Assessment Brief

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 the human brain. 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 datasets that contain 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 (Class 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 different neurons often fire simultaneously, and some of the spikes will be partially overlapping.

The goal is to process the recordings and automatically find when each spike occurs, and which neuron produced it (often called *spike sorting* in the literature). This is akin to the MNIST classification problem, except that you also need to detect when in time each spike has occurred 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 the same morphology. The sample rate for all the recordings is 25 kHz.

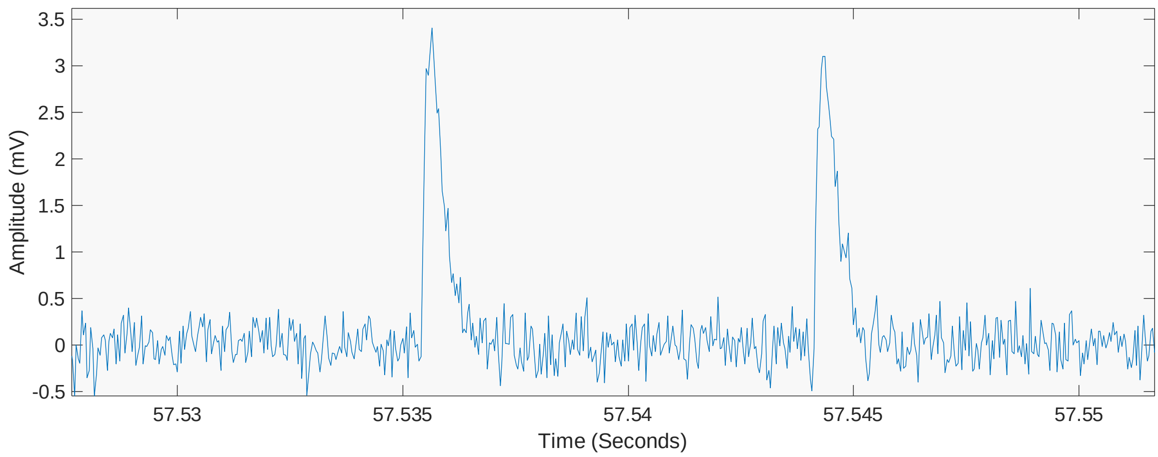

Figure 1 An example of two spikes, both are from the same class of neuron and have approximately the same morphology.

Dataset Overview

There are six datasets that you will use for this coursework, they are all in the same format and are available on Moodle. Each dataset is a MATLAB data file that contains some or all of 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 the start of each spike.
Class	The class (1, 2, 3, 4 or 5), i.e the type of neuron that generated each spike.

For example, the first element in the *Index* vector states where (in samples) the first spike occurs in the *d* vector; the corresponding first element in the *Class* vector states the class of this 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('D1.mat', squeeze_me=True)
d = mat['d']
Index = mat['Index']
Class = mat['Class']
```

Datasets

The following datasets are available on Moodle.

Dataset	SNR (approx..)	Notes
D1	80 dB	This is a low-noise dataset that is fully labelled, it has both Index and Class vectors that are correct. You should use this for training your classifier.
D2	60 dB	This is an unlabelled dataset; it only contains the d vector.
D3	40 dB	This is an unlabelled dataset; it only contains the d vector.
D4	20 dB	This is an unlabelled dataset; it only contains the d vector.
D5	0 dB	This is an unlabelled dataset; it only contains the d vector.
D6	< 0 dB	This is an unlabelled dataset; it only contains the d vector.

Tasks

The following tasks form the coursework. 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. Load dataset D1 into Python and, using it as training data, devise a method for finding and classifying the individual spikes in the *d* vector and thus regenerate the *Index* and *Class* vectors.

2. Load datasets D2 to D6 into Python. Detect and classify each spike in these datasets using any CI technique you wish and thus generate the vectors *Index* and *Class* for each dataset. You should upload the corresponding .mat files to Moodle in a **single ZIP file** that includes one or more files. For example, once you have a submission for dataset 2 you may upload a ZIP file containing the file D2.mat (in which should be the vectors *Index* and *Class*). Note: do not include the vector *d* in your submissions as this may exceed the maximum file size on Moodle. Once you have a submission for dataset 3 you may upload a ZIP file containing the files D2.mat and D3.mat and so on.
   Submission: D2, D3, D4, D5 and D6 .mat file containing the vectors *Index* and *Class*.

Feedback: Auto marked feedback provided.
Weighting: 60%

3. Once you have completed task 2 write a summary (no more than 400 words) in the free text area of the submission portal on Moodle that describes the CI methods you have used and reflects on the overall performance and sensitivity to SNR.
   Submission: 400-word commentary
   Feedback: Free text feedback.
   Weighting: 30%

4. Once you have completed tasks 2 & 3 you are required to upload your Python code to Moodle, this will be marked on the overall code quality (e.g., structure, variable names, meaningful comments). Your Python code needs to be in a **single ZIP file** with clear instructions on how it can be run (e.g., list of libraries to be installed).
   Submission: All Python files and instructions.
   Weighting: 10%

Feedback & Feedforward

An automarker will run on any submissions uploaded to the data submission portal by 12:00 on Monday and Wednesday each week. You may make as many attempts as you wish by uploading revised submissions. The final run will be on Wednesday 10th December, you can continue to upload revised submissions until the final deadline on Friday 12th of December. Two additional runs will occur at 12:00 on Friday the 28th of November and Friday 5th of December (i.e. the two Fridays before the deadline)

The automarker will grade your task 2 ZIP file against the known ground-truth of spike location and class for each dataset. For the Index vector, we will apply a window of +/- 50 samples to allow for small errors. Automarked feedback will be in the form of precision, recall and confusion matrices. Examples of how these are computed for multi-class classification are available [here] and an example feedback file is available on Moodle.

The automarker provides feedback and feedforward information as you are able to improve your submission using the feedback. For example, you may find that your solution consistently mis-classifies spikes belonging to Class 3 and may choose to focus on this as an area for improvement between submissions.

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.

During the coursework C task, you will have lots of opportunities to submit labelled data and receive feedback that may be used to improve your solution. We are always asked how the submitted data will eventually lead to a final mark and this document is intended to shed some light into that.

Dataset Performance Marking Breakdown

The following section details how your labelled dataset will be marked.

Marking a Dataset

There are two distinct problems that you must solve to accurately identify a neuron pulse and its class in the recordings. The first is pulse location identification and the second is pulse classification. Now it may be that your approach solves both in one go – but this does not change the underlying problem. To recognise that some students may thrive in one of these tasks at the cost of the other, we produce a grade for each task separately and then combine them into one final grade.

F1 Score

We use the F1 score to quantify the success of both pulse identification and classification. This is a statistical measure that combines **precision** and **recall** into a single metric using their harmonic mean.

This is mathematically defined as:

$$F = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Once we have two F1 scores, one for identification and one for classification respectively, we combine them with a weighting as follows:

$$F_{Dataset} = 0.3F_{Ident} + 0.7F_{Class}$$

Weighting Datasets

There are 6 datasets provided – with D2-D6 unlabelled, and therefore suitable for marking purposes. As you move through the datasets the Signal to Noise ratio (SNR) decreases – meaning it is harder to accurately classify. Therefore, to combine the F1 scores from each dataset we use an additional weighting scheme as follows:

$$F_{Final} = 0.1F_{D2} + 0.15F_{D3} + 0.2F_{D4} + 0.25F_{D5} + 0.3F_{D6}$$

This produces a single value that may be considered a measure of success at the task.

Marking Caveats

Remember, your grade is formed from a combination of this marking and other elements. All grades are considered provisional and are subject to change until they have gone through the board of studies.