# Plans

Create preprocessing functions to preprocess a file, directory, or dataset. The preprocessing function should supper options to remove constant offset from the signal, normalize each channel to a constant average (or perhaps median) power, and apply a lowpass anti-aliasing filter, and decimate the data by a specified ratio. The file processing function should save each preprocessed matfile as a pickled object.

Preprocess the dataset with offset removed, amplitudes normalized, antialiasing filter applied, and sample rates forced to 400Hz. Note that the dog data is already sampled at 400Hz, while the human data is sampled at 5000Hz, so only the human data sample rate will be changed in this case.

Preprocess the dataset with offset removed, amplitudes normalized, antialiasing filter applied, and sample rates forced to 20Hz.

Create a function to load the data for an individual into memory and assemble a specified number of epochs, where each epoch contains a specified number of training and validation samples with a specified contiguous pattern length. For each output pattern, it should randomly select a file and then draw a training pattern from a random position within that file. The function should optionally apply a specified multilayer temporal decimating neural network to the training data. The function should compile the transformed output into a 2d numpy array where each row is a training pattern, each column a feature, and a 1d numpy class membership vector. The function should save the completed batch of training and test data as a pickled object.

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

# Accomplishments