Coursework 1 – Data Preparation and Classification

Date Released: Friday, 20th October 2023

Date Due: Friday 10th November 2023

This coursework is worth 20% of your overall mark

1. Introduction

In this coursework, you are required to carry out a mini research project in Machine Learning. This will take the form of supervised image classification on the EMNIST dataset. You will be required to load the data, create training and testing subsets, and use these to train several machine learning models and perform inference on the testing data. You will then be required to evaluate and compare the results, using this to determine the most suitable model.

The submission will include the code as a single .m file and the report as a 1-page extended abstract. An abstract template is included for use and an example abstract submitted to a conference is included for reference.

2. Dataset

The EMNIST dataset contains 26,000 images of hand-written letters along with labels showing the letter that they represent. The dataset is contained in the file `dataset-letters.mat`. This contains a variable `dataset` holding the images and labels. (**Hint: We access this data in the same way we access table columns.**) Each image is of size 28 x 28 pixels, stored as a reshaped 1 x 784 vector. To view this as an image, you can use the `reshape` command.

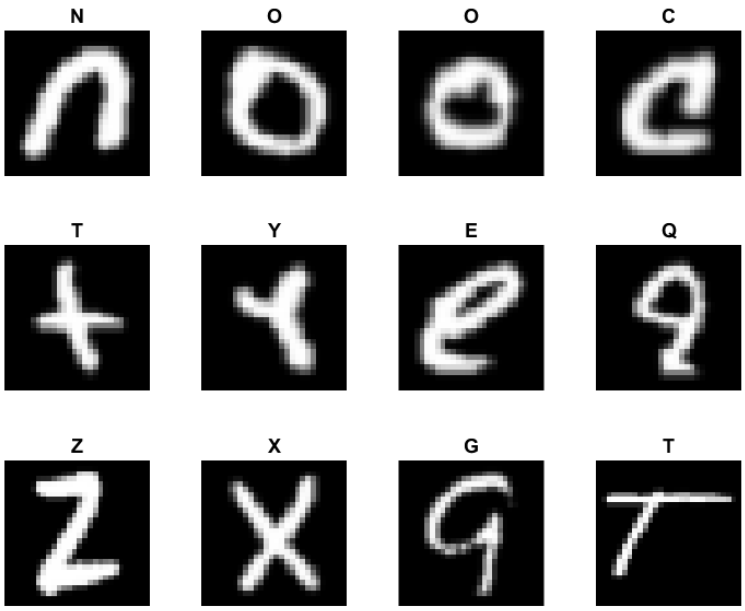

Figure 1: Example Data from EMNIST

3. Data Preparation

To use the images with your method effectively, you will need to load the data into MATLAB, and convert the **images** to a double data type once imported.

Produce a 3-row and 4-column array of figures (**Hint: the subplot command will be useful here**), showing randomly-chosen examples and including the corresponding labels as the subfigure titles. Save this figure as a portable network graphic (png) file and use it in your report to show what your data looks like.

Split your dataset into training and testing subsets, each containing 50% of the data. Be careful to ensure that the labels still correspond to the correct images. You may find the `randperm` command useful for this. Your data and labels should retain the same row and column format, i.e. your data should have one image per row and 784 columns containing the intensity data, and your label should have one label per row.

Check your label distribution to ensure that you have representation in of each class in training and testing.

4. Model Training and Evaluation

4.1. Model Training with *K*-Nearest Neighbour

In this section, you will be required to classify the test data using your own implementation of *k*-nearest neighbour (see Week 3 Lab for reference). This should be carried out separately using two different distance metrics to provide one set of results per distance metric:

1. L^2-distance, also known as Euclidean distance.
2. Another distance metric of your choice. For example, you could use L^1, Bray-Curtis or Cosine Distance.

Be sure to save the prediction results and computation times for evaluation and comparison later.

4.2. Model Training with Existing Models

It is important to compare with existing models and we often do this by implementing the author's code. We will use some existing classification algorithms that are already implemented in MATLAB. You will need to train and compare **any two** classification algorithms implemented in MATLAB. It is sufficient for this coursework to use the models' default parameters but you are welcome to optimise these parameters to improve performance. Please note that, given the dataset size, the models make take several minutes to run.

Table 1 shows some examples of classification algorithms in MATLAB and their function names as implemented in MATLAB's Statistics and Machine Learning Toolbox™. You may find it useful to have a look at this MATLAB documentation page on Supervised Learning Workflow and Algorithms for some ideas on how to structure your solution and select your algorithms. You are encouraged to make use of the suggestions on this page but feel free to explore further.

Table 1: MATLAB Classification of Algorithms and their function Names

Algorithm	**Function**
K-Nearest Neighbour	fitcknn()
SVM for Multiclass	fitcecoc()
Decision Tree	fitctree()
Ensembles	fitcensemble()

To train a machine learning model, you need to use the syntax:

```
>> model = function(features,label)
>> predictions = predict(model,features)
```

For example, you might try:

```
>> knnmodel = fitcknn(training_features,training_label)
>> predictions = predict(knnmodel,testing_features)
```

4.3. Evaluation

It is often important to evaluate methods from multiple points of view. For this work, please evaluate the models in terms of their accuracy and time taken to train and test. These can then be discussed in your report to select the most suitable model for your problem.

Accuracy is measured by comparing the predictions from your models to the test labels in the dataset. It will be sufficient to calculate the accuracy as

$$accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ labels}$$

The time taken to train the models can be computed using the `tic` and `toc` commands.

5. Report

As part of this coursework, you are asked to write a 1-page report in the form of an Extended Abstract, which is a common way of submitting work to a conference or workshop. A template is provided in the file `SCC361-CW1-Report-Template.docx` and an example abstract accepted at BioMedEng23 is included in `Example-ExtendedAbstract-BioMedEng23.docx`.

The report must include the following sections:

1. **Introduction:** This should be a paragraph, describing the problem that you are solving and potential uses for it.
2. **Data and Preparation:** Describe the content of the dataset and the data preparation method. Please include the figure that you generated in Section 3.
3. **Methodology:** Describe the model training and evaluation methods used, including your reasons for choosing the models. You should briefly describe each method but you do not need to give a detailed technical description.
4. **Results:** This should include presentation of your results in a table, error analysis and observations
5. **Conclusion:** In this section, you should identify the model that you would recommend for use with your problem and justify your reasoning.

Most conference submissions will not permit you to change the margins, font or font size. Beyond the report title, please use the font Arial at size 10.

Ordinarily, in a paper submission, you would include the names and affiliations of the authors at the top, as in the template. Please do not do this; this submission should be anonymous.

6. Submission

Please submit the code and report on Moodle on or before the **deadline of 6pm on Friday, 10th November, 2023**. There will be separate submission points for the code and for the report. If for

some reason, you cannot upload your submission, please zip it up and e-mail it to b.williams6@lancaster.ac.uk and scc-teaching-office@lancaster.ac.uk as soon as possible. You do not need to submit the data.

Your submission should be anonymous and not include your name in the code or report.

You code should be submitted as a **single** MATLAB script (.m file), which is running without any errors, structured and includes documentation/ detailed comments.

Your 1-page report should be submitted as a pdf file, without your name.

7. Marking

Marks and feedback will be returned by 1st December 2022. There are a total of 20 marks for this coursework and the coursework mark constitutes 20% of your overall mark for this module.

- 10 marks are allocated to your code, including correct implementation, appropriate structure and suitable annotation/documentation of your code using comments.
- 10 marks are allocated to your report, with 2 marks per section mentioned above.