Coursework I FAQs

Nirmalie Wiratunga, Sadiq Sani and Kyle Martin

October 15, 2018

Update - 15/10/18

1 Are the datasets uniform?

No. Each dataset is completely unique, and will have to be loaded into your python script as such. You should examine each of the datasets carefully. Much of the individual work from this coursework stems from understanding how to load the datasets and apply them to different algorithms. Though only a few changes to the code should be necessary, your results will be poor if the data is loaded incorrectly.
When loading the dataset into your code, you should think about:

- How many examples are there in the dataset?
- What are the indices of the class label and features?
- How many features are there?
- How will you normalize your features? (more info in the next question)

2 How do I normalize each dataset?

Normalization of each dataset will improve your results. When you normalize the features, you are standardizing them to a set of values between 0 and 1. As each dataset is unique, there is no 'one size fits all' normalization method (meaning that if you simply copy and paste the MNIST normalization, it will not work on other datasets). Instead, you have these options:

Min-Max Normalization:

$$Normalized(x) = \frac{x - min(X)}{max(X) - min(X)} \tag{1}$$

Standard Deviation Normalization:

$$Normalized(x) = \frac{x - mean(X)}{StDev(X)} \tag{2}$$

Min-Max normalization is demonstrated in the MNIST labs. We suggest Standard Deviation here as an alternative, as some of you will have done that previously in some of the data science courses.

It is important to remember that you can only normalize a dataset in this way if the data is consistent across features (i.e. features exist within a similar range). If the features exist across different ranges (for example, in the breast cancer dataset), then you should normalize each feature individually. This can be done either in the code, or using Excel on the dataset csv files.

Please note: Train and test sets should be normalized independently and obviously must use the same normalisation strategy (i.e. min-max or std based).

3 How should I split into train and test sets within datasets?

The train-test split of the dataset is an important consideration in your evaluation. If train-test split is already provided then you can use them as is. If not, then you should consider it carefully, ensuring that (a) your training set offers enough data to allow effective training of the algorithm and (b) that the test set is large enough to convincingly demonstrate your results. Having a test set size of 10 examples may offer 100% accuracy on a given algorithm, but this is hardly a meaningful evaluation. You should justify your choice of train-test split (if creating your own) in the first section of the coursework.

When implementing the train-test split, keep in mind that you should ensure the disjoint between your train and test set is random. Otherwise, you may have classes which appear in your test set but not your training set, meaning your algorithm's accuracy will suffer. Look at the train-test split during loading of the dataset in the kNN lab from week 4 to get ideas about how to do this.

In some situations (depending on the power of your machine) you may have to work with a smaller samples of the train or test sets (as we had to do in Lab 3 with the training set). What ever strategy you choose simply state it clearly and use the same strategy consistently when applying different configurations of your algorithm to that dataset. For more information about the importance of train-test splits to the specific algorithms, you should refer to the lab and lecture from week 3 (ANN), as well as the lecture from week 4 (kNN).

There will also be further examples in week 5 in the lab.

For the more keen and ambitious student; you might even want to work with several disjoint train - test splits and average the accuracy results over several runs.

Update - 12/10/18

4 What should I write in the abstract?

The abstract is an important component of any research document. It acts as a summary of the entire paper, allowing researchers to quickly pinpoint sections of the document which are relevant to their work. You should provide an abstract for your coursework submission.

An abstract is not simply a 'copy and paste' of the introduction. The abstract should provide an overview of the entire coursework document (i.e. the selected datasets, the examined algorithms and the observed results). A good rule of thumb is to write the abstract last, so that you can summarise the contents of each section in one or two sentences each.

5 How long should the CW be?

We have suggested a length of approximately four pages, divided as so:

- **Part 1 - Introduction:** Half a page.
- **Part 2 - The ANN Classifier:** One full page.
- **Part 3 - The k-NN Classifier:** One full page.
- **Part 4 - The Hybrid Classifier:** One full page.
- **References:** Half a page (where applicable).

The template is in dual-column style, meaning that 'one full page' means both columns on a single page of A4, formatted as per the template.

Students will not be penalised for going slightly over the recommended limit. The goal of this exercise is to measure your understanding of the algorithms, as well as your practical ability to apply them in a simple evaluation task. You should therefore take the time and space that you feel are required to demonstrate that knowledge. That being said, you should avoid waffling, as it is likely to make your argument less clear.

We do not believe that this exercise can be appropriately completed to a 4th year standard if you were to go under the suggested limit.

Update - 6/10/18

6 What am I expected to do for CW1?

For full details of the coursework, please refer to the coursework specification document on Moodle. However we summarise a few key points next.

In coursework 1, we are expecting you to demonstrate an understanding of two algorithms; the Artificial Neural Network (ANN) and k Nearest Neighbour (k-NN). We would like you to demonstrate a full understanding of the theory behind these algorithms, and show that you know how to adjust their hyperparameters to find an optimal solution to a given problem. This will culminate in a hybrid system which combines both algorithms.

When submitting, you should submit both a written report and your full working code in your zipped submission. The code should be structured as three ipynb files - ann.ipynb, knn.ipynb and hybrid.ipynb. The report should be broken down as follows.

- **Part 1 - Introduction:**
  1. Select two datasets from the provided list.
  2. Discuss your choices and how you will split them into train and test sets, considering what factors have led to this decision.
  3. Note that all of the sections below should be completed for **both** datasets.
- **Part 2 - The ANN Classifier:**
  1. Paragraph detailing students understanding of hyperparamaters.
  2. A graph for each hyperparameter comparing the different values.
  3. Paragraph explaining the results.
- **Part 3 - The k-NN Classifier:**
  1. Paragraph detailing understanding of weighted and unweighted kNN.
  2. Graph comparing weighted and unweighted kNN at different values of k.
  3. Paragraph explaining the results.
- **Part 4 - The Hybrid Classifier:**
  1. Paragraph detailing understanding of the hybrid system functionality.
  2. Table comparing the results of the ANN, kNN and hybrid.
  3. Paragraph explaining the results.

There is no need for a conclusion - the report should finish with the discussion of your hybrid classifier results.

7 How do I create the graphs for the CW?

We encourage the use of the matplotlib package when creating the graphs for your coursework. Though we will not penalise the use of Excel (or similar packages), you will find it much easier to export your results directly as a graph, rather than saving them to a file before exporting them to a spreadsheet.

8 What template should I use for the written components of the CW?

A template has been provided for the written component of the coursework. It is on moodle and in a word format. Additionally, we have provided formatting guidelines to keep you right. You should use the existing template (see cm4017-cw1-template.doc), rather than attempting to recreate it from scratch.

9 Am I expected to comment my code?

Since we have provided the code which will act as a basis for your experimentation, we will already have knowledge of its functionality and background mechanics. Therefore, we do not require strict documentation or commented code. However, if you feel that any of the methods you have adopted or code you have used is confusing or unclear, it would be helpful if you could comment that part. This will help us to understand your thought process, and might enable us to grant marks for that (even in situations where the code itself may be incorrect).

Please Note: The process of your experimentation should be completely clear from your report for both pieces of coursework - we should not have to refer to the original code for clarity. **Documentation of code should never be used as an excuse for an unclear report.**