CSCI316 – Big Data Mining Techniques and Implementation
Laboratory 8 (Assessed)
Autumn 2023

6 Marks
Deadline: Refer to the submission link of this assignment on Moodle

<u>One task</u> is included in this laboratory. The specification of each task starts in a separate page.

You must implement and run all your Python code in Jupyter Notebook. ***The deliverables include one Jupyter Notebook source file (with .ipybn extension) and one PDF document for each task.***

To generate a PDF file for a notebook source file, you can either (i) use the Web browser's PDF printing function, or (ii) click "File" on top of the notebook, choose "Download as" and then "PDF via LaTex".

The submitted source file(s) and PDF document(s) must show that all of your code has been executed successfully. Otherwise, they will not be assessed.

This is an <u>individual assessment</u>. Plagiarism of any part of this assessment will result in having 0 mark for this assessment and for all students involved.

The correctness of your implementation and the clearness of your explanations will be assessed.

Task Specification

(6 marks)
Data set: MAGIC Gamma Telescope Dataset
(Source: https://archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope)

Data set information
The data are Monte-Carlo generated to simulate registration of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique. The dataset contains 19,020 records. Attribute information:

1. fLength: continuous # major axis of ellipse [mm]
2. fWidth: continuous # minor axis of ellipse [mm]
3. fSize: continuous # 10-log of sum of content of all pixels [in #phot]
4. fConc: continuous # ratio of sum of two highest pixels over fSize [ratio]
5. fConc1: continuous # ratio of highest pixel over fSize [ratio]
6. fAsym: continuous # distance from highest pixel to center, projected onto major axis [mm]
7. fM3Long: continuous # 3rd root of third moment along major axis [mm]
8. fM3Trans: continuous # 3rd root of third moment along minor axis [mm]
9. fAlpha: continuous # angle of major axis with vector to origin [deg]
10. fDist: continuous # distance from origin to center of ellipse [mm]
11. class: g,h # gamma (signal), hadron (background)

g = gamma (signal): 12332
h = hadron (background): 6688

Objective
Develop an Artificial Neural Network (ANN) in TensorFlow/Keras to predict the class. You must use TensorFlow to train the ANN.

Requirements
(1) Randomly separate the data into two subsets: ~70% for training and ~30% for test.
(2) The training process includes a hyperparameter fine-tunning step. Define a grid including at least three hyperparameters: (a) the number of hidden layers, (b) the number of neurons in each layer, and (c) the regularization parameters for L1 and L2. Each hyperparameter has at least two candidate values. All other parameters (e.g., activation functions and learning rates) are up to you. (Note. You can use Scikit-Learn for hyperparameter tuning, i.e., by using a Keras wrapper.)
(3) Report the learning curve and test accuracy.

Deliverables
- A Jupiter Notebook source file named `your_name_lab8.ipybn` which contains your implementation source code in Python
- A PDF document named `your_name_lab8.pdf` which is generated from your Jupiter Notebook source file, and which includes clear and accurate explanation of your implementation and results.

---END---