Machine Learning Engineer Nanodegree

Capstone Proposal

David Lopez Rubio

February 24, 2017

Proposal

(approx. 2-3 pages)

Domain Background

(approx. 1-2 paragraphs)

In this section, provide brief details on the background information of the domain from which the project is proposed. Historical information relevant to the project should be included. It should be clear how or why a problem in the domain can or should be solved. Related academic research should be appropriately cited in this section, including why that research is relevant. Additionally, a discussion of your personal motivation for investigating a particular problem in the domain is encouraged but not required.

This problem is englobed into activity recognition by accelerometers

data.

https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Usi

https://archive.ics.uci.edu/ml/datasets/PAMAP2+Physical+Activity+Monit

For example this two datasets have been widely used.

Also this paper is a perfect resume of all the work in this field.

https://www.mdpi.com/2227-9709/5/2/27

You have two ways to solve, and many things to keep testing if you obtain no results. You can treat data as “independently and identically distributed” with any standard ML algorithm after windowing data and applying a transform to frequency or other domains. But you can also use the time dependency and use raw data without domain transformation and feed to a RNN for example, a machine learning technique where cells have memory about past inputs. Example of this (using one of the datasets above)

https://medium.com/@curiousily/human-activity-recognition-using-lstms-on-android-tensorflow-for-hackers-part-vi-492da5adef64

However, this approach differ from mine because the sensor is body-worn and I intend to install sensor in a furniture to detect human activity. Also, lab approaches differ from real life in power consumption of electronics. For a marketeable product, you should keep the power

use under reasonable figures. Using a gyro is feasible only for example if you plan to have a product you recharge at most weekly.

Problem Statement

(approx. 1 paragraph)

The final aim of this project is to detect whether a chair is being used of not. Inputs are accelerometer data. Computing power to classification task needed goal should approach in the end the ARM Cortex-M4 domain. No more. Simple goal. Special considerations.

_1) Low power, only accelerometer data at a 50Hz will be used. By Nyquist-Shannon theorem we will be able to sample up to 25Hz signals. No gyro data. Gyro burns from 1mA upwards meanwhile accelerometer works as low as 6uA. That is a full magnitude order.
_2) If our classificator could run at the end in a very basic embedded microprocessor that would be a huge goal met.

In this section, clearly describe the problem that is to be solved. The problem described should be well defined and should have at least one relevant potential solution. Additionally, describe the problem thoroughly such that it is clear that the problem is quantifiable (the problem can be expressed in mathematical or logical terms) , measurable (the problem can be measured by some metric and clearly observed), and replicable (the problem can be reproduced and occurs more than once).

Datasets and Inputs

(approx. 2-3 paragraphs)

I will use my own dataset. I have attached a ARM Cortex M4 with LSM6DSL accelerometer/gyro board that saves on SD-card raw axes data at 50Hz. Data is tri-axial, X,Y,Z. Board have a LiPo battery and is taped to a standard office chair. With that I have recorded around 1 hour of raw data labeled by me as sitting, and a lot of time not sitting.

See dataset exploration.ipynb notebook.

Dataset is unbalanced in raw because I just capture data and stop manually. However since the annotated data is captured separately in different data structures, is very easy to just crop the structures to have a balanced 50%/50% dataset if this helps.

Solution Statement

(approx. 1 paragraph)

In terms of black-box, the solution will be something that after feeding raw or preprocessed accelerometers data in a given time window (example 4-7 seconds) that black-box will return if the chair is being used or not.

Benchmark Model

(approximately 1-2 paragraphs)

Output will be obvious just looking at data for a human. From a rest position, only acceleration from Earth will appear on accelerometer data plus some noise. Then if we use a fairly high-pass filter (more than 2Hz for example), and compute x,y,z vector absolute value, if it differs from 9.8 m/s2 then other forces are moving the accelerometer and is likely chair is being used.

But also values are previously labeled. Output will be true/false so we can compare against labeled data.

Benchmark will start with naive prediction. Is important for this to have the percentage of data of each class in the corpus. If 50/50, naive predictor will approach 0.5 just by random guessing. If we beat that, we are (starting) to do well. If I do worst, my approach is just wrong.

Evaluation Metrics

(approx. 1-2 paragraphs)

F1 score as weighted average of the precision and recall. In my project is important both to have minimum false positives and false positives.

That can be computed from the predictions and the labeled data in the validation cut of data.

Project Design

(approx. 1 page)

First dataset checks, cleaning and fitting in NumPy data structures for easy work from the txt.
Since this is a temporal series data, time axis should be consistent with data. Specially if at the end I have to use transform domain techniques.

I will write a function to work with the time axis. Inputs are time in seconds and window length. Return is an numpy structure with the raw values, size acc sample rate in Hz * window length in seconds and true or false if the chair has been used for more than 50% of of that time window. Time axis work is very important, I usually work with pandas timestamp64 and that allows me to compute timedeltas easily.

Data cuts for training, test and validation should be done at the beginning. Test data should never be used to test hyperparameters. Since this is a time series with non uniform distribution of data, cuts should be supervised just to not choose a full datapoints of just one state. Ideal should be 50% chair being used, 50% chair empty.

Next I will try to train a very simple classificator algorithm that works with simple data preprocessing. Only algebraic operations in a given window of time of data.

I will try several processing prior to feeding to a classificator:

Compute data variability for each axis and feed to classifier (3 inputs, after normalizing they go from 0 to 1) Compute resultant acceleration vector modulus (1 input, after normalizing it goes from 0 to 1)
If that fails I will start training with time series transformed to frequency domain. Remember I want to keep computation as simple as possible so moving to this point will be a little failure. For such an easy task this seems impossible to fail.
Preprocessing this time would be:
Fast Fourier Transform coefficients computation for a given time window. Then spectrum from 0 -25 Hz will be divided in bucks, and a fixed number of this bucks will be feed to classifier. I expect no more than 2 - 3. Direct cosine transformation coefficients computation for a given time window. I am less confident with DCT.
At the end I will evaluate the models by the F1 score and wrap up the cons and pros of every one favoring the ones that require less computing resources.

And to finish I will provide a working classifier code with my choice of algorithms.

Algorithms to explore include classic classification ones, starting with decission trees, towards random forest (I think random forest could be a solid algorithm for this) and also I can try if I have time a simple RNN based deep learning net. RNN will require a different approach and data preprocessing since we can feed it with discrete values and it will (hopefully) use its memory abilities to choose using previous feed values.

As I am trying to be consistent and to attain real results, RNN presents a huge drawback, and that is that running a RNN classificator in a very limited ARM Cortex M4 with 256kB RAM at most is difficult.

However, for RNN a clear hardware target could be this cheap chip that a former mining IC ASIC company has just baked. This chip, besides MCU has a neural network hardware based part that can be programmed. It is specially designed for CNN, but I will check anyways.
https://github.com/kendryte