Contents

List of Figures

Accelerometer data classification using Neural Networks

Sameer, Shantanu, Rekah, Eera

May 1, 2014

Abstract

Human activity is a great source for building new kind of services that take the current status of the user into consideration. Equipping a service provided by a phone application with the context of the user results in richer and more engaging services. In this work we present a neural based classifier that is able to detect the status of the human body from readings of accelerometers placed on different parts of the body. We show how different configuration of the neural network and data preprocessing can change the accuracy of our prediction, we also show how we can predict the movement using one accelerometer reading to simulate the effect of having a mobile device in users pocket or a smart watch attached to the user's wrist. To the best of our knowledge this is the first study that considers a single source of data and achieves (96%) accuracy.

1 Introduction

With the increasing use of smart mobile devices new opportunities are present for richer and more engaging services that cater to every need of its users. Most of these smart devices are equipped with accelerometers that give an indication of of the relative position of the device and its user at any point in time. Using this data to identify the status of body movement provides an accurate record of the physical activity of the user which has applications related to the health care and interactive design for mobile devices' applications.

In this work we seek to build a neural network based classifier that is able to correctly identify a pattern of human body movement from a series of readings, we discuss different design choices and report their impact on prediction accuracy. The rest of this report goes as follows; section 3 describes the dataset we are using, section 2 provides a simple description of the inner workings of accelerometers for the inexperienced reader, section 5 describes the methodology we used, section 6 discusses a summary of the results we are getting and includes ideas for future work.

2 Accelerometer

MAGIC MAGIC MAGIC OH MY GOD !!

3 Data Set

Raw data we are using are based on the work by [7]. The data set report 8 hours of activity of 4 subject tests of varying age , weight, height and gender. Each subject provided two hours of

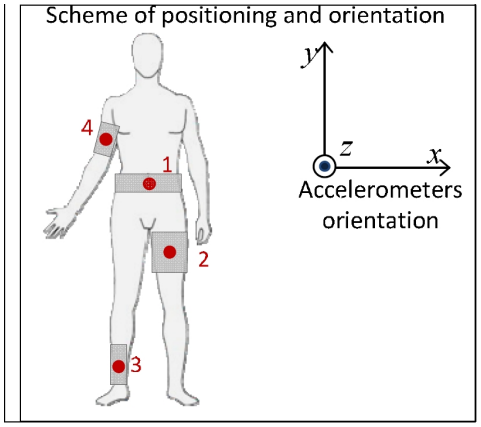

Figure 1: Accelerometer Placement

activity of postures varying between (Standing, sitting down, sitting , standing up and walking). The data provided is a continuous stream of reading from sensors placed of four parts of the body, waist, arm, leg and foot. Figure 1 shows the places of these accelerators [7]. Figure 2 shows the distribution of the postures among the samples.

Along with the three dimensional input from each accelerometer,which is adds up to twelve readings for each time step, the data set also provides some biometrics that are subject specific. This could help improving the prediction accuracy given that different users might move differently and having these metrics helps the neural network account for that. These biometrics include the following:

1. Age: Humans movement pattern tend to change
   evolve in different ages.
2. Weight: Weight affects the way humans move which also affects the accelerometer data.
3. Height: Reading along the Y-direction (height direction) are affected by subject's height.
4. Gender: Males and females have different patterns for movement, especially for walking.

4 Frameworks

During the course of our experiments we tried different tools and libraries before settling on doing all of out experiments using Matlab [3]. We used used the following tools:

4.1 PyBrain

PyBrain [6] is a strong and modular Python library for building machine learning applications. It support for building reinforcement learning and Neural network based learning. It allows for building flexible architecture for different kind of networks (Feed Forward, Recurrent) it also has predefined layers for different layer for network architectures like LSTM [2] and Linear layers. It also allows for different kinds of activation functions. We used PyBrain to train FFN on raw entries of the data with different number of layers and number of neurons in hidden layers, the maximum classification accuracy we got was 73% after around 200 epoch steps,

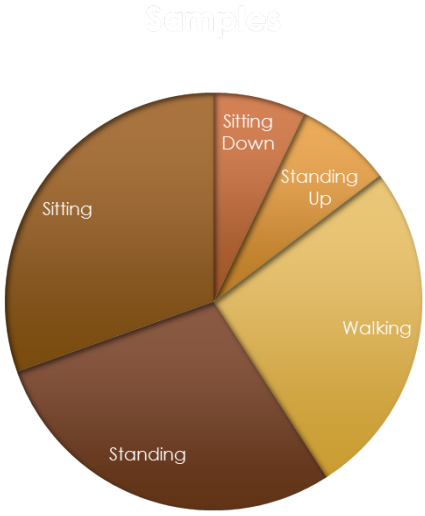

Figure 2: Sample data distribution

feeding the network seven consecutive rows at a time raised the accuracy to 80%. We stopped experimenting with PyBrain because of the high execution time we were facing. Training a thousand steps for the FFN with seven row input took about 5 hours, we had to look for some other platform.

4.2 NeuroLab

NeuroLab [5] is a simple and powerful SciPy Python Python based library for building Neural Network. It is not as configurable or developed as PyBrain. Unfortunately, given the large data set we had and due the fact that this is a Pure Python based library we faced the same problems of high execution time.

4.3 Other libraries

These include Bain.js [1] and Cuda-Convert [4]. Initial use of these two libraries didn't provide any promising results they showed to be not up to the speed we needed as well.

5 Methodology

Previous work in detecting the body posture [7] used decision tree algorithms, namely C4.5, with intensive data preprocessing and feature extraction before actually using the data for classification; work include finding the roll and pitch angles for the body and also the variance in roll and pitch. In this work we opted for doing as possible of data preprocessing and to let the neural network the feature extraction. We try to train our network with the following variants:

1. Using unprocessed rows and feeding them to the network, section 5.1 discusses that.

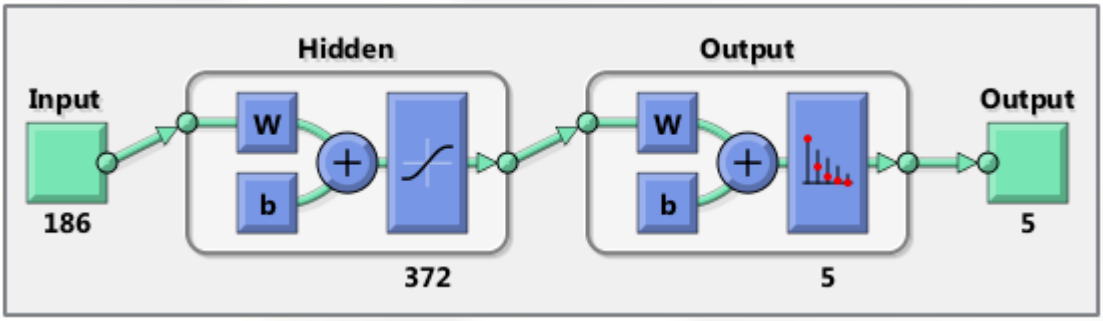

Figure 3: Neural Network for moving window

2. Using deltas of two consecutive rows or more than two rows, section 5.2 discusses that.
3. A person specific learner where we only do the training for one person for increased accuracy.
4. Use only one accelerometer to detect body posture. This gives a more realistic use case where the user only has one source of data; a smart phone. Section 5.3 discusses that.

Doing these operations and trying different versions we managed to achieve up to 99.85% classification accuracy which is higher than the one achieved by [7] which accounted to 99.4%.

5.1 Classification using raw rows

For an initial start we built Neural networks that take all the readings in separate rows. We relied on the idea that even though the human goes through the same stages between standing up and sitting down the subject might actually have a subtle difference in the exact points that the accelerometer reads the data at TODO SHANTANU I NEED YOUR RUN FOR THIS !!!

5.2 A Moving window

As described in section 3 the rows of data we have are actually consecutive in time, mashing more than one row into single input to the Neural Network gives a sense of of this movement for the network to learn; for this we came up with the idea of a moving window. To build this moving window we take 8 consecutive row, accounting for 1 second of time, and we find the difference in readings between the beginning and the end of the window. We tried different network configuration and architectures, increasing the number of Neurons gave some good results with best prediction accuracy for validation data of 99.7%. Figure 4 shows the confusion matrix and accuracy for a network of 372 neurons in the hidden layer shown in in figure 3.

5.3 Single source of data

In this section we try to use recurrent Neural networks instead of feed forward networks to build a learner for the continuous data. Wu use Continuous time recurrent Neural networks.

Figure 4: Confusion matrix results for moving window

6 Conclusion

7 Future Step for this work

MAGIC MAGIC

References

[1] BrainJS. Brain: Neural networks in javascript, May 2014.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[3] MATLAB. *version 8.3.0 (R2014a)*. The MathWorks Inc., Natick, Massachusetts, 2014.

[4] NeuroLab. High-performance c++/cuda implementation of convolutional neural networks, May 2014.

[5] NeuroLab. Neurolab: Simple and powerful neural network library for python, May 2014.

[6] Tom Schaul, Justin Bayer, Daan Wierstra, Yi Sun, Martin Felder, Frank Sehnke, Thomas Rückstieß, and Jürgen Schmidhuber. PyBrain. *Journal of Machine Learning Research*, 11:743–746, 2010.

[7] Wallace Ugulino, Débora Cardador, Katia Vega, Eduardo Velloso, Ruy Milidiú, and Hugo Fuks. Wearable computing: accelerometers data classification of body postures and movements. In *Advances in Artificial Intelligence-SBIA 2012*, pages 52–61. Springer, 2012.