Human Activity Recognition using Accelerometers and Gyroscopic sensors

Nihir Chadderwala
Indiana Bloomington University
Indiana, USA
nchadder@iu.edu

Lenin Arivukadal
Indiana Bloomington University
Indiana, USA
larivuka@iu.edu

Abstract: Smartphones are quickly becoming an important part of a person's day to day life. Smartphones are also becoming advanced day by day with the help of advanced features and sensors that are available in smartphones. Smartphones now contains sensors that are capable of measuring heart rate, blood pressure, physical activities, etc. Smartphones have inbuild sensors such as accelerometer and gyroscope that continuously record data related to person's physical activities. This has appealing use in healthcare applications such as monitoring physical activities for elderly people to ensure that they stay healthy. In this paper, we will be making use of supervised deep learning techniques on dataset that contains recordings of accelerometer data obtained from 30 different individuals using Samsung SII phones tied to their waist. Results are obtained using deep learning neural networks, convolutional neural networks and long-short term memory recurrent neural networks. The research question that is discussed in this paper deals with which deep learning technique is best suited for recognition of human activity.

1. Introduction

There is a huge demand and market for wearable devices such as smartwatch, Fitbit, etc. People who are motivated to live a healthy lifestyle uses such devices that can track their physical activities and health information. Having the same information in smartphones helps everyone to keep track of healthy lifestyle without having to buy additional devices. Human activity recognition is changing the daily routine of people by health and fitness tracking. However, human activity recognition is a highly challenging subject. Recognition can be accomplished by exploiting the information retrieved from inertial sensors such as accelerometers embedded into the smartphones.

2. Related Work

Currently, there is lot of research being performed in human activity recognition by "Wireless Sensor Data Mining Lab" at Fordham University which is funded by National Science Foundation and Google Inc. Their research focuses on resolving many challenges that are faced when dealing with human activity recognition. Starting from the gathering data to using techniques for manipulating data. Picking the right algorithm for human activity recognition is utmost important. ML methods have been previously employed for recognition include Naïve Bayes, SVMs, Threshold-based and Markov chains [1]. In particular SVMs are used for classification for recognition [2]. In addition to picking the right algorithm, it is also important to perform dimensionality reduction. Performing feature reduction is important as it allows more amount of data to be processed faster.

3. Methodology

The primary objective is to determine the deep learning technique that can perform human activity recognition with the highest accuracy. In order to achieve this, we will be performing below list of tasks –

i. Study the dataset and understand the features that represent the data. This will help us understand how the data is correlated.
ii. Understand how the data is distributed using data visualization techniques.
iii. Implement various deep learning techniques and use loss function and accuracy to determine the best model
iv. Perform dimension reduction and run the model on reduced feature data.
v. Evaluate models and provide analysis

3.1 Experimental Setup

The experiment has been carried out with a group of 30 volunteers within an age bracket of 19 to 48 years. Each person performed six activities – walking, walking upstairs, walking downstairs, laying, sitting and standing. The experiment was video recorded to facilitate the data labeling. The dataset was then randomly partitioned into two sets where 70% of the dataset is used for training and rest for testing the model. A Samsung SII smartphone with accelerometer and gyroscope sensors were used to measure 3-axial linear acceleration and angular velocity respectively at a constant rate of 50Hz. The sensor signals were pre-processed by applying noise filters and then sampled in a fixed width sliding windows of 2.56 seconds.

The dataset contains 10299 observations and 563 features. The signals are prepended with t and f to denote that they are time and frequency. The features related to body acceleration can be classified in time and frequency variables. The Fourier transformation is applied on the fixed sample signals to generate frequency variables. In addition to Fourier transformation various statistical functions were applied such as mean, standard deviation, mean absolution difference, etc.

The signals related to gravitational acceleration were gathered by applying Butterworth low-pass filter. This helps to remove the noise from the sample data generated by sensors. In addition to body acceleration and gravitational acceleration, gyroscopic signals were also gathered. Fourier transformation was applied on gyroscope signals as well to generate frequency variables. Similar to body acceleration and gravitation acceleration, various statistical functions were applied over the data.

3.2 Data Exploration – Lenin's Contribution

Data exploration and data pre-processing is an important stage to prep the data for machine learning algorithm. In addition, the data exploration provides insight into the data that can be useful to formulate hypothesis, transform data, remove outliers, etc.

The first task in preparing the dataset is to get the dimensions of the data and ensure that there are no missing values. The dataset contains 561 features and we started out by exploring how these are related to each other and whether there are some which can be safely ignored for our problem. In addition, for checking for correlation, we also checked our variable for zero or low variance so that they can be removed before running any analysis. We checked for any missing values in our columns, which might lead to errors in any future analysis but didn't find any variables with missing values.

After ensuring that there were no missing data, we determined the target feature. The target feature 'Activity' is a categorical feature that contains multiple classes or labels such as – Walking, Walking Upstairs, Walking Downstairs, Laying, Sitting, and Standing. It was important to determine that we have balanced number of classes in the dataset. We started with basic visual exploration of the dataset by plotting distributions for the variables for each activity.

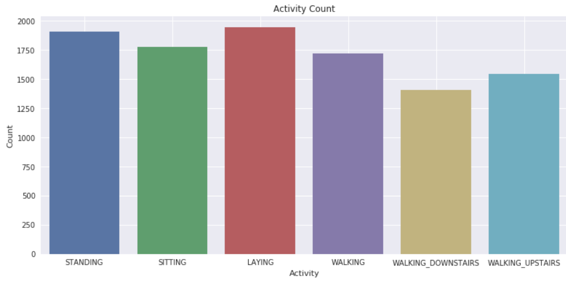


Figure 1: Bar plot presenting number of data points for each activity

The bar plot presented in figure 1, helps us understand that we have enough dataset for each class. Bar plot was very useful to help us understand and ensure that we do not have imbalanced target classes. Once the target feature was identified as appropriate for our algorithm, it was important that we perform similar checks for input variables as well. It is important that input variables are normally distributed and they are not skewed.

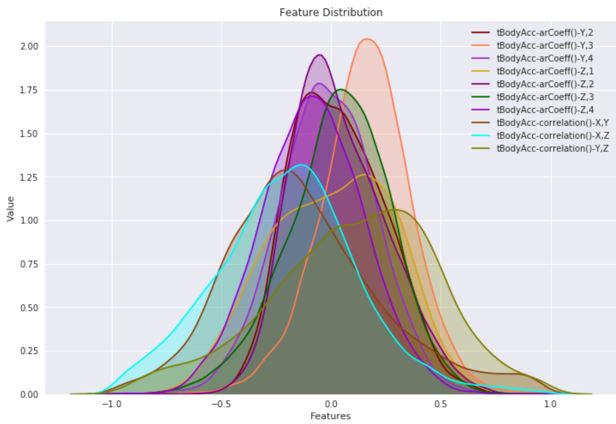


Figure 2: Feature distribution plot

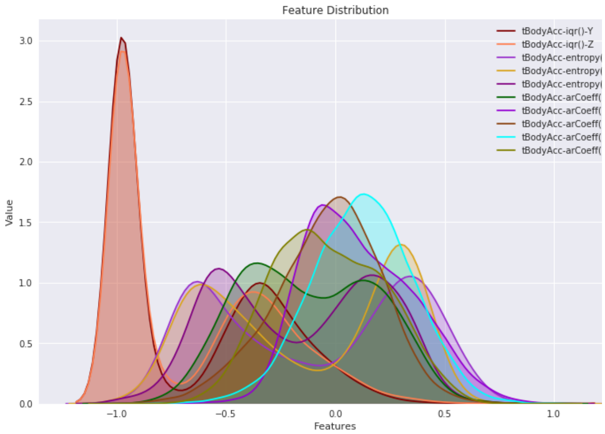


Figure 3: Feature distribution plot

The histogram plots presented in figure 2 and 3 clearly shows that the input variables are normally distributed. Therefore, the input variables do not require any transformation. Here histograms plots were very helpful as they gave us understanding of the input variables and how they are distributed. In addition to understanding how the data is distributed, we also need to understand the correlation of data with the labels.

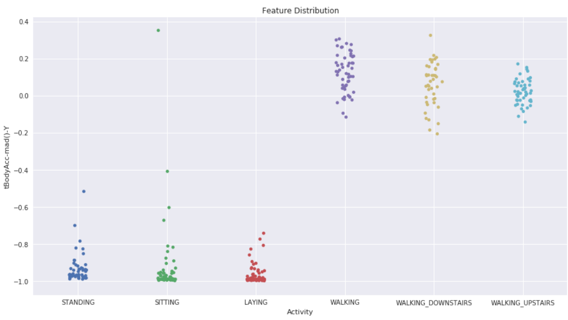


Figure 4: Strip plot showing body acceleration for the activities

The strip-plot represented in figure 4 helps us understand how body acceleration is related to the activities. We can see that the body acceleration is in negative for activities such as standing, sitting and laying. On the other hand, the body acceleration is positive for activities such as walking, walking downstairs and walking upstairs.

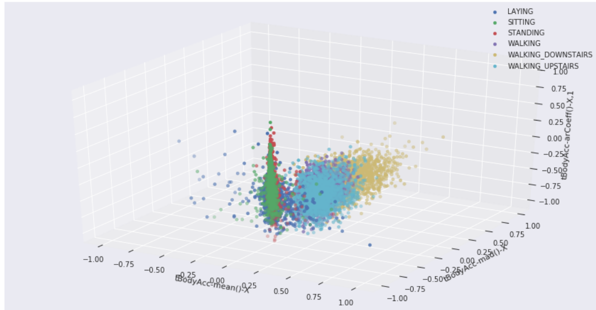


Figure 5: 3-D plot to identify distribution of activities in correlation with body acceleration

In figure 5, we have 3D plot that provides information on distribution of activities in correlation with body acceleration. 3D plot is very helpful to establish the correlation between body acceleration and human activity. We can easily separate static activities such as sitting, standing and laying from the dynamic activities such as walking, walking upstairs and walking downstairs. The information will be helpful to our models in differentiating the activities.

3.3 Neural network model - Nihir

For our base neural network model, we used multilayer perceptron model. Our input layer accepted had about twice the number of neurons to the features. Our input layer had 1024 neurons with 561 features for each observation. Weights and bias for all of our layers were selected randomly from a normal distribution. We selected tf.truncated_normal to select random numbers from a normal distribution where standard deviation is 0.1. We used truncated normal distribution to prevent generating dead neurons with our activation function. For our layers, we used ReLU as the activation function. Mathematically ReLU is defined as:

$$y = \max(0, x)$$

Since ReLU function converges faster than other activation function, we decided to use ReLU. The output layer of the network contains 6 neurons. Suppose if the first neuron is fired i.e. output probability is approximately around 1, then it indicates that the observation belongs to the first activity. We have labelled dataset that contains multiple classes. So, for our output layer, we decided to use softmax as an activation function. Softmax assigns decimal probabilities to each class where the decimal probabilities add up to 1.0 [3]. The training over the full dataset was fairly quick.

3.4 Dimension reduction with PCA - Lenin

Dimensionality reduction or reducing features is useful as it helps eliminate features that has very less or no impact on the model and also helps to improve model performance. For feature reduction, we will be making use of Principal Component Analysis (PCA) algorithm.

PCA is used to filter or remove the features which have low or zero variance. It uses orthogonal transformation to convert a set of observations of correlated variables into set of values of linearly uncorrelated variables which is called principal component ("Principal component analysis", n.d.).

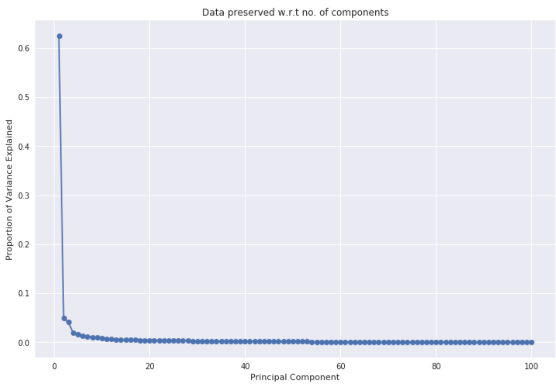


Figure 6: Proportion of variance explained

The principal component analysis graph presented in figure 6 shows the amount of variance saturates after around 20 components. Therefore, it is more likely that components above 20 will not have much impact on the classification model. Hence, we will be making use of 20 components and transform dataset PCA algorithm with 20 components.

3.5 t-SNE dimensionality reduction - Nihir

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear technique for dimensionality reduction. The reduction is very useful for data such as image processing, NLP, genomic data and speech processing.

The algorithm starts by calculating the probability of points in high dimensional space and calculating the probability of similarity of points in the corresponding low-dimensional space. The similarity of points is calculated as the conditional probability that a point A would choose point B as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian (normal distribution) centered at A.

It then tries to minimize the difference between these conditional probabilities (or similarities) in higher-dimensional and lower-dimensional space for a perfect representation of data points in lower-dimensional space. To measure the minimization of the sum of difference of conditional probability t-SNE minimizes the sum of Kullback-Leibler divergence of overall data points using a gradient descent method.

3.6 Convolutional Neural Network - Nihir

We constructed 2D convolutional neural network for the dataset. We had multiple convolutional layers with ReLU as an activation function. We had set filter (feature detector) at 64. We started with less number of filters based on PCA components, however, we found that 64 appears to works best. A pooling layer is used to reduce the complexity of the output and prevent overfitting of the data. The output is then fed into second CNN layer with the same number of filters. One more pooling layer is added to reduce any additional complexity and overfitting of the data. The data is then reshaped to 32 * 32 * 2, 1000 before feeding to the fully connected layer with ReLU activation. Dropout layer is used within fully connected layer to randomly assign weights to the neurons. With dropout layer, the network becomes less sensitive to react to small variations in the data. Thus, increasing the accuracy of unseen data.

3.7 Long-short term memory – Recurrent Neural Network - Nihir

We did not get good accuracy with CNN, so we also decided to implement LSTM RNN model. LSTMs are explicitly designed to avoid long-term dependency problem. Remembering information for long periods of time is practically their default behavior. Recurrent Neural networks have the form of a chain of repeating modules of neural network. RNN allows the data to flow in both directions within the network units and forms a directed cycle. Additionally, RNN can use their internal memory for processing arbitrary input sequences and all the information usually circulates within the hidden states. LSTM is well suited for various classification problems.

With 32 hidden neurons, 2 LSTM cells and 6 output classes, the model was trained using LSTM for random normal network initialization. For calculating loss, we used L2-norm loss function.

3.8 Optimizing Models - Nihir

In order to optimize the models, we tested our models with various hyperparameters. We provided various number of hidden neurons, various epoch values, various stripes and kernel size for CNN, and various optimization algorithms such as gradient descent, Adam and Momentum optimizer. We found that Adam optimizer worked better than other optimizer algorithms.

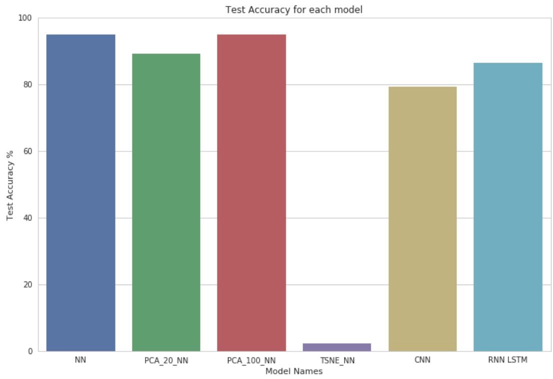
4. Analysis - Lenin

For evaluating the performance of the model, we calculated accuracy using mathematical formula –

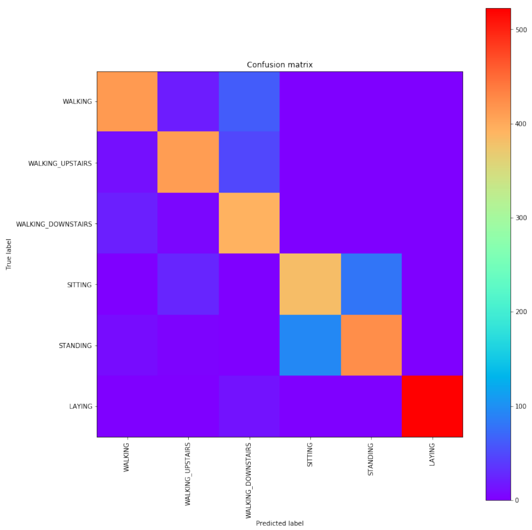
$$Accuracy = \frac{number\ of\ samples\ correctly\ predicted}{total\ number\ of\ samples}$$

and we also looked at the confusion matrix to understand true positives, true negative, false positive and false negatives.

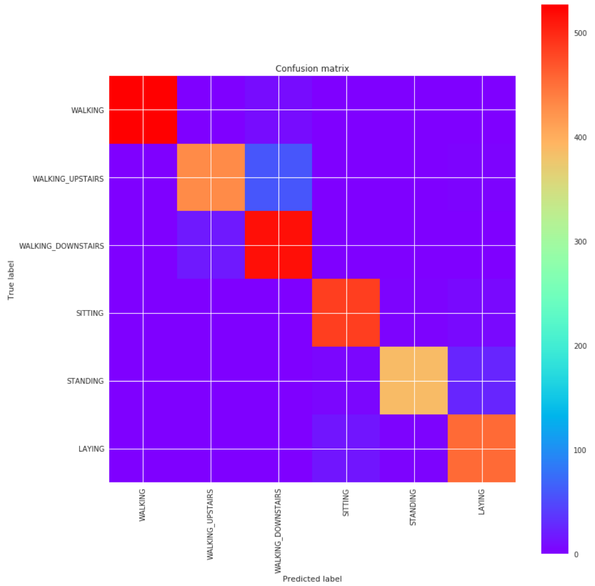
Model Accuracy results



RNN Confusion Matrix



Multi-Layer Perceptron Confusion Matrix



As evident from the confusion matrix plot, RNN model had lot of confusion over multiple activities on the other hand, multi-layer perceptron model had less confusion between the activities.

5. Conclusion

We tested different deep learning techniques to classify human activities. We used multi-layer perceptron neural network, reduced features using PCA and T-SNE, convolutional neural network and long-short term memory recurrent neural network. We identified that multi-layer perceptron with principal components of 100 performed best. It gave best accuracy with less amount of time. Future work could include using video dataset in order to classify human activities.

6. References

1. Mannini, A., Sabatini, A.M.: Machine learning methods for classifying human physical activity from on-body accelerometers. Sensors 10(2) (2010) 1154-1175
2. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. SIGKDD Explor. 12(2) (March 2011) 74-82
3. “Multi-Class Neural Networks: Softmax | Machine Learning Crash Course | Google Developers.” Google,

Google, developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax.
4. https://github.com/aymericdamien/TensorFlow-Examples