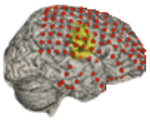

Completed • $8,000 • 202 teams

UPenn and Mayo Clinic's Seizure Detection Challenge

Mon 19 May 2014 – Tue 19 Aug 2014 (15 days ago)

Features for seizure detection

Michael Hills

Hey everyone, I've posted up my documentation and code in another thread, but I thought I'd post here as well as I also wanted to give my answer on the question of how did people handle the size of the data with respect to computational speed etc.

For feature selection I used FFT 1-47Hz, concatenated with correlation coefficients (and their eigenvalues) of both the FFT output data, as well as the input time data. The data was then trained on per-patient Random Forest classifiers (3000 trees).

My background is Computer Engineering, so I spent a lot of time optimising my development cycle to be fast. As an example using 4-cores I can do a cross-validation run against all patients on a 150-tree Random Forest using FFT 1-47Hz in under 4 minutes. The processed data is also cached for re-use if I wanted to try another classifier on it. You can check out my code in the other thread, but to summarise the techniques that I used to streamline the code:

- Process each piece of training data as it comes in rather than loading it all and transforming all of it at once. This keeps memory usage down. Before doing this I used to crash my laptop a LOT even with 16GB of RAM.
- Cache all processed data to save recomputing it again later
- For python numpy arrays use hickle (h5py) instead of pickle, loads most data in 0 seconds
- Never hold in memory data you don't need, e.g. for making predictions, first load training data to train the classifier, then drop that data and load the test data for classification. Loading data from hickle format is fast enough that you can forget about load times
- Dump scores to disk so I can re-run previous runs to get immediate scoring output if I forgot to copy and paste it out
- Can specify lists of different data pipelines to try and lists of classifiers to try. The code would run all combinations and print out the cross-validation scores in sorted order at the end. Additionally if I decided to cancel a run, I could run it again from roughly where it left off as any intermediate results were already cached to disk.

For final classification though I would use 3000 estimators in my Random Forest which blew out run times to 2-3 hours.

Having a fast SSD is also a huge benefit.

Thanked by Lalit , Jonathan Street , Matthew Roos , rcarson , TheAnalyticProphet , and 2 others

#21 / Posted 9 days ago / Edited 9 days ago

Reply / Quote / Thank / Flag / Email User

< 1 2

Reply

File Edit Insert View Format

Formats

p

Words: 0

+ Add attachment(s) ...

Post Reply

Email me when someone replies

Start Watching View all posts « Back to forum