Classification of Voltage Spikes from Neurons using Machine Learning

Computational Intelligence - EE40098 | Candidate: 12627 | Date: 13/12/2019

By using machine learning (ML) techniques, the identification of voltage spikes, generated by 1 of 4 neurons, in a set of brainwave can be made using supervised learning. This is done by feeding a machine learning algorithm with a set of pre-classified data generated by similar neurons. The ML technique used to to carry this out includes a combination of Principle Component Analysis (PCA) and K-Nearest Neighbors (KNN). Other ML

PCA is a method of dimensionality reduction within the data which allows a comparison among the more variant characteristics in the data. Important characteristics will likely include the height and with of a peak. **KNN** in a method of grouping similar structured data, to each other and classifying them based on where they lie relative to each other. So once again, taller spikes will likely have a more similar data structure to each other and thus will likely lie closer to each other. By combining these methods together, dimensionality reduction can be done using PCA before implementing the data into KNN for classification, this allows for faster classification and memory efficient classification technique. Artificial Neural Networks (ANN) can also be using rather than KNN to classify this data, however it was found that with a shallow ANN the performance was worse than KNN and a deeper ANN, required much more tuning to performed better than a KNN, and was also slower to train than KNN. For this reason, a combination of PCA and KNN was used to classify the data.

Before ML is applied to the brainwave data, it needs to be filtered so all the peaks beginnings and ends can be normalised to zero. This was done using a bandpass filter with a band of 100-2k Hz. Along with this, a peak detection algorithm was used to find the indexes for all the peaks generated by the neurons.

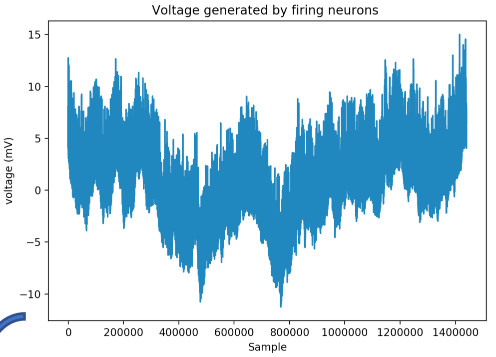

Figure1: Submission Data

BP Filter
100 – 2k Hz

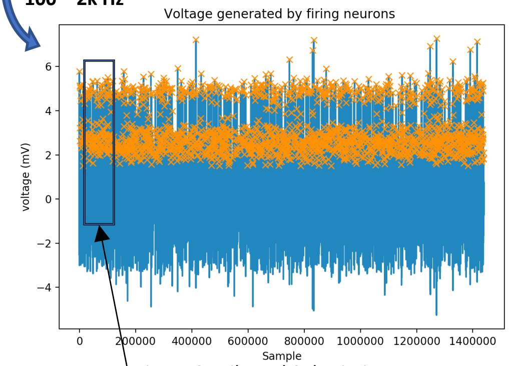

Figure 2: Filtered Submission Data

An up-close section of the data, showing the peaks (X) clearly defined.

The submission data was found to produce 1998 peaks using this filter and peak detection method

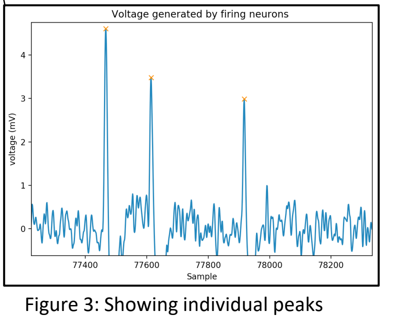

Figure 3: Showing individual peaks

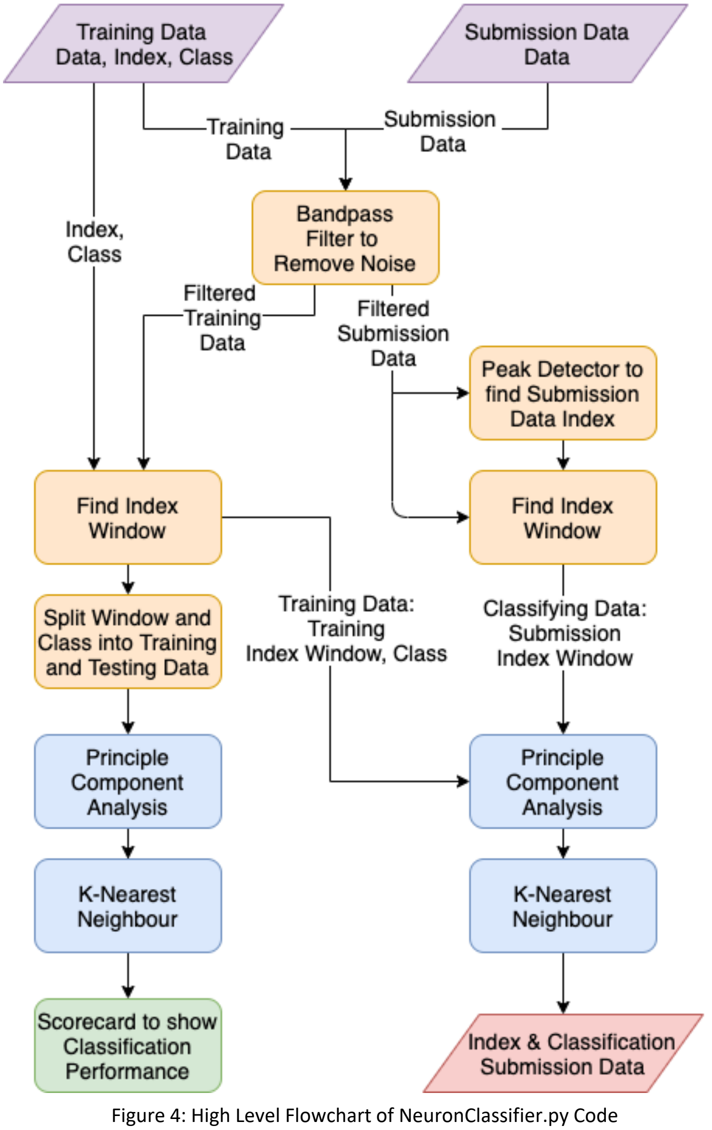

Figure 4: High Level Flowchart of NeuronClassifier.py Code

The ML algorithm was first classified and tested using a range of the Testing.mat data provided before applying it to the Submission.mat data.

The Training.mat data was split into a of training and a set of testing data (Code line 150) to observe the performance of the classification on control data. An output performance of 93.3% was observed this performance was satisfactory so now the remaining Training.mat data was trained on the ML algorithm and classification of the submission data was completed. This data can be found in the reproduced Submission.mat file.

I believe the training and classification within the Training.mat data performs well, however, due to the extra noise in the Submission.mat data as well as the extra implementation of a peak detector to find the index, I am uncertain about the performance of the classification on the Submission.mat data.

It was found that small differences in the window used to classify the peaks had a big impact on the percentage of each neuron found in the submission data. Since each neuron was found to appear approximately equally in the training data, it was assumed that the submission data would have a similar ratio of neurons being fired and thus the window sizes were tuned to find a ratio that came close to this.