INTERIM REPORT

Human Activity Recognition

AREA

Deep Learning

Submitted to

Prof. Sonia Abraham

Submitted by

Anlin Albert

1. Abstract

Human Activity Recognition (HAR) is classifying the activity of a person using responsive sensors that are affected by human movement. Both users and capabilities(sensors) of smartphones increase and users usually carry their smartphones with them. These facts make HAR more important and popular. This work focuses on the recognition of human activity using smartphone sensors using different machine learning classification approaches. Data retrieved from smartphones’ accelerometer sensors are classified to recognize the human activity. Results of the approaches used are compared in terms of efficiency and precision.

1. Introduction

Human activity recognition (HAR) is an ability to interpret human body gestures or motion via sensors and determine human activity or action. It is based on an inertial measurement unit (IMU) that has become the de facto method for continuously monitoring not only what human beings are up to but also in monitoring the activities of devices, machine parts, pets, and others. This has made HAR based on IMU sensors a hot area for research. Not to mention that these maintain high levels of privacy and comfort for the user. To understand human behavior and intrinsically anticipate human intentions, research into human activity recognition HAR) using sensors in wearable and handheld devices has intensified. The ability of a system to use as few resources as possible to recognize a user’s activity from raw data is what many researchers are striving for attention.

Human activity analysis is one of the most important problems that has received considerable attention from the computer vision community in recent years. It has various applications, spanning from activity understanding for intelligent surveillance systems to improving human-computer interactions. Recent approaches have demonstrated great performance in recognizing individual actions. However, in reality, human activity can involve multiple people, and to recognize such group activities and their interactions would require information more than the motion of individuals.

Most human daily tasks can be simplified or automated if they can be recognized via the HAR system. Typically, the HAR system can be either supervised or unsupervised. A supervised HAR system required some prior training with dedicated datasets while an unsupervised HAR system is configured with a set of rules during development. HAR is considered an important component in various scientific research contexts i.e. surveillance, healthcare, and human-computer interaction (HCI).

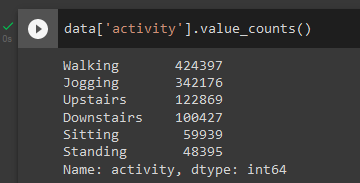
1. Data exploration

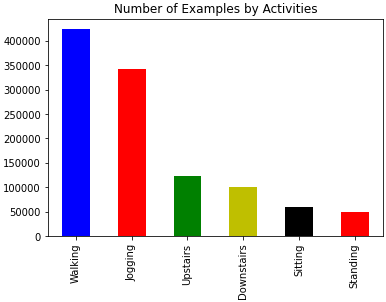
The WISDM dataset has a total of 1098209 samples, and the percentage of the total samples associated with each activity. It can be seen that WISDM is an unbalanced dataset. Activity walking takes up the most, reaching 38.6% while standing only accounts for 4.4%. Its experimental object consists of 36 subjects. These subjects performed certain daily activities with an Android phone in their front leg pockets. The sensor used is an accelerometer with a sampling frequency of 20 Hz. It is also a built-in motion sensor of the smartphone. Six activities were recorded: standing (Std), sitting (Sit), walking (Walk), upstairs (Up), downstairs (Down), and jogging (Jog). The data collection was supervised by a dedicated person to ensure the quality of data.

This is the summary of the dataset that I have used. This dataset contains 6 activities having nearly 11 lakhs records.



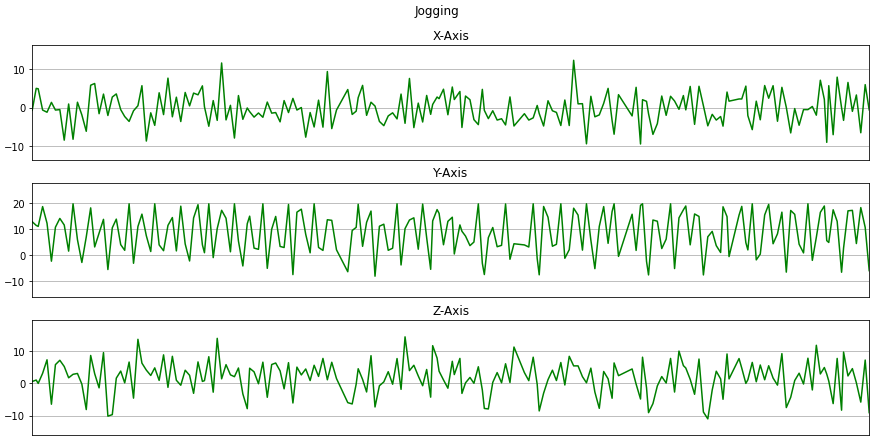
From the below two snapshots, we can see that this dataset contains highly unbalanced data. Means here walking and jogging have more no of records i.e. 424397 and 342176 records respectively while standing has 48395 records only. If we use this dataset then it is going to be highly overfitted and skewed towards walking and jogging.So we need to balance the dataset, for that what I did I took only 48395 records from each activity.

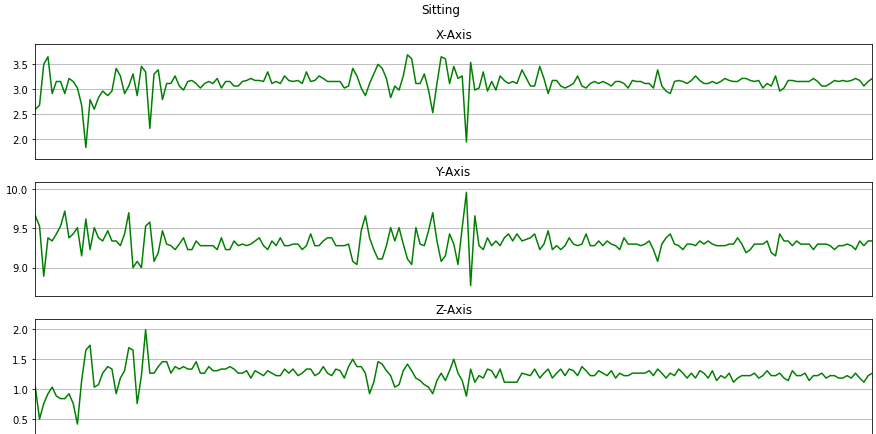




After exploring the dataset I tried to plot these accelerometer values for timestamp 10 sec so that we can see how the accelerometer data looks visually for each activity. Because each activity follows a specific pattern and by looking at these patterns we can classify which accelerometer values belong to which class.







1. Architecture



CNN follows a hierarchical model which works on building a network, like a funnel, and finally gives out a fully connected layer where all the networks are connected and output is processed.

Three types of layers make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function which are defined below.

[1] Convolutional Layer - This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image to the size of the filter (MxM). The output is termed the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

[2] Pooling Layer - In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce computational costs. This is performed by decreasing the connections between layers and independently operating on each feature map. Depending upon the method used, there are several types of Pooling operations. In Max Pooling, the largest element is taken from the feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer

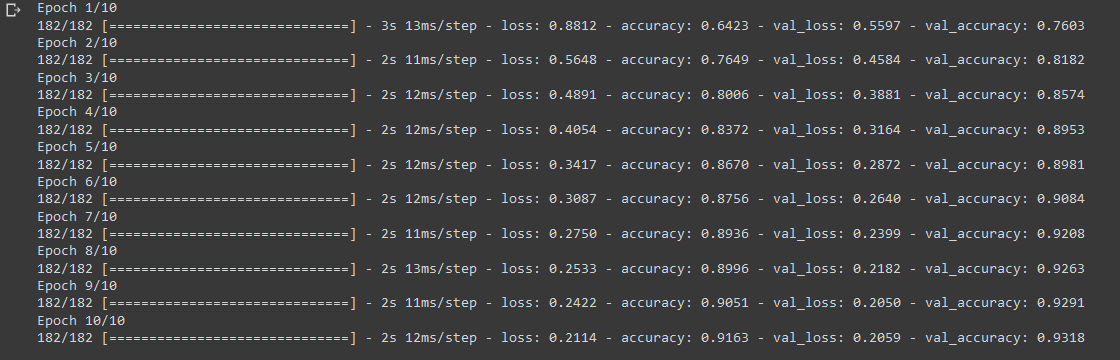
[3] Fully Connected Layer - The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture. In this, the input image from the previous layers is flattened and fed to the FC layer. The flattened vector then undergoes a few more FC layers where the mathematical function operations usually take place. In this stage, the classification process begins to take place.

The main advantage of CNN compared to other neural networks is that it automatically detects the important features without any human supervision. Little dependence on preprocessing and it is easy to understand and fast to implement. It has the highest accuracy among all algorithms that predict image

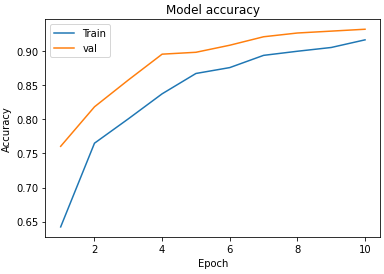
1. Implementation of the CNN model

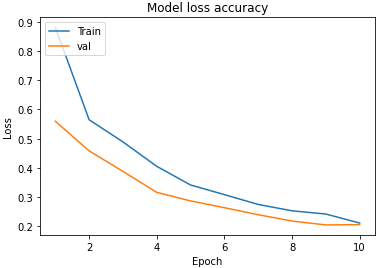
Here I have called sequential layer 1st. In 1st layer, I have added a 2-dimensional convolution layer that is Con2D. Then 16 filters are passed having [2,2] kernel size and activation function relu is used. For 1st layer input shape is x-train. Then I added the dropout layer means randomly 0.1 or 10% of neurons have been dropped. Another layer of CNN having 32 layers and a size of [2,2] with activation function relu is added.

In the hidden Convolution layer, we don't need to provide input shapes because it automatically matches the preceding layers. Then 20% dropout will be added. Then add flatten then dense layer having 64 and activation function relu and drop 50% neuron randomly. Then add the final layer. As we need 6 classes and it is a multi-class classification we are taking softmax as an activation function. For compilation Adam optimizer is used and the loss function is sparse categorical cross-entropy. Then the training process will start. I have used several epochs to reach the final accuracy of approx 94%.

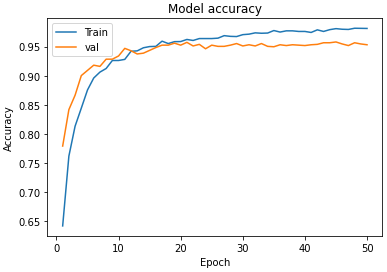


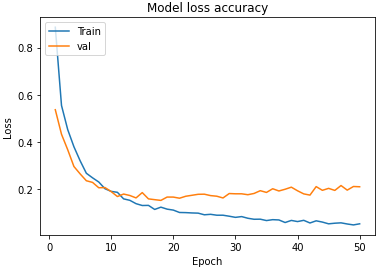
The snapshots below represent the learning curves for 10 epochs. 1st graph is plotted between no of epochs and the accuracy of the model. Then the 2nd graph is plotted between no of epochs and loss of a model. Here we got quite a good accuracy. As validation loss is less than training loss we can say that our model is neither overfitting nor underfitting.





The snapshots below represent the learning curves for 50 epochs.





1. UI Design

The application is developed using android.

