**VANHACKHATHON 2016**

**Challenge:** Machine Learning Challenge

**Company:** Thalmic Labs

**Author:** Henrique Thor Küsel

**E-mail:** hoofos@hoofos.com

**The Problem**

A gesture recognition company has provided you with a list of 2000 examples of 6 different types of hand gestures.

The data is supplied to you as a collection of CSV files named as follows:

*GestureM\_ExampleN.txt*

where M is the gesture number (1-6) and N is the example number (1-2000). Each CSV file contains a 50x8 matrix (50 time domain samples, 8 channels of data). The value of the sample represents the amplitude of the muscle activation.

Your task is to develop a classification model to differentiate between these gestures so that when a new example of 50x8 data comes in your model will have a high accuracy in classifying it as the correct gesture

Please describe your process of solving this problem, the models you experimented with the results each model achieved and your final results for your best model.

Feel free to code in whatever language you are most comfortable in.

You can find the data here:

<https://www.dropbox.com/s/igoyvp6jqf9y30n/VanHackathon%20Challenges.zip?dl=0>

**The Solution**

**Introduction**

Each csv example contains eight channels of data, what means eight variations of the associated gesture. To solve the problem we must calculate the *mean gesture*, which is the gesture made out of the mean value of all channels.

To get a more precise measurement, I discard the highest and the lowest valued channels. This way it’s possible to avoid undesired discrepancies.

**Technology Applied**

As a Web Developer, I decided to try something different and created a very simple page where it’s possible to interact with all the test files and my solution, checking on the fly the result it’s computing. The url to access the solution is <http://www.hoofos.com/projects/learning_machine> .

All the code is was done in plain Javascript. Just one simple file (*scripts/gesture\_handler.js*) with some well documented (lots of comments) functions. I really hope you like and enjoy the idea.

**The Testing**

To test out the solution I’ve created a factor for each gesture based on the csv files. Basically the test is done by generating the same factor for the file to be tested and comparing it to the known factor previously saved. To make the test simple and user friendly, there is only one button to send the file to be tested. After the test is done, the solution outputs a fictitious name for the gesture, an fictitious image to illustrate it, the factor computed for the file and the factor previously saved for that specific gesture.

Here are the fictitious names related to gestures 1 to 6, as in the name of the files:

|  |  |
| --- | --- |
| **Fictitious Gesture** | **Gesture 1 to 6** |
| Pinch | Gesture 1 |
| Rotate Left | Gesture 2 |
| Rotate Right | Gesture 3 |
| Spread | Gesture 4 |
| Swipe Left | Gesture 5 |
| Swipe Right | Gesture 6 |

**Calculating the Factor**

The factor is created based on the data of the csv files. Each file contains a 50x8 matrix: 50 different entry for each of the 8 different channels. To compute the factor, the system follows these steps:

* analyzes row by row, discarding the highest and the lowest values for the purpose of avoiding discrepancies;
* create a mean value of the other 6 channels;
* the mean value is then added to the mean values of the other rows;
* the factor is created by getting the mean value of this addition.

**Is It Bullet Proof?**

Well, unfortunately the system does not work for all the cases. The main reason is that the known factor saved in the system was created using random test files, and there are several files with data that just does not fit there. The best solution would be to use the whole set of files to create a more reliable standard factor, but still we would probably stay with something between 80% and 90% of success rate.

**Some Tests Results**

|  |  |
| --- | --- |
| **Test File** | **Result Gesture** |
| Gesture1\_Example154.txt | Pinch (Gesture 1) |
| Gesture6\_Example30.txt | Swipe Right (Gesture 6) |
| Gesture1\_Example20.txt | Pinch (Gesture 1) |
| Gesture5\_Example1242.txt | Swipe Left (Gesture 5) |
| Gesture4\_Example612.txt | Spread (Gesture 4) |
| Gesture2\_Example1049.txt | Rotate Left (Gesture 2) |
| Gesture3\_Example902.txt | Rotate Right (Gesture 3) |