2223-CT5165 Principles of Machine Learning
- Announcements
- Module Overview
- Instructor
- Information
- Discussion Board
- Learning Materials
- Assessment
- Virtual Classroom
- My Grades
- Help
- Qwickly Attendance

Machine Learning Assignment 2 - due 8th November 22 23:59

Machine Learning Assignment 2 Description

Attached Files: wildfires.txt (8.354 KB)

Overview: The goal of this assignment is to **read**, **understand, implement** and **evaluate** the machine learning algorithm called Perceptron and Multi-Layer Perceptron.

You are encouraged to do this assignment in a pair. See below for the policy on 2-person assignments.

Note that this assignment is likely to involve effort over the full period that you have to do it.

You must write your own implementation from scratch **and** compare them against each other on same dataset and against a reference implementation of each from an open-source machine learning package.

Implementation Details:

- You may use any programming/scripting language you wish, but it must be entirely your own work, NOT based in any part on an implementation found elsewhere.
- You can use libraries for basic operations such as multiplying or inverting matrices, and for non-core operations such as plotting, but the implementation of your algorithm must be "from scratch" – you cannot just call an existing implementation in a library, nor modify/extend one.
- You may make any reasonable design decisions, and should document them (e.g. which activation function to use, or which loss function to use).
- It must be possible to input training data and testing data to your program, for example from a file or files, in a reasonable format of your choosing. Your program must be able to handle different numbers of attributes and training cases. For testing data, it must be able to output predicted and actual values to a file.

To Test Your Program:

- You will be provided with a file called **wildfires.txt**. The data is not splited into training, validation, and test sets. You need to do it yourself.
- Columns are separated by tabs and rows are separated by newlines. Each row describes one instance in the dataset. The attributes are in columns in the following order: fire, year, temp, humidity, rainfall, drought_code, buildup_index, day, month, wind_speed.
- The goal of your classifier is to predict **fire** (which may be one of two classes: yes or no) based on the other attributes.
- You (or your program) should randomly divide the file e.g. 2/3 for training, 1/3 for testing. The training data are the cases your machine learning algorithm will use. It will attempt to classify the testing cases, and should compute the accuracy of classifications.
- This procedure should be repeated with different random divisions 5 times, and the individual results and average accuracy reported.
- You should test your own implementation **and** a reference implementation of the same algorithm. The reference implementation can be selected from any ML package of your choosing. The reference implementation should be trained and evaluated in the same way as your own implementation (i.e. randomly divide data e.g. in to 2/3 for training, 1/3 for testing; repeat 5 times).

Report Contents (approximately 3-4 pages of main report plus code in an appendix):

- Names of the one or two team members, and your degree programme (e.g. 1MAI)
- A description of your algorithm and design decisions
- Your tests and results: you should compare the results achieved by your implementations against reference implementations of the same algorithm.
- Conclusions and observations
- If a 2-person assignment, write down exactly what each person was responsible for
- Your code, attached as an appendix. Suggestion: to generate the code appendix, print your code files to .pdf from Notepad++, and then merge the code .pdf with your main report .pdf
- Note: if your report is excessively long, I reserve the right to stop reading/grading after 5 pages of the main report.

Submission Instructions:

Note: you will lose marks if these submission instructions are not followed correctly. You must submit the report and your code separately:

1. Submit your report (including your code as an appendix) as a single PDF file to the Turnitin link below
2. Submit a separate .zip file with all of your source code files to the link below. This zip file should include all files needed to compile your solution, and nothing more: include source files, project files and external dependencies such as jars, but not any files that are recreated when your solution is compiled.

Policy on Two-Person Submissions:

- You can work in pairs, but you cannot collaborate with other people in the class beyond that.
- Your report documentation must include a section detailing what contributions each person made. It is not sufficient to say something along the lines of "we both worked on everything together": if you worked on some parts together, then please say so, but also identify the parts that you did separately (and there must be some).
- Each person must do some of the coding. There must be comments (e.g., on every function) so that it is clear for every line of code who was responsible for it. I will not accept any statements that two people worked on one line of code.
- I expect that people on a 2-person assignment each put in as much work as a person on a one-person assignment would.
- Across all parts of the assignment, we may award different marks to each person doing a 2-person assignment, if we feel that their contributions are not of equal value.
- If each person's contribution is not sufficiently clear, you are both likely to lose marks. Since the class is so large, I will not be able to get back to you to ask for clarification on who did what.
- To avoid any version conflicts, **only one of the two people must submit the assignment via Blackboard**. It is therefore important to have both contributors' names and IDs on the report and code.

Policy on Plagiarism:

As you are all postgraduate students, I will treat any plagiarism (from another student or other sources) very seriously. If **any** aspect of your work is plagiarised or is otherwise dishonest, you will receive 0 for the full assignment.

Marking Scheme:

Marks breakdown (out of 10):

- Implementation: 1 for code with significant flaws; **4** for fully correct code of both; or in between.
- Testing: 0 for no testing or couldn't get it to work; **3** for high quality and comprehensive tests,
- Particularly high quality implementation and detailed report documenting all aspects as specified above: **3**

A "particularly high quality" implementation may meet the criterion for one of many reasons, e.g.: very comprehensive in the algorithm options; nice user interface; an interesting choice of language or other implementation choice; or others.

"High quality and comprehensive tests" should include testing your implementation against a reference implementation. For example, you could present either a learning curve or ROC curve which could be used to compare the performance of your code against the chosen reference implementation. It's not necessary to write the code to generate a learning curve/ROC curve yourself - you could just generate the data and then plot it using a spreadsheet application such as Excel.

Assignment 2 - pdf upload here

Upload your reportin pdf file here.

assignment 2 code zip file upload here