Presentation:
Human Activity Recognition using Smartphone Sensor Data

Aniket Shirke	150100012
Purav Gandhi	150050007
Abhishek Kumar	150050020

Project Introduction

We intend to classify the physical activities performed by a user based on accelerometer and gyroscope sensor data collected by a smartphone in the user’s pocket.

To implement the above, we will be using a number of machine learning concepts and use a variety of classifying techniques to figure out which methods will best classify our data.

Motivation

Phone applications nowadays can show you how many steps you have walked, ran, flights of stairs you have climbed, calories burnt, etc. On a similar line, we also intend to build such a classifier from scratch which based on data from sensors already present in smartphones is able to identify the user activity. By doing this project we intend to gain practical knowledge of building classifiers and learn & implement machine learning concepts.

Currently the accuracy of these human activity classifiers are about 85% and improving them has many hurdles. Some of these being:

- High sampling rate of data is required, so more data needs to be processed every second.
- Also since the physical attributes of users, the sensors used in the smartphones vary greatly, a bias creeps in which decreases accuracy

If we can discover a novel method to handle this data or to decrease user bias or something else which can potentially increase accuracy, it would be a great step forward :D

Review of Literature

1] **Smart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity** -Research paper (http://userpages.umbc.edu/~nroy/courses/spring2016/cmisr/papers/Smart_Devices_Different_SenSys15.pdf)

This research was conducted on the same dataset we are using. They have implemented different types of classifiers and cross validation techniques and also tried to minimize the device and user bias.

2] **Using Machine Learning on Sensor Data** -Research Paper (Journal of Computing and Information Technology - CIT 18, 2010, 4, 341–347 doi:10.2498/cit.1001913)

This research is unrelated to smartphone activity classification. But it gave us an idea as to how we can use sensor data and train our neural network with it.

Our DataSet

We have used the "Heterogeneity Human Activity Recognition Dataset" which can be obtained from:

https://archive.ics.uci.edu/ml/datasets/Heterogeneity+Activity+Recognition

Data Set Information:

- **The Heterogeneity Dataset for Human Activity Recognition from Smartphone and Smartwatch sensors consists of two datasets (one containing accelerometer readings, the other containing gyroscope readings)**
- **Number of Attributes : 16 Number of Instances: 26,000,000+**
- **Users executed activities scripted in no specific order while carrying smartphones.**
- **Activities: 'Biking', 'Sitting', 'Standing', 'Walking', 'Stairs Up' and 'Stairs down'.**
- **Sensors: Sensors: Two embedded sensors, i.e., Accelerometer and Gyroscope, sampled at the highest frequency the respective device allows (100 Hz)**
- **Recordings: 9 users and 8 smartphones (2 Samsung Galaxy S3 mini, 2 Samsung Galaxy S3, 2 LG Nexus 4, 2 Samsung Galaxy S+)**

Project Timeline : Pre Midsem

28.02.17

First meeting with project mentors. Realized a lot more research was required

7.3.17

Got a better picture of how to approach the project. Found suitable text editor able to show the enormous dataset.

14.3.17

Parsed the data into workable files. Implemented basic neural network for the first time by training it with every tenth instance

22.3.17

Thought of peak detection over a sliding window to extract more features. Also tried implementation of random forest but results were unsatisfactory

30.03.17

Varied the number of nodes and layers in NN and collected and summarized results

Project Timeline : Post Midsem

10.4.17

Processed the data better. Managed to combine the accelerometer and gyroscope datasets to

(15-25).4.17

End-sem break. In the meantime studied RNN, LSTM and decided to implement it later

27.4.17

Implemented basic version of LSTM using Keras with Tensor-flow backend

28.04.17

Implemented GRU.

Normalized the data, varied various parameters of LSTM, and increased accuracy by 5%

What All We Did In this Project

Since the number of instances was extremely large (> 26 million), we faced a lot of issues in running code on such a large dataset. In order to overcome this we took every tenth instance reducing the sampling rate to one tenth of original sampling rate.

Initially instead of feeding a window of data points to a neural network, we fed individual data points to the network.

However, since this was conceptually incorrect it led to very low training and test accuracy and led to some activities getting 90%+ false negatives.

What All We Did In this Project

After having realized that a number of things could be improved upon, we started by doing the most important thing first: Found a way to correctly merge the accelerometer and gyroscope files.

We now wanted to implement a neural network which is affected by previous inputs along with the current inputs.

So, we switched from scikit_learn library to keras and used keras Sequential model to implement LSTM.

What All We Did In this Project

We got a good accuracy of 78% using LSTM in the first attempt itself. To further improve the accuracy, we normalized the accelerometer and gyroscope readings and applied one-hot encoding for different devices.
In order to make our model better, we added some LSTM layers to our Sequential model which increased the number of parameters and led to higher scores.

We later tried out a number of models by varying the batch size, loss function, number of neurons, number of epochs, optimizer, activation function and came up with a model achieving 85.3% accuracy.

How We Tested Data

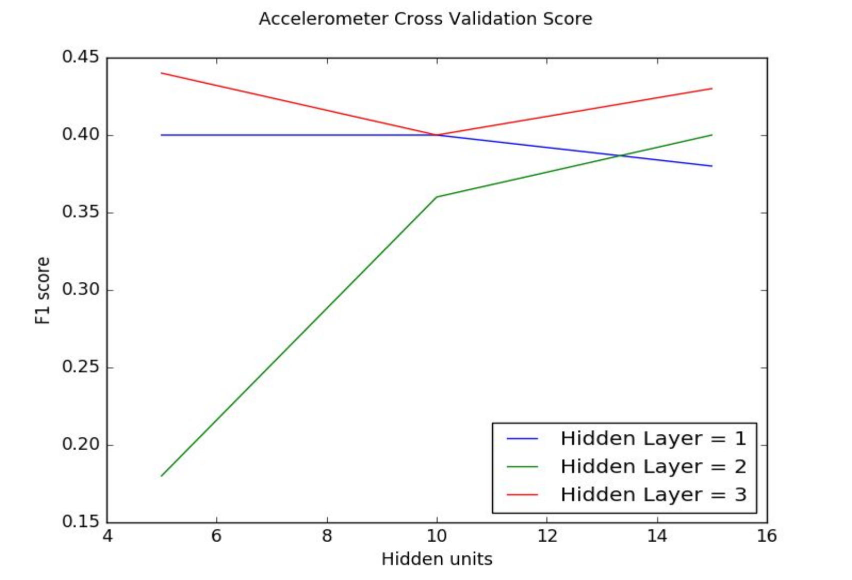
We split the data as 3/4th for training and 1/4th for test.

In the dataset, the data of user1 was followed by that of user2, which was followed by that of user3 and so on. So it is very likely that the test set corresponds to only one or 2 users. Most probably, the network had not even yet trained on these users. Knowing this, we did not send ‘user’ as an input feature.

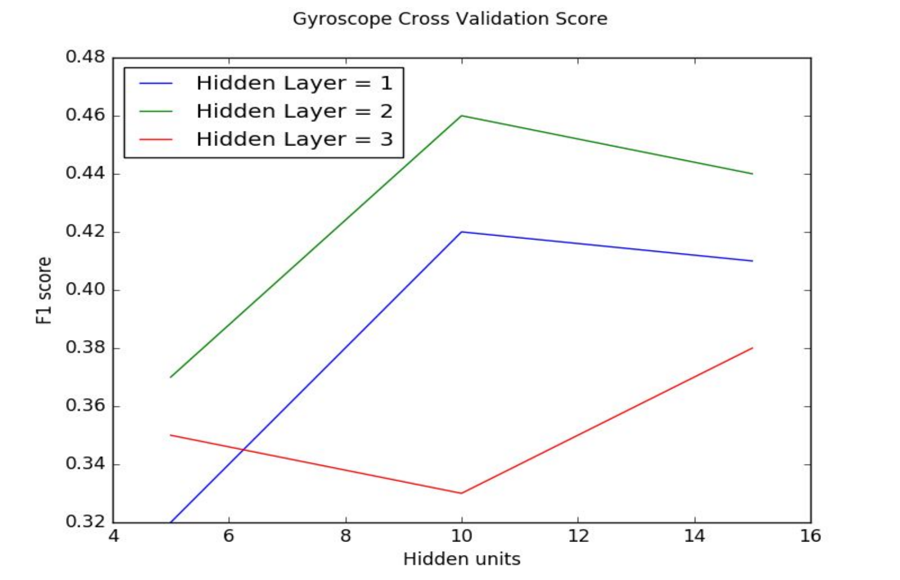
Effectively, one can see it as training on most of the users and testing on the last user

Neural Network Results (pre midsem)

Accelerometer



Gyroscope



P.S. : The accuracy of testing did not even cross 50%

Our Sequential Model

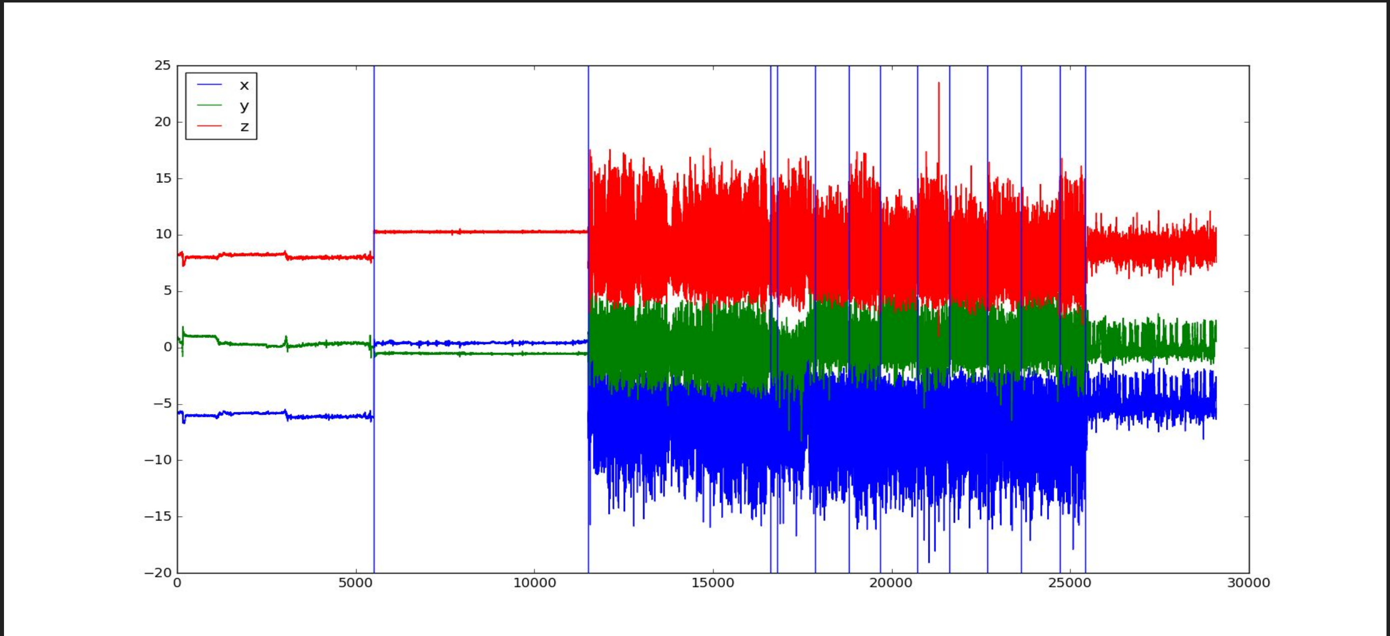
We are implementing a stateless LSTM. We added a LSTM layer with 24 neurons, which is followed by another LSTM layer with 12 neurons. This now goes to the final classifying layer, a DENSE layer with 6 neurons whose activation function is sigmoid.

We have kept Adam optimization as it converges very quickly and it is better than rmsprop.

The most surprising result was that keeping the loss function as “binary-crossentropy” gives 20% better accuracy than other loss functions.

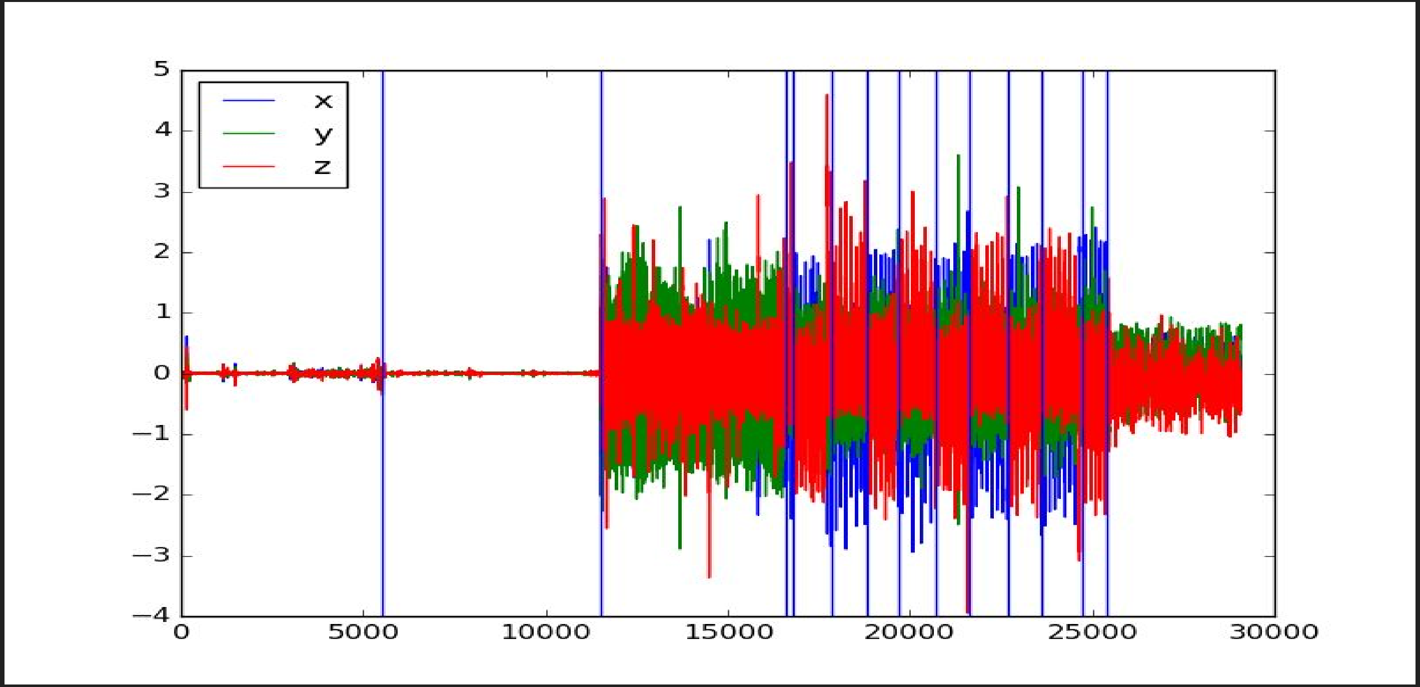
Our GRU implementation produced results extremely similar to LSTM.

Plots:



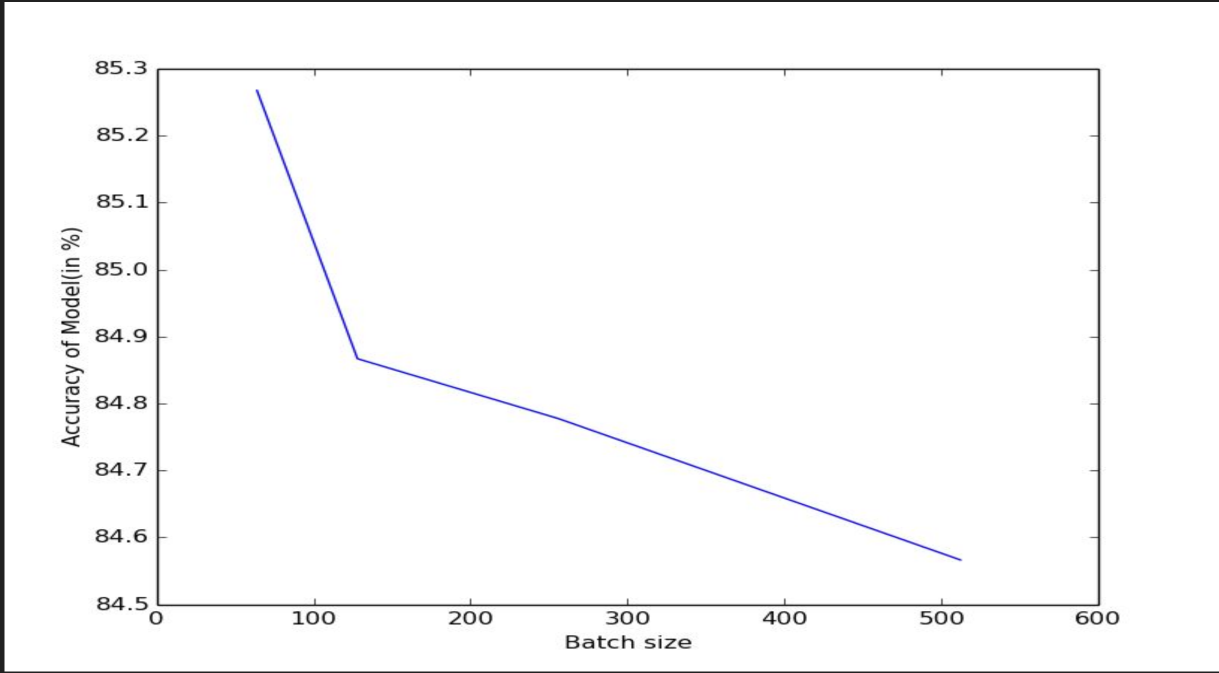
Accelerometer readings for different activities

Plots:



Gyroscope readings for different activities

Plots:



Accuracy vs Batch Size

Scope for Further Improvement on Project

We can further improve accuracy by applying a noise filter to remove outliers and smooth values.

Better and more concentrated feature extraction in time and frequency domain from a given window would lead to better results.

Can develop an application which can classify activities and give the user the output in real time.

By removing gravity components from captured data so that only user body movement data is captured by sensor.

THANK YOU..!!!!!!