc: 0.9560

Epoch 18/35
- 2s - loss: 0.0017 - acc: 0.9994 - val_loss: 0.4649 - val_acc: 0.9632

Epoch 19/35
- 2s - loss: 9.0610e-04 - acc: 0.9997 - val_loss: 0.4664 - val_acc: 0.9553

Epoch 20/35
- 2s - loss: 3.9084e-05 - acc: 1.0000 - val_loss: 0.4709 - val_acc: 0.9474

Epoch 21/35
- 2s - loss: 5.3738e-04 - acc: 0.9997 - val_loss: 0.4958 - val_acc: 0.9488

Epoch 22/35
- 2s - loss: 0.0040 - acc: 0.9991 - val_loss: 0.5521 - val_acc: 0.9430

Epoch 23/35
- 2s - loss: 1.3702e-04 - acc: 1.0000 - val_loss: 0.5429 - val_acc: 0.9452

Epoch 24/35
- 2s - loss: 1.4376e-06 - acc: 1.0000 - val_loss: 0.6310 - val_acc: 0.9394

Epoch 25/35
- 2s - loss: 0.0044 - acc: 0.9991 - val_loss: 0.5280 - val_acc: 0.9503

Epoch 26/35
- 2s - loss: 0.0019 - acc: 0.9997 - val_loss: 0.5767 - val_acc: 0.9466

Epoch 27/35
- 2s - loss: 0.0018 - acc: 0.9997 - val_loss: 0.5291 - val_acc: 0.9531

Epoch 28/35
- 2s - loss: 2.5490e-07 - acc: 1.0000 - val_loss: 1.0195 - val_acc: 0.8904

Epoch 29/35
- 2s - loss: 0.0024 - acc: 0.9991 - val_loss: 0.5087 - val_acc: 0.9582

Epoch 30/35
- 2s - loss: 4.3033e-06 - acc: 1.0000 - val_loss: 0.5709 - val_acc: 0.9495

Epoch 31/35
- 2s - loss: 5.1196e-05 - acc: 1.0000 - val_loss: 0.5390 - val_acc: 0.9524

Epoch 32/35
- 2s - loss: 2.7047e-06 - acc: 1.0000 - val_loss: 0.6054 - val_acc: 0.9416

Epoch 33/35
- 2s - loss: 3.1454e-05 - acc: 1.0000 - val_loss: 0.5516 - val_acc: 0.9553

Epoch 34/35
- 2s - loss: 1.1196e-05 - acc: 1.0000 - val_loss: 0.5516 - val_acc: 0.9553
```

```
- 2s - loss: 1.3173e-07 - acc: 1.0000 - val_loss: 0.5942 - val_acc: 0.9459
```

Epoch 35/35

```
- 2s - loss: 0.0019 - acc: 0.9997 - val_loss: 0.5441 - val_acc: 0.9438
```

Test accuracy:

0.9437635183850036

Model: "sequential\_51"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_101 (Conv1D)	(None, 125, 16)	592
conv1d_102 (Conv1D)	(None, 122, 64)	4160
dropout_51 (Dropout)	(None, 122, 64)	0
max_pooling1d_51 (MaxPooling)	(None, 24, 64)	0
flatten_51 (Flatten)	(None, 1536)	0
dense_101 (Dense)	(None, 32)	49184
dense_102 (Dense)	(None, 3)	99
<hr/>		

Total params: 54,035

Trainable params: 54,035

Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/30

```
- 7s - loss: 10.8115 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 2/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 3/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 4/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 5/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 6/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 7/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 8/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 9/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 10/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 11/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 12/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 13/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 14/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 15/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

Epoch 16/30

```
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396
```

```

Epoch 17/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 18/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 19/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 20/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 21/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 22/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 23/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 24/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 25/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 26/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 27/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 28/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 29/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Epoch 30/30
- 2s - loss: 10.8533 - acc: 0.3266 - val_loss: 10.6447 - val_acc: 0.3396

Test accuracy:
0.3395818315054241
Model: "sequential_52"

Layer (type)          Output Shape         Param #
=====
conv1d_103 (Conv1D)    (None, 126, 32)      896
conv1d_104 (Conv1D)    (None, 122, 16)       2576
dropout_52 (Dropout)   (None, 122, 16)       0
max_pooling1d_52 (MaxPooling) (None, 61, 16)  0
flatten_52 (Flatten)   (None, 976)           0
dense_103 (Dense)     (None, 32)            31264
dense_104 (Dense)     (None, 3)              99
=====
Total params: 34,835
Trainable params: 34,835
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples
Epoch 1/30
- 6s - loss: 9.9927 - acc: 0.3714 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 2/30
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 3/30
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

```



Epoch 30/30

- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Test accuracy:

0.357606344628695

Model: "sequential\_53"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_105 (Conv1D)	(None, 125, 32)	1184
conv1d_106 (Conv1D)	(None, 124, 64)	4160
dropout_53 (Dropout)	(None, 124, 64)	0
max_pooling1d_53 (MaxPooling)	(None, 41, 64)	0
flatten_53 (Flatten)	(None, 2624)	0
dense_105 (Dense)	(None, 64)	168000
dense_106 (Dense)	(None, 3)	195
<hr/>		

Total params: 173,539

Trainable params: 173,539

Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/25

- 7s - loss: 5.1909 - acc: 0.5358 - val\_loss: 0.5431 - val\_acc: 0.7686

Epoch 2/25

- 2s - loss: 0.0857 - acc: 0.9702 - val\_loss: 0.3579 - val\_acc: 0.9019

Epoch 3/25

- 2s - loss: 0.0233 - acc: 0.9921 - val\_loss: 0.1574 - val\_acc: 0.9445

Epoch 4/25

- 2s - loss: 0.0108 - acc: 0.9960 - val\_loss: 0.2173 - val\_acc: 0.9466

Epoch 5/25

- 2s - loss: 0.0049 - acc: 0.9985 - val\_loss: 0.3071 - val\_acc: 0.9459

Epoch 6/25

- 2s - loss: 0.0064 - acc: 0.9982 - val\_loss: 0.2743 - val\_acc: 0.9517

Epoch 7/25

- 2s - loss: 0.0049 - acc: 0.9985 - val\_loss: 0.4136 - val\_acc: 0.9416

Epoch 8/25

- 2s - loss: 4.6925e-05 - acc: 1.0000 - val\_loss: 0.3826 - val\_acc: 0.9445

Epoch 9/25

- 2s - loss: 0.0011 - acc: 0.9997 - val\_loss: 0.3341 - val\_acc: 0.9474

Epoch 10/25

- 2s - loss: 0.0052 - acc: 0.9985 - val\_loss: 0.3756 - val\_acc: 0.9517

Epoch 11/25

- 2s - loss: 1.6328e-04 - acc: 1.0000 - val\_loss: 0.3823 - val\_acc: 0.9495

Epoch 12/25

- 2s - loss: 0.0021 - acc: 0.9997 - val\_loss: 0.3724 - val\_acc: 0.9452

Epoch 13/25

- 2s - loss: 1.1995e-04 - acc: 1.0000 - val\_loss: 0.3103 - val\_acc: 0.9589

Epoch 14/25

- 2s - loss: 3.8604e-04 - acc: 1.0000 - val\_loss: 0.4412 - val\_acc: 0.9416

Epoch 15/25

- 2s - loss: 6.8220e-05 - acc: 1.0000 - val\_loss: 0.3831 - val\_acc: 0.9524

Epoch 16/25

- 2s - loss: 0.0013 - acc: 0.9997 - val\_loss: 0.4131 - val\_acc: 0.9481

```
Epoch 17/25
- 2s - loss: 2.0236e-04 - acc: 1.0000 - val_loss: 0.3926 - val_acc: 0.9510

Epoch 18/25
- 2s - loss: 8.8953e-05 - acc: 1.0000 - val_loss: 0.4052 - val_acc: 0.9445

Epoch 19/25
- 2s - loss: 0.0022 - acc: 0.9991 - val_loss: 0.3975 - val_acc: 0.9503

Epoch 20/25
- 2s - loss: 7.8336e-06 - acc: 1.0000 - val_loss: 0.5373 - val_acc: 0.9344

Epoch 21/25
- 2s - loss: 0.0025 - acc: 0.9997 - val_loss: 0.4019 - val_acc: 0.9517

Epoch 22/25
- 2s - loss: 9.4612e-05 - acc: 1.0000 - val_loss: 0.4253 - val_acc: 0.9503

Epoch 23/25
- 2s - loss: 3.7265e-04 - acc: 0.9997 - val_loss: 0.5297 - val_acc: 0.9402

Epoch 24/25
- 2s - loss: 2.3664e-06 - acc: 1.0000 - val_loss: 0.4476 - val_acc: 0.9546

Epoch 25/25
- 2s - loss: 6.9779e-04 - acc: 0.9997 - val_loss: 0.6682 - val_acc: 0.9293
```

Test accuracy:  
0.9293439077144917  
Model: "sequential\_54"

Layer (type)	Output Shape	Param #
conv1d_107 (Conv1D)	(None, 126, 32)	896
conv1d_108 (Conv1D)	(None, 125, 32)	2080
dropout_54 (Dropout)	(None, 125, 32)	0
max_pooling1d_54 (MaxPooling)	(None, 41, 32)	0
flatten_54 (Flatten)	(None, 1312)	0
dense_107 (Dense)	(None, 64)	84032
dense_108 (Dense)	(None, 3)	195

Total params: 87,203  
Trainable params: 87,203  
Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples  
Epoch 1/25  
- 7s - loss: 9.9376 - acc: 0.3729 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 2/25  
- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 3/25  
- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 4/25  
- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 5/25  
- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 6/25  
- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 7/25  
- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 8/25  
- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 9/25

```

- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 10/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 11/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 12/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 13/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 14/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 15/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 16/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 17/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 18/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 19/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 20/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 21/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 22/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 23/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 24/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 25/25
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Test accuracy:
0.357606344628695
Model: "sequential_55"



| Layer (type)                                                           | Output Shape    | Param # |
|------------------------------------------------------------------------|-----------------|---------|
| <hr/>                                                                  |                 |         |
| conv1d_109 (Conv1D)                                                    | (None, 124, 64) | 2944    |
| <hr/>                                                                  |                 |         |
| conv1d_110 (Conv1D)                                                    | (None, 121, 32) | 8224    |
| <hr/>                                                                  |                 |         |
| dropout_55 (Dropout)                                                   | (None, 121, 32) | 0       |
| <hr/>                                                                  |                 |         |
| max_pooling1d_55 (MaxPooling)                                          | (None, 40, 32)  | 0       |
| <hr/>                                                                  |                 |         |
| flatten_55 (Flatten)                                                   | (None, 1280)    | 0       |
| <hr/>                                                                  |                 |         |
| dense_109 (Dense)                                                      | (None, 64)      | 81984   |
| <hr/>                                                                  |                 |         |
| dense_110 (Dense)                                                      | (None, 3)       | 195     |
| <hr/>                                                                  |                 |         |
| Total params:                                                          | 93,347          |         |
| Trainable params:                                                      | 93,347          |         |
| Non-trainable params:                                                  | 0               |         |
| <hr/>                                                                  |                 |         |
| Train on 3285 samples, validate on 1387 samples                        |                 |         |
| Epoch 1/35                                                             |                 |         |
| - 7s - loss: 0.4659 - acc: 0.8280 - val loss: 0.2167 - val acc: 0.9293 |                 |         |


```

Epoch 2/35  
- 1s - loss: 0.0503 - acc: 0.9857 - val\_loss: 0.1466 - val\_acc: 0.9546

Epoch 3/35  
- 1s - loss: 0.0271 - acc: 0.9924 - val\_loss: 0.1610 - val\_acc: 0.9625

Epoch 4/35  
- 1s - loss: 0.0177 - acc: 0.9973 - val\_loss: 0.1640 - val\_acc: 0.9690

Epoch 5/35  
- 1s - loss: 0.0126 - acc: 0.9979 - val\_loss: 0.1827 - val\_acc: 0.9647

Epoch 6/35  
- 1s - loss: 0.0123 - acc: 0.9963 - val\_loss: 0.1693 - val\_acc: 0.9640

Epoch 7/35  
- 1s - loss: 0.0086 - acc: 0.9979 - val\_loss: 0.2965 - val\_acc: 0.9603

Epoch 8/35  
- 1s - loss: 6.6320e-05 - acc: 1.0000 - val\_loss: 0.1885 - val\_acc: 0.9697

Epoch 9/35  
- 1s - loss: 0.0089 - acc: 0.9976 - val\_loss: 0.2611 - val\_acc: 0.9625

Epoch 10/35  
- 1s - loss: 2.9329e-06 - acc: 1.0000 - val\_loss: 0.2226 - val\_acc: 0.9661

Epoch 11/35  
- 1s - loss: 0.0156 - acc: 0.9970 - val\_loss: 0.2122 - val\_acc: 0.9668

Epoch 12/35  
- 1s - loss: 0.0068 - acc: 0.9982 - val\_loss: 0.3630 - val\_acc: 0.9474

Epoch 13/35  
- 1s - loss: 4.0137e-04 - acc: 0.9997 - val\_loss: 0.2297 - val\_acc: 0.9647

Epoch 14/35  
- 1s - loss: 0.0087 - acc: 0.9988 - val\_loss: 0.4004 - val\_acc: 0.9423

Epoch 15/35  
- 1s - loss: 4.5843e-06 - acc: 1.0000 - val\_loss: 0.3137 - val\_acc: 0.9603

Epoch 16/35  
- 1s - loss: 1.7177e-07 - acc: 1.0000 - val\_loss: 0.3433 - val\_acc: 0.9618

Epoch 17/35  
- 1s - loss: 0.0111 - acc: 0.9976 - val\_loss: 0.5313 - val\_acc: 0.9459

Epoch 18/35  
- 1s - loss: 3.7062e-06 - acc: 1.0000 - val\_loss: 0.4438 - val\_acc: 0.9560

Epoch 19/35  
- 1s - loss: 1.4918e-07 - acc: 1.0000 - val\_loss: 0.2840 - val\_acc: 0.9683

Epoch 20/35  
- 1s - loss: 1.2073e-07 - acc: 1.0000 - val\_loss: 0.3419 - val\_acc: 0.9654

Epoch 21/35  
- 1s - loss: 0.0020 - acc: 0.9991 - val\_loss: 0.4433 - val\_acc: 0.9582

Epoch 22/35  
- 1s - loss: 1.7675e-07 - acc: 1.0000 - val\_loss: 0.3540 - val\_acc: 0.9632

Epoch 23/35  
- 1s - loss: 1.1999e-07 - acc: 1.0000 - val\_loss: 0.3240 - val\_acc: 0.9654

Epoch 24/35  
- 1s - loss: 1.1937e-07 - acc: 1.0000 - val\_loss: 0.3011 - val\_acc: 0.9661

Epoch 25/35  
- 1s - loss: 1.1930e-07 - acc: 1.0000 - val\_loss: 0.3226 - val\_acc: 0.9654

Epoch 26/35  
- 1s - loss: 1.1921e-07 - acc: 1.0000 - val\_loss: 0.3300 - val\_acc: 0.9654

Epoch 27/35

```

- 1s - loss: 1.1925e-07 - acc: 1.0000 - val_loss: 0.3070 - val_acc: 0.9661

Epoch 28/35
- 1s - loss: 1.1921e-07 - acc: 1.0000 - val_loss: 0.3123 - val_acc: 0.9654

Epoch 29/35
- 1s - loss: 1.1921e-07 - acc: 1.0000 - val_loss: 0.3258 - val_acc: 0.9668

Epoch 30/35
- 1s - loss: 1.1921e-07 - acc: 1.0000 - val_loss: 0.3318 - val_acc: 0.9647

Epoch 31/35
- 1s - loss: 1.1921e-07 - acc: 1.0000 - val_loss: 0.3318 - val_acc: 0.9647

Epoch 32/35
- 1s - loss: 1.1921e-07 - acc: 1.0000 - val_loss: 0.3145 - val_acc: 0.9661

Epoch 33/35
- 1s - loss: 1.1937e-07 - acc: 1.0000 - val_loss: 0.3302 - val_acc: 0.9661

Epoch 34/35
- 1s - loss: 1.1923e-07 - acc: 1.0000 - val_loss: 0.3619 - val_acc: 0.9647

Epoch 35/35
- 1s - loss: 1.1923e-07 - acc: 1.0000 - val_loss: 0.3205 - val_acc: 0.9668

Test accuracy:
0.9668348954578226
Model: "sequential_56"

Layer (type)          Output Shape         Param #
=====
conv1d_111 (Conv1D)    (None, 126, 64)      1792
=====
conv1d_112 (Conv1D)    (None, 125, 32)       4128
=====
dropout_56 (Dropout)   (None, 125, 32)       0
=====
max_pooling1d_56 (MaxPooling) (None, 41, 32)  0
=====
flatten_56 (Flatten)   (None, 1312)          0
=====
dense_111 (Dense)     (None, 64)            84032
=====
dense_112 (Dense)     (None, 3)              195
=====

Total params: 90,147
Trainable params: 90,147
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples
Epoch 1/35
- 7s - loss: 10.0303 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 2/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 3/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 4/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 5/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 6/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 7/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 8/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 9/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

```



```
-r---. --,--  
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Test accuracy:  
0.357606344628695  
Model: "sequential\_57"

Layer (type)	Output Shape	Param #
conv1d_113 (Conv1D)	(None, 124, 16)	736
conv1d_114 (Conv1D)	(None, 122, 16)	784
dropout_57 (Dropout)	(None, 122, 16)	0
max_pooling1d_57 (MaxPooling)	(None, 61, 16)	0
flatten_57 (Flatten)	(None, 976)	0
dense_113 (Dense)	(None, 64)	62528
dense_114 (Dense)	(None, 3)	195

Total params: 64,243

Trainable params: 64,243

Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/25

```
- 7s - loss: 9.9066 - acc: 0.3753 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 2/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 3/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 4/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 5/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 6/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 7/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 8/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 9/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 10/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 11/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 12/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 13/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 14/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 15/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 16/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

Epoch 17/25

```
- 0s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576
```

-- loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 18/25

- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 19/25

- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 20/25

- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 21/25

- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 22/25

- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 23/25

- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 24/25

- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 25/25

- 0s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Test accuracy:

0.357606344628695

Model: "sequential\_58"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_115 (Conv1D)	(None, 126, 64)	1792
conv1d_116 (Conv1D)	(None, 124, 64)	12352
dropout_58 (Dropout)	(None, 124, 64)	0
max_pooling1d_58 (MaxPooling)	(None, 41, 64)	0
flatten_58 (Flatten)	(None, 2624)	0
dense_115 (Dense)	(None, 64)	168000
dense_116 (Dense)	(None, 3)	195
<hr/>		

Total params: 182,339

Trainable params: 182,339

Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/30

- 7s - loss: 1.5162 - acc: 0.5534 - val\_loss: 0.7796 - val\_acc: 0.7123

Epoch 2/30

- 1s - loss: 0.4246 - acc: 0.8371 - val\_loss: 0.4130 - val\_acc: 0.8673

Epoch 3/30

- 1s - loss: 0.1984 - acc: 0.9312 - val\_loss: 0.3016 - val\_acc: 0.9099

Epoch 4/30

- 1s - loss: 0.1264 - acc: 0.9595 - val\_loss: 0.3302 - val\_acc: 0.9171

Epoch 5/30

- 1s - loss: 0.0826 - acc: 0.9699 - val\_loss: 0.2722 - val\_acc: 0.9214

Epoch 6/30

- 1s - loss: 0.0816 - acc: 0.9781 - val\_loss: 0.2494 - val\_acc: 0.9394

Epoch 7/30

- 1s - loss: 0.0514 - acc: 0.9848 - val\_loss: 0.3298 - val\_acc: 0.9092

Epoch 8/30

- 1s - loss: 0.0577 - acc: 0.9823 - val\_loss: 0.3442 - val\_acc: 0.9200

Epoch 9/30

- 1s - loss: 0.0394 - acc: 0.9884 - val\_loss: 0.3661 - val\_acc: 0.9135

```

Epoch 10/30
- 1s - loss: 0.0344 - acc: 0.9903 - val_loss: 0.4247 - val_acc: 0.9156

Epoch 11/30
- 1s - loss: 0.0550 - acc: 0.9878 - val_loss: 0.4341 - val_acc: 0.9221

Epoch 12/30
- 1s - loss: 0.0316 - acc: 0.9903 - val_loss: 0.3792 - val_acc: 0.9358

Epoch 13/30
- 1s - loss: 0.0409 - acc: 0.9921 - val_loss: 0.4544 - val_acc: 0.9193

Epoch 14/30
- 1s - loss: 0.0420 - acc: 0.9915 - val_loss: 0.3237 - val_acc: 0.9430

Epoch 15/30
- 1s - loss: 0.0432 - acc: 0.9890 - val_loss: 0.3416 - val_acc: 0.9322

Epoch 16/30
- 1s - loss: 0.0284 - acc: 0.9921 - val_loss: 0.3373 - val_acc: 0.9560

Epoch 17/30
- 1s - loss: 0.0370 - acc: 0.9933 - val_loss: 0.3652 - val_acc: 0.9322

Epoch 18/30
- 1s - loss: 0.0295 - acc: 0.9924 - val_loss: 0.3867 - val_acc: 0.9301

Epoch 19/30
- 1s - loss: 0.0319 - acc: 0.9936 - val_loss: 0.3481 - val_acc: 0.9329

Epoch 20/30
- 1s - loss: 0.0261 - acc: 0.9927 - val_loss: 0.3015 - val_acc: 0.9387

Epoch 21/30
- 1s - loss: 0.0227 - acc: 0.9951 - val_loss: 0.4957 - val_acc: 0.9243

Epoch 22/30
- 1s - loss: 0.0287 - acc: 0.9936 - val_loss: 0.3458 - val_acc: 0.9546

Epoch 23/30
- 1s - loss: 0.0184 - acc: 0.9957 - val_loss: 0.5827 - val_acc: 0.9214

Epoch 24/30
- 1s - loss: 0.0148 - acc: 0.9963 - val_loss: 0.4104 - val_acc: 0.9459

Epoch 25/30
- 1s - loss: 0.0299 - acc: 0.9933 - val_loss: 0.3711 - val_acc: 0.9387

Epoch 26/30
- 1s - loss: 0.0063 - acc: 0.9988 - val_loss: 0.4380 - val_acc: 0.9402

Epoch 27/30
- 1s - loss: 0.0287 - acc: 0.9945 - val_loss: 0.4527 - val_acc: 0.9373

Epoch 28/30
- 1s - loss: 0.0295 - acc: 0.9936 - val_loss: 0.3457 - val_acc: 0.9539

Epoch 29/30
- 1s - loss: 0.0224 - acc: 0.9954 - val_loss: 0.4421 - val_acc: 0.9409

Epoch 30/30
- 1s - loss: 0.0183 - acc: 0.9963 - val_loss: 0.3638 - val_acc: 0.9603

Test accuracy:
0.9603460706560922
Model: "sequential_59"

```

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_117 (Conv1D)	(None, 127, 32)	608
<hr/>		
conv1d_118 (Conv1D)	(None, 125, 64)	6208
<hr/>		
dropout_59 (Dropout)	(None, 125, 64)	0
<hr/>		
max_pooling1d_59 (MaxPooling)	(None, 41, 64)	0

flatten_59 (Flatten)	(None, 2624)	0
dense_117 (Dense)	(None, 16)	42000
dense_118 (Dense)	(None, 3)	51
=====		
Total params: 48,867		
Trainable params: 48,867		
Non-trainable params: 0		

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/35  
- 8s - loss: 11.1465 - acc: 0.3005 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 2/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 3/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 4/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 5/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 6/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 7/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 8/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 9/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 10/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 11/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 12/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 13/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 14/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 15/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 16/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 17/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 18/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 19/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 20/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 21/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

Epoch 22/35  
- 1s - loss: 11.2802 - acc: 0.3002 - val\_loss: 11.2373 - val\_acc: 0.3028

```
Epoch 23/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 24/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 25/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 26/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 27/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 28/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 29/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 30/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 31/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 32/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 33/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 34/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 35/35
- 1s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
```

Test accuracy:

0.3028118241022367

Model: "sequential\_60"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_119 (Conv1D)	(None, 126, 16)	448
<hr/>		
conv1d_120 (Conv1D)	(None, 122, 32)	2592
<hr/>		
dropout_60 (Dropout)	(None, 122, 32)	0
<hr/>		
max_pooling1d_60 (MaxPooling)	(None, 40, 32)	0
<hr/>		
flatten_60 (Flatten)	(None, 1280)	0
<hr/>		
dense_119 (Dense)	(None, 16)	20496
<hr/>		
dense_120 (Dense)	(None, 3)	51
<hr/>		

Total params: 23,587

Trainable params: 23,587

Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/25

- 8s - loss: 10.0551 - acc: 0.3720 - val\_loss: 10.1239 - val\_acc: 0.3576

Epoch 2/25

- 2s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 3/25

- 2s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

Epoch 4/25

- 2s - loss: 10.1026 - acc: 0.3732 - val\_loss: 10.3542 - val\_acc: 0.3576

```

Epoch 5/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 6/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 7/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 8/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 9/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 10/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 11/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 12/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 13/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 14/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 15/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 16/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 17/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 18/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 19/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 20/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 21/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 22/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 23/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 24/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 25/25
- 2s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Test accuracy:
0.357606344628695
Model: "sequential_61"

```

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_121 (Conv1D)	(None, 126, 64)	1792
conv1d_122 (Conv1D)	(None, 123, 16)	4112
dropout_61 (Dropout)	(None, 123, 16)	0
max_pooling1d_61 (MaxPooling)	(None, 61, 16)	0

flatten_61 (Flatten)	(None, 9/6)	0
dense_121 (Dense)	(None, 64)	62528
dense_122 (Dense)	(None, 3)	195

---

Total params: 68,627  
Trainable params: 68,627  
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples

Epoch 1/25  
- 8s - loss: 0.4836 - acc: 0.8046 - val\_loss: 0.3139 - val\_acc: 0.8911

Epoch 2/25  
- 2s - loss: 0.0497 - acc: 0.9845 - val\_loss: 0.3353 - val\_acc: 0.9286

Epoch 3/25  
- 2s - loss: 0.0231 - acc: 0.9915 - val\_loss: 0.2652 - val\_acc: 0.9481

Epoch 4/25  
- 2s - loss: 0.0195 - acc: 0.9936 - val\_loss: 0.2535 - val\_acc: 0.9567

Epoch 5/25  
- 2s - loss: 0.0109 - acc: 0.9970 - val\_loss: 0.5397 - val\_acc: 0.9012

Epoch 6/25  
- 2s - loss: 0.0115 - acc: 0.9976 - val\_loss: 0.2785 - val\_acc: 0.9596

Epoch 7/25  
- 2s - loss: 0.0084 - acc: 0.9979 - val\_loss: 0.3226 - val\_acc: 0.9654

Epoch 8/25  
- 2s - loss: 0.0040 - acc: 0.9985 - val\_loss: 0.3776 - val\_acc: 0.9603

Epoch 9/25  
- 2s - loss: 0.0098 - acc: 0.9991 - val\_loss: 0.2602 - val\_acc: 0.9481

Epoch 10/25  
- 2s - loss: 0.0012 - acc: 0.9997 - val\_loss: 0.3454 - val\_acc: 0.9553

Epoch 11/25  
- 2s - loss: 0.0043 - acc: 0.9988 - val\_loss: 0.3971 - val\_acc: 0.9503

Epoch 12/25  
- 2s - loss: 2.5287e-04 - acc: 1.0000 - val\_loss: 0.3819 - val\_acc: 0.9524

Epoch 13/25  
- 2s - loss: 0.0100 - acc: 0.9979 - val\_loss: 0.3908 - val\_acc: 0.9618

Epoch 14/25  
- 2s - loss: 0.0012 - acc: 0.9997 - val\_loss: 0.4159 - val\_acc: 0.9625

Epoch 15/25  
- 2s - loss: 0.0110 - acc: 0.9985 - val\_loss: 0.4285 - val\_acc: 0.9560

Epoch 16/25  
- 2s - loss: 0.0031 - acc: 0.9988 - val\_loss: 0.5563 - val\_acc: 0.9481

Epoch 17/25  
- 2s - loss: 0.0020 - acc: 0.9991 - val\_loss: 0.3734 - val\_acc: 0.9654

Epoch 18/25  
- 2s - loss: 0.0031 - acc: 0.9991 - val\_loss: 0.4119 - val\_acc: 0.9603

Epoch 19/25  
- 2s - loss: 0.0044 - acc: 0.9994 - val\_loss: 0.5306 - val\_acc: 0.9214

Epoch 20/25  
- 2s - loss: 0.0014 - acc: 0.9997 - val\_loss: 0.4654 - val\_acc: 0.9510

Epoch 21/25  
- 2s - loss: 0.0051 - acc: 0.9994 - val\_loss: 0.5850 - val\_acc: 0.9495

Epoch 22/25  
- 2s - loss: 0.0034 - acc: 0.9985 - val\_loss: 0.4033 - val\_acc: 0.9546

- - -

```
Epoch 23/25
- 2s - loss: 0.0023 - acc: 0.9997 - val_loss: 0.4191 - val_acc: 0.9539

Epoch 24/25
- 2s - loss: 0.0059 - acc: 0.9994 - val_loss: 0.3182 - val_acc: 0.9690

Epoch 25/25
- 2s - loss: 0.0029 - acc: 0.9994 - val_loss: 0.2942 - val_acc: 0.9567

Test accuracy:
0.9567411679884643
Model: "sequential_62"
```

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_123 (Conv1D)	(None, 125, 64)	2368
conv1d_124 (Conv1D)	(None, 123, 64)	12352
dropout_62 (Dropout)	(None, 123, 64)	0
max_pooling1d_62 (MaxPooling)	(None, 61, 64)	0
flatten_62 (Flatten)	(None, 3904)	0
dense_123 (Dense)	(None, 32)	124960
dense_124 (Dense)	(None, 3)	99
<hr/>		

```
Total params: 139,779
Trainable params: 139,779
Non-trainable params: 0
```

---

```
Train on 3285 samples, validate on 1387 samples
Epoch 1/35
- 8s - loss: 0.5688 - acc: 0.8024 - val_loss: 0.4414 - val_acc: 0.8500

Epoch 2/35
- 1s - loss: 0.0683 - acc: 0.9802 - val_loss: 0.3062 - val_acc: 0.9250

Epoch 3/35
- 1s - loss: 0.0244 - acc: 0.9945 - val_loss: 0.1927 - val_acc: 0.9524

Epoch 4/35
- 1s - loss: 0.0142 - acc: 0.9967 - val_loss: 0.2593 - val_acc: 0.9416

Epoch 5/35
- 1s - loss: 0.0278 - acc: 0.9954 - val_loss: 0.2516 - val_acc: 0.9531

Epoch 6/35
- 1s - loss: 0.0036 - acc: 0.9988 - val_loss: 0.4118 - val_acc: 0.9279

Epoch 7/35
- 1s - loss: 0.0058 - acc: 0.9985 - val_loss: 0.2490 - val_acc: 0.9474

Epoch 8/35
- 1s - loss: 3.5041e-04 - acc: 1.0000 - val_loss: 0.4269 - val_acc: 0.9402

Epoch 9/35
- 1s - loss: 0.0072 - acc: 0.9970 - val_loss: 0.3600 - val_acc: 0.9387

Epoch 10/35
- 1s - loss: 0.0011 - acc: 0.9994 - val_loss: 0.3560 - val_acc: 0.9495

Epoch 11/35
- 1s - loss: 3.0010e-06 - acc: 1.0000 - val_loss: 0.3559 - val_acc: 0.9618

Epoch 12/35
- 1s - loss: 0.0112 - acc: 0.9982 - val_loss: 0.3250 - val_acc: 0.9567

Epoch 13/35
- 1s - loss: 0.0216 - acc: 0.9982 - val_loss: 0.3694 - val_acc: 0.9589

Epoch 14/35
- 1s - loss: 7.8641e-05 - acc: 1.0000 - val_loss: 1.3198 - val_acc: 0.8118

Epoch 15/35
```

```

- 1s - loss: 0.0199 - acc: 0.9976 - val_loss: 0.3036 - val_acc: 0.9510

Epoch 16/35
- 1s - loss: 8.9263e-06 - acc: 1.0000 - val_loss: 0.7063 - val_acc: 0.9113

Epoch 17/35
- 1s - loss: 0.0057 - acc: 0.9985 - val_loss: 0.3733 - val_acc: 0.9611

Epoch 18/35
- 1s - loss: 0.0025 - acc: 0.9994 - val_loss: 0.4013 - val_acc: 0.9459

Epoch 19/35
- 1s - loss: 3.2886e-04 - acc: 0.9997 - val_loss: 0.4346 - val_acc: 0.9452

Epoch 20/35
- 1s - loss: 0.0096 - acc: 0.9973 - val_loss: 0.7573 - val_acc: 0.8846

Epoch 21/35
- 1s - loss: 4.1761e-04 - acc: 0.9997 - val_loss: 0.3699 - val_acc: 0.9647

Epoch 22/35
- 1s - loss: 1.6896e-07 - acc: 1.0000 - val_loss: 0.3544 - val_acc: 0.9632

Epoch 23/35
- 1s - loss: 0.0017 - acc: 0.9997 - val_loss: 0.3883 - val_acc: 0.9618

Epoch 24/35
- 1s - loss: 0.0016 - acc: 0.9997 - val_loss: 0.3823 - val_acc: 0.9603

Epoch 25/35
- 1s - loss: 2.6440e-06 - acc: 1.0000 - val_loss: 0.3345 - val_acc: 0.9603

Epoch 26/35
- 1s - loss: 0.0027 - acc: 0.9991 - val_loss: 0.3219 - val_acc: 0.9553

Epoch 27/35
- 1s - loss: 2.3110e-05 - acc: 1.0000 - val_loss: 0.2775 - val_acc: 0.9611

Epoch 28/35
- 1s - loss: 4.2968e-05 - acc: 1.0000 - val_loss: 0.2461 - val_acc: 0.9712

Epoch 29/35
- 1s - loss: 3.1439e-04 - acc: 0.9997 - val_loss: 0.3206 - val_acc: 0.9647

Epoch 30/35
- 1s - loss: 1.8616e-07 - acc: 1.0000 - val_loss: 0.2975 - val_acc: 0.9596

Epoch 31/35
- 1s - loss: 0.0022 - acc: 0.9991 - val_loss: 0.3590 - val_acc: 0.9575

Epoch 32/35
- 1s - loss: 1.2280e-07 - acc: 1.0000 - val_loss: 0.3564 - val_acc: 0.9589

Epoch 33/35
- 1s - loss: 0.0052 - acc: 0.9988 - val_loss: 0.4413 - val_acc: 0.9466

Epoch 34/35
- 1s - loss: 1.1289e-06 - acc: 1.0000 - val_loss: 0.3637 - val_acc: 0.9560

Epoch 35/35
- 1s - loss: 0.0018 - acc: 0.9997 - val_loss: 0.3957 - val_acc: 0.9546

Test accuracy:
0.9545782263878875
Model: "sequential_63"


```

Layer (type)	Output Shape	Param #
conv1d_125 (Conv1D)	(None, 126, 16)	448
conv1d_126 (Conv1D)	(None, 125, 32)	1056
dropout_63 (Dropout)	(None, 125, 32)	0
max_pooling1d_63 (MaxPooling)	(None, 41, 32)	0
flatten_63 (Flatten)	(None, 1312)	0

dense_125 (Dense)	(None, 32)	42016
dense_126 (Dense)	(None, 3)	99
=====		
Total params: 43,619		
Trainable params: 43,619		
Non-trainable params: 0		

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/35  
- 8s - loss: 5.1380 - acc: 0.3409 - val\_loss: 1.0915 - val\_acc: 0.3663

Epoch 2/35  
- 1s - loss: 1.9354 - acc: 0.3790 - val\_loss: 1.0955 - val\_acc: 0.3764

Epoch 3/35  
- 1s - loss: 1.2631 - acc: 0.3942 - val\_loss: 1.0961 - val\_acc: 0.3663

Epoch 4/35  
- 0s - loss: 1.1517 - acc: 0.4128 - val\_loss: 1.0932 - val\_acc: 0.3778

Epoch 5/35  
- 0s - loss: 1.1148 - acc: 0.4137 - val\_loss: 1.0904 - val\_acc: 0.3893

Epoch 6/35  
- 0s - loss: 1.0653 - acc: 0.4588 - val\_loss: 1.0800 - val\_acc: 0.4174

Epoch 7/35  
- 0s - loss: 1.0454 - acc: 0.4758 - val\_loss: 1.0683 - val\_acc: 0.4369

Epoch 8/35  
- 0s - loss: 1.0172 - acc: 0.4974 - val\_loss: 1.0490 - val\_acc: 0.4369

Epoch 9/35  
- 0s - loss: 0.9530 - acc: 0.5282 - val\_loss: 1.0295 - val\_acc: 0.4369

Epoch 10/35  
- 0s - loss: 0.8998 - acc: 0.5723 - val\_loss: 0.9944 - val\_acc: 0.4398

Epoch 11/35  
- 0s - loss: 0.8163 - acc: 0.6262 - val\_loss: 0.9658 - val\_acc: 0.4521

Epoch 12/35  
- 0s - loss: 0.7546 - acc: 0.6569 - val\_loss: 0.9083 - val\_acc: 0.5386

Epoch 13/35  
- 0s - loss: 0.6657 - acc: 0.7148 - val\_loss: 0.8400 - val\_acc: 0.5926

Epoch 14/35  
- 1s - loss: 0.6037 - acc: 0.7330 - val\_loss: 0.7974 - val\_acc: 0.6157

Epoch 15/35  
- 1s - loss: 0.5434 - acc: 0.7705 - val\_loss: 0.7370 - val\_acc: 0.6590

Epoch 16/35  
- 0s - loss: 0.4951 - acc: 0.7924 - val\_loss: 0.7235 - val\_acc: 0.6669

Epoch 17/35  
- 1s - loss: 0.4449 - acc: 0.8155 - val\_loss: 0.7077 - val\_acc: 0.6770

Epoch 18/35  
- 0s - loss: 0.4212 - acc: 0.8262 - val\_loss: 0.6592 - val\_acc: 0.7152

Epoch 19/35  
- 0s - loss: 0.3629 - acc: 0.8578 - val\_loss: 0.6211 - val\_acc: 0.7462

Epoch 20/35  
- 1s - loss: 0.3444 - acc: 0.8697 - val\_loss: 0.5995 - val\_acc: 0.7844

Epoch 21/35  
- 1s - loss: 0.2914 - acc: 0.8898 - val\_loss: 0.5750 - val\_acc: 0.7859

Epoch 22/35  
- 0s - loss: 0.2861 - acc: 0.8913 - val\_loss: 0.5568 - val\_acc: 0.7952

Epoch 23/35

```

- 1s - loss: 0.2738 - acc: 0.9008 - val_loss: 0.5419 - val_acc: 0.7851

Epoch 24/35
- 0s - loss: 0.2485 - acc: 0.9072 - val_loss: 0.5265 - val_acc: 0.8284

Epoch 25/35
- 0s - loss: 0.2267 - acc: 0.9166 - val_loss: 0.5011 - val_acc: 0.8255

Epoch 26/35
- 0s - loss: 0.2048 - acc: 0.9245 - val_loss: 0.5024 - val_acc: 0.8356

Epoch 27/35
- 0s - loss: 0.1936 - acc: 0.9279 - val_loss: 0.4629 - val_acc: 0.8450

Epoch 28/35
- 0s - loss: 0.1783 - acc: 0.9367 - val_loss: 0.4653 - val_acc: 0.8399

Epoch 29/35
- 1s - loss: 0.1460 - acc: 0.9464 - val_loss: 0.4522 - val_acc: 0.8515

Epoch 30/35
- 1s - loss: 0.1471 - acc: 0.9431 - val_loss: 0.4225 - val_acc: 0.8609

Epoch 31/35
- 0s - loss: 0.1350 - acc: 0.9562 - val_loss: 0.3994 - val_acc: 0.8709

Epoch 32/35
- 0s - loss: 0.1157 - acc: 0.9607 - val_loss: 0.3691 - val_acc: 0.8825

Epoch 33/35
- 1s - loss: 0.1004 - acc: 0.9610 - val_loss: 0.3310 - val_acc: 0.8998

Epoch 34/35
- 0s - loss: 0.1043 - acc: 0.9616 - val_loss: 0.3461 - val_acc: 0.8955

Epoch 35/35
- 0s - loss: 0.0969 - acc: 0.9647 - val_loss: 0.3774 - val_acc: 0.8782

Test accuracy:
0.8781542898341744
Model: "sequential_64"

Layer (type)          Output Shape         Param #
=================================================================
conv1d_127 (Conv1D)    (None, 127, 64)      1216
conv1d_128 (Conv1D)    (None, 124, 32)       8224
dropout_64 (Dropout)   (None, 124, 32)       0
max_pooling1d_64 (MaxPooling) (None, 24, 32)   0
flatten_64 (Flatten)   (None, 768)           0
dense_127 (Dense)     (None, 16)            12304
dense_128 (Dense)     (None, 3)              51
=================================================================
Total params: 21,795
Trainable params: 21,795
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples
Epoch 1/35
- 9s - loss: 0.7785 - acc: 0.6301 - val_loss: 0.5675 - val_acc: 0.8032

Epoch 2/35
- 1s - loss: 0.1983 - acc: 0.9373 - val_loss: 0.3270 - val_acc: 0.9034

Epoch 3/35
- 1s - loss: 0.0394 - acc: 0.9912 - val_loss: 0.2615 - val_acc: 0.9337

Epoch 4/35
- 1s - loss: 0.0138 - acc: 0.9973 - val_loss: 0.2356 - val_acc: 0.9402

Epoch 5/35
- 1s - loss: 0.0054 - acc: 1.0000 - val_loss: 0.2269 - val_acc: 0.9423

```

```
Epoch 6/35
- 1s - loss: 0.0025 - acc: 1.0000 - val_loss: 0.2296 - val_acc: 0.9402

Epoch 7/35
- 1s - loss: 0.0021 - acc: 1.0000 - val_loss: 0.2328 - val_acc: 0.9481

Epoch 8/35
- 1s - loss: 0.0014 - acc: 1.0000 - val_loss: 0.2473 - val_acc: 0.9488

Epoch 9/35
- 1s - loss: 0.0013 - acc: 1.0000 - val_loss: 0.2307 - val_acc: 0.9452

Epoch 10/35
- 1s - loss: 7.8832e-04 - acc: 1.0000 - val_loss: 0.2411 - val_acc: 0.9466

Epoch 11/35
- 1s - loss: 5.8660e-04 - acc: 1.0000 - val_loss: 0.2397 - val_acc: 0.9517

Epoch 12/35
- 1s - loss: 4.5671e-04 - acc: 1.0000 - val_loss: 0.2279 - val_acc: 0.9517

Epoch 13/35
- 1s - loss: 5.4822e-04 - acc: 1.0000 - val_loss: 0.2399 - val_acc: 0.9531

Epoch 14/35
- 1s - loss: 4.1710e-04 - acc: 1.0000 - val_loss: 0.2404 - val_acc: 0.9524

Epoch 15/35
- 1s - loss: 3.0823e-04 - acc: 1.0000 - val_loss: 0.2360 - val_acc: 0.9539

Epoch 16/35
- 1s - loss: 2.5737e-04 - acc: 1.0000 - val_loss: 0.2476 - val_acc: 0.9524

Epoch 17/35
- 1s - loss: 2.3934e-04 - acc: 1.0000 - val_loss: 0.2484 - val_acc: 0.9560

Epoch 18/35
- 1s - loss: 1.9373e-04 - acc: 1.0000 - val_loss: 0.2434 - val_acc: 0.9560

Epoch 19/35
- 1s - loss: 1.8286e-04 - acc: 1.0000 - val_loss: 0.2524 - val_acc: 0.9495

Epoch 20/35
- 1s - loss: 1.2346e-04 - acc: 1.0000 - val_loss: 0.2503 - val_acc: 0.9560

Epoch 21/35
- 1s - loss: 1.2583e-04 - acc: 1.0000 - val_loss: 0.2502 - val_acc: 0.9553

Epoch 22/35
- 1s - loss: 1.2442e-04 - acc: 1.0000 - val_loss: 0.2441 - val_acc: 0.9546

Epoch 23/35
- 1s - loss: 1.1025e-04 - acc: 1.0000 - val_loss: 0.2576 - val_acc: 0.9546

Epoch 24/35
- 1s - loss: 9.1053e-05 - acc: 1.0000 - val_loss: 0.2618 - val_acc: 0.9531

Epoch 25/35
- 1s - loss: 8.6485e-05 - acc: 1.0000 - val_loss: 0.2564 - val_acc: 0.9553

Epoch 26/35
- 1s - loss: 7.4719e-05 - acc: 1.0000 - val_loss: 0.2638 - val_acc: 0.9553

Epoch 27/35
- 1s - loss: 6.9832e-05 - acc: 1.0000 - val_loss: 0.2591 - val_acc: 0.9553

Epoch 28/35
- 1s - loss: 1.1117e-04 - acc: 1.0000 - val_loss: 0.2452 - val_acc: 0.9567

Epoch 29/35
- 1s - loss: 7.0542e-05 - acc: 1.0000 - val_loss: 0.2562 - val_acc: 0.9560

Epoch 30/35
- 1s - loss: 5.2335e-05 - acc: 1.0000 - val_loss: 0.2706 - val_acc: 0.9539

Epoch 31/35
```

```

- 1s - loss: 5.1004e-05 - acc: 1.0000 - val_loss: 0.2728 - val_acc: 0.9575

Epoch 32/35
- 1s - loss: 4.5192e-05 - acc: 1.0000 - val_loss: 0.2716 - val_acc: 0.9560

Epoch 33/35
- 1s - loss: 4.5574e-05 - acc: 1.0000 - val_loss: 0.2770 - val_acc: 0.9546

Epoch 34/35
- 1s - loss: 4.3755e-05 - acc: 1.0000 - val_loss: 0.2692 - val_acc: 0.9560

Epoch 35/35
- 1s - loss: 4.1398e-05 - acc: 1.0000 - val_loss: 0.2697 - val_acc: 0.9589

Test accuracy:
0.958904109589041
Model: "sequential_65"

Layer (type)          Output Shape         Param #
=====
conv1d_129 (Conv1D)    (None, 124, 64)      2944
conv1d_130 (Conv1D)    (None, 121, 32)       8224
dropout_65 (Dropout)   (None, 121, 32)       0
max_pooling1d_65 (MaxPooling) (None, 24, 32)  0
flatten_65 (Flatten)   (None, 768)           0
dense_129 (Dense)     (None, 32)            24608
dense_130 (Dense)     (None, 3)              99
=====
Total params: 35,875
Trainable params: 35,875
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples
Epoch 1/35
- 8s - loss: 0.6002 - acc: 0.7482 - val_loss: 0.3756 - val_acc: 0.8782

Epoch 2/35
- 1s - loss: 0.0687 - acc: 0.9826 - val_loss: 0.2662 - val_acc: 0.9257

Epoch 3/35
- 1s - loss: 0.0115 - acc: 0.9982 - val_loss: 0.2730 - val_acc: 0.9207

Epoch 4/35
- 1s - loss: 0.0107 - acc: 0.9967 - val_loss: 0.1983 - val_acc: 0.9423

Epoch 5/35
- 1s - loss: 0.0053 - acc: 0.9991 - val_loss: 0.2630 - val_acc: 0.9229

Epoch 6/35
- 1s - loss: 0.0021 - acc: 0.9997 - val_loss: 0.2399 - val_acc: 0.9560

Epoch 7/35
- 1s - loss: 5.7017e-04 - acc: 1.0000 - val_loss: 0.2355 - val_acc: 0.9546

Epoch 8/35
- 1s - loss: 3.5665e-04 - acc: 1.0000 - val_loss: 0.2199 - val_acc: 0.9575

Epoch 9/35
- 1s - loss: 2.8043e-04 - acc: 1.0000 - val_loss: 0.2265 - val_acc: 0.9560

Epoch 10/35
- 1s - loss: 2.7281e-04 - acc: 1.0000 - val_loss: 0.2339 - val_acc: 0.9575

Epoch 11/35
- 1s - loss: 1.9570e-04 - acc: 1.0000 - val_loss: 0.2346 - val_acc: 0.9575

Epoch 12/35
- 1s - loss: 1.4770e-04 - acc: 1.0000 - val_loss: 0.2296 - val_acc: 0.9575

Epoch 13/35
- 1s - loss: 1.4061e-04 - acc: 1.0000 - val_loss: 0.2266 - val_acc: 0.9589

```

Epoch 14/35  
 - 1s - loss: 1.3250e-04 - acc: 1.0000 - val\_loss: 0.2277 - val\_acc: 0.9589

Epoch 15/35  
 - 1s - loss: 1.3081e-04 - acc: 1.0000 - val\_loss: 0.2495 - val\_acc: 0.9531

Epoch 16/35  
 - 1s - loss: 1.1101e-04 - acc: 1.0000 - val\_loss: 0.2345 - val\_acc: 0.9589

Epoch 17/35  
 - 1s - loss: 7.4442e-05 - acc: 1.0000 - val\_loss: 0.2300 - val\_acc: 0.9603

Epoch 18/35  
 - 1s - loss: 6.9407e-05 - acc: 1.0000 - val\_loss: 0.2284 - val\_acc: 0.9603

Epoch 19/35  
 - 1s - loss: 6.3131e-05 - acc: 1.0000 - val\_loss: 0.2323 - val\_acc: 0.9603

Epoch 20/35  
 - 1s - loss: 5.6836e-05 - acc: 1.0000 - val\_loss: 0.2301 - val\_acc: 0.9611

Epoch 21/35  
 - 1s - loss: 5.2437e-05 - acc: 1.0000 - val\_loss: 0.2342 - val\_acc: 0.9603

Epoch 22/35  
 - 1s - loss: 5.0980e-05 - acc: 1.0000 - val\_loss: 0.2523 - val\_acc: 0.9582

Epoch 23/35  
 - 1s - loss: 4.5336e-05 - acc: 1.0000 - val\_loss: 0.2431 - val\_acc: 0.9582

Epoch 24/35  
 - 1s - loss: 3.7504e-05 - acc: 1.0000 - val\_loss: 0.2440 - val\_acc: 0.9611

Epoch 25/35  
 - 1s - loss: 3.2357e-05 - acc: 1.0000 - val\_loss: 0.2487 - val\_acc: 0.9603

Epoch 26/35  
 - 1s - loss: 3.1211e-05 - acc: 1.0000 - val\_loss: 0.2456 - val\_acc: 0.9611

Epoch 27/35  
 - 1s - loss: 2.9749e-05 - acc: 1.0000 - val\_loss: 0.2504 - val\_acc: 0.9596

Epoch 28/35  
 - 1s - loss: 2.7178e-05 - acc: 1.0000 - val\_loss: 0.2448 - val\_acc: 0.9611

Epoch 29/35  
 - 1s - loss: 2.3088e-05 - acc: 1.0000 - val\_loss: 0.2425 - val\_acc: 0.9618

Epoch 30/35  
 - 1s - loss: 2.2196e-05 - acc: 1.0000 - val\_loss: 0.2499 - val\_acc: 0.9611

Epoch 31/35  
 - 1s - loss: 2.0657e-05 - acc: 1.0000 - val\_loss: 0.2455 - val\_acc: 0.9618

Epoch 32/35  
 - 1s - loss: 2.3585e-05 - acc: 1.0000 - val\_loss: 0.2468 - val\_acc: 0.9632

Epoch 33/35  
 - 1s - loss: 1.8300e-05 - acc: 1.0000 - val\_loss: 0.2492 - val\_acc: 0.9618

Epoch 34/35  
 - 1s - loss: 1.6857e-05 - acc: 1.0000 - val\_loss: 0.2594 - val\_acc: 0.9611

Epoch 35/35  
 - 1s - loss: 1.5557e-05 - acc: 1.0000 - val\_loss: 0.2481 - val\_acc: 0.9632

Test accuracy:  
0.9632299927901946

Model: "sequential\_66"

Layer (type)	Output Shape	Param #
conv1d_131 (Conv1D)	(None, 124, 64)	2944
conv1d_132 (Conv1D)	(None, 121, 32)	8224

dropout_66 (Dropout)	(None, 121, 32)	0
max_pooling1d_66 (MaxPooling)	(None, 24, 32)	0
flatten_66 (Flatten)	(None, 768)	0
dense_131 (Dense)	(None, 16)	12304
dense_132 (Dense)	(None, 3)	51

---

Total params: 23,523  
Trainable params: 23,523  
Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/35

- 8s - loss: 0.5420 - acc: 0.7836 - val\_loss: 0.2487 - val\_acc: 0.9279

Epoch 2/35

- 1s - loss: 0.0383 - acc: 0.9933 - val\_loss: 0.2672 - val\_acc: 0.9084

Epoch 3/35

- 1s - loss: 0.0099 - acc: 0.9991 - val\_loss: 0.1556 - val\_acc: 0.9531

Epoch 4/35

- 1s - loss: 0.0042 - acc: 1.0000 - val\_loss: 0.1520 - val\_acc: 0.9625

Epoch 5/35

- 1s - loss: 0.0016 - acc: 1.0000 - val\_loss: 0.1517 - val\_acc: 0.9654

Epoch 6/35

- 1s - loss: 0.0013 - acc: 1.0000 - val\_loss: 0.1642 - val\_acc: 0.9704

Epoch 7/35

- 1s - loss: 8.4406e-04 - acc: 1.0000 - val\_loss: 0.1725 - val\_acc: 0.9712

Epoch 8/35

- 1s - loss: 6.5376e-04 - acc: 1.0000 - val\_loss: 0.1668 - val\_acc: 0.9726

Epoch 9/35

- 1s - loss: 4.8128e-04 - acc: 1.0000 - val\_loss: 0.1615 - val\_acc: 0.9748

Epoch 10/35

- 1s - loss: 4.3219e-04 - acc: 1.0000 - val\_loss: 0.1662 - val\_acc: 0.9755

Epoch 11/35

- 1s - loss: 3.7282e-04 - acc: 1.0000 - val\_loss: 0.1795 - val\_acc: 0.9733

Epoch 12/35

- 1s - loss: 2.7745e-04 - acc: 1.0000 - val\_loss: 0.1756 - val\_acc: 0.9748

Epoch 13/35

- 1s - loss: 2.2960e-04 - acc: 1.0000 - val\_loss: 0.1809 - val\_acc: 0.9726

Epoch 14/35

- 1s - loss: 1.8287e-04 - acc: 1.0000 - val\_loss: 0.1803 - val\_acc: 0.9755

Epoch 15/35

- 1s - loss: 1.6256e-04 - acc: 1.0000 - val\_loss: 0.1792 - val\_acc: 0.9748

Epoch 16/35

- 1s - loss: 1.4400e-04 - acc: 1.0000 - val\_loss: 0.1896 - val\_acc: 0.9726

Epoch 17/35

- 1s - loss: 1.2923e-04 - acc: 1.0000 - val\_loss: 0.1850 - val\_acc: 0.9762

Epoch 18/35

- 1s - loss: 1.0776e-04 - acc: 1.0000 - val\_loss: 0.1862 - val\_acc: 0.9762

Epoch 19/35

- 1s - loss: 9.7743e-05 - acc: 1.0000 - val\_loss: 0.1939 - val\_acc: 0.9740

Epoch 20/35

- 1s - loss: 8.5694e-05 - acc: 1.0000 - val\_loss: 0.1899 - val\_acc: 0.9748

Epoch 21/35

- 1s - loss: 6.8919e-05 - acc: 1.0000 - val\_loss: 0.1883 - val\_acc: 0.9762

Epoch 22/35  
 - 1s - loss: 6.9654e-05 - acc: 1.0000 - val\_loss: 0.1865 - val\_acc: 0.9755

Epoch 23/35  
 - 1s - loss: 6.8056e-05 - acc: 1.0000 - val\_loss: 0.1889 - val\_acc: 0.9762

Epoch 24/35  
 - 1s - loss: 5.9913e-05 - acc: 1.0000 - val\_loss: 0.1892 - val\_acc: 0.9762

Epoch 25/35  
 - 1s - loss: 6.0503e-05 - acc: 1.0000 - val\_loss: 0.1908 - val\_acc: 0.9762

Epoch 26/35  
 - 1s - loss: 4.1488e-05 - acc: 1.0000 - val\_loss: 0.1933 - val\_acc: 0.9769

Epoch 27/35  
 - 1s - loss: 4.0724e-05 - acc: 1.0000 - val\_loss: 0.1993 - val\_acc: 0.9755

Epoch 28/35  
 - 1s - loss: 3.8581e-05 - acc: 1.0000 - val\_loss: 0.1945 - val\_acc: 0.9748

Epoch 29/35  
 - 1s - loss: 3.7215e-05 - acc: 1.0000 - val\_loss: 0.1993 - val\_acc: 0.9762

Epoch 30/35  
 - 1s - loss: 3.2876e-05 - acc: 1.0000 - val\_loss: 0.2035 - val\_acc: 0.9762

Epoch 31/35  
 - 1s - loss: 2.8926e-05 - acc: 1.0000 - val\_loss: 0.2083 - val\_acc: 0.9748

Epoch 32/35  
 - 1s - loss: 3.3692e-05 - acc: 1.0000 - val\_loss: 0.2046 - val\_acc: 0.9755

Epoch 33/35  
 - 1s - loss: 2.4852e-05 - acc: 1.0000 - val\_loss: 0.2047 - val\_acc: 0.9748

Epoch 34/35  
 - 1s - loss: 2.0616e-05 - acc: 1.0000 - val\_loss: 0.2065 - val\_acc: 0.9748

Epoch 35/35  
 - 1s - loss: 1.9275e-05 - acc: 1.0000 - val\_loss: 0.2068 - val\_acc: 0.9748

Test accuracy:  
0.9747656813266041

Model: "sequential\_67"

Layer (type)	Output Shape	Param #
conv1d_133 (Conv1D)	(None, 124, 64)	2944
conv1d_134 (Conv1D)	(None, 121, 32)	8224
dropout_67 (Dropout)	(None, 121, 32)	0
max_pooling1d_67 (MaxPooling)	(None, 24, 32)	0
flatten_67 (Flatten)	(None, 768)	0
dense_133 (Dense)	(None, 16)	12304
dense_134 (Dense)	(None, 3)	51

Total params: 23,523  
 Trainable params: 23,523  
 Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/35  
 - 9s - loss: 0.6686 - acc: 0.7279 - val\_loss: 0.4579 - val\_acc: 0.8421

Epoch 2/35  
 - 1s - loss: 0.0740 - acc: 0.9823 - val\_loss: 0.1798 - val\_acc: 0.9445

Epoch 3/35  
 - 1s - loss: 0.0151 - acc: 0.9970 - val\_loss: 0.1919 - val\_acc: 0.9366

```
Epoch 4/35
- 1s - loss: 0.0065 - acc: 1.0000 - val_loss: 0.1478 - val_acc: 0.9575

Epoch 5/35
- 1s - loss: 0.0024 - acc: 1.0000 - val_loss: 0.1384 - val_acc: 0.9625

Epoch 6/35
- 1s - loss: 0.0012 - acc: 1.0000 - val_loss: 0.1268 - val_acc: 0.9668

Epoch 7/35
- 1s - loss: 8.6910e-04 - acc: 1.0000 - val_loss: 0.1340 - val_acc: 0.9647

Epoch 8/35
- 1s - loss: 6.3069e-04 - acc: 1.0000 - val_loss: 0.1422 - val_acc: 0.9668

Epoch 9/35
- 1s - loss: 4.9860e-04 - acc: 1.0000 - val_loss: 0.1309 - val_acc: 0.9668

Epoch 10/35
- 1s - loss: 4.0354e-04 - acc: 1.0000 - val_loss: 0.1336 - val_acc: 0.9676

Epoch 11/35
- 1s - loss: 3.4553e-04 - acc: 1.0000 - val_loss: 0.1395 - val_acc: 0.9683

Epoch 12/35
- 1s - loss: 2.8472e-04 - acc: 1.0000 - val_loss: 0.1365 - val_acc: 0.9676

Epoch 13/35
- 1s - loss: 2.0976e-04 - acc: 1.0000 - val_loss: 0.1373 - val_acc: 0.9661

Epoch 14/35
- 1s - loss: 2.0142e-04 - acc: 1.0000 - val_loss: 0.1382 - val_acc: 0.9683

Epoch 15/35
- 1s - loss: 1.8581e-04 - acc: 1.0000 - val_loss: 0.1363 - val_acc: 0.9683

Epoch 16/35
- 1s - loss: 1.4463e-04 - acc: 1.0000 - val_loss: 0.1418 - val_acc: 0.9676

Epoch 17/35
- 1s - loss: 1.4055e-04 - acc: 1.0000 - val_loss: 0.1423 - val_acc: 0.9683

Epoch 18/35
- 1s - loss: 1.1840e-04 - acc: 1.0000 - val_loss: 0.1421 - val_acc: 0.9683

Epoch 19/35
- 1s - loss: 9.8407e-05 - acc: 1.0000 - val_loss: 0.1411 - val_acc: 0.9690

Epoch 20/35
- 1s - loss: 8.9737e-05 - acc: 1.0000 - val_loss: 0.1479 - val_acc: 0.9683

Epoch 21/35
- 1s - loss: 8.0326e-05 - acc: 1.0000 - val_loss: 0.1480 - val_acc: 0.9668

Epoch 22/35
- 1s - loss: 7.6505e-05 - acc: 1.0000 - val_loss: 0.1458 - val_acc: 0.9683

Epoch 23/35
- 1s - loss: 6.9195e-05 - acc: 1.0000 - val_loss: 0.1453 - val_acc: 0.9690

Epoch 24/35
- 1s - loss: 7.2113e-05 - acc: 1.0000 - val_loss: 0.1431 - val_acc: 0.9697

Epoch 25/35
- 1s - loss: 5.5074e-05 - acc: 1.0000 - val_loss: 0.1441 - val_acc: 0.9690

Epoch 26/35
- 1s - loss: 4.5347e-05 - acc: 1.0000 - val_loss: 0.1473 - val_acc: 0.9690

Epoch 27/35
- 1s - loss: 4.2791e-05 - acc: 1.0000 - val_loss: 0.1504 - val_acc: 0.9683

Epoch 28/35
- 1s - loss: 3.9949e-05 - acc: 1.0000 - val_loss: 0.1480 - val_acc: 0.9690

Epoch 29/35
- 1s - loss: 3.6378e-05 - acc: 1.0000 - val_loss: 0.1492 - val_acc: 0.9683
```

```
-- loss: 0.0000 - acc: 1.0000 - val_loss: 0.1466 - val_acc: 0.9690
Epoch 30/35
- 1s - loss: 3.7849e-05 - acc: 1.0000 - val_loss: 0.1466 - val_acc: 0.9690

Epoch 31/35
- 1s - loss: 3.0879e-05 - acc: 1.0000 - val_loss: 0.1522 - val_acc: 0.9668

Epoch 32/35
- 1s - loss: 2.9801e-05 - acc: 1.0000 - val_loss: 0.1546 - val_acc: 0.9683

Epoch 33/35
- 1s - loss: 2.7975e-05 - acc: 1.0000 - val_loss: 0.1533 - val_acc: 0.9683

Epoch 34/35
- 1s - loss: 2.8407e-05 - acc: 1.0000 - val_loss: 0.1524 - val_acc: 0.9683

Epoch 35/35
- 1s - loss: 2.3273e-05 - acc: 1.0000 - val_loss: 0.1536 - val_acc: 0.9683
```

Test accuracy:

0.9682768565248738

Model: "sequential\_68"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_135 (Conv1D)	(None, 124, 64)	2944
conv1d_136 (Conv1D)	(None, 121, 32)	8224
dropout_68 (Dropout)	(None, 121, 32)	0
max_pooling1d_68 (MaxPooling)	(None, 24, 32)	0
flatten_68 (Flatten)	(None, 768)	0
dense_135 (Dense)	(None, 16)	12304
dense_136 (Dense)	(None, 3)	51
<hr/>		
Total params:	23,523	
Trainable params:	23,523	
Non-trainable params:	0	

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/35  
- 9s - loss: 0.7101 - acc: 0.6843 - val\_loss: 0.3609 - val\_acc: 0.9041

Epoch 2/35  
- 1s - loss: 0.0521 - acc: 0.9866 - val\_loss: 0.2013 - val\_acc: 0.9524

Epoch 3/35  
- 1s - loss: 0.0093 - acc: 0.9985 - val\_loss: 0.2027 - val\_acc: 0.9640

Epoch 4/35  
- 1s - loss: 0.0050 - acc: 0.9994 - val\_loss: 0.2865 - val\_acc: 0.9337

Epoch 5/35  
- 1s - loss: 0.0022 - acc: 0.9997 - val\_loss: 0.2251 - val\_acc: 0.9596

Epoch 6/35  
- 1s - loss: 0.0011 - acc: 1.0000 - val\_loss: 0.1843 - val\_acc: 0.9683

Epoch 7/35  
- 1s - loss: 5.1621e-04 - acc: 1.0000 - val\_loss: 0.2019 - val\_acc: 0.9683

Epoch 8/35  
- 1s - loss: 4.8855e-04 - acc: 1.0000 - val\_loss: 0.2030 - val\_acc: 0.9668

Epoch 9/35  
- 1s - loss: 3.7579e-04 - acc: 1.0000 - val\_loss: 0.2075 - val\_acc: 0.9661

Epoch 10/35  
- 1s - loss: 2.9026e-04 - acc: 1.0000 - val\_loss: 0.2065 - val\_acc: 0.9683

Epoch 11/35  
- 1s - loss: 2.7072e-04 - acc: 1.0000 - val\_loss: 0.2129 - val\_acc: 0.9661

```
Epoch 12/35
- 1s - loss: 2.3506e-04 - acc: 1.0000 - val_loss: 0.1869 - val_acc: 0.9690

Epoch 13/35
- 1s - loss: 2.1516e-04 - acc: 1.0000 - val_loss: 0.2216 - val_acc: 0.9668

Epoch 14/35
- 1s - loss: 1.4506e-04 - acc: 1.0000 - val_loss: 0.2297 - val_acc: 0.9661

Epoch 15/35
- 1s - loss: 1.4895e-04 - acc: 1.0000 - val_loss: 0.2161 - val_acc: 0.9676

Epoch 16/35
- 1s - loss: 1.4750e-04 - acc: 1.0000 - val_loss: 0.2217 - val_acc: 0.9676

Epoch 17/35
- 1s - loss: 1.1951e-04 - acc: 1.0000 - val_loss: 0.2220 - val_acc: 0.9683

Epoch 18/35
- 1s - loss: 1.1539e-04 - acc: 1.0000 - val_loss: 0.2070 - val_acc: 0.9712

Epoch 19/35
- 1s - loss: 1.1605e-04 - acc: 1.0000 - val_loss: 0.2191 - val_acc: 0.9683

Epoch 20/35
- 1s - loss: 6.9681e-05 - acc: 1.0000 - val_loss: 0.2239 - val_acc: 0.9683

Epoch 21/35
- 1s - loss: 5.9597e-05 - acc: 1.0000 - val_loss: 0.2333 - val_acc: 0.9676

Epoch 22/35
- 1s - loss: 6.2056e-05 - acc: 1.0000 - val_loss: 0.2273 - val_acc: 0.9683

Epoch 23/35
- 1s - loss: 5.0726e-05 - acc: 1.0000 - val_loss: 0.2350 - val_acc: 0.9676

Epoch 24/35
- 1s - loss: 4.5918e-05 - acc: 1.0000 - val_loss: 0.2251 - val_acc: 0.9712

Epoch 25/35
- 1s - loss: 4.8049e-05 - acc: 1.0000 - val_loss: 0.2423 - val_acc: 0.9676

Epoch 26/35
- 1s - loss: 5.5797e-05 - acc: 1.0000 - val_loss: 0.2350 - val_acc: 0.9668

Epoch 27/35
- 1s - loss: 4.2466e-05 - acc: 1.0000 - val_loss: 0.2255 - val_acc: 0.9712

Epoch 28/35
- 1s - loss: 2.7534e-05 - acc: 1.0000 - val_loss: 0.2388 - val_acc: 0.9676

Epoch 29/35
- 1s - loss: 3.1584e-05 - acc: 1.0000 - val_loss: 0.2348 - val_acc: 0.9719

Epoch 30/35
- 1s - loss: 2.7215e-05 - acc: 1.0000 - val_loss: 0.2356 - val_acc: 0.9704

Epoch 31/35
- 1s - loss: 2.4039e-05 - acc: 1.0000 - val_loss: 0.2445 - val_acc: 0.9683

Epoch 32/35
- 1s - loss: 2.7575e-05 - acc: 1.0000 - val_loss: 0.2328 - val_acc: 0.9704

Epoch 33/35
- 1s - loss: 2.4115e-05 - acc: 1.0000 - val_loss: 0.2357 - val_acc: 0.9719

Epoch 34/35
- 1s - loss: 1.9032e-05 - acc: 1.0000 - val_loss: 0.2446 - val_acc: 0.9690

Epoch 35/35
- 1s - loss: 1.7457e-05 - acc: 1.0000 - val_loss: 0.2454 - val_acc: 0.9697

Test accuracy:
0.969718817591925
Model: "sequential_69"
```

Layer (type)	Output Shape	Param #
--------------	--------------	---------

layer type	output shape	param #
conv1d_137 (Conv1D)	(None, 124, 64)	2944
conv1d_138 (Conv1D)	(None, 120, 32)	10272
dropout_69 (Dropout)	(None, 120, 32)	0
max_pooling1d_69 (MaxPooling)	(None, 24, 32)	0
flatten_69 (Flatten)	(None, 768)	0
dense_137 (Dense)	(None, 16)	12304
dense_138 (Dense)	(None, 3)	51

Total params: 25,571

Trainable params: 25,571

Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples

Epoch 1/35

- 9s - loss: 0.7300 - acc: 0.6840 - val\_loss: 0.4469 - val\_acc: 0.8709

Epoch 2/35

- 1s - loss: 0.0800 - acc: 0.9814 - val\_loss: 0.1891 - val\_acc: 0.9553

Epoch 3/35

- 1s - loss: 0.0143 - acc: 0.9976 - val\_loss: 0.2384 - val\_acc: 0.9495

Epoch 4/35

- 1s - loss: 0.0049 - acc: 1.0000 - val\_loss: 0.1679 - val\_acc: 0.9654

Epoch 5/35

- 1s - loss: 0.0024 - acc: 1.0000 - val\_loss: 0.2075 - val\_acc: 0.9589

Epoch 6/35

- 1s - loss: 0.0017 - acc: 1.0000 - val\_loss: 0.1964 - val\_acc: 0.9632

Epoch 7/35

- 1s - loss: 0.0014 - acc: 1.0000 - val\_loss: 0.1714 - val\_acc: 0.9668

Epoch 8/35

- 1s - loss: 8.1996e-04 - acc: 1.0000 - val\_loss: 0.1957 - val\_acc: 0.9661

Epoch 9/35

- 1s - loss: 5.0831e-04 - acc: 1.0000 - val\_loss: 0.1872 - val\_acc: 0.9676

Epoch 10/35

- 1s - loss: 4.7692e-04 - acc: 1.0000 - val\_loss: 0.1698 - val\_acc: 0.9690

Epoch 11/35

- 1s - loss: 4.2934e-04 - acc: 1.0000 - val\_loss: 0.1776 - val\_acc: 0.9683

Epoch 12/35

- 1s - loss: 3.2642e-04 - acc: 1.0000 - val\_loss: 0.1933 - val\_acc: 0.9676

Epoch 13/35

- 1s - loss: 3.1290e-04 - acc: 1.0000 - val\_loss: 0.2030 - val\_acc: 0.9676

Epoch 14/35

- 1s - loss: 2.3516e-04 - acc: 1.0000 - val\_loss: 0.1992 - val\_acc: 0.9676

Epoch 15/35

- 1s - loss: 1.7920e-04 - acc: 1.0000 - val\_loss: 0.1994 - val\_acc: 0.9668

Epoch 16/35

- 1s - loss: 1.4993e-04 - acc: 1.0000 - val\_loss: 0.2189 - val\_acc: 0.9654

Epoch 17/35

- 1s - loss: 1.3266e-04 - acc: 1.0000 - val\_loss: 0.2165 - val\_acc: 0.9661

Epoch 18/35

- 1s - loss: 1.3659e-04 - acc: 1.0000 - val\_loss: 0.2010 - val\_acc: 0.9683

Epoch 19/35

- 1s - loss: 1.3855e-04 - acc: 1.0000 - val\_loss: 0.2080 - val\_acc: 0.9668

```

Epoch 20/35
- 1s - loss: 1.2409e-04 - acc: 1.0000 - val_loss: 0.2141 - val_acc: 0.9654

Epoch 21/35
- 1s - loss: 9.8059e-05 - acc: 1.0000 - val_loss: 0.2137 - val_acc: 0.9668

Epoch 22/35
- 1s - loss: 7.8302e-05 - acc: 1.0000 - val_loss: 0.2217 - val_acc: 0.9661

Epoch 23/35
- 1s - loss: 6.3008e-05 - acc: 1.0000 - val_loss: 0.2141 - val_acc: 0.9676

Epoch 24/35
- 1s - loss: 6.0488e-05 - acc: 1.0000 - val_loss: 0.2155 - val_acc: 0.9676

Epoch 25/35
- 1s - loss: 5.0435e-05 - acc: 1.0000 - val_loss: 0.2185 - val_acc: 0.9676

Epoch 26/35
- 1s - loss: 6.3524e-05 - acc: 1.0000 - val_loss: 0.2093 - val_acc: 0.9683

Epoch 27/35
- 1s - loss: 4.6562e-05 - acc: 1.0000 - val_loss: 0.2395 - val_acc: 0.9640

Epoch 28/35
- 1s - loss: 5.2329e-05 - acc: 1.0000 - val_loss: 0.2391 - val_acc: 0.9640

Epoch 29/35
- 1s - loss: 4.0505e-05 - acc: 1.0000 - val_loss: 0.2348 - val_acc: 0.9640

Epoch 30/35
- 1s - loss: 4.1174e-05 - acc: 1.0000 - val_loss: 0.2386 - val_acc: 0.9661

Epoch 31/35
- 1s - loss: 2.9101e-05 - acc: 1.0000 - val_loss: 0.2308 - val_acc: 0.9661

Epoch 32/35
- 1s - loss: 3.1876e-05 - acc: 1.0000 - val_loss: 0.2267 - val_acc: 0.9668

Epoch 33/35
- 1s - loss: 3.3921e-05 - acc: 1.0000 - val_loss: 0.2350 - val_acc: 0.9661

Epoch 34/35
- 1s - loss: 2.7614e-05 - acc: 1.0000 - val_loss: 0.2454 - val_acc: 0.9654

Epoch 35/35
- 1s - loss: 2.3987e-05 - acc: 1.0000 - val_loss: 0.2468 - val_acc: 0.9661

```

Test accuracy:  
0.966113914924297  
Model: "sequential\_70"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_139 (Conv1D)	(None, 127, 64)	1216
<hr/>		
conv1d_140 (Conv1D)	(None, 124, 32)	8224
<hr/>		
dropout_70 (Dropout)	(None, 124, 32)	0
<hr/>		
max_pooling1d_70 (MaxPooling)	(None, 24, 32)	0
<hr/>		
flatten_70 (Flatten)	(None, 768)	0
<hr/>		
dense_139 (Dense)	(None, 16)	12304
<hr/>		
dense_140 (Dense)	(None, 3)	51
<hr/>		

Total params: 21,795  
Trainable params: 21,795  
Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples  
Epoch 1/35  
- 9s - loss: 0.7197 - acc: 0.6858 - val\_loss: 0.5135 - val\_acc: 0.7866

Epoch 2/35

```
Epoch 2/35
- 1s - loss: 0.1457 - acc: 0.9540 - val_loss: 0.3475 - val_acc: 0.9092

Epoch 3/35
- 1s - loss: 0.0351 - acc: 0.9909 - val_loss: 0.2765 - val_acc: 0.9351

Epoch 4/35
- 1s - loss: 0.0121 - acc: 0.9979 - val_loss: 0.2475 - val_acc: 0.9459

Epoch 5/35
- 1s - loss: 0.0041 - acc: 1.0000 - val_loss: 0.2591 - val_acc: 0.9488

Epoch 6/35
- 1s - loss: 0.0020 - acc: 1.0000 - val_loss: 0.2429 - val_acc: 0.9517

Epoch 7/35
- 1s - loss: 0.0013 - acc: 1.0000 - val_loss: 0.2544 - val_acc: 0.9503

Epoch 8/35
- 1s - loss: 9.6578e-04 - acc: 1.0000 - val_loss: 0.2618 - val_acc: 0.9517

Epoch 9/35
- 1s - loss: 7.7513e-04 - acc: 1.0000 - val_loss: 0.2668 - val_acc: 0.9524

Epoch 10/35
- 1s - loss: 6.1936e-04 - acc: 1.0000 - val_loss: 0.2660 - val_acc: 0.9539

Epoch 11/35
- 1s - loss: 5.2469e-04 - acc: 1.0000 - val_loss: 0.2813 - val_acc: 0.9539

Epoch 12/35
- 1s - loss: 3.9670e-04 - acc: 1.0000 - val_loss: 0.2754 - val_acc: 0.9546

Epoch 13/35
- 1s - loss: 3.8219e-04 - acc: 1.0000 - val_loss: 0.2957 - val_acc: 0.9495

Epoch 14/35
- 1s - loss: 3.0491e-04 - acc: 1.0000 - val_loss: 0.2756 - val_acc: 0.9546

Epoch 15/35
- 1s - loss: 3.1360e-04 - acc: 1.0000 - val_loss: 0.2760 - val_acc: 0.9553

Epoch 16/35
- 1s - loss: 2.1285e-04 - acc: 1.0000 - val_loss: 0.2735 - val_acc: 0.9582

Epoch 17/35
- 1s - loss: 2.1847e-04 - acc: 1.0000 - val_loss: 0.2730 - val_acc: 0.9553

Epoch 18/35
- 1s - loss: 1.7922e-04 - acc: 1.0000 - val_loss: 0.2814 - val_acc: 0.9560

Epoch 19/35
- 1s - loss: 1.6010e-04 - acc: 1.0000 - val_loss: 0.2868 - val_acc: 0.9567

Epoch 20/35
- 1s - loss: 1.1866e-04 - acc: 1.0000 - val_loss: 0.2880 - val_acc: 0.9567

Epoch 21/35
- 1s - loss: 1.1056e-04 - acc: 1.0000 - val_loss: 0.2935 - val_acc: 0.9553

Epoch 22/35
- 1s - loss: 9.9627e-05 - acc: 1.0000 - val_loss: 0.2934 - val_acc: 0.9567

Epoch 23/35
- 1s - loss: 1.0224e-04 - acc: 1.0000 - val_loss: 0.3021 - val_acc: 0.9553

Epoch 24/35
- 1s - loss: 9.7775e-05 - acc: 1.0000 - val_loss: 0.2958 - val_acc: 0.9560

Epoch 25/35
- 1s - loss: 7.1050e-05 - acc: 1.0000 - val_loss: 0.2972 - val_acc: 0.9582

Epoch 26/35
- 1s - loss: 6.5534e-05 - acc: 1.0000 - val_loss: 0.2971 - val_acc: 0.9575

Epoch 27/35
- 1s - loss: 6.0170e-05 - acc: 1.0000 - val_loss: 0.3052 - val_acc: 0.9560
```

```

Epoch 28/35
- 1s - loss: 5.9840e-05 - acc: 1.0000 - val_loss: 0.3064 - val_acc: 0.9560

Epoch 29/35
- 1s - loss: 5.1002e-05 - acc: 1.0000 - val_loss: 0.2992 - val_acc: 0.9567

Epoch 30/35
- 1s - loss: 5.0336e-05 - acc: 1.0000 - val_loss: 0.3044 - val_acc: 0.9575

Epoch 31/35
- 1s - loss: 4.5944e-05 - acc: 1.0000 - val_loss: 0.2928 - val_acc: 0.9596

Epoch 32/35
- 1s - loss: 3.7508e-05 - acc: 1.0000 - val_loss: 0.3051 - val_acc: 0.9582

Epoch 33/35
- 1s - loss: 3.6136e-05 - acc: 1.0000 - val_loss: 0.3099 - val_acc: 0.9560

Epoch 34/35
- 1s - loss: 3.3276e-05 - acc: 1.0000 - val_loss: 0.3204 - val_acc: 0.9560

Epoch 35/35
- 1s - loss: 2.8993e-05 - acc: 1.0000 - val_loss: 0.3061 - val_acc: 0.9582

Test accuracy:
0.9581831290555155
Model: "sequential_71"

Layer (type)          Output Shape         Param #
=====
conv1d_141 (Conv1D)    (None, 124, 64)      2944
conv1d_142 (Conv1D)    (None, 121, 32)       8224
dropout_71 (Dropout)   (None, 121, 32)       0
max_pooling1d_71 (MaxPooling) (None, 24, 32)  0
flatten_71 (Flatten)   (None, 768)           0
dense_141 (Dense)     (None, 32)            24608
dense_142 (Dense)     (None, 3)              99
=====
Total params: 35,875
Trainable params: 35,875
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples
Epoch 1/35
- 9s - loss: 0.4999 - acc: 0.8058 - val_loss: 0.2244 - val_acc: 0.9351

Epoch 2/35
- 1s - loss: 0.0271 - acc: 0.9936 - val_loss: 0.2184 - val_acc: 0.9387

Epoch 3/35
- 1s - loss: 0.0052 - acc: 0.9997 - val_loss: 0.1936 - val_acc: 0.9503

Epoch 4/35
- 1s - loss: 0.0027 - acc: 1.0000 - val_loss: 0.2070 - val_acc: 0.9445

Epoch 5/35
- 1s - loss: 0.0010 - acc: 1.0000 - val_loss: 0.1893 - val_acc: 0.9553

Epoch 6/35
- 1s - loss: 5.9447e-04 - acc: 1.0000 - val_loss: 0.1942 - val_acc: 0.9575

Epoch 7/35
- 1s - loss: 4.1962e-04 - acc: 1.0000 - val_loss: 0.1933 - val_acc: 0.9582

Epoch 8/35
- 1s - loss: 3.1583e-04 - acc: 1.0000 - val_loss: 0.2102 - val_acc: 0.9567

Epoch 9/35
- 1s - loss: 2.6344e-04 - acc: 1.0000 - val_loss: 0.1969 - val_acc: 0.9589

```

```
Epoch 10/35
- 1s - loss: 4.6239e-04 - acc: 1.0000 - val_loss: 0.1409 - val_acc: 0.9654

Epoch 11/35
- 1s - loss: 3.4270e-04 - acc: 1.0000 - val_loss: 0.2045 - val_acc: 0.9582

Epoch 12/35
- 1s - loss: 1.3590e-04 - acc: 1.0000 - val_loss: 0.2162 - val_acc: 0.9582

Epoch 13/35
- 1s - loss: 1.2091e-04 - acc: 1.0000 - val_loss: 0.2435 - val_acc: 0.9531

Epoch 14/35
- 1s - loss: 1.2703e-04 - acc: 1.0000 - val_loss: 0.2454 - val_acc: 0.9546

Epoch 15/35
- 1s - loss: 8.5140e-05 - acc: 1.0000 - val_loss: 0.2302 - val_acc: 0.9567

Epoch 16/35
- 1s - loss: 9.0118e-05 - acc: 1.0000 - val_loss: 0.2498 - val_acc: 0.9539

Epoch 17/35
- 1s - loss: 8.3200e-05 - acc: 1.0000 - val_loss: 0.2412 - val_acc: 0.9560

Epoch 18/35
- 1s - loss: 6.0608e-05 - acc: 1.0000 - val_loss: 0.2367 - val_acc: 0.9575

Epoch 19/35
- 1s - loss: 4.8959e-05 - acc: 1.0000 - val_loss: 0.2365 - val_acc: 0.9575

Epoch 20/35
- 1s - loss: 5.6453e-05 - acc: 1.0000 - val_loss: 0.2289 - val_acc: 0.9575

Epoch 21/35
- 1s - loss: 4.5174e-05 - acc: 1.0000 - val_loss: 0.2281 - val_acc: 0.9582

Epoch 22/35
- 1s - loss: 4.1135e-05 - acc: 1.0000 - val_loss: 0.2341 - val_acc: 0.9582

Epoch 23/35
- 1s - loss: 3.2666e-05 - acc: 1.0000 - val_loss: 0.2486 - val_acc: 0.9567

Epoch 24/35
- 1s - loss: 4.5202e-05 - acc: 1.0000 - val_loss: 0.2387 - val_acc: 0.9603

Epoch 25/35
- 1s - loss: 4.3286e-05 - acc: 1.0000 - val_loss: 0.2619 - val_acc: 0.9553

Epoch 26/35
- 1s - loss: 2.4153e-05 - acc: 1.0000 - val_loss: 0.2658 - val_acc: 0.9546

Epoch 27/35
- 1s - loss: 2.7712e-05 - acc: 1.0000 - val_loss: 0.2471 - val_acc: 0.9567

Epoch 28/35
- 1s - loss: 1.7245e-05 - acc: 1.0000 - val_loss: 0.2567 - val_acc: 0.9567

Epoch 29/35
- 1s - loss: 2.1375e-05 - acc: 1.0000 - val_loss: 0.2383 - val_acc: 0.9603

Epoch 30/35
- 1s - loss: 2.3673e-05 - acc: 1.0000 - val_loss: 0.2486 - val_acc: 0.9567

Epoch 31/35
- 1s - loss: 1.7151e-05 - acc: 1.0000 - val_loss: 0.2567 - val_acc: 0.9575

Epoch 32/35
- 1s - loss: 1.3803e-05 - acc: 1.0000 - val_loss: 0.2606 - val_acc: 0.9575

Epoch 33/35
- 1s - loss: 1.6064e-05 - acc: 1.0000 - val_loss: 0.2488 - val_acc: 0.9582

Epoch 34/35
- 1s - loss: 1.3905e-05 - acc: 1.0000 - val_loss: 0.2456 - val_acc: 0.9589

Epoch 35/35
- 1s - loss: 1.2424e-05 - acc: 1.0000 - val_loss: 0.2622 - val_acc: 0.9575
```

Test accuracy:  
0.9574621485219899  
Model: "sequential\_72"

Layer (type)	Output Shape	Param #
conv1d_143 (Conv1D)	(None, 124, 64)	2944
conv1d_144 (Conv1D)	(None, 121, 32)	8224
dropout_72 (Dropout)	(None, 121, 32)	0
max_pooling1d_72 (MaxPooling)	(None, 24, 32)	0
flatten_72 (Flatten)	(None, 768)	0
dense_143 (Dense)	(None, 16)	12304
dense_144 (Dense)	(None, 3)	51

Total params: 23,523

Trainable params: 23,523

Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples

Epoch 1/35

- 9s - loss: 0.4771 - acc: 0.7991 - val\_loss: 0.2377 - val\_acc: 0.9452

Epoch 2/35

- 1s - loss: 0.0349 - acc: 0.9930 - val\_loss: 0.1652 - val\_acc: 0.9618

Epoch 3/35

- 1s - loss: 0.0046 - acc: 1.0000 - val\_loss: 0.2012 - val\_acc: 0.9719

Epoch 4/35

- 1s - loss: 0.0018 - acc: 1.0000 - val\_loss: 0.1966 - val\_acc: 0.9676

Epoch 5/35

- 1s - loss: 0.0011 - acc: 1.0000 - val\_loss: 0.2110 - val\_acc: 0.9704

Epoch 6/35

- 1s - loss: 6.7445e-04 - acc: 1.0000 - val\_loss: 0.2219 - val\_acc: 0.9712

Epoch 7/35

- 1s - loss: 4.7044e-04 - acc: 1.0000 - val\_loss: 0.2118 - val\_acc: 0.9712

Epoch 8/35

- 1s - loss: 3.4510e-04 - acc: 1.0000 - val\_loss: 0.2175 - val\_acc: 0.9690

Epoch 9/35

- 1s - loss: 2.6537e-04 - acc: 1.0000 - val\_loss: 0.2285 - val\_acc: 0.9690

Epoch 10/35

- 1s - loss: 2.0637e-04 - acc: 1.0000 - val\_loss: 0.2391 - val\_acc: 0.9712

Epoch 11/35

- 1s - loss: 1.6598e-04 - acc: 1.0000 - val\_loss: 0.2440 - val\_acc: 0.9712

Epoch 12/35

- 1s - loss: 1.3637e-04 - acc: 1.0000 - val\_loss: 0.2470 - val\_acc: 0.9712

Epoch 13/35

- 1s - loss: 1.1563e-04 - acc: 1.0000 - val\_loss: 0.2402 - val\_acc: 0.9712

Epoch 14/35

- 1s - loss: 9.7541e-05 - acc: 1.0000 - val\_loss: 0.2537 - val\_acc: 0.9704

Epoch 15/35

- 1s - loss: 8.2719e-05 - acc: 1.0000 - val\_loss: 0.2519 - val\_acc: 0.9712

Epoch 16/35

- 1s - loss: 7.1740e-05 - acc: 1.0000 - val\_loss: 0.2612 - val\_acc: 0.9704

Epoch 17/35

- 1s - loss: 6.2588e-05 - acc: 1.0000 - val\_loss: 0.2619 - val\_acc: 0.9704

- - - - -

```

Epoch 18/35
- 1s - loss: 5.5914e-05 - acc: 1.0000 - val_loss: 0.2508 - val_acc: 0.9712

Epoch 19/35
- 1s - loss: 4.8257e-05 - acc: 1.0000 - val_loss: 0.2555 - val_acc: 0.9712

Epoch 20/35
- 1s - loss: 4.2577e-05 - acc: 1.0000 - val_loss: 0.2589 - val_acc: 0.9712

Epoch 21/35
- 1s - loss: 3.8916e-05 - acc: 1.0000 - val_loss: 0.2653 - val_acc: 0.9704

Epoch 22/35
- 1s - loss: 3.4474e-05 - acc: 1.0000 - val_loss: 0.2675 - val_acc: 0.9704

Epoch 23/35
- 1s - loss: 3.1199e-05 - acc: 1.0000 - val_loss: 0.2707 - val_acc: 0.9704

Epoch 24/35
- 1s - loss: 2.7792e-05 - acc: 1.0000 - val_loss: 0.2671 - val_acc: 0.9712

Epoch 25/35
- 1s - loss: 2.5385e-05 - acc: 1.0000 - val_loss: 0.2603 - val_acc: 0.9712

Epoch 26/35
- 1s - loss: 2.2828e-05 - acc: 1.0000 - val_loss: 0.2721 - val_acc: 0.9704

Epoch 27/35
- 1s - loss: 2.0929e-05 - acc: 1.0000 - val_loss: 0.2710 - val_acc: 0.9712

Epoch 28/35
- 1s - loss: 1.9214e-05 - acc: 1.0000 - val_loss: 0.2767 - val_acc: 0.9704

Epoch 29/35
- 1s - loss: 1.7361e-05 - acc: 1.0000 - val_loss: 0.2785 - val_acc: 0.9704

Epoch 30/35
- 1s - loss: 1.5777e-05 - acc: 1.0000 - val_loss: 0.2741 - val_acc: 0.9712

Epoch 31/35
- 1s - loss: 1.4533e-05 - acc: 1.0000 - val_loss: 0.2740 - val_acc: 0.9712

Epoch 32/35
- 1s - loss: 1.3389e-05 - acc: 1.0000 - val_loss: 0.2758 - val_acc: 0.9712

Epoch 33/35
- 1s - loss: 1.2512e-05 - acc: 1.0000 - val_loss: 0.2824 - val_acc: 0.9704

Epoch 34/35
- 1s - loss: 1.1397e-05 - acc: 1.0000 - val_loss: 0.2852 - val_acc: 0.9704

Epoch 35/35
- 1s - loss: 1.0537e-05 - acc: 1.0000 - val_loss: 0.2812 - val_acc: 0.9712

```

Test accuracy:

0.9711607786589762

Model: "sequential\_73"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_145 (Conv1D)	(None, 127, 64)	1216
conv1d_146 (Conv1D)	(None, 123, 32)	10272
dropout_73 (Dropout)	(None, 123, 32)	0
max_pooling1d_73 (MaxPooling)	(None, 24, 32)	0
flatten_73 (Flatten)	(None, 768)	0
dense_145 (Dense)	(None, 16)	12304
dense_146 (Dense)	(None, 3)	51
<hr/>		

Total params: 23,843

Trainable params: 23,843

Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples  
Epoch 1/35  
- 10s - loss: 0.6672 - acc: 0.7102 - val\_loss: 0.4619 - val\_acc: 0.8399

Epoch 2/35  
- 1s - loss: 0.0601 - acc: 0.9857 - val\_loss: 0.1758 - val\_acc: 0.9503

Epoch 3/35  
- 1s - loss: 0.0087 - acc: 1.0000 - val\_loss: 0.2470 - val\_acc: 0.9445

Epoch 4/35  
- 1s - loss: 0.0036 - acc: 1.0000 - val\_loss: 0.1970 - val\_acc: 0.9560

Epoch 5/35  
- 1s - loss: 0.0017 - acc: 1.0000 - val\_loss: 0.2037 - val\_acc: 0.9575

Epoch 6/35  
- 1s - loss: 0.0011 - acc: 1.0000 - val\_loss: 0.2296 - val\_acc: 0.9582

Epoch 7/35  
- 1s - loss: 7.7039e-04 - acc: 1.0000 - val\_loss: 0.2238 - val\_acc: 0.9582

Epoch 8/35  
- 1s - loss: 6.1616e-04 - acc: 1.0000 - val\_loss: 0.2132 - val\_acc: 0.9567

Epoch 9/35  
- 1s - loss: 4.5827e-04 - acc: 1.0000 - val\_loss: 0.2402 - val\_acc: 0.9596

Epoch 10/35  
- 1s - loss: 3.7885e-04 - acc: 1.0000 - val\_loss: 0.2419 - val\_acc: 0.9582

Epoch 11/35  
- 1s - loss: 3.0392e-04 - acc: 1.0000 - val\_loss: 0.2587 - val\_acc: 0.9575

Epoch 12/35  
- 1s - loss: 2.6918e-04 - acc: 1.0000 - val\_loss: 0.2431 - val\_acc: 0.9589

Epoch 13/35  
- 1s - loss: 2.1750e-04 - acc: 1.0000 - val\_loss: 0.2420 - val\_acc: 0.9589

Epoch 14/35  
- 1s - loss: 1.8923e-04 - acc: 1.0000 - val\_loss: 0.2416 - val\_acc: 0.9603

Epoch 15/35  
- 1s - loss: 1.6418e-04 - acc: 1.0000 - val\_loss: 0.2482 - val\_acc: 0.9596

Epoch 16/35  
- 1s - loss: 1.3875e-04 - acc: 1.0000 - val\_loss: 0.2509 - val\_acc: 0.9596

Epoch 17/35  
- 1s - loss: 1.1965e-04 - acc: 1.0000 - val\_loss: 0.2501 - val\_acc: 0.9596

Epoch 18/35  
- 1s - loss: 1.0575e-04 - acc: 1.0000 - val\_loss: 0.2559 - val\_acc: 0.9596

Epoch 19/35  
- 1s - loss: 9.4306e-05 - acc: 1.0000 - val\_loss: 0.2464 - val\_acc: 0.9611

Epoch 20/35  
- 1s - loss: 8.5323e-05 - acc: 1.0000 - val\_loss: 0.2574 - val\_acc: 0.9596

Epoch 21/35  
- 1s - loss: 7.6860e-05 - acc: 1.0000 - val\_loss: 0.2703 - val\_acc: 0.9575

Epoch 22/35  
- 1s - loss: 6.8285e-05 - acc: 1.0000 - val\_loss: 0.2683 - val\_acc: 0.9575

Epoch 23/35  
- 1s - loss: 6.1742e-05 - acc: 1.0000 - val\_loss: 0.2652 - val\_acc: 0.9589

Epoch 24/35  
- 1s - loss: 5.6221e-05 - acc: 1.0000 - val\_loss: 0.2703 - val\_acc: 0.9582

Epoch 25/35  
- 1s - loss: 5.0008e-05 - acc: 1.0000 - val\_loss: 0.2748 - val\_acc: 0.9575

```

Epoch 26/35
- 1s - loss: 4.5212e-05 - acc: 1.0000 - val_loss: 0.2733 - val_acc: 0.9596

Epoch 27/35
- 1s - loss: 4.0680e-05 - acc: 1.0000 - val_loss: 0.2753 - val_acc: 0.9575

Epoch 28/35
- 1s - loss: 3.7987e-05 - acc: 1.0000 - val_loss: 0.2744 - val_acc: 0.9596

Epoch 29/35
- 1s - loss: 3.4272e-05 - acc: 1.0000 - val_loss: 0.2731 - val_acc: 0.9589

Epoch 30/35
- 1s - loss: 3.1242e-05 - acc: 1.0000 - val_loss: 0.2844 - val_acc: 0.9596

Epoch 31/35
- 1s - loss: 2.9613e-05 - acc: 1.0000 - val_loss: 0.2838 - val_acc: 0.9596

Epoch 32/35
- 1s - loss: 2.6408e-05 - acc: 1.0000 - val_loss: 0.2825 - val_acc: 0.9582

Epoch 33/35
- 1s - loss: 2.4943e-05 - acc: 1.0000 - val_loss: 0.2858 - val_acc: 0.9596

Epoch 34/35
- 1s - loss: 2.1728e-05 - acc: 1.0000 - val_loss: 0.2855 - val_acc: 0.9596

Epoch 35/35
- 1s - loss: 2.0694e-05 - acc: 1.0000 - val_loss: 0.2853 - val_acc: 0.9582

Test accuracy:
0.9581831290555155
Model: "sequential_74"

Layer (type)          Output Shape         Param #
=====
conv1d_147 (Conv1D)    (None, 124, 64)      2944
conv1d_148 (Conv1D)    (None, 121, 32)       8224
dropout_74 (Dropout)   (None, 121, 32)       0
max_pooling1d_74 (MaxPooling) (None, 24, 32)   0
flatten_74 (Flatten)   (None, 768)           0
dense_147 (Dense)     (None, 32)            24608
dense_148 (Dense)     (None, 3)              99
=====
Total params: 35,875
Trainable params: 35,875
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples
Epoch 1/35
- 10s - loss: 0.6205 - acc: 0.7458 - val_loss: 0.3695 - val_acc: 0.8962

Epoch 2/35
- 1s - loss: 0.0540 - acc: 0.9890 - val_loss: 0.2253 - val_acc: 0.9337

Epoch 3/35
- 1s - loss: 0.0070 - acc: 0.9997 - val_loss: 0.1661 - val_acc: 0.9531

Epoch 4/35
- 1s - loss: 0.0040 - acc: 0.9997 - val_loss: 0.1496 - val_acc: 0.9596

Epoch 5/35
- 1s - loss: 0.0016 - acc: 1.0000 - val_loss: 0.1401 - val_acc: 0.9654

Epoch 6/35
- 1s - loss: 9.2150e-04 - acc: 1.0000 - val_loss: 0.1706 - val_acc: 0.9567

Epoch 7/35
- 1s - loss: 6.7712e-04 - acc: 1.0000 - val_loss: 0.1633 - val_acc: 0.9625

Epoch 8/35

```

```
- 1s - loss: 5.0960e-04 - acc: 1.0000 - val_loss: 0.1534 - val_acc: 0.9647

Epoch 9/35
- 1s - loss: 4.2568e-04 - acc: 1.0000 - val_loss: 0.1503 - val_acc: 0.9668

Epoch 10/35
- 1s - loss: 3.2907e-04 - acc: 1.0000 - val_loss: 0.1572 - val_acc: 0.9676

Epoch 11/35
- 1s - loss: 2.6477e-04 - acc: 1.0000 - val_loss: 0.1611 - val_acc: 0.9661

Epoch 12/35
- 1s - loss: 2.3021e-04 - acc: 1.0000 - val_loss: 0.1640 - val_acc: 0.9668

Epoch 13/35
- 1s - loss: 1.9439e-04 - acc: 1.0000 - val_loss: 0.1799 - val_acc: 0.9647

Epoch 14/35
- 1s - loss: 1.6572e-04 - acc: 1.0000 - val_loss: 0.1666 - val_acc: 0.9668

Epoch 15/35
- 1s - loss: 1.4672e-04 - acc: 1.0000 - val_loss: 0.1661 - val_acc: 0.9668

Epoch 16/35
- 1s - loss: 1.2125e-04 - acc: 1.0000 - val_loss: 0.1686 - val_acc: 0.9676

Epoch 17/35
- 1s - loss: 1.1266e-04 - acc: 1.0000 - val_loss: 0.1629 - val_acc: 0.9690

Epoch 18/35
- 1s - loss: 1.0722e-04 - acc: 1.0000 - val_loss: 0.1656 - val_acc: 0.9683

Epoch 19/35
- 1s - loss: 9.0253e-05 - acc: 1.0000 - val_loss: 0.1700 - val_acc: 0.9683

Epoch 20/35
- 1s - loss: 7.6906e-05 - acc: 1.0000 - val_loss: 0.1785 - val_acc: 0.9654

Epoch 21/35
- 1s - loss: 7.1785e-05 - acc: 1.0000 - val_loss: 0.1791 - val_acc: 0.9668

Epoch 22/35
- 1s - loss: 6.3796e-05 - acc: 1.0000 - val_loss: 0.1768 - val_acc: 0.9683

Epoch 23/35
- 1s - loss: 6.1629e-05 - acc: 1.0000 - val_loss: 0.1756 - val_acc: 0.9676

Epoch 24/35
- 1s - loss: 6.5220e-05 - acc: 1.0000 - val_loss: 0.1846 - val_acc: 0.9683

Epoch 25/35
- 1s - loss: 4.2295e-05 - acc: 1.0000 - val_loss: 0.1840 - val_acc: 0.9661

Epoch 26/35
- 1s - loss: 3.9931e-05 - acc: 1.0000 - val_loss: 0.1793 - val_acc: 0.9676

Epoch 27/35
- 1s - loss: 3.6143e-05 - acc: 1.0000 - val_loss: 0.1832 - val_acc: 0.9676

Epoch 28/35
- 1s - loss: 3.1580e-05 - acc: 1.0000 - val_loss: 0.1758 - val_acc: 0.9676

Epoch 29/35
- 1s - loss: 3.1947e-05 - acc: 1.0000 - val_loss: 0.1790 - val_acc: 0.9704

Epoch 30/35
- 1s - loss: 2.8816e-05 - acc: 1.0000 - val_loss: 0.1875 - val_acc: 0.9690

Epoch 31/35
- 1s - loss: 2.5558e-05 - acc: 1.0000 - val_loss: 0.1866 - val_acc: 0.9676

Epoch 32/35
- 1s - loss: 2.1975e-05 - acc: 1.0000 - val_loss: 0.1909 - val_acc: 0.9676

Epoch 33/35
- 1s - loss: 2.1611e-05 - acc: 1.0000 - val_loss: 0.1865 - val_acc: 0.9676
```

```

Epoch 34/35
- 1s - loss: 1.9213e-05 - acc: 1.0000 - val_loss: 0.1926 - val_acc: 0.9676

Epoch 35/35
- 1s - loss: 1.7839e-05 - acc: 1.0000 - val_loss: 0.1908 - val_acc: 0.9676

Test accuracy:
0.9675558759913482
Model: "sequential_75"

Layer (type)          Output Shape         Param #
=====
conv1d_149 (Conv1D)    (None, 124, 64)      2944
conv1d_150 (Conv1D)    (None, 121, 32)       8224
dropout_75 (Dropout)   (None, 121, 32)       0
max_pooling1d_75 (MaxPooling) (None, 24, 32)  0
flatten_75 (Flatten)   (None, 768)           0
dense_149 (Dense)     (None, 16)            12304
dense_150 (Dense)     (None, 3)              51
=====
Total params: 23,523
Trainable params: 23,523
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples
Epoch 1/25
- 10s - loss: 0.7361 - acc: 0.6922 - val_loss: 0.4450 - val_acc: 0.8414

Epoch 2/25
- 1s - loss: 0.0806 - acc: 0.9769 - val_loss: 0.1829 - val_acc: 0.9380

Epoch 3/25
- 1s - loss: 0.0172 - acc: 0.9970 - val_loss: 0.1943 - val_acc: 0.9488

Epoch 4/25
- 1s - loss: 0.0057 - acc: 0.9994 - val_loss: 0.1699 - val_acc: 0.9531

Epoch 5/25
- 1s - loss: 0.0027 - acc: 1.0000 - val_loss: 0.1815 - val_acc: 0.9503

Epoch 6/25
- 1s - loss: 0.0024 - acc: 1.0000 - val_loss: 0.1953 - val_acc: 0.9524

Epoch 7/25
- 1s - loss: 0.0010 - acc: 1.0000 - val_loss: 0.1695 - val_acc: 0.9582

Epoch 8/25
- 1s - loss: 6.7315e-04 - acc: 1.0000 - val_loss: 0.1742 - val_acc: 0.9567

Epoch 9/25
- 1s - loss: 6.4138e-04 - acc: 1.0000 - val_loss: 0.1722 - val_acc: 0.9546

Epoch 10/25
- 1s - loss: 4.5515e-04 - acc: 1.0000 - val_loss: 0.1652 - val_acc: 0.9589

Epoch 11/25
- 1s - loss: 3.9644e-04 - acc: 1.0000 - val_loss: 0.1696 - val_acc: 0.9582

Epoch 12/25
- 1s - loss: 4.8667e-04 - acc: 1.0000 - val_loss: 0.1979 - val_acc: 0.9560

Epoch 13/25
- 1s - loss: 2.4702e-04 - acc: 1.0000 - val_loss: 0.1849 - val_acc: 0.9575

Epoch 14/25
- 1s - loss: 1.9472e-04 - acc: 1.0000 - val_loss: 0.1824 - val_acc: 0.9582

Epoch 15/25
- 1s - loss: 3.2868e-04 - acc: 1.0000 - val_loss: 0.1853 - val_acc: 0.9596

Epoch 16/25

```

```

- 1s - loss: 2.7069e-04 - acc: 1.0000 - val_loss: 0.1791 - val_acc: 0.9618

Epoch 17/25
- 1s - loss: 2.2287e-04 - acc: 1.0000 - val_loss: 0.1634 - val_acc: 0.9618

Epoch 18/25
- 1s - loss: 1.9772e-04 - acc: 1.0000 - val_loss: 0.1724 - val_acc: 0.9625

Epoch 19/25
- 1s - loss: 1.9612e-04 - acc: 1.0000 - val_loss: 0.1872 - val_acc: 0.9589

Epoch 20/25
- 1s - loss: 9.7451e-05 - acc: 1.0000 - val_loss: 0.1818 - val_acc: 0.9603

Epoch 21/25
- 1s - loss: 7.2756e-05 - acc: 1.0000 - val_loss: 0.1921 - val_acc: 0.9603

Epoch 22/25
- 1s - loss: 6.1329e-05 - acc: 1.0000 - val_loss: 0.1755 - val_acc: 0.9603

Epoch 23/25
- 1s - loss: 6.0502e-05 - acc: 1.0000 - val_loss: 0.1869 - val_acc: 0.9603

Epoch 24/25
- 1s - loss: 5.9009e-05 - acc: 1.0000 - val_loss: 0.1861 - val_acc: 0.9603

Epoch 25/25
- 1s - loss: 4.7366e-05 - acc: 1.0000 - val_loss: 0.1751 - val_acc: 0.9618

Test accuracy:
0.9617880317231434
Model: "sequential_76"

Layer (type)          Output Shape         Param #
=====
conv1d_151 (Conv1D)    (None, 125, 64)      2368
conv1d_152 (Conv1D)    (None, 122, 32)       8224
dropout_76 (Dropout)   (None, 122, 32)       0
max_pooling1d_76 (MaxPooling) (None, 24, 32)  0
flatten_76 (Flatten)   (None, 768)           0
dense_151 (Dense)     (None, 32)            24608
dense_152 (Dense)     (None, 3)              99
=====
Total params: 35,299
Trainable params: 35,299
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples
Epoch 1/30
- 1s - loss: 1.0441 - acc: 0.5747 - val_loss: 0.6944 - val_acc: 0.7678

Epoch 2/30
- 1s - loss: 0.2532 - acc: 0.9190 - val_loss: 0.3040 - val_acc: 0.9236

Epoch 3/30
- 1s - loss: 0.0601 - acc: 0.9857 - val_loss: 0.2046 - val_acc: 0.9524

Epoch 4/30
- 1s - loss: 0.0301 - acc: 0.9918 - val_loss: 0.1800 - val_acc: 0.9589

Epoch 5/30
- 1s - loss: 0.0125 - acc: 0.9973 - val_loss: 0.1735 - val_acc: 0.9510

Epoch 6/30
- 1s - loss: 0.0265 - acc: 0.9915 - val_loss: 0.1678 - val_acc: 0.9618

Epoch 7/30
- 1s - loss: 0.0071 - acc: 0.9988 - val_loss: 0.1635 - val_acc: 0.9582

Epoch 8/30
- 1s - loss: 0.0024 - acc: 1.0000 - val_loss: 0.1654 - val_acc: 0.9596

```

Epoch 9/30  
 - 1s - loss: 0.0020 - acc: 1.0000 - val\_loss: 0.1384 - val\_acc: 0.9618

Epoch 10/30  
 - 1s - loss: 0.0024 - acc: 0.9997 - val\_loss: 0.1501 - val\_acc: 0.9661

Epoch 11/30  
 - 1s - loss: 0.0013 - acc: 1.0000 - val\_loss: 0.1486 - val\_acc: 0.9647

Epoch 12/30  
 - 1s - loss: 0.0013 - acc: 1.0000 - val\_loss: 0.1677 - val\_acc: 0.9611

Epoch 13/30  
 - 1s - loss: 0.0031 - acc: 0.9991 - val\_loss: 0.1700 - val\_acc: 0.9654

Epoch 14/30  
 - 1s - loss: 0.0014 - acc: 0.9997 - val\_loss: 0.1639 - val\_acc: 0.9676

Epoch 15/30  
 - 1s - loss: 8.7924e-04 - acc: 0.9997 - val\_loss: 0.1675 - val\_acc: 0.9640

Epoch 16/30  
 - 1s - loss: 4.8147e-04 - acc: 1.0000 - val\_loss: 0.1711 - val\_acc: 0.9647

Epoch 17/30  
 - 1s - loss: 4.5849e-04 - acc: 1.0000 - val\_loss: 0.1900 - val\_acc: 0.9632

Epoch 18/30  
 - 1s - loss: 4.5356e-04 - acc: 1.0000 - val\_loss: 0.2016 - val\_acc: 0.9625

Epoch 19/30  
 - 1s - loss: 4.1266e-04 - acc: 1.0000 - val\_loss: 0.1806 - val\_acc: 0.9647

Epoch 20/30  
 - 1s - loss: 3.0333e-04 - acc: 1.0000 - val\_loss: 0.1816 - val\_acc: 0.9632

Epoch 21/30  
 - 1s - loss: 4.1352e-04 - acc: 1.0000 - val\_loss: 0.1980 - val\_acc: 0.9632

Epoch 22/30  
 - 1s - loss: 2.7290e-04 - acc: 1.0000 - val\_loss: 0.1876 - val\_acc: 0.9640

Epoch 23/30  
 - 1s - loss: 1.9472e-04 - acc: 1.0000 - val\_loss: 0.1750 - val\_acc: 0.9640

Epoch 24/30  
 - 1s - loss: 4.1607e-04 - acc: 1.0000 - val\_loss: 0.2079 - val\_acc: 0.9575

Epoch 25/30  
 - 1s - loss: 6.1308e-04 - acc: 1.0000 - val\_loss: 0.2110 - val\_acc: 0.9611

Epoch 26/30  
 - 1s - loss: 0.0035 - acc: 0.9991 - val\_loss: 0.4790 - val\_acc: 0.8767

Epoch 27/30  
 - 1s - loss: 0.0137 - acc: 0.9957 - val\_loss: 0.1709 - val\_acc: 0.9676

Epoch 28/30  
 - 1s - loss: 0.0040 - acc: 0.9988 - val\_loss: 0.2098 - val\_acc: 0.9668

Epoch 29/30  
 - 1s - loss: 4.4654e-04 - acc: 1.0000 - val\_loss: 0.1965 - val\_acc: 0.9647

Epoch 30/30  
 - 1s - loss: 4.2316e-04 - acc: 1.0000 - val\_loss: 0.2060 - val\_acc: 0.9618

Test accuracy:  
0.9617880317231434  
Model: "sequential\_77"

Layer (type)	Output Shape	Param #
conv1d_153 (Conv1D)	(None, 124, 32)	1472
conv1d_154 (Conv1D)	(None, 120, 64)	10304

dropout_77 (Dropout)	(None, 120, 64)	0
max_pooling1d_77 (MaxPooling)	(None, 24, 64)	0
flatten_77 (Flatten)	(None, 1536)	0
dense_153 (Dense)	(None, 32)	49184
dense_154 (Dense)	(None, 3)	99
=====		
Total params:	61,059	
Trainable params:	61,059	
Non-trainable params:	0	
=====		
Train on 3285 samples, validate on 1387 samples		
Epoch 1/35		
- 1s - loss: 0.6226 - acc: 0.7394 - val_loss: 0.2166 - val_acc: 0.9438		
Epoch 2/35		
- 1s - loss: 0.0322 - acc: 0.9963 - val_loss: 0.1840 - val_acc: 0.9575		
Epoch 3/35		
- 1s - loss: 0.0069 - acc: 0.9997 - val_loss: 0.1533 - val_acc: 0.9647		
Epoch 4/35		
- 1s - loss: 0.0028 - acc: 1.0000 - val_loss: 0.1607 - val_acc: 0.9748		
Epoch 5/35		
- 1s - loss: 0.0013 - acc: 1.0000 - val_loss: 0.1583 - val_acc: 0.9719		
Epoch 6/35		
- 1s - loss: 8.5321e-04 - acc: 1.0000 - val_loss: 0.1589 - val_acc: 0.9748		
Epoch 7/35		
- 1s - loss: 6.8438e-04 - acc: 1.0000 - val_loss: 0.1695 - val_acc: 0.9726		
Epoch 8/35		
- 1s - loss: 5.0310e-04 - acc: 1.0000 - val_loss: 0.1662 - val_acc: 0.9748		
Epoch 9/35		
- 1s - loss: 4.1674e-04 - acc: 1.0000 - val_loss: 0.1729 - val_acc: 0.9755		
Epoch 10/35		
- 1s - loss: 3.0810e-04 - acc: 1.0000 - val_loss: 0.1629 - val_acc: 0.9748		
Epoch 11/35		
- 1s - loss: 2.5163e-04 - acc: 1.0000 - val_loss: 0.1763 - val_acc: 0.9755		
Epoch 12/35		
- 1s - loss: 2.3358e-04 - acc: 1.0000 - val_loss: 0.1761 - val_acc: 0.9755		
Epoch 13/35		
- 1s - loss: 1.7180e-04 - acc: 1.0000 - val_loss: 0.1720 - val_acc: 0.9762		
Epoch 14/35		
- 1s - loss: 1.5421e-04 - acc: 1.0000 - val_loss: 0.1801 - val_acc: 0.9755		
Epoch 15/35		
- 1s - loss: 1.3132e-04 - acc: 1.0000 - val_loss: 0.1800 - val_acc: 0.9755		
Epoch 16/35		
- 1s - loss: 1.2422e-04 - acc: 1.0000 - val_loss: 0.1865 - val_acc: 0.9748		
Epoch 17/35		
- 1s - loss: 9.6231e-05 - acc: 1.0000 - val_loss: 0.1848 - val_acc: 0.9755		
Epoch 18/35		
- 1s - loss: 8.8946e-05 - acc: 1.0000 - val_loss: 0.1892 - val_acc: 0.9762		
Epoch 19/35		
- 1s - loss: 7.6806e-05 - acc: 1.0000 - val_loss: 0.1822 - val_acc: 0.9755		
Epoch 20/35		
- 1s - loss: 6.9056e-05 - acc: 1.0000 - val_loss: 0.1904 - val_acc: 0.9755		
Epoch 21/35		
- 1s - loss: 6.3501e-05 - acc: 1.0000 - val_loss: 0.1921 - val_acc: 0.9755		

```
Epoch 22/35
- 1s - loss: 5.8704e-05 - acc: 1.0000 - val_loss: 0.1942 - val_acc: 0.9791

Epoch 23/35
- 1s - loss: 5.8294e-05 - acc: 1.0000 - val_loss: 0.1914 - val_acc: 0.9762

Epoch 24/35
- 1s - loss: 4.8645e-05 - acc: 1.0000 - val_loss: 0.1935 - val_acc: 0.9769

Epoch 25/35
- 1s - loss: 4.1485e-05 - acc: 1.0000 - val_loss: 0.1988 - val_acc: 0.9769

Epoch 26/35
- 1s - loss: 3.6657e-05 - acc: 1.0000 - val_loss: 0.1967 - val_acc: 0.9776

Epoch 27/35
- 1s - loss: 3.5319e-05 - acc: 1.0000 - val_loss: 0.1963 - val_acc: 0.9762

Epoch 28/35
- 1s - loss: 4.0433e-05 - acc: 1.0000 - val_loss: 0.1941 - val_acc: 0.9762

Epoch 29/35
- 1s - loss: 2.9387e-05 - acc: 1.0000 - val_loss: 0.2028 - val_acc: 0.9776

Epoch 30/35
- 1s - loss: 2.6898e-05 - acc: 1.0000 - val_loss: 0.1967 - val_acc: 0.9776

Epoch 31/35
- 1s - loss: 2.5395e-05 - acc: 1.0000 - val_loss: 0.2060 - val_acc: 0.9776

Epoch 32/35
- 1s - loss: 2.2393e-05 - acc: 1.0000 - val_loss: 0.2057 - val_acc: 0.9762

Epoch 33/35
- 1s - loss: 2.0767e-05 - acc: 1.0000 - val_loss: 0.2062 - val_acc: 0.9784

Epoch 34/35
- 1s - loss: 1.9838e-05 - acc: 1.0000 - val_loss: 0.2071 - val_acc: 0.9762

Epoch 35/35
- 1s - loss: 1.9337e-05 - acc: 1.0000 - val_loss: 0.2091 - val_acc: 0.9784
```

Test accuracy:  
0.9783705839942322  
Model: "sequential\_78"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_155 (Conv1D)	(None, 124, 32)	1472
conv1d_156 (Conv1D)	(None, 120, 64)	10304
dropout_78 (Dropout)	(None, 120, 64)	0
max_pooling1d_78 (MaxPooling)	(None, 24, 64)	0
flatten_78 (Flatten)	(None, 1536)	0
dense_155 (Dense)	(None, 32)	49184
dense_156 (Dense)	(None, 3)	99
<hr/>		
Total params:	61,059	
Trainable params:	61,059	
Non-trainable params:	0	

---

```
Train on 3285 samples, validate on 1387 samples
Epoch 1/35
- 10s - loss: 0.6036 - acc: 0.7406 - val_loss: 0.3478 - val_acc: 0.9012

Epoch 2/35
- 1s - loss: 0.0525 - acc: 0.9890 - val_loss: 0.1445 - val_acc: 0.9575

Epoch 3/35
- 1s - loss: 0.0081 - acc: 0.9997 - val_loss: 0.1573 - val_acc: 0.9575
```

```
Epoch 4/35
- 1s - loss: 0.0123 - acc: 0.9967 - val_loss: 0.1963 - val_acc: 0.9438

Epoch 5/35
- 1s - loss: 0.0022 - acc: 1.0000 - val_loss: 0.1525 - val_acc: 0.9618

Epoch 6/35
- 1s - loss: 7.1749e-04 - acc: 1.0000 - val_loss: 0.1525 - val_acc: 0.9676

Epoch 7/35
- 1s - loss: 4.5241e-04 - acc: 1.0000 - val_loss: 0.1504 - val_acc: 0.9676

Epoch 8/35
- 1s - loss: 3.7000e-04 - acc: 1.0000 - val_loss: 0.1508 - val_acc: 0.9683

Epoch 9/35
- 1s - loss: 2.9113e-04 - acc: 1.0000 - val_loss: 0.1641 - val_acc: 0.9654

Epoch 10/35
- 1s - loss: 2.2040e-04 - acc: 1.0000 - val_loss: 0.1654 - val_acc: 0.9676

Epoch 11/35
- 1s - loss: 1.8391e-04 - acc: 1.0000 - val_loss: 0.1687 - val_acc: 0.9668

Epoch 12/35
- 1s - loss: 1.6111e-04 - acc: 1.0000 - val_loss: 0.1627 - val_acc: 0.9683

Epoch 13/35
- 1s - loss: 1.4420e-04 - acc: 1.0000 - val_loss: 0.1691 - val_acc: 0.9668

Epoch 14/35
- 1s - loss: 1.2822e-04 - acc: 1.0000 - val_loss: 0.1723 - val_acc: 0.9668

Epoch 15/35
- 1s - loss: 1.1983e-04 - acc: 1.0000 - val_loss: 0.1808 - val_acc: 0.9654

Epoch 16/35
- 1s - loss: 9.2028e-05 - acc: 1.0000 - val_loss: 0.1750 - val_acc: 0.9676

Epoch 17/35
- 1s - loss: 8.4962e-05 - acc: 1.0000 - val_loss: 0.1733 - val_acc: 0.9690

Epoch 18/35
- 1s - loss: 8.1213e-05 - acc: 1.0000 - val_loss: 0.1795 - val_acc: 0.9676

Epoch 19/35
- 1s - loss: 6.1916e-05 - acc: 1.0000 - val_loss: 0.1806 - val_acc: 0.9668

Epoch 20/35
- 1s - loss: 5.9880e-05 - acc: 1.0000 - val_loss: 0.1881 - val_acc: 0.9647

Epoch 21/35
- 1s - loss: 5.7620e-05 - acc: 1.0000 - val_loss: 0.1796 - val_acc: 0.9668

Epoch 22/35
- 1s - loss: 4.5296e-05 - acc: 1.0000 - val_loss: 0.1839 - val_acc: 0.9690

Epoch 23/35
- 1s - loss: 4.8156e-05 - acc: 1.0000 - val_loss: 0.1887 - val_acc: 0.9654

Epoch 24/35
- 1s - loss: 5.3652e-05 - acc: 1.0000 - val_loss: 0.1860 - val_acc: 0.9676

Epoch 25/35
- 1s - loss: 4.0736e-05 - acc: 1.0000 - val_loss: 0.1890 - val_acc: 0.9676

Epoch 26/35
- 1s - loss: 3.4427e-05 - acc: 1.0000 - val_loss: 0.1924 - val_acc: 0.9661

Epoch 27/35
- 1s - loss: 2.9042e-05 - acc: 1.0000 - val_loss: 0.1908 - val_acc: 0.9690

Epoch 28/35
- 1s - loss: 2.4911e-05 - acc: 1.0000 - val_loss: 0.1872 - val_acc: 0.9690

Epoch 29/35
- 1s - loss: 2.5244e-05 - acc: 1.0000 - val_loss: 0.1882 - val_acc: 0.9683
```

```
1s    loss: 2.0041e-05 - acc: 1.0000 - val_loss: 0.1902 - val_acc: 0.9683
Epoch 30/35
- 1s - loss: 2.4842e-05 - acc: 1.0000 - val_loss: 0.1901 - val_acc: 0.9683
Epoch 31/35
- 1s - loss: 2.4851e-05 - acc: 1.0000 - val_loss: 0.1926 - val_acc: 0.9683
Epoch 32/35
- 1s - loss: 2.0101e-05 - acc: 1.0000 - val_loss: 0.1952 - val_acc: 0.9683
Epoch 33/35
- 1s - loss: 1.9813e-05 - acc: 1.0000 - val_loss: 0.2048 - val_acc: 0.9647
Epoch 34/35
- 1s - loss: 1.6645e-05 - acc: 1.0000 - val_loss: 0.1934 - val_acc: 0.9690
Epoch 35/35
- 1s - loss: 1.6090e-05 - acc: 1.0000 - val_loss: 0.1934 - val_acc: 0.9690
```

Test accuracy:  
0.9689978370583994  
Model: "sequential\_79"

Layer (type)	Output Shape	Param #
conv1d_157 (Conv1D)	(None, 124, 32)	1472
conv1d_158 (Conv1D)	(None, 120, 64)	10304
dropout_79 (Dropout)	(None, 120, 64)	0
max_pooling1d_79 (MaxPooling)	(None, 24, 64)	0
flatten_79 (Flatten)	(None, 1536)	0
dense_157 (Dense)	(None, 16)	24592
dense_158 (Dense)	(None, 3)	51

Total params: 36,419  
Trainable params: 36,419  
Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples  
Epoch 1/35

- 1s - loss: 0.6813 - acc: 0.7254 - val\_loss: 0.4253 - val\_acc: 0.8717

Epoch 2/35

- 1s - loss: 0.1265 - acc: 0.9674 - val\_loss: 0.2856 - val\_acc: 0.9322

Epoch 3/35

- 1s - loss: 0.0328 - acc: 0.9942 - val\_loss: 0.3470 - val\_acc: 0.9293

Epoch 4/35

- 1s - loss: 0.0118 - acc: 0.9997 - val\_loss: 0.3523 - val\_acc: 0.9308

Epoch 5/35

- 1s - loss: 0.0063 - acc: 1.0000 - val\_loss: 0.3400 - val\_acc: 0.9380

Epoch 6/35

- 1s - loss: 0.0040 - acc: 1.0000 - val\_loss: 0.3446 - val\_acc: 0.9358

Epoch 7/35

- 1s - loss: 0.0026 - acc: 1.0000 - val\_loss: 0.3720 - val\_acc: 0.9366

Epoch 8/35

- 1s - loss: 0.0018 - acc: 1.0000 - val\_loss: 0.3457 - val\_acc: 0.9402

Epoch 9/35

- 1s - loss: 0.0013 - acc: 1.0000 - val\_loss: 0.3455 - val\_acc: 0.9438

Epoch 10/35

- 1s - loss: 0.0012 - acc: 1.0000 - val\_loss: 0.3562 - val\_acc: 0.9445

Epoch 11/35

- 1s - loss: 8.2949e-04 - acc: 1.0000 - val\_loss: 0.3772 - val\_acc: 0.9402

```
Epoch 12/35
- 1s - loss: 7.1628e-04 - acc: 1.0000 - val_loss: 0.3716 - val_acc: 0.9416

Epoch 13/35
- 1s - loss: 5.7696e-04 - acc: 1.0000 - val_loss: 0.3699 - val_acc: 0.9430

Epoch 14/35
- 1s - loss: 4.4178e-04 - acc: 1.0000 - val_loss: 0.3632 - val_acc: 0.9459

Epoch 15/35
- 1s - loss: 4.0005e-04 - acc: 1.0000 - val_loss: 0.3592 - val_acc: 0.9481

Epoch 16/35
- 1s - loss: 3.4103e-04 - acc: 1.0000 - val_loss: 0.3707 - val_acc: 0.9466

Epoch 17/35
- 1s - loss: 2.9113e-04 - acc: 1.0000 - val_loss: 0.3765 - val_acc: 0.9474

Epoch 18/35
- 1s - loss: 2.4257e-04 - acc: 1.0000 - val_loss: 0.3677 - val_acc: 0.9459

Epoch 19/35
- 1s - loss: 2.3954e-04 - acc: 1.0000 - val_loss: 0.3707 - val_acc: 0.9466

Epoch 20/35
- 1s - loss: 2.0241e-04 - acc: 1.0000 - val_loss: 0.3739 - val_acc: 0.9474

Epoch 21/35
- 1s - loss: 1.7593e-04 - acc: 1.0000 - val_loss: 0.3712 - val_acc: 0.9495

Epoch 22/35
- 1s - loss: 1.5610e-04 - acc: 1.0000 - val_loss: 0.3700 - val_acc: 0.9503

Epoch 23/35
- 1s - loss: 1.4256e-04 - acc: 1.0000 - val_loss: 0.3693 - val_acc: 0.9488

Epoch 24/35
- 1s - loss: 1.2724e-04 - acc: 1.0000 - val_loss: 0.3707 - val_acc: 0.9503

Epoch 25/35
- 1s - loss: 1.1458e-04 - acc: 1.0000 - val_loss: 0.3743 - val_acc: 0.9495

Epoch 26/35
- 1s - loss: 1.0133e-04 - acc: 1.0000 - val_loss: 0.3762 - val_acc: 0.9481

Epoch 27/35
- 1s - loss: 1.0124e-04 - acc: 1.0000 - val_loss: 0.3734 - val_acc: 0.9510

Epoch 28/35
- 1s - loss: 8.5018e-05 - acc: 1.0000 - val_loss: 0.3722 - val_acc: 0.9524

Epoch 29/35
- 1s - loss: 7.9770e-05 - acc: 1.0000 - val_loss: 0.3737 - val_acc: 0.9517

Epoch 30/35
- 1s - loss: 6.9941e-05 - acc: 1.0000 - val_loss: 0.3804 - val_acc: 0.9495

Epoch 31/35
- 1s - loss: 7.0058e-05 - acc: 1.0000 - val_loss: 0.3802 - val_acc: 0.9503

Epoch 32/35
- 1s - loss: 6.0016e-05 - acc: 1.0000 - val_loss: 0.3739 - val_acc: 0.9503

Epoch 33/35
- 1s - loss: 5.5544e-05 - acc: 1.0000 - val_loss: 0.3794 - val_acc: 0.9510

Epoch 34/35
- 1s - loss: 5.3617e-05 - acc: 1.0000 - val_loss: 0.3774 - val_acc: 0.9531

Epoch 35/35
- 1s - loss: 4.6523e-05 - acc: 1.0000 - val_loss: 0.3761 - val_acc: 0.9510

Test accuracy:
0.9509733237202596
Model: "sequential_80"
```

Layer (Type)	Output Shape	Param #

layer (type)	output shape	param #
conv1d_159 (Conv1D)	(None, 127, 32)	608
conv1d_160 (Conv1D)	(None, 123, 64)	10304
dropout_80 (Dropout)	(None, 123, 64)	0
max_pooling1d_80 (MaxPooling)	(None, 24, 64)	0
flatten_80 (Flatten)	(None, 1536)	0
dense_159 (Dense)	(None, 32)	49184
dense_160 (Dense)	(None, 3)	99

Total params: 60,195

Trainable params: 60,195

Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples

Epoch 1/30

- 1s - loss: 0.7319 - acc: 0.6779 - val\_loss: 0.5013 - val\_acc: 0.7902

Epoch 2/30

- 1s - loss: 0.0918 - acc: 0.9714 - val\_loss: 0.1663 - val\_acc: 0.9531

Epoch 3/30

- 1s - loss: 0.0177 - acc: 0.9970 - val\_loss: 0.1889 - val\_acc: 0.9315

Epoch 4/30

- 1s - loss: 0.0066 - acc: 0.9997 - val\_loss: 0.2016 - val\_acc: 0.9207

Epoch 5/30

- 1s - loss: 0.0033 - acc: 1.0000 - val\_loss: 0.1903 - val\_acc: 0.9351

Epoch 6/30

- 1s - loss: 0.0046 - acc: 0.9988 - val\_loss: 0.3384 - val\_acc: 0.8825

Epoch 7/30

- 1s - loss: 0.0124 - acc: 0.9957 - val\_loss: 0.2054 - val\_acc: 0.9257

Epoch 8/30

- 1s - loss: 0.0014 - acc: 1.0000 - val\_loss: 0.1777 - val\_acc: 0.9488

Epoch 9/30

- 1s - loss: 0.0020 - acc: 1.0000 - val\_loss: 0.1231 - val\_acc: 0.9589

Epoch 10/30

- 1s - loss: 4.9512e-04 - acc: 1.0000 - val\_loss: 0.1412 - val\_acc: 0.9546

Epoch 11/30

- 1s - loss: 3.2664e-04 - acc: 1.0000 - val\_loss: 0.1075 - val\_acc: 0.9647

Epoch 12/30

- 1s - loss: 3.1838e-04 - acc: 1.0000 - val\_loss: 0.1328 - val\_acc: 0.9596

Epoch 13/30

- 1s - loss: 2.4681e-04 - acc: 1.0000 - val\_loss: 0.1317 - val\_acc: 0.9589

Epoch 14/30

- 1s - loss: 2.0141e-04 - acc: 1.0000 - val\_loss: 0.1321 - val\_acc: 0.9596

Epoch 15/30

- 1s - loss: 1.6680e-04 - acc: 1.0000 - val\_loss: 0.1337 - val\_acc: 0.9596

Epoch 16/30

- 1s - loss: 2.1162e-04 - acc: 1.0000 - val\_loss: 0.1983 - val\_acc: 0.9344

Epoch 17/30

- 1s - loss: 1.2564e-04 - acc: 1.0000 - val\_loss: 0.1366 - val\_acc: 0.9596

Epoch 18/30

- 1s - loss: 1.1683e-04 - acc: 1.0000 - val\_loss: 0.1338 - val\_acc: 0.9589

Epoch 19/30

- 1s - loss: 1.1285e-04 - acc: 1.0000 - val\_loss: 0.1406 - val\_acc: 0.9589

```

Epoch 20/30
- 1s - loss: 1.8244e-04 - acc: 1.0000 - val_loss: 0.1125 - val_acc: 0.9647

Epoch 21/30
- 1s - loss: 8.4381e-05 - acc: 1.0000 - val_loss: 0.1206 - val_acc: 0.9632

Epoch 22/30
- 1s - loss: 7.3453e-05 - acc: 1.0000 - val_loss: 0.1208 - val_acc: 0.9640

Epoch 23/30
- 1s - loss: 6.9764e-05 - acc: 1.0000 - val_loss: 0.1399 - val_acc: 0.9603

Epoch 24/30
- 1s - loss: 7.8638e-05 - acc: 1.0000 - val_loss: 0.1503 - val_acc: 0.9575

Epoch 25/30
- 1s - loss: 6.5670e-05 - acc: 1.0000 - val_loss: 0.1404 - val_acc: 0.9603

Epoch 26/30
- 1s - loss: 5.5513e-05 - acc: 1.0000 - val_loss: 0.1325 - val_acc: 0.9611

Epoch 27/30
- 1s - loss: 5.1641e-05 - acc: 1.0000 - val_loss: 0.1617 - val_acc: 0.9582

Epoch 28/30
- 1s - loss: 8.1605e-05 - acc: 1.0000 - val_loss: 0.1511 - val_acc: 0.9589

Epoch 29/30
- 1s - loss: 8.1628e-05 - acc: 1.0000 - val_loss: 0.2122 - val_acc: 0.9387

Epoch 30/30
- 1s - loss: 5.7057e-05 - acc: 1.0000 - val_loss: 0.1269 - val_acc: 0.9647

Test accuracy:
0.9646719538572458
Model: "sequential_81"

```

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_161 (Conv1D)	(None, 124, 32)	1472
conv1d_162 (Conv1D)	(None, 120, 64)	10304
dropout_81 (Dropout)	(None, 120, 64)	0
max_pooling1d_81 (MaxPooling)	(None, 24, 64)	0
flatten_81 (Flatten)	(None, 1536)	0
dense_161 (Dense)	(None, 32)	49184
dense_162 (Dense)	(None, 3)	99
<hr/>		

```

Total params: 61,059
Trainable params: 61,059
Non-trainable params: 0

```

---

```

Train on 3285 samples, validate on 1387 samples
Epoch 1/35
- 10s - loss: 9.9059 - acc: 0.3714 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 2/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 3/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 4/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 5/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 6/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

```



```

Epoch 33/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 34/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Epoch 35/35
- 1s - loss: 10.1026 - acc: 0.3732 - val_loss: 10.3542 - val_acc: 0.3576

Test accuracy:
0.357606344628695
Model: "sequential_82"

Layer (type)          Output Shape         Param #
=====
conv1d_163 (Conv1D)    (None, 125, 32)      1184
conv1d_164 (Conv1D)    (None, 121, 64)       10304
dropout_82 (Dropout)   (None, 121, 64)       0
max_pooling1d_82 (MaxPooling) (None, 60, 64)  0
flatten_82 (Flatten)   (None, 3840)        0
dense_163 (Dense)     (None, 16)           61456
dense_164 (Dense)     (None, 3)            51
=====
Total params: 72,995
Trainable params: 72,995
Non-trainable params: 0

Train on 3285 samples, validate on 1387 samples
Epoch 1/30
- 1s - loss: 11.2442 - acc: 0.2992 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 2/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 3/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 4/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 5/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 6/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 7/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 8/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 9/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 10/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 11/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 12/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 13/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 14/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
-
```

```
Epoch 15/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 16/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 17/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 18/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 19/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 20/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 21/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 22/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 23/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 24/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 25/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 26/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 27/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 28/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 29/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028

Epoch 30/30
- 2s - loss: 11.2802 - acc: 0.3002 - val_loss: 11.2373 - val_acc: 0.3028
```

Test accuracy:  
0.3028118241022367  
Model: "sequential\_83"

Layer (type)	Output Shape	Param #
<hr/>		
conv1d_165 (Conv1D)	(None, 124, 32)	1472
conv1d_166 (Conv1D)	(None, 120, 16)	2576
dropout_83 (Dropout)	(None, 120, 16)	0
max_pooling1d_83 (MaxPooling)	(None, 24, 16)	0
flatten_83 (Flatten)	(None, 384)	0
dense_165 (Dense)	(None, 32)	12320
dense_166 (Dense)	(None, 3)	99
<hr/>		

Total params: 16,467  
Trainable params: 16,467  
Non-trainable params: 0

---

Train on 3285 samples, validate on 1387 samples  
Epoch 1/35  
- 11s - loss: 0.6256 - acc: 0.7346 - val\_loss: 0.4509 - val\_acc: 0.8529

Epoch 2/35

```
- 1s - loss: 0.0835 - acc: 0.9802 - val_loss: 0.2829 - val_acc: 0.9279

Epoch 3/35
- 1s - loss: 0.0199 - acc: 0.9973 - val_loss: 0.3142 - val_acc: 0.9380

Epoch 4/35
- 1s - loss: 0.0062 - acc: 0.9997 - val_loss: 0.3096 - val_acc: 0.9466

Epoch 5/35
- 1s - loss: 0.0028 - acc: 1.0000 - val_loss: 0.3044 - val_acc: 0.9488

Epoch 6/35
- 1s - loss: 0.0017 - acc: 1.0000 - val_loss: 0.3104 - val_acc: 0.9524

Epoch 7/35
- 1s - loss: 0.0011 - acc: 1.0000 - val_loss: 0.3303 - val_acc: 0.9539

Epoch 8/35
- 1s - loss: 8.4941e-04 - acc: 1.0000 - val_loss: 0.3318 - val_acc: 0.9517

Epoch 9/35
- 1s - loss: 6.3896e-04 - acc: 1.0000 - val_loss: 0.3465 - val_acc: 0.9517

Epoch 10/35
- 1s - loss: 5.0760e-04 - acc: 1.0000 - val_loss: 0.3498 - val_acc: 0.9510

Epoch 11/35
- 1s - loss: 4.2461e-04 - acc: 1.0000 - val_loss: 0.3553 - val_acc: 0.9517

Epoch 12/35
- 1s - loss: 3.6201e-04 - acc: 1.0000 - val_loss: 0.3456 - val_acc: 0.9531

Epoch 13/35
- 1s - loss: 2.8418e-04 - acc: 1.0000 - val_loss: 0.3554 - val_acc: 0.9517

Epoch 14/35
- 1s - loss: 2.4319e-04 - acc: 1.0000 - val_loss: 0.3541 - val_acc: 0.9524

Epoch 15/35
- 1s - loss: 2.0595e-04 - acc: 1.0000 - val_loss: 0.3566 - val_acc: 0.9524

Epoch 16/35
- 1s - loss: 1.7746e-04 - acc: 1.0000 - val_loss: 0.3580 - val_acc: 0.9517

Epoch 17/35
- 1s - loss: 1.5212e-04 - acc: 1.0000 - val_loss: 0.3569 - val_acc: 0.9517

Epoch 18/35
- 1s - loss: 1.3672e-04 - acc: 1.0000 - val_loss: 0.3605 - val_acc: 0.9531

Epoch 19/35
- 1s - loss: 1.1886e-04 - acc: 1.0000 - val_loss: 0.3687 - val_acc: 0.9517

Epoch 20/35
- 1s - loss: 1.0349e-04 - acc: 1.0000 - val_loss: 0.3669 - val_acc: 0.9524

Epoch 21/35
- 1s - loss: 9.5473e-05 - acc: 1.0000 - val_loss: 0.3664 - val_acc: 0.9531

Epoch 22/35
- 1s - loss: 8.4267e-05 - acc: 1.0000 - val_loss: 0.3781 - val_acc: 0.9510

Epoch 23/35
- 1s - loss: 7.6998e-05 - acc: 1.0000 - val_loss: 0.3747 - val_acc: 0.9517

Epoch 24/35
- 1s - loss: 6.9237e-05 - acc: 1.0000 - val_loss: 0.3694 - val_acc: 0.9531

Epoch 25/35
- 1s - loss: 6.0344e-05 - acc: 1.0000 - val_loss: 0.3685 - val_acc: 0.9531

Epoch 26/35
- 1s - loss: 5.8712e-05 - acc: 1.0000 - val_loss: 0.3750 - val_acc: 0.9531

Epoch 27/35
- 1s - loss: 5.0876e-05 - acc: 1.0000 - val_loss: 0.3834 - val_acc: 0.9503
```

```
Epoch 28/35
- 1s - loss: 4.6789e-05 - acc: 1.0000 - val_loss: 0.3845 - val_acc: 0.9503

Epoch 29/35
- 1s - loss: 4.2268e-05 - acc: 1.0000 - val_loss: 0.3859 - val_acc: 0.9495

Epoch 30/35
- 1s - loss: 3.8651e-05 - acc: 1.0000 - val_loss: 0.3778 - val_acc: 0.9524

Epoch 31/35
- 1s - loss: 3.5481e-05 - acc: 1.0000 - val_loss: 0.3811 - val_acc: 0.9510

Epoch 32/35
- 1s - loss: 3.2223e-05 - acc: 1.0000 - val_loss: 0.3767 - val_acc: 0.9531

Epoch 33/35
- 1s - loss: 3.0199e-05 - acc: 1.0000 - val_loss: 0.3736 - val_acc: 0.9546

Epoch 34/35
- 1s - loss: 2.7868e-05 - acc: 1.0000 - val_loss: 0.3832 - val_acc: 0.9517

Epoch 35/35
- 1s - loss: 2.5313e-05 - acc: 1.0000 - val_loss: 0.3823 - val_acc: 0.9517

Test accuracy:
0.9516943042537851
100%|██████████| 40/40 [27:42<00:00, 47.52s/it, best loss: -0.9783705839942322]
```

In [52]:

```
X_train_dynamic, Y_train_dynamic , X_val_dynamic, Y_val_dynamic = data_dynamic()

score, acc =best_model_dynamic.evaluate(X_train_dynamic,Y_train_dynamic)
print("Train Accuracy:",acc)

score, acc =best_model_dynamic.evaluate(X_val_dynamic,Y_val_dynamic)
print("Test Accuracy:",acc)

3285/3285 [=====] - 0s 128us/step
Train Accuracy: 1.0
1387/1387 [=====] - 0s 131us/step
Test Accuracy: 0.9783705839942322
```

In [0]:

```
##saving static model
best_model_dynamic.save(data_path + 'dynamic_classification.h5')
```

## Compiling the classification models

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 = []

    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"
    ]
```

```

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

# 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))

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 = data_path + f'y_{subset}.txt'
    y = _read_csv(filename)[0]

    return pd.get_dummies(y).values

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

    return X_train, X_test, y_train, y_test

```

In [0]:

```
X_train, X_test, Y_train, Y_test = load_data()
```

In [0]:

```
from keras.models import load_model
import pickle
```

In [0]:

```

def predict_activity(X):
    ##predicting dynamic or static
    st_dy_class_classifier = load_model(data_path+'2class_classification.h5')
    st_dy_prediction = st_dy_class_classifier.predict(X)
    st_dy_Y_prediction = np.argmax(st_dy_prediction, axis=1)
    #separating static and dynamic data
    static_X_data = X[st_dy_Y_prediction==1]
    dynamic_X_data = X[st_dy_Y_prediction==0]

    #static prediction
    static_classifier = load_model(data_path+'static_classification.h5')
    static_scale = pickle.load(open(data_path+'Scale_static.p','rb'))
    temp = static_X_data.reshape((static_X_data.shape[0] * static_X_data.shape[1], static_X_data.shape[2]))
    temp = static_scale.transform(temp)
    static_X_data = temp.reshape(static_X_data.shape)
    static_Y_prediction = static_classifier.predict(static_X_data)
    static_Y_prediction = np.argmax(static_Y_prediction, axis=1)
    #adding 4 because static activities were labelled from 4 in the dataset
    static_Y_prediction = static_Y_prediction + 4

    #dynamic prediction
    dynamic_classifier = load_model(data_path+'dynamic_classification.h5')
    dynamic_scale = pickle.load(open(data_path+'Scale_dynamic.p','rb'))
    temp = dynamic_X_data.reshape((dynamic_X_data.shape[0] * dynamic_X_data.shape[1], dynamic_X_data.shape[2]))
    temp = dynamic_scale.transform(temp)

```

```

dynamic_X_data = temp.reshape(dynamic_X_data.shape)
dynamic_Y_prediction = dynamic_classifier.predict(dynamic_X_data)
dynamic_Y_prediction = np.argmax(dynamic_Y_prediction, axis=1)
#adding 1 because dynamic activities were labelled from 1 in the dataset
dynamic_Y_prediction = dynamic_Y_prediction + 1

##getting final output in same order
static_counter, dynamic_counter = 0, 0
pred_output = []
for value in st_dy_Y_prediction:
    if value == 1:
        pred_output.append(static_Y_prediction[static_counter])
        static_counter += 1
    else:
        pred_output.append(dynamic_Y_prediction[dynamic_counter])
        dynamic_counter += 1
return pd.get_dummies(pred_output).values

```

In [75]:

```

pred_y_train = predict_activity(X_train)
pred_y_test = predict_activity(X_test)
from sklearn.metrics import accuracy_score
print('Train Accuracy: ',accuracy_score(Y_train,pred_y_train))
print('Test Accuracy: ',accuracy_score(Y_test,pred_y_test))

```

Train Accuracy: 0.9993199129488575  
Test Accuracy: 0.9647098744485918

## Conclusion

- Initially to understand the data better Exploratory data analysis was performed and the results were taken into account.
- After the initial step the domain specific data which has great features selected by domain experts of signal processing is then given to different models and the output is then considered which gave us great accuracy of 96.61(by Linear SVM)
- Then the time series features is directly taken as input without any feature engineering done on it.
- This input is then given to deep learning models with LSTM to generate accuracy of 89.07% on test data.
- Hyperparameter tuning is then performed on the model using 'Hyperas' (a hyperparameter tuning library for neural networks)and the accuracy captured after applying the hypertuned model to the data is 91.6%.
- To further improve the test accuracy I went with a divide and conquer approach :
  - In this I first classified the activity as static or a dynamic one.
  - Then I created 2 classifiers, one for static and the other for dynamic, to classify the sub activities in the data.
  - The final approach resulted in 96.47% of test accuracy and 99.93% of train accuracy.