Assignment 1 - Deep Learning Fundamentals

Due: 11:59pm, 27/09/2024

Delivered by Manish Gupta
The University of Adelaide
manish.gupta@adelaide.edu.au

Abstract

The first assignment is to implement, describe, and test a specific algorithm called Perceptron (which can be interpreted as a dense layer neural network) for predicting diabetes (using the diabetes dataset provided). The submission will take the form of a short conference paper.

1. Task and dataset

The task is to use Perceptron to predict diabetes. The original data are from https://www.kaggle.com/uciml/pima-indians-diabetes-database with description, and the pre-processed data are available from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html. The pre-processed data are easier for you to handle — someone else have already turned them to a typically binary class classification dataset with the first column of the data being the class id $+1$ or -1 and the rest being the features. I recommend you to use the pre-processed data (diabetes, or diabetes scale) for your experiment and coding, and use the original site for more information to better understand the problem.

2. Submission

The submission takes the form of a conference paper, and your code.

2.1. Report/paper submission

You need to write a shortened version of a paper such as might be submitted to a conference, and specifically a paper such as might be submitted to CVPR[1], which is one of the best conferences in Computer Vision. The paper must be in the CVPR format, and submitted as a pdf document. By far the easiest way to achieve that is to use LaTeX. LaTeXis a very powerful document formatting package, it's free, and it is the only way to generate well formatted documents that contain maths. It's also the easiest way to generate well formatted documents in general. If you find too difficult to use LaTeX, you can use the word CVPR template too.

All the information about the CVPR paper format is available on their web site[1]. The paper must be all your own work, with no text copied from any other document. The paper you submit must be in the format specified for CVPR 2020, which is specified as part of the author instructions[1]. The easiest way to achieve this is to download the LaTeXtemplate and use that. You can use some other means if you really want to, but your paper needs to conform to the CVPR style specification. The only exception is that I don't mind if you use a4 paper rather than their preference for letter paper (it's a US conference).

As a guide, I would expect your report to be in the range 2-5 pages including figures, references etc., depending on the size and number of results included. There is no maximum or minimum page count requirement. Do not pad your report with material that does not contribute to the assessment criteria below—this makes your paper harder to read which may result in a lower mark!

Good paper exemplars? Go to the CVPR websites (not limited to 2020), and read the BEST PAPER AWARD papers and the oral papers in the past. You may not fully understand the the papers, but they should give you some idea what a paper should look like.

2.2. Assessment

The purpose of the paper is to demonstrate that you understand the problem, that you can solve it with your Perceptron, and that you can understand and analyse your results. This means that your submission should have sections which broadly cover the following

- An introduction, which describes the problem, and the method/algorithm, citing sources (papers or websites) where appropriate (10 points);

- A description of the method. This will typically require explaining some part of the algorithm in detail, and providing examples illustrating its effects and deficiencies. If you propose an improvement then you should describe how your method works, in enough detail that a reasonably skilled person would be able to implement it (30 points).

- Experimental Analysis. Describe the tests you have run, and your motivation for having run them. Report the results of the tests and the conclusions that you have drawn. The goal is not to show that your method outperforms all comparators, but rather that you understand what the method aims to achieve, and can devise, execute, and report upon a set of tests which demonstrate whether it does so. If you have improved upon the base method then you have an opportunity here to show that your improvement is well motivated, and possibly even that it works (30 points);

- Code. In order to be able to test the method you will need to implement it. You can implement it in Python (preferred) or Matlab. You will also need to submit your code in https://github.com, which leaves the time stamp. You don't have to make your github code publicly available, but it is important for you to give access to the tutors at a1844883@adelaide.edu.au and a1812913@adelaide.edu.au to check. The tutors will not be marking the quality of your code, only checking that it shows enough evidence that you wrote it yourself, submitted in time, and no obvious error(s). Please make a separate section (Code section) in the report with the link to your github code (20 points).

- Conclusion. Summarise what you have learned from the process, including ideas about what you could do in the future to improve the method you are reporting on (10 points).

References

[1] CVPR. Ieee computersociety conference on computer vision and pattern recognition. See http://cvpr2020.thecvf.com/submission/ main-conference/author-guidelines.