# Terminology

Subsample – a set of EEG readings collected at the same instant.

Sample – a set of contiguous subsamples. Samples are flattened into a vector for presentation to the network algorithms. The number of subsamples in a sample can be just a few, as when training the network input layer, up to an entire 10 minute file, as when classifying a test file.

Minibatch – the number of training samples processed between network parameter updates.

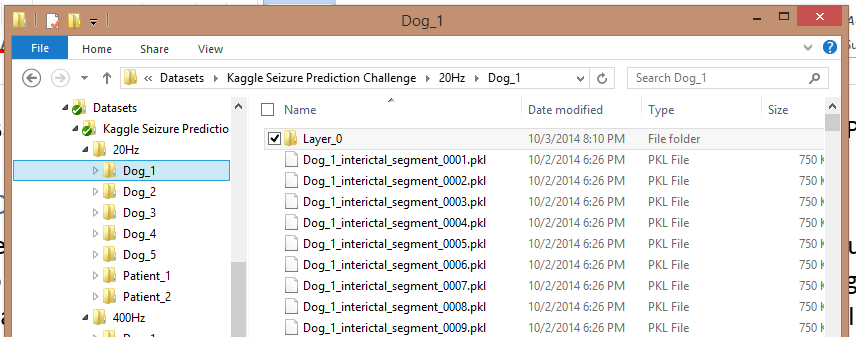
GPU Batch – a set of training and validation samples that can be loaded onto the GPU during training.

# Accomplishments

Created preprocessing functions to preprocess mat files in a single directory and multiple directories for the complete dataset. The preprocessing removes linear trends from the data along the time axis, decimates the data to a specified sampling frequency, and normalizes each channel to a constant standard deviation of 1/24 = 0.0417 centered about 0.5 and clipped at 0 and 1. Only a very small fraction of the samples are clipped with this scaling. The file processing function saves each preprocessed matfile as a pickled tuple object.

Preprocessed the entire dataset for 20Hz sample rate.

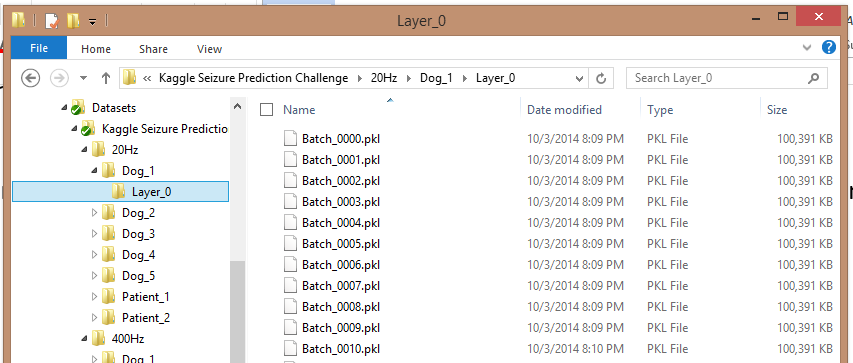
Preprocessed the entire dataset for 400Hz sample rate. Note that the dog data is already sampled at 400Hz, but the human data is sampled at 5000Hz, so only the human data sample rate was changed in this case.



Created functions to assemble sets of pickled GPU batch files. Each GPU batch file contains both training and validation samples with specified subsample counts. The functions partition the dataset into training and validation samples at the 10 minute file level according to a user specified split ratio. All training samples are drawn from the training partition, and all validation samples are drawn from validation partition. Each sample is drawn by randomly selecting a file and then drawing a sample with the specified number of contiguous subsamples from a random position within that file. Each GPU batch of training and validation sets is pickled.

Created training & validation data with an 80/20 split using the 20Hz sampled dataset, 16 samples per pattern, about 100MB per batch, and 20 batches.

Created training & validation data with an 80/20 split using three of the seven individuals from the 400Hz sampled dataset, 16 samples per pattern, about 100MB per batch, and about 20 batches. (I had to interrupt processing after more than 12 hours to use my computer again.)



Experimented with training a one layer RBM with varying numbers of neurons on one 20Hz and one 400Hz Dog\_1 GPU batch file. The following results are based on training for 50 epochs in each case.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hidden Neurons | 20Hz RMSE | | 400Hz RMSE | |
|  | Training | Validation | Training | Validation |
| 10 | 0.0348 | 0.0347 | 0.0276 | 0.0280 |
| 20 | 0.0304 | 0.0304 | 0.0218 | 0.0223 |
| 50 | 0.0219 | 0.0221 | 0.0139 | 0.0142 |
| 100 | 0.0145 | 0.0149 | 0.0089 | 0.0091 |
| 200 | 0.0081 | 0.0086 | 0.0065 | 0.0067 |
| 500 | 0.0046 | 0.0050 | 0.0046 | 0.0047 |
| 1000 | 0.0031 | 0.0031 | 0.0039 | 0.0040 |
| 2000 | 0.0031 | 0.0031 | 0.0037 | 0.0038 |

Experimented with training a two layer RBM where the first layer had a fixed size of 1000 neurons and the second layer has various sizes. In this case, I processed only the 20Hz Dog\_1 GPU batch file and again trained for 50 epochs.

|  |  |  |
| --- | --- | --- |
| Hidden Neurons | 20Hz RMSE | |
|  | Training | Validation |
| 10 | 0.0379 | 0.0377 |
| 20 | 0.0354 | 0.0352 |
| 50 | 0.0316 | 0.0315 |
| 100 | 0.0246 | 0.0245 |
| 200 | 0.0187 | 0.0186 |
| 500 | 0.0089 | 0.0090 |
| 1000 | 0.0059 | 0.0059 |
| 2000 | 0.0055 | 0.0055 |

# Plans

Generate GPU batch files for 16, 256, 4096, and 65536 samples.

Modify the RBM training program to pretrain an hierarchical autoencoder with constant depth per layer.

Compare its RMSE to a single layer network with the same hidden layer dimension.

