CLASSIFICATION OF HUMAN ACTIVITIES

Deep Neural Network

NOVEMBER 24, 2021

Table of Contents

Abstract

Human Activity Recognition is the problem of predicting what a person is doing based on a trace of their movement using sensors. It aims to understand human behavior and integrate users and their social context with computer systems.

In 2012, Davide Anguita, from the University of Genova, made available the dataset A Public Domain Dataset for Human Activity Recognition Using Smartphones.”. A rich dataset, containing human body motion and their activities, it was collected based on the smartphones to gather context information about people actions.

In this project, we will predict the human activity based on 4 information collected by the sensors of the smartphone: Acceleration, Orientation, Marginal and Velocity.

With the advancements in Deep Learning, Deep Neural Networks (DNN) are becoming very increasingly effective on this type of task. Therefore, this report aims to investigate the viability of Deep Neural Networks using human motions data to predict human activity.

We were very successful on the first experiment to achieve high performance, with accuracy 1 on the validation dataset, we will present in the next parts the dataset used in the experiment, the pre-processing, and the successful model.

Keywords: Human Activity Recognition, Deep Learning, Prediction of human activity, Sensors.

Introduction

Aim

This project aims to study and investigate the applicability of Deep Neural Networks to predict human activity based on human motions, more precisely the features that has been used are Acceleration, orientation, marginal and velocity.

Problem Statement

To solve the task of human motion recognition, there are a lot of methods that can be successful to do so: like video processing, and the previous machine learning models like decision tree.

The available dataset that will conduct the following study contains information about Acceleration, orientation, marginal, and velocity.

- **Which DNN is superior in the context of predicting human motions?**

Social and Ethical Aspects

The dataset is completely based on the perception of the individual (the subject of the study) and the environment. Therefore, it is very hard to have a general model that works for everyone. For example, the human motion of an injured person will be completely different than a normal person. Another example will be the difference based on gender; women have q center gravity completely different than men.

There is a lot of factors that play key role in this aspect like: age, gender, normal/injured….

This project can have more humanistic and positive applications, in our world, like for example the democratization of the analysis of physical efforts, so everyone can analyze his movements during a workout and his performance without spending too much money on medical tests.

Structure

This report is divided into 3 Chapters. Chapter 2: “Method” discusses the details regarding the dataset along with parameter selection for the model. Chapter 3 will present different results obtained during the experiment. Chapter 4 will analyze the previous results. And finally, the conclusion and the future work.

Method

Dataset

Data Exploration

The dataset contains 5 files:

- **Acc.csv**
- **Labels.csv**
- **Orin.csv**
- **Mag.csv**
- **Velo.csv**

For all the files, we transformed the timestamp to seconds, and we took the first t ime as our reference '2019-10-11 19:05:56.381000'.

Acc.csv

- Columns of the file

	Timestamp	Acc_X	Acc_Y	Acc_Z
0	10-11-2019 19:05:56.381	-0.131576	-0.086370	9.398655
1	10-11-2019 19:05:56.391	-0.122146	0.239352	9.314530
2	10-11-2019 19:05:56.400	0.095950	0.429307	9.773775
3	10-11-2019 19:05:56.410	-0.278271	0.395927	10.230476
4	10-11-2019 19:05:56.420	-0.530796	0.440384	9.580527

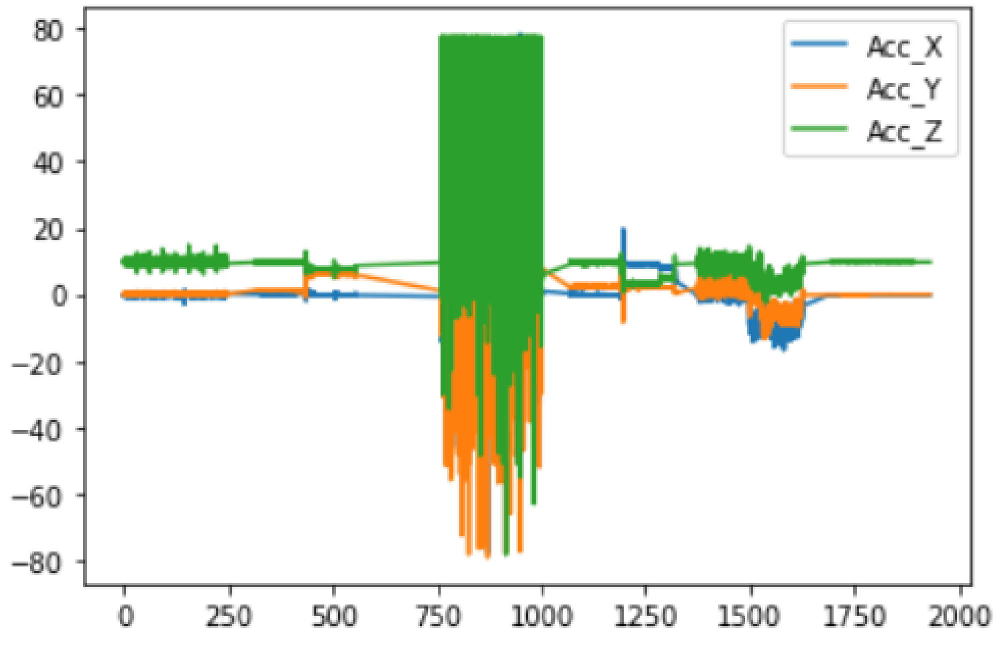
- Data type of the columns of the file

```
Int64Index: 148777 entries, 0 to 148776
Data columns (total 4 columns):
 #   Column     Non-Null Count   Dtype
---  ------     --------------   -----
 0   Timestamp  148777 non-null  object
 1   Acc_X      148777 non-null  float64
 2   Acc_Y      148777 non-null  float64
 3   Acc_Z      148777 non-null  float64
dtypes: float64(3), object(1)
memory usage: 5.7+ MB
```

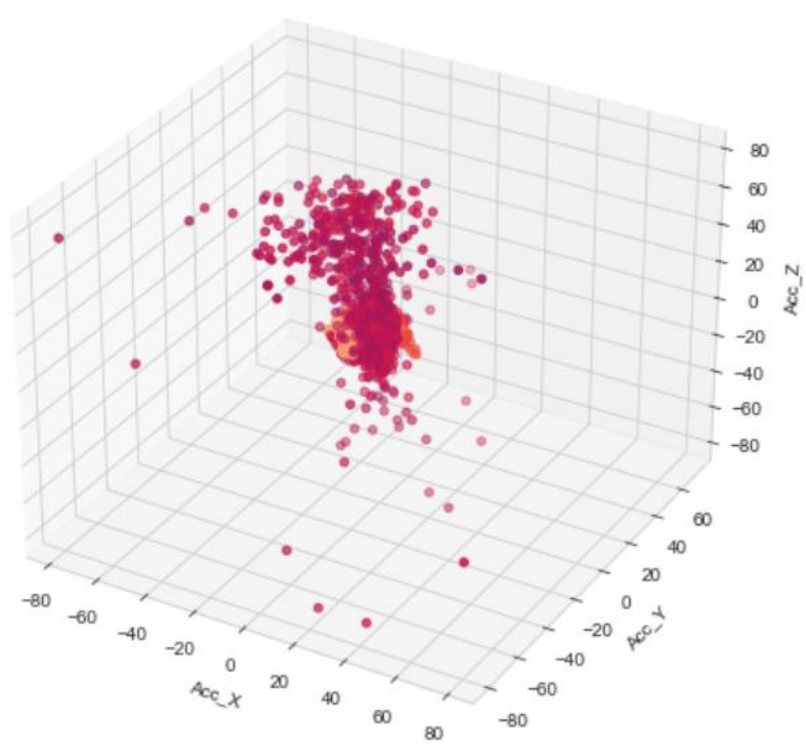
- Missing values

```
Timestamp    0
Acc_X        0
Acc_Y        0
Acc_Z        0
dtype: int64
```

- Plot 2d of the values Acc_X, Acc_Y and Acc_Z of over time



- Plot 3d scatter cloud points



Velocity.csv

- Columns of the dataset

	Timestamp	AngVelocity_Timestamp	AngV_X	AngV_Y	AngV_Z
0	10-11-2019 19:05:56.381	10-11-2019 19:05:56.374	-0.008043	0.006485	-0.013777
1	10-11-2019 19:05:56.391	10-11-2019 19:05:56.384	-0.006196	-0.003392	-0.014942
2	10-11-2019 19:05:56.400	10-11-2019 19:05:56.394	0.002659	0.011626	-0.012663
3	10-11-2019 19:05:56.410	10-11-2019 19:05:56.404	-0.000111	0.012432	-0.011048
4	10-11-2019 19:05:56.420	10-11-2019 19:05:56.414	-0.007276	-0.002988	-0.009649

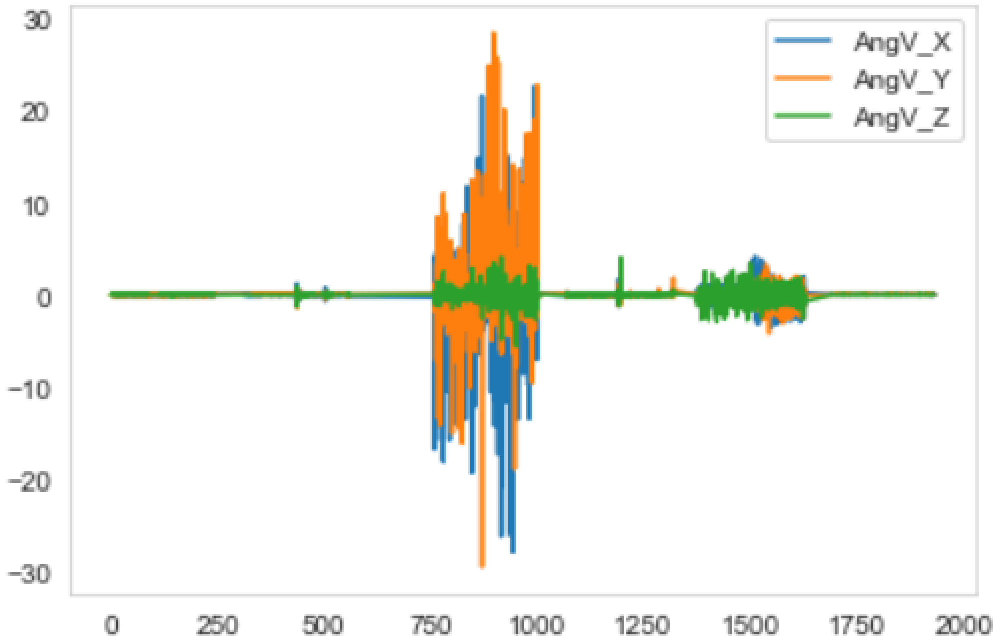
- Datatype of the dataset

```
Int64Index: 148777 entries, 0 to 148776
Data columns (total 5 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   Timestamp              148777 non-null  object
 1   AngVelocity_Timestamp  148777 non-null  object
 2   AngV_X                 148777 non-null  float64
 3   AngV_Y                 148777 non-null  float64
 4   AngV_Z                 148777 non-null  float64
dtypes: float64(3), object(2)
memory usage: 6.8+ MB
```

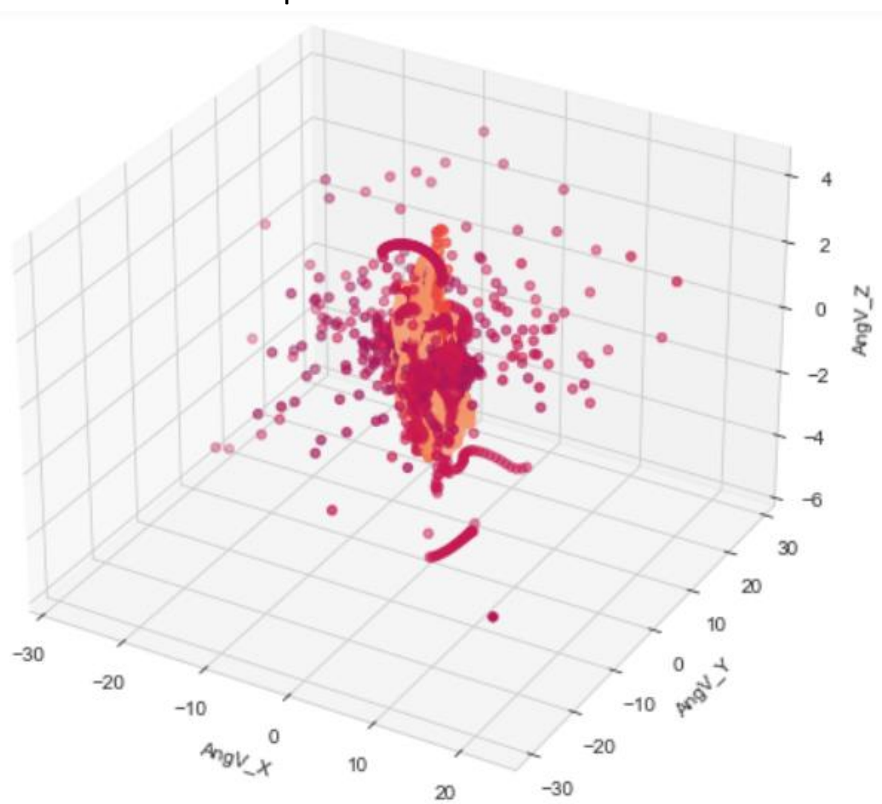
- Missing values

```
Timestamp                0
AngVelocity_Timestamp    0
AngV_X                   0
AngV_Y                   0
AngV_Z                   0
dtype: int64
```

- Plot 2d of the values AngV_X, AngV_Y and AngV_Z of over time



- Plot 3d scatter cloud points



Orin.csv

- Columns of the dataset

	Timestamp	Orientation_Timestamp	Orin_X	Orin_Y	Orin_Z
0	10-11-2019 19:05:56.381	10-11-2019 19:05:56.381	-77.395704	-0.891910	1.271720
1	10-11-2019 19:05:56.391	10-11-2019 19:05:56.391	-77.394480	-0.890001	1.269079
2	10-11-2019 19:05:56.400	10-11-2019 19:05:56.401	-77.393229	-0.891062	1.267604
3	10-11-2019 19:05:56.410	10-11-2019 19:05:56.410	-77.392690	-0.894940	1.271933
4	10-11-2019 19:05:56.420	10-11-2019 19:05:56.420	-77.393355	-0.894859	1.271206

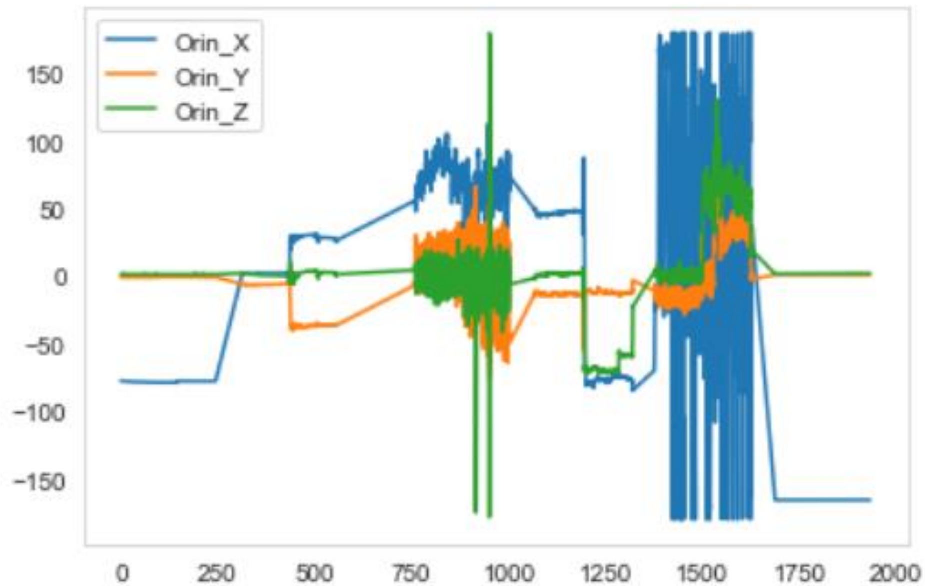
- Datatype of the dataset

```
Int64Index: 148777 entries, 0 to 148776
Data columns (total 5 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   Timestamp              148777 non-null  object
 1   Orientation_Timestamp  148777 non-null  object
 2   Orin_X                 148777 non-null  float64
 3   Orin_Y                 148777 non-null  float64
 4   Orin_Z                 148777 non-null  float64
dtypes: float64(3), object(2)
memory usage: 6.8+ MB
```

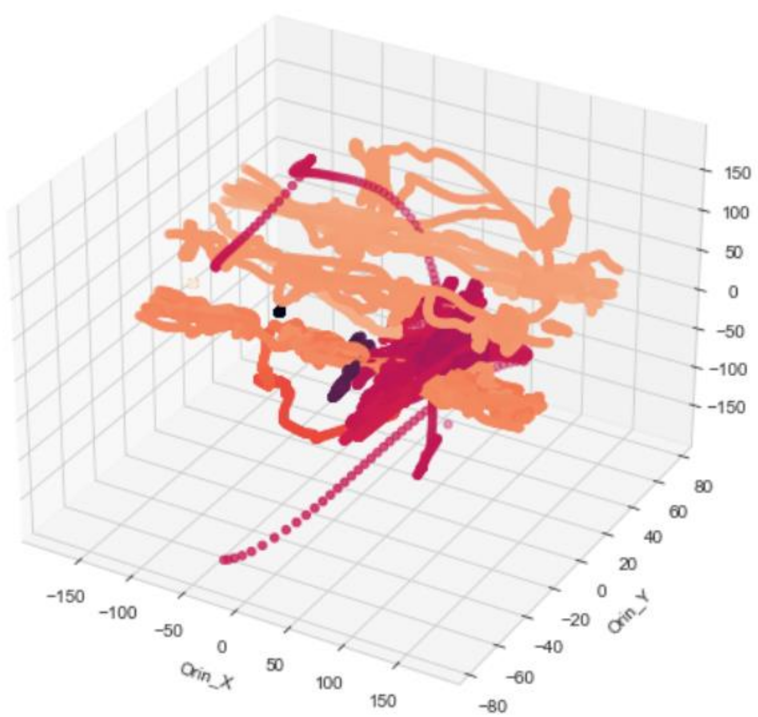
- Missing Values

```
Timestamp                0
Orientation_Timestamp    0
Orin_X                   0
Orin_Y                   0
Orin_Z                   0
dtype: int64
```

- Plot 2d of the values Orin_X, Orin_Y and Orin_Z of over time



- Plot 3d scatter cloud points



Mag.csv

- Columns of the dataset

	Timestamp	MagField_Timestamp	Mag_X	Mag_Y	Mag_Z
0	10-11-2019 19:05:56.381	10-11-2019 19:05:56.381	21.740547	-3.269182	-26.693754
1	10-11-2019 19:05:56.391	10-11-2019 19:05:56.391	17.138367	-10.707533	-22.636873
2	10-11-2019 19:05:56.400	10-11-2019 19:05:56.401	17.138367	-10.707533	-22.636873
3	10-11-2019 19:05:56.410	10-11-2019 19:05:56.410	15.912338	-12.276485	-21.495737
4	10-11-2019 19:05:56.420	10-11-2019 19:05:56.420	15.912338	-12.276485	-21.495737

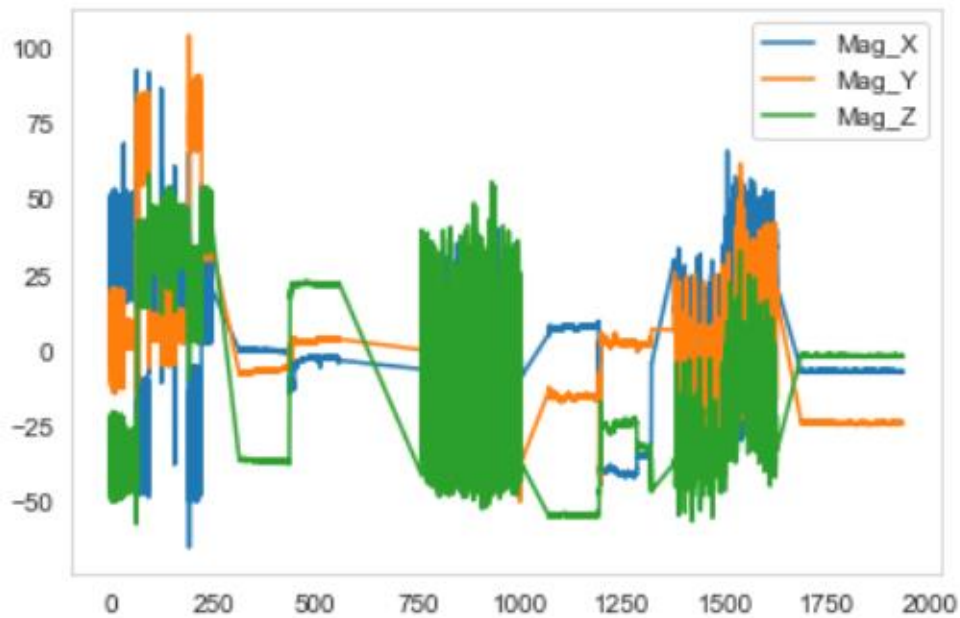
- Datatype of the dataset

```
Int64Index: 148777 entries, 0 to 148776
Data columns (total 5 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   Timestamp           148777 non-null  object 
 1   MagField_Timestamp  148777 non-null  object 
 2   Mag_X               148777 non-null  float64
 3   Mag_Y               148777 non-null  float64
 4   Mag_Z               148777 non-null  float64
dtypes: float64(3), object(2)
memory usage: 6.8+ MB
```

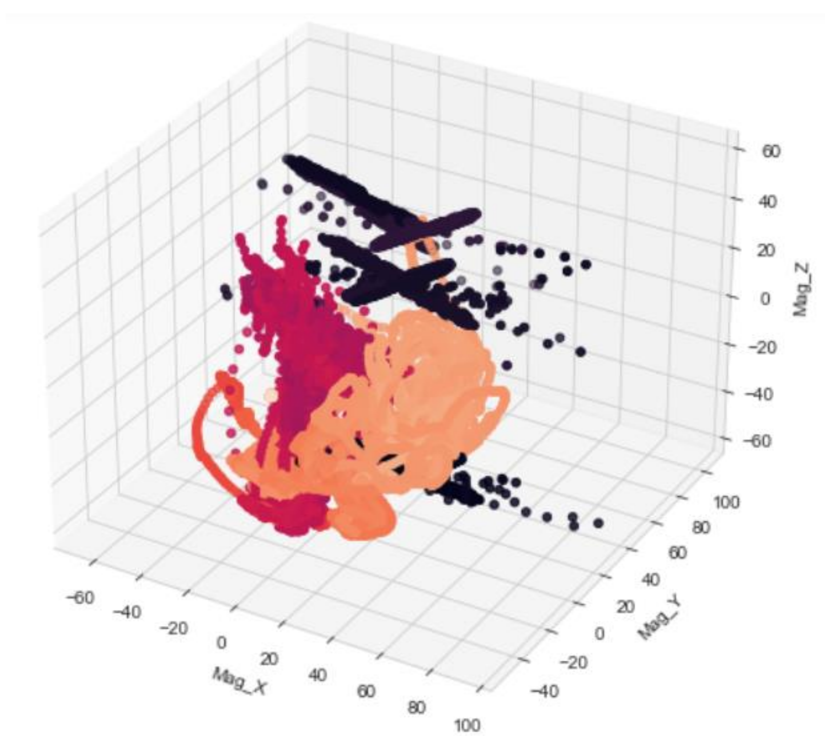
- Missing Values

```
Timestamp             0
MagField_Timestamp    0
Mag_X                 0
Mag_Y                 0
Mag_Z                 0
dtype: int64
```

- Plot 2d of the values Orin_X, Orin_Y and Orin_Z of over time



- Plot 3d scatter cloud points



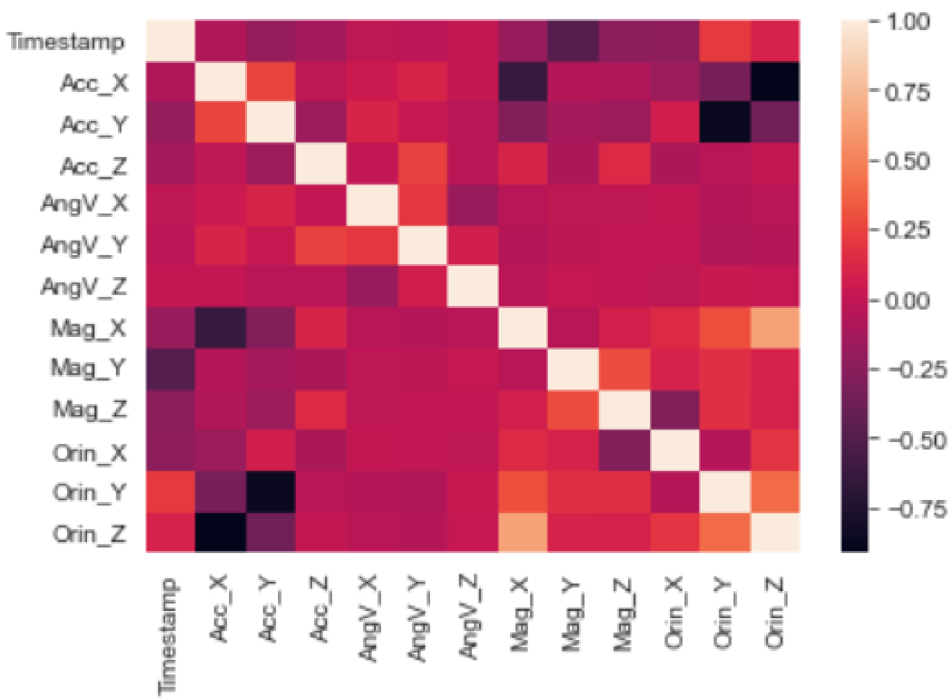
Final Table

After doing a left join to all tables on Timestamp, we obtain the final table.

The timestamp column corresponds to the transformation of the timestamp to seconds, using the first time as a reference '2019-10-11 19:05:56.381000'.

	Timestamp	Acc_X	Acc_Y	Acc_Z	AngV_X	AngV_Y	AngV_Z	Mag_X	Mag_Y	Mag_Z	Orin_X	Orin_Y	Orin_Z	Activity
0	0.000	-0.131576	-0.086370	9.398655	-0.008043	0.006485	-0.013777	21.740547	-3.269182	-26.693754	-77.395704	-0.891910	1.271720	Vibration
1	0.010	-0.122146	0.239352	9.314530	-0.006196	-0.003392	-0.014942	17.138367	-10.707533	-22.636873	-77.394480	-0.890001	1.269079	Vibration
2	0.019	0.095950	0.429307	9.773775	0.002659	0.011626	-0.012663	17.138367	-10.707533	-22.636873	-77.393229	-0.891062	1.267604	Vibration
3	0.029	-0.278271	0.395927	10.230476	-0.000111	0.012432	-0.011048	15.912338	-12.276485	-21.495737	-77.392690	-0.894940	1.271933	Vibration
4	0.039	-0.530796	0.440384	9.580527	-0.007276	-0.002988	-0.009649	15.912338	-12.276485	-21.495737	-77.393355	-0.894859	1.271206	Vibration

Correlation between all the Features



Pre-processing

Standardization

We transform the 12 columns ['Acc_X', 'Acc_Y', 'Acc_Z', 'AngV_X', 'AngV_Y', 'AngV_Z', 'Mag_X', 'Mag_Y', 'Mag_Z', 'Orin_X', 'Orin_Y', 'Orin_Z'] using the standardization technique.

$$\mathrm{SD} = \sqrt{\frac{\sum |x - \bar{x}|^2}{n}}$$

Label Encoding

As an instruction on the beginning of this project, we have to choose only 4 activities, our choice was : ['Walking', 'Vibration','Standing','Sitting'].

We did a Label Encoding to the column Activity on the final table, we assigned:

Sitting : 0

Standing : 1

Vibration : 2

Walking : 3

The final table becomes:

Timestamp	Acc_X	Acc_Y	Acc_Z	AngV_X	AngV_Y	AngV_Z	Mag_X	Mag_Y	Mag_Z	Orin_X	Orin_Y	Orin_Z	Activity
453.493	0.048499	6.051466	8.063582	-0.013392	0.031700	0.016064	-4.583858	2.781290	21.426144	28.993054	-36.606981	0.173703	0
1220.879	8.767418	2.167493	3.746409	0.022829	0.000355	-0.002668	-38.999340	0.783661	-27.333921	-78.422191	-13.026828	-67.104218	1
364.458	-0.051792	1.068927	9.680968	-0.006064	0.002177	-0.015156	-0.462933	-6.958204	-36.808449	1.961038	-6.402093	0.527455	0
206.222	-0.498463	-0.347877	10.282268	-0.002345	-0.007054	-0.011669	-29.075233	76.698303	19.127728	-77.607032	-0.783468	1.259422	2
520.170	-0.322130	5.855374	7.761510	-0.014333	0.009075	-0.019064	-2.745640	3.334394	21.533928	28.490657	-36.199034	2.720874	0

Split Train/ Test / Validation

We split the dataset into Train/ Test/ Validation with the proportion 60/20/20.

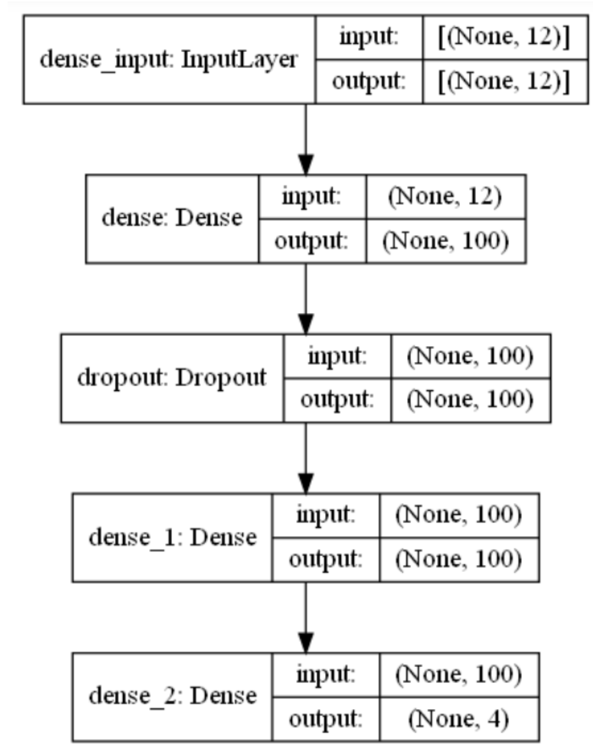
Deep Neural Network

Deep Neural Network (DNN) is a collection of connected processing units known as neurons, which are modeled to reproduce the flexibility and power of a biological brain. A neuron is a generalization of the idea of perceptron, which was developed by Frank Rosenblatt. A perceptron takes multiple binary inputs to produce a single binary output. To compute the output, the weights are introduced to represent the significance of the input a perceptron.

A neuron produces a graded value between 0 and 1. This graded value at the output allows the network to learn the weights with the help of backpropagation. Further, the output values are biased to generate extreme values towards 0 or 1. A DNN is a complex network of these neurons which take various input values to generate decisions that has the potential to mimic human thinking. Since its creation, DNN has been widely used to solve complex tasks.

Model Selection

For our model, we choose the following architecture:



1. In the beginning, a Dense Layer with 100 hidden units.
2. Followed by a DropOut Layer with 50% activation, for regularization.
3. Followed by a Dense Layer with 100 hidden units.
4. Followed by the final Dense with 4 hidden units, and softmax activation.
5. The model takes as input the 12 features ['Acc_X', 'Acc_Y', 'Acc_Z', 'AngV_X', 'AngV_Y', 'AngV_Z', 'Mag_X', 'Mag_Y', 'Mag_Z', 'Orin_X', 'Orin_Y', 'Orin_Z'] and predict the probability of one of the four human activity ['Walking', 'Vibration','Standing','Sitting'].
6.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 100)               1300
_________________________________________________________________
dropout (Dropout)            (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 100)               10100
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 404
=================================================================
Total params: 11,804
Trainable params: 11,804
Non-trainable params: 0
```

Training and Inference

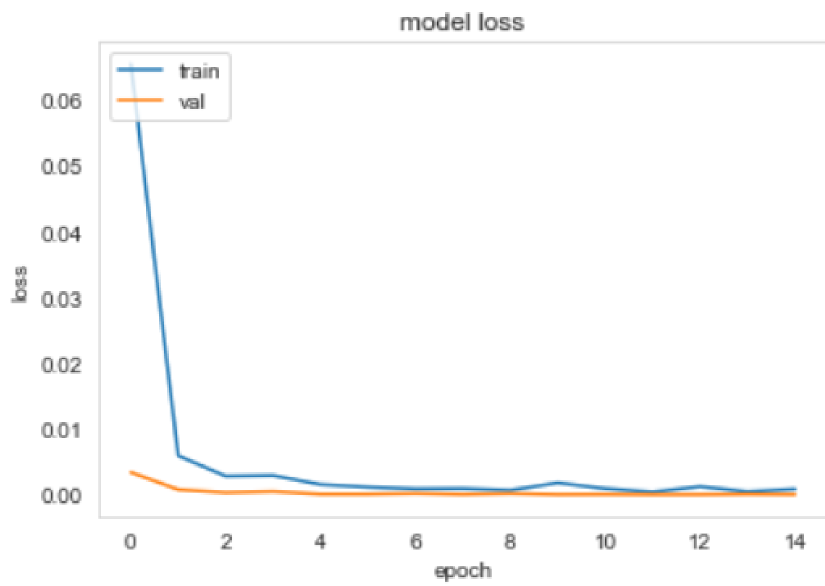
We choose as initial parameters for parameters:

- Batch size = 32
- Epochs = 128
- Early Stopping in call back section: we stop the training of the model after 3 iterations with no improvements.

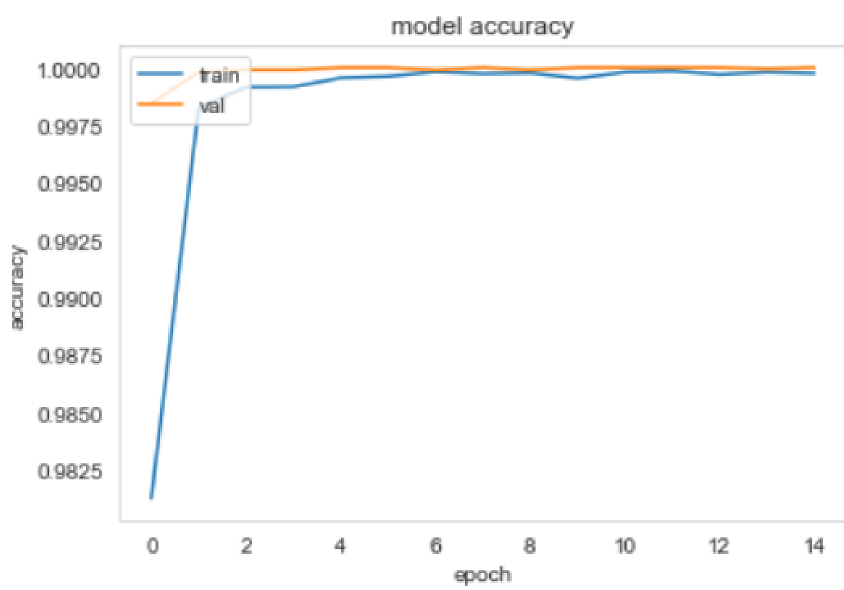
We compile the model with sparse categorical cross-entropy function.

Then, we launch the training.

Loss Function



Accuracy

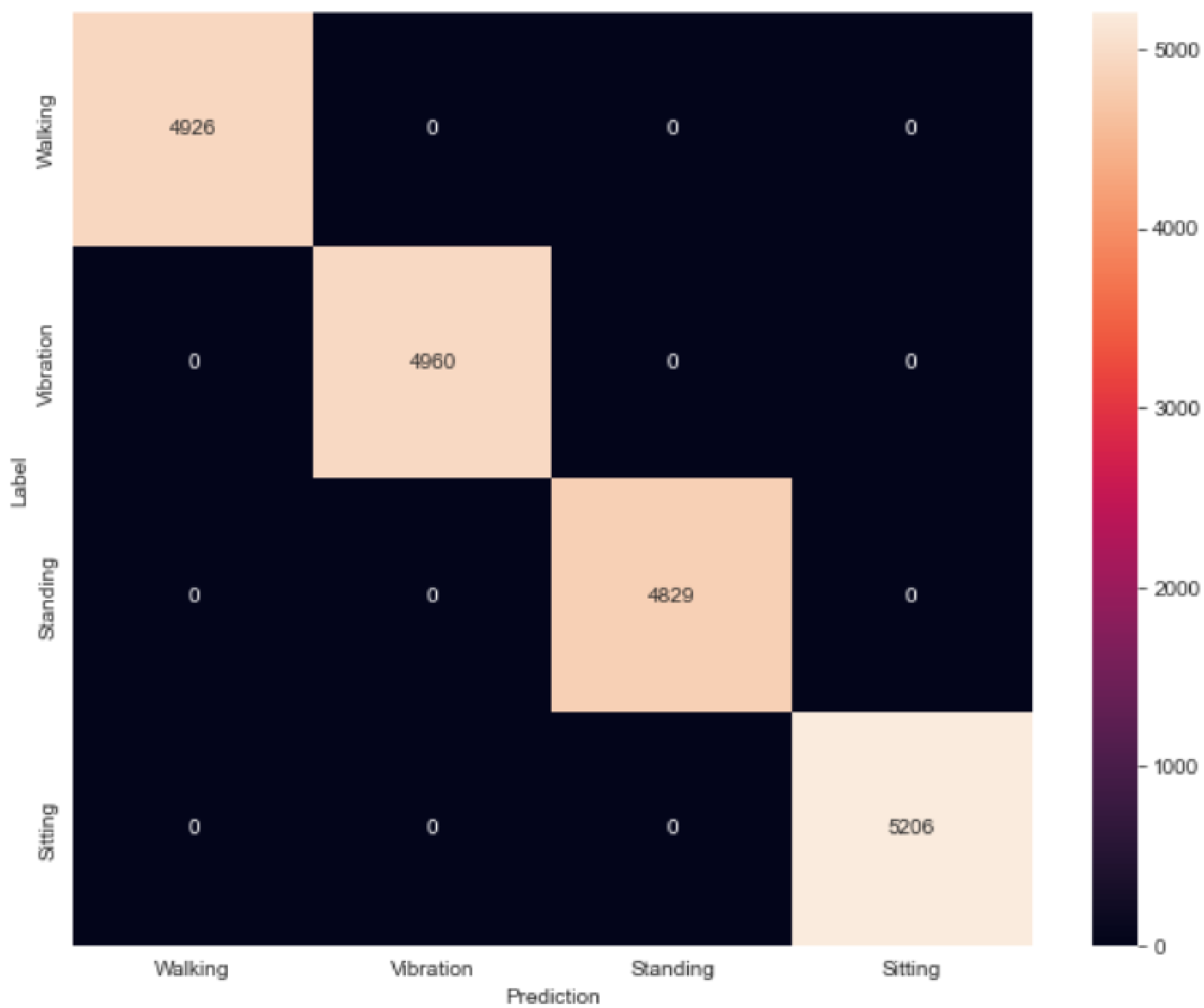


Evaluation on Validation set

Loss and accuracy

```
Evaluate on Validation data
623/623 [==============================] - 0s 446us/step - loss: 4.0338e-05 - accuracy: 1.0000
Validation loss, Validation accuracy: [4.0338243707083166e-05, 1.0]
```

Confusion Matrix



Analysis and Discussion

- The accuracy of the model is 1, on train set, test set and validation set. We were successful to obtain this result without the need to launch a lot of experiments to find the best hyperparameters for the model.
- The model achieved the aim goal. Based on the information of the sensors, our model can predict the activity of the body.
- The accuracy and loss function are performing a little bit better on the test set compared to the train set. This is completely normal behavior of the layer Dropout, that has the functionality to disactivate a portion of the number of neurons and activate them on the inference, which makes the model much smoother and more performing.

Conclusions and Future Work

To conclude, I gained a lot of experiences doing this project, I have got more familiar with Matlab Mobile, and I can use it to analyze my physical performances. Also, I have gained more knowledge of the framework Tensorflow and Keras, and it opened my sights to be able to see what can be done in the field of AI. Because, it will takes too much time, to program everything from the theoretical perspective.