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.

Stage Gating

This coursework uses a stage gating approach to help you build towards the final submission. There are 4 tasks in total, each of which will allow you to make a submission and receive automated feedback. This gives you the opportunity for multiple points of feedback (on work submitted so far) and feedforward (areas to improve in future submissions) throughout the course.

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 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.

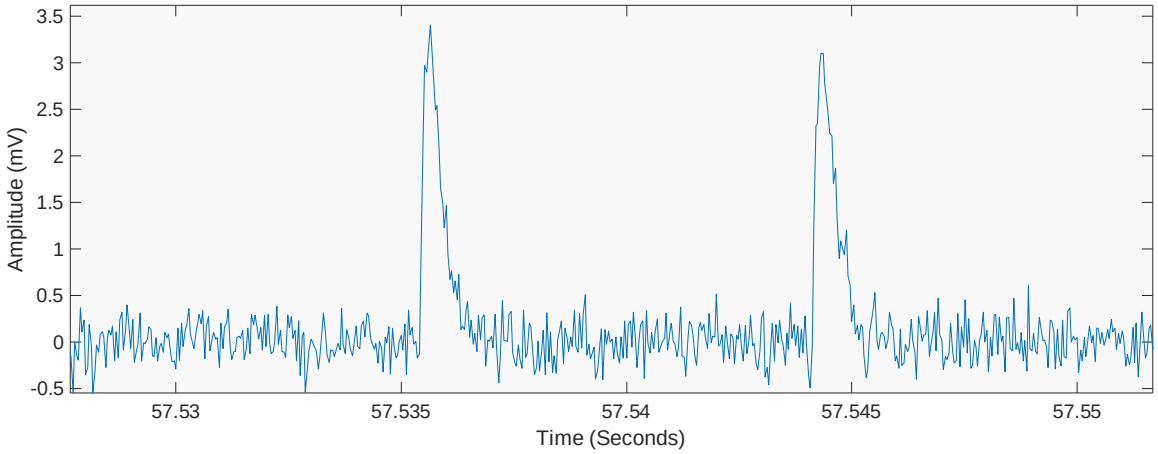

Figure 1 An example of two spikes, both are from the same class of neuron and have approximately the same shape and amplitude.

Dataset Overview

There are four datasets that you will use for this coursework, they are all in the same basic 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. It is up to you how you wish to split this data into training, testing, and validation datasets.

Dataset	Description
D1	This is a low noise recording that is fully labelled, it has both Index and Class vectors that are correct. You should use this for training your classifier.
D2	This is a low noise recording that does not contain labels (i.e., it only contains the *d* vector).
D3	This is a high noise recording that does not contain labels (i.e., it only contains the *d* vector).
D4	This is a high noise recording that does not contain labels (i.e., it only contains the *d* vector).

Tasks

The following tasks form the coursework, and you should attempt them in order. 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. Each task will require you submit a .mat file to Moodle in the format specified.

1. Load dataset D2 into Python and devise a method for finding the individual spikes in the *d* vector and thus generate the *Index* vector.
   Submission: D2.mat file containing the vectors D and Index.
   Feedback: Precision and recall of the spike locations.
   Weighting: 10%

2. Load dataset D2 into Python and detect and classify each spike using any CI technique of your choosing. You may wish to use dataset D1 to help you with training your classifier.
   Submission: D2.mat file containing the vectors D, Index and Class.
   Feedback: A confusion matrix outlining the performance.
   Weighting: 20%

3. Load dataset D3 into Python, this is a more realistic dataset that contains more noise so will be harder to process. Detect and classify each spike using any CI method of your choosing.
   Submission: D3.mat file containing the vectors D, Index and Class.
   Feedback: A confusion matrix outlining the performance.
   Weighting: 30%

4. Load dataset D4 into Python, this is the final dataset and has similar noise as D3. Detect and classify each spike using any CI technique of your choosing. Once you have completed this task you should also write a brief (no more than 200 word) summary in the free text area of the submission portal that describes the CI methods you have used and reflects on the overall performance you have achieved. As a rule, your summary should focus on the high-level design of your solutions and must include a statement explaining why the CI solutions were appropriate.
   Submission: D4.mat file containing the vectors D, Index and Class. 200-word summary.
   Feedback: Detailed feedback provided only after submission closes.
   Weighting: 40%

Once you are satisfied that you have completed a task you should upload the corresponding .mat files to Moodle. Your electronic submission to Moodle should be a **single ZIP file** that includes one or more files. For example, once you have completed task 1 you may upload a ZIP file containing the file D2.mat. Once you have completed task 3 you may upload a ZIP file containing the files D2.mat and D3.mat and so on. You will also be required to upload your Python code to Moodle, this will not be marked but will be checked for plagiarism and will be considered alongside your 200-word summary. 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).

Marking

For tasks 1 to 3 an automarker will run twice a week on Tuesday and Thursday, with a scheduled timeline on Moodle, this automarker will download your ZIP file (if provided) and upload the technical feedback detailed above for any tasks that you have submitted. You will be able to submit as many attempts for automarking as you wish so long as each attempt is submitted before midday on a Tuesday or Thursday. Automarking will be used to provide you with rapid feedback about your performance, however task 4 will only be marked after the submission portal closes. Final marks will be awarded based on a sliding scale with the best performance in your cohort receiving the highest marks. Marks will be deducted for false positive spike detections. 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 either precision and recall scores or a confusion matrix. Examples of how these are computed for multi-class classification are available here.

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.