# Accomplishments

* Cleaned up and commented the CPU version of the SequenceDecimatingNetwork.
* Modified the API to use 3D (pattern, sample, feature) arrays for both network inputs and outputs. A pattern is defined as the sequence of samples which contribute to a single output sample.



* Converted the CPU version to a GPU version using the cudamat library. This was a little more difficult than expected for two reasons: 1) the cudamat library uses Fortran style memory organization while my python numpy code depends on C style memory organization to look at the same memory in two different ways without moving it, and 2) the cudamat library only supports 2d matrices. These required more intrusive modification of the code than might otherwise have been required.
* Found that I could exploit buffered representations in Python to create a linear parameter vector view of the weight and bias matrices. This allows very efficient (no cost) serialization and deserialization of the model parameters.

# Plans

* Complete the workflow for submission preparation – create ROC plots, AUC metrics and then assemble everything into a procedure to train and test a network for submission
* Make an initial submission to the contest before the closing date

# Question that we never discussed

The training data has a large class imbalance as described in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| Dir | Interictal Files=0 | Preictal Files=1 | E(Class==1) |
| Dog\_1 | 480 | 24 | 0.0476 |
| Dog\_2 | 500 | 42 | 0.0775 |
| Dog\_3 | 1440 | 72 | 0.0476 |
| Dog\_4 | 804 | 97 | 0.1077 |
| Dog\_5 | 450 | 30 | 0.0625 |
| Patient\_1 | 50 | 18 | 0.2647 |
| Patient\_2 | 42 | 18 | 0.3000 |

When training and validating the classifier, is it better to present training examples drawn from these classes in this ratio or to present an equal number of examples from the two classes?