Sensor Music Player

István Szőllősi

Faculty of Sciences and Letters, "Petru Maior" University of Târgu Mureș

August 25, 2018

Contents

1 About

The project is committed to the GitHub, you can find here.

The main structure of the repository is *a valid Android project* with several additionals folders, like the:

- **backend** folder where the *Python* and *JavaScript* codes are stored
- **docs** folder where the documents about the project are stored

2 Node.js

In Node.js is very simple to create a small web server for REST calls.

2.1 Installation

2.2 Configuration

Used tutorial: Build Node.js RESTful APIs in 10 Minutes

2.2.1 Mongoose

2.2.2 Express

2.2.3 Nodemon

3 MongoDB

MongoDB to store signal data from the *Y axis* of the accelerometer from the Android devices.

3.1 Drop collection

Code:

```
show dbs
use <db>
show collections
db.<collection>.drop()
```

Listing 1: MongoDB shell commands to drop a collection

4 Matplotlib

Install module from here.

Optionally, you need to install the **python-tk** package also.

4.1 Examples

According to this official tutorial you can easily generate a plot about an array using this Python script:

```
1 import matplotlib.pyplot as plt
2 plt.plot([ 5.733, 1.704, -2.713, -1.343, 2.604, 3.922, 2.157, -0.910, -2.414,
      -2.943, -1.526, -0.823])
3 plt.ylabel('some numbers')
4 plt.show()
```

Result:

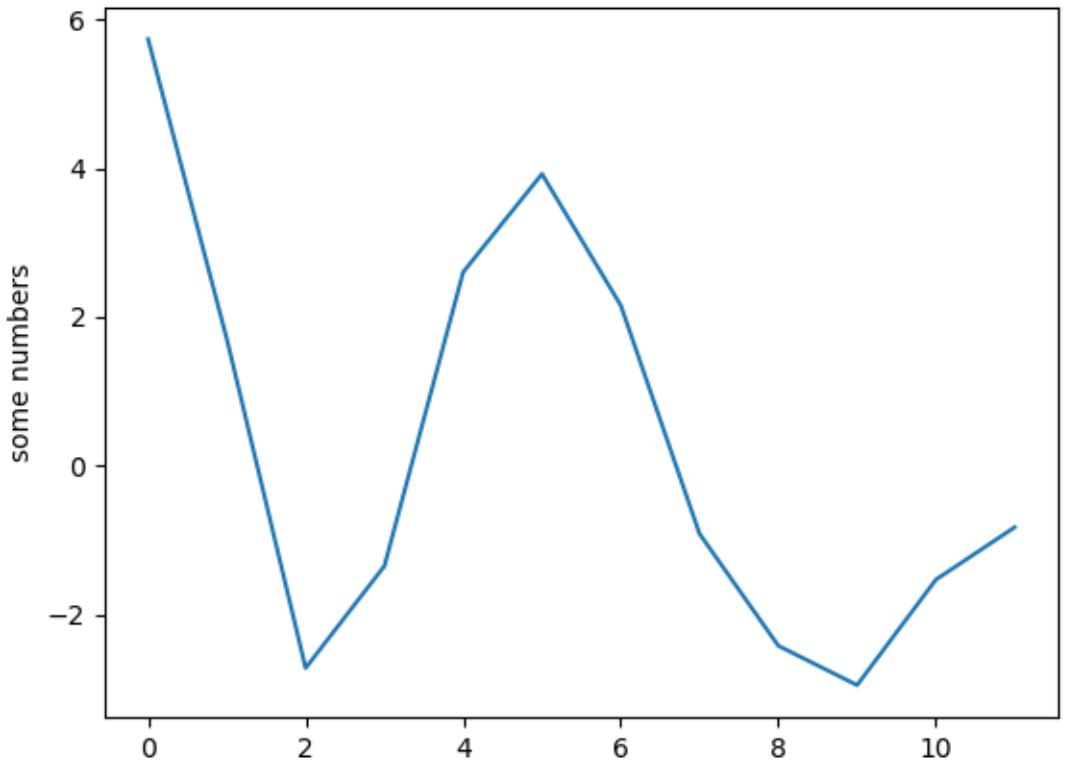

Figure 1: The result of the script

This example was very easy, so here is a *normal* signal from the accelerometer:

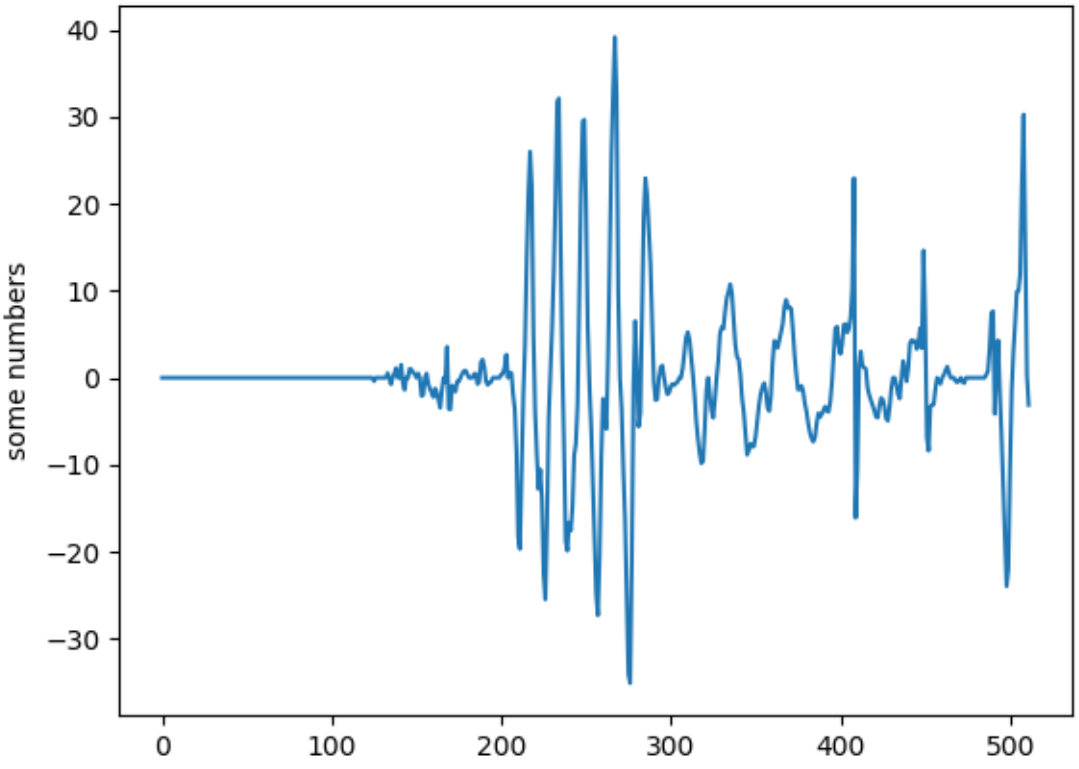

Figure 2: A section of the signal of accelerometer in real usage

5 dtaidistance

FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space

dtaidistance

5.1 Installation

Run the follow codes in terminal:

```
sudo apt install python3-pip python3-setuptools
sudo apt-get install python3-dev
sudo apt-get install python3-tk
pip3 install wheel
pip3 install dtw
pip3 install dtaidistance
```

6 Postman

6.1 Installation

Installed according to this article: How to install Postman native app in Linux Mint 18.3

Used to test the main functionalities of the Node.js server.

6.2 Usage

6.2.1 GET

To get all buffers from database run this code in Postman/Linux terminal

```
curl -X GET http://localhost:3000/buffers
```

Listing 2: Get all buffers

The response is or an empty list, if no items in the database or a list like this:


```
[
    {
        "value": [
            5.733050346374512,
            1.704751968383789,
            -2.7134790420532227,
            -1.343064308166504,
            2.6042985916137695,
            3.92281436920166,
            2.15725040435791,
            -0.9106369018554688,
            -2.4146032333374023,
            -2.943338394165039,
            -1.5269522666931152,
            -0.8230304718017578
        ],
        "_id": "5b82607f5601ec575d3bf0e4",
        "__v": 0
    }
]
```


Listing 3: A sub section of the signal to process

6.2.2 POST

Post a new buffer a.k.a a sub section of the signal to store and process. Run this code in Postman or in a Linux terminal to post a new buffer to the Node.js server.

```
curl -X POST http://localhost:3000/buffers  -d '{
  "value":[-2.2, -1.1, 0, 1.1, 2.2]
}'
```

Listing 4: Send signal data via REST