CCE2502: Pattern Recognition and Machine Learning - Assignment II:

IMPORTANT Notes:

This assignment contains three tasks and the maximum score is 100.

The assignment contributes to 30% of the final study unit mark.

Submit the answers in a jupyter notebook (add explanations, discussions and diagrams in markdown cells within the jupyter notebook itself, except where indicated).

Cite any blogs, code repositories and/or generative AI tools (e.g chatgpt) used in completing this assignment. In the case of generative AI tools, explain how these tools were used, i.e submit a document with all the prompts and answers generated by the GenAI tools.

This work is to be attempted individually. It is essential that the work you eventually submit and present for your assignment consists only of your own work; use of copied material (including material obtained from Generative AI tools) will be treated as plagiarism. Discussion is only permitted on general issues, and it is absolutely forbidden to discuss specific details with anyone and/or share results.

Please sign the plagiarism form that can be found here:

https://www.um.edu.mt/ict/students/formsguidelines/

and submit the jupyter notebook (with all cells executed) zipped together with the signed plagiarism form.

Aim of assignment

The main aim of this assignment is to get an understanding of (a) the gradient descent method when used to train a logistic regression model, (b) the usefulness of higher order terms of the input features in case of non-linear separable datasets, and (c) the automation of higher order terms discovery using a network of non-linear regressors. The experiments will be carried out in Python within a Jupyter Notebook. The **sci-kit learn** package should ONLY be used for implementing the muti-layer perceptron (MLP) model in task 3.

In addition to this document you should find two datasets in csv file format and a jupyter notebook that contains helper functions. You can use this juypter notebook as a basis for completing the tasks.

Task 1:

This task is on adding the gradient descent update equations to the lr_train_model() function provided in the attached jupyter notebook and gaining experience in using the lr_train_model(), the associated functions and the helper functions, namely data_scatter_plot() and display_results().

Note that the lr_train_model() function automatically splits the dataset into train and validation and the display_results() plots loss against epochs and computes accuracy. Take some time to understand the lr_train_model() function.

a) Use the csv python module to load the data in the binary_classification_basic.csv file. The first two columns are the input features (X1, X2) and the third column is the output variable (y). **[2 marks]**

b) Using the function provided, plot the scatter graph. **[2 marks]**

c) The regularized categorical cross entropy loss function is given below. Derive the partial derivatives of the loss function with-respect-to the model parameters, w. (Note: include the result in the juypter notebook using mathjax and scan and attach all workings in the zip file).

$$L(\boldsymbol{w}) = -\frac{1}{2m}\left[\sum_{i=1}^{m}\{t^i \log y(\boldsymbol{x}^i, \boldsymbol{w}) + (1 - t^i)\log(1 - y(\boldsymbol{x}^i, \boldsymbol{w}))\}\right] + \phi \sum_{j=1}^{N} w_j^2$$

Where *m* is the number of examples in the dataset, *i* is an index to the i^{th} example, *t* is the target variable, $\boldsymbol{x}$ is the input vector and $\boldsymbol{w}$ is the weight vector. **[10 marks]**

d) Complete the lr_train_model() function by adding the gradient descent update rules in the space indicated. Note that the bias term should not be regularized. **[10 marks]**

e) Using the lr_train_model() function train a model on the binary_classification_basic dataset. Remember to make use of the display_results() function to get insight and debug the model. **[10 marks]**

f) Superimpose the decision boundary learnt by the model onto the dataset scatter plot. **[5 marks]**

Task 2:

This task is on adapting the logistic regression model to a non-linearly separable dataset.

a) Use the csv python module to load the data in the binary_classification_advanced.csv file. The first two columns are the input features (X1, X2) and the third column is the output variable (y). **[2 marks]**

b) Using the function provided plot the scatter graph. **[2 marks]**

c) Using the lr_train_model() function, train a model on the binary_classification_advanced dataset and explain any poor result. (Note: to get some insight you can also superimpose the decision boundary learnt by the model onto the dataset scatter plot). **[5 marks]**

d) Manually, find higher order terms of the input features (X1, X2) that improve the classification accuracy. Explain your answer and your reasoning. (Note: consider using scatter plots in your explanation) **[20 marks]**

Task 3:

In this task you will be implementing a neural network (multi-layer perceptron) to solve the binary_classification_advanced dataset.

a) In no more than 100 words, explain how a neural network can learn non-linear separable dataset and obtains an acceptable prediction accuracy. **[7 marks]**

b) Use the sklearn.neural_network.MLPClassifier() module (see jupyter notebook for link to API) to learn a non-linear model for the binary_classification_advanced dataset. (Note: you are expected to find an efficient structure that models the dataset) **[15 marks]**

c) Explain how you would check whether the neural network is performing as it should, without computing the accuracy or any other quantitative evaluation metric. **[10 marks]**