Overview

Within this course you have learnt about classification and optimisation techniques that are inspired by the intelligence seen within nature; you will now use these techniques to further our understanding of natural intelligence. The analysis and classification of large datasets is one of the primary applications of AI/CI methods. For example, deep convolutional neural networks have extensive applications in computer vision and image processing, where they can classify features and objects within many thousand different images. This coursework will test your knowledge, and ability to apply this knowledge, to a classification task that is inspired by current biomedical engineering research. You will create a system to automatically analyse a set of recordings that have been made from the human brain – one of the most complicated structures known to exist. This document details the datasets that you will be working with, the submission details, and the marking criteria.

Recordings

You will be working with recordings made using a simple bipolar electrode inserted within the cortical region of the brain, this is a typical experimental setup that is frequently employed by neuroscientists. The recordings contain several *spikes* (extracellular action potentials) that are from five different types of neuron (Type 1, 2, 3, 4 & 5). Each neuron produces spikes that have *a subtly different morphology*, and each neuron can only produce one spike at a time. One of the challenges with this type of recording is that neurons often fire together, and some of the spikes will be partially overlapping.

The goal is to process the recordings and automatically identify when each spike occurs, and which neuron produced it (often called *spike sorting*). This is akin to the MNIST classification problem, except that you also need to detect when in time each spike has occurred in order to extract them for classification. This information will enable the selective recording from *individual neurons*, a critically unmet need in modern neuroscience. The recordings are time records and Figure 1 illustrates an example of two spikes, both are from the same type of neuron and have approximately the same morphology. The sample rate for all of the recordings is 25 kHz.

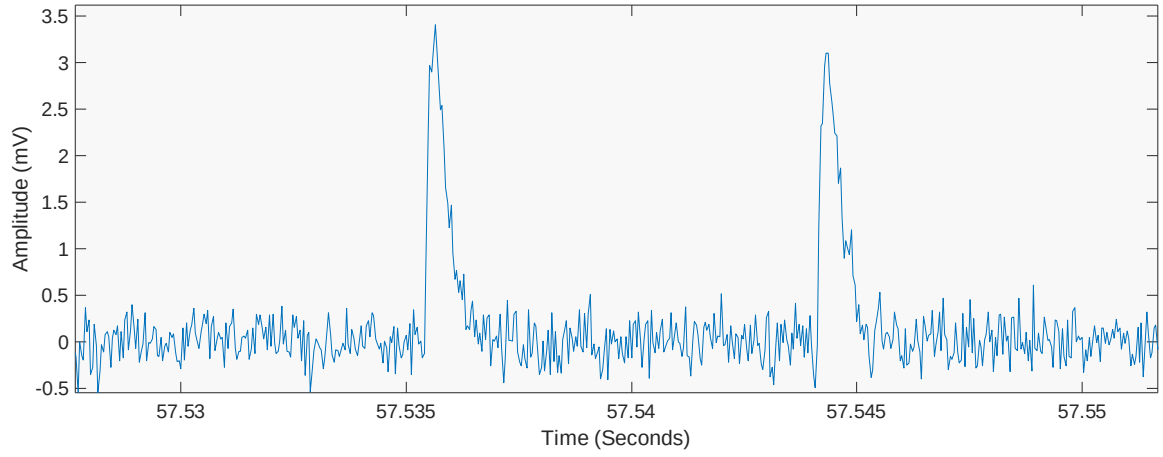

Figure 1 An example of two spikes within the training dataset, both spikes are from the same type of neuron and have approximately the same shape and amplitude.

Training Dataset

There is a training dataset on Moodle (training.mat) that you can use to develop your algorithms. The training dataset has been generated using a detailed simulation model and contains a single time domain recording of the spikes from the five types of neurons. It is up to you how you wish to split this data into training, testing, and validation datasets.

The dataset is available on Moodle, it is a MATLAB data file that contains the following three vectors:

Vector	Description
d	Raw time domain recording (1440000 samples), 25 kHz sampling frequency.
Index	The location in the recording (in samples) of each spike.
Class	The class (1, 2, 3, 4 or 5), i.e the type of neuron that generated each spike.

The Index and Class vectors can be used to train your algorithms and to assess their performance on unseen data. You should firstly consider how you can detect the spikes, then consider classification as the next step. You should use the Python tools developed in the laboratory sessions to solve this challenge, and your final solution must be written in Python.

Note: You can import a .mat file into Python using the following code snippet:

```
import scipy.io as spio

mat = spio.loadmat('lowNoise.mat', squeeze_me=True)

d = mat['d']

Index = mat['Index']

Class = mat['Class']
```

Submission Datasets

Once you have trained and tested your algorithms using the training dataset, you should run them on the submission dataset that is also on Moodle. This dataset contains a real recording made from the cortical region of the brain of a human and is also expected to contain spikes from five neurons, of the same type in the simulation (i.e., the morphology should be *similar* to the simulated recordings). Unfortunately, the subject was moving when these recordings were being made, and so the noise is much worse than the training dataset. You will notice that the Class and Index vectors are missing from the submission dataset, it is your job to analyse the dataset and generate a Class vector and an Index vector.

Tasks

The following tasks form the core coursework components, you should attempt them in order and the results should form the basis for a final report. You are free to use whatever CI techniques you wish, alongside any standard signal processing or mathematical tools. You may use CI techniques that have not been taught in the course.

1. Identify suitable performance metrics for this classification problem from the literature and write code that will compute these metrics for you.
2. Implement and test one CI technique (perhaps a simple MLP) on the training dataset and assess the performance using your metrics from (1).
3. Based on the results from (2), identify and implement a second, different, CI technique and repeat the assessment.
4. Selecting the result from either (2) or (3) use an optimisation technique (such as simulated annealing) to optimise the parameters of your chosen technique until you have reached the best performance you can.

Submission Details

Once you are satisfied that your best classifier has identified and classified all of the spikes within the submission dataset, you should upload your code and your classifications to Moodle using the strict guidelines below. Your electronic submission to Moodle should be a **single ZIP file** that includes:

1. All of your Python source code, for each of the four tasks, with appropriate comments. This should be runnable, so that your code can by tested by the examiners. If you used any packages that are not in Anaconda then please provide simple installation instructions for these packages.
2. A MATLAB file named yourCandidateNumber.mat that contains an Index and Class vector that you have generated for the submission dataset using whichever technique you predict will give the best result. This should be in the format detailed in the table above (i.e., with vectors called *Class* and *Index*).
3. A report in **PDF formant,** with a maximum of **10 pages,** that contains the following sections:
    a. Introduction
    b. Performance Metrics
    c. First Solution
    d. Second Solution
    e. Optimisation Approach
    f. Conclusion & Confidence Level

Marking Criteria

Your coursework will be marked against three criteria, marks will be deducted for failure to follow the submission details defined above.

1. **20% - Readability and quality of your code and comments**. You may use third party libraries, but your comments should explain how the function calls to these libraries work. Your comments should be sufficiently detailed that the code could be recreated using just the comments.

2. **20% - Performance of your classifier.** Your Index and Class vectors will be compared against those detected using a state-of-the-art detector and a performance mark will be assigned based on the number of true positive detections. Marks will be deducted for false positive detections. For the Index vector, we will apply a window of +/- 50 samples to allow for small errors.

3. **60% - Quality of justifications and explanations of the methodology in the report.** As a rule, your report should focus on the high-level design of your solutions and must include a critical analysis of **why the solutions were appropriate**. You should include a statement of your own confidence in your final classifier performance. You will be marked based on your justification of the methods you used, higher marks will be awarded for innovative methods or solutions that are based on observations from the data.

Hints

Start out by plotting the recordings (like Fig. 1), and make sure you understand what is being asked of you. You may find it helpful to spend some time reading around the problem, and you should consider a range of different approaches. This is a real-world research problem where many of the CI techniques you have been taught are being used – there is no perfect solution.