# COS711 Assignment 2

## Sign-Based Adaptive Learning Rate for Neural Network Galaxy Classification

Due date: 3 October 2025, at 23h30

## 1 General instructions

For this assignment, you have to submit an archive (zip file) containing (1) a Jupyter notebook with all of your code, and (2) a video presentation, wherein you describe what you have done, present and discuss your findings. Guidelines for preparing the video presentation are provided in this specification document.

## 2 Galaxy Classification

The night sky is host to numerous celestial bodies. The Sloan Digital Sky Survey (SDSS) has searched about one-third of the sky and found around one billion objects, almost three million of which are galaxies. An automated system that can check photometric data and make predictions can save time for astronomers and researchers in their efforts to classify galaxies.

For this assignment, you will work with a dataset that contains photometric features extracted from images of the night sky. Your task is to optimise and train a neural network (NN) to perform galaxy classification. As part of this task, you will need to preprocess the data, optimise NN's hyperparameters, and investigate the utility of using the gradient sign for the purpose of learning rate adaptation.

### 2.1 Data set

Download the data set from ClickUP. The data set is also available from Kaggle. The data is provided in a `csv` format, where each row represents a data set entry, and each column represents a data attribute. There are 100,000 rows in total. Each galaxy is described via over 30 numeric features.

All data set entries belong to the same class (GALAXY), and your task is rather to predict the *subclass* for each data point. Subclass is listed per entry in the 'subclass' column, and the two subclasses you will be differentiating between are STARFORMING versus STARBURST galaxies. A star-forming galaxy is a celestial object undergoing an active process of converting gas and dust into new stars. A starburst galaxy is one undergoing an exceptionally high rate of star formation, as compared to the long-term average rate of star formation in the galaxy. Your task is to construct a NN to perform galaxy *classification* into one of these two types.

### 2.1.1 Data preparation

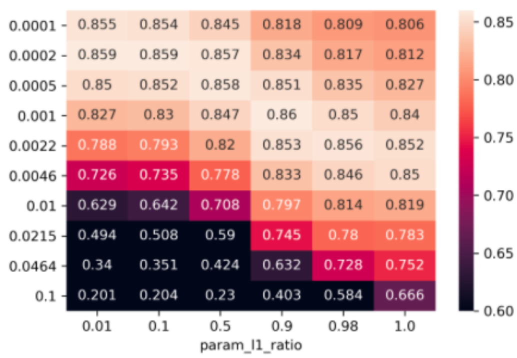
The given data contains numeric attributes that lie in various ranges. Analyse the data set, and pre-process it in a way that will make it possible for a NN to effectively discover the hidden relationships between inputs and outputs. Give extra thought to the following data set properties:

1. Some of the columns are not informative. Identify and remove them.
2. The dataset contains outliers. Develop a sensible strategy to both identify and handle the outliers.
3. The classes are not present in equal proportion – STARFORMING constitutes about 75% of the data, while STARBURST constitutes the remaining 25%. How should this imbalance be handled during training, testing and evaluation?

## 2.2 Hyperparameter Optimisation

As discussed in class, the performance of your NN model greatly depends on various hyperparameters, such as the NN architecture, activation functions, error (loss) function, training algorithm, etc. You will have to choose the hyperparameter values for your NN model. Your video presentation **must** contain a section justifying all hyperparameter choices. Two justifications are acceptable: (1) theoretical insight; (2) empirical evidence. I.e., if you cannot decide on a value for a certain hyperparameter analytically, you have to run some experiments to see which value performs better than others.

You must empirically compare at least two different hyperparameters of your choice, excluding epoch count. While you are welcome to thoroughly optimise more that just two hyperparameters, it is required of you to perform a **grid search** over any two hyperparameters of your choice. Grid search implies selecting a set of values for each hyperparameter, and evaluating the model for every combination of values between the two sets. Visualise the results of the grid search using a heatmap such as the one shown below, where x-axis represents the values of hyperparameter A, y-axis represents the values of hyperparameter B, and each cell is colourised according to the NN performance for a combination of (A,B):




NB: to compare the performance of any two hyperparameter values, you must obtain average performance for each hyperparameter across a few independent runs, as well as standard deviation. K-fold cross-validation is an excellent technique to perform hyperparameter optimisation. Discuss whether there is statistical evidence that one hyperparameter value is more performant than another value. Consider using appropriate hypothesis testing to make an informed conclusion. Hint: if standard deviations between two hyperparameter values do not overlap, you can safely conclude that the difference in performance is significant.

Since we are working on a classification task, remember to compare your models in terms of accuracy, precision and recall in addition to the NN loss value. Hint: figure out what metric is most appropriate when there is class imbalance. Show confusion matrices in your video presentation.

### 2.3 Sign-Based Adaptive Learning Rate

For this part of the assignment, you will evaluate the effectiveness of scaling the learning rate based of gradient sign consistency versus using a static learning rate.

As discussed in class, having one learning rate value across all weights in the NN is a suboptimal strategy, since the loss landscape is known to be ill-conditioned, i.e., significantly steeper in some dimensions. For this assignment, you will develop a novel adaptive learning rate algorithm by considering the change in the sign of the gradient for each weight, and adjusting the learning rate accordingly. Specifically, you are required to implement the following strategy:

1. Start all weights off with identical learning rates.
2. For each weight, prior to applying the gradient update, check whether the gradient **sign** (positive/negative) has changed compared to the previous weight update.
3. If the sign has not changed, **increase** the learning rate. Consistent sign is indicative of a stable trajectory which can be traversed faster.
4. If the sign has changed, **decrease** the learning rate. If the sign is alternating, then oscillatory behaviour is taking place, therefore slow-and-steady is likely to yield better results.

Note that the algorithm outlined above does not prescribe **how** the learning rate must increase or decrease – only **when**. It is up to you to propose a viable approach. Feel free to experiment with a few variants.

Collect sufficient data (i.e. results from multiple independent runs) for both the adaptive learning rate schedule and standard static learning rate. How do the algorithms compare? Discuss and interpret all your results thoroughly. Remember that simply pasting a table with numbers in it, or a graph with no explanation, will **not yield any marks**. Visualise the experimental data where appropriate: it is much easier to analyse your results when you can see them plotted next to one another in different colours. If you see that one approach is doing better than another, provide a hypothesis for why it is the case. Running the experiments is only half of the research process, the other and more important half is interpretation. Aim to derive as many insights from your results as you can.

## 3 Notes

- Implementation
  - You are required to use Python programming language for this assignment, and submit your code in Jupyter notebook format.
  - You may use a machine learning/neural network library/framework.
- Video Report
  - You must report on all data preparation steps taken.
  - You must report on all NN hyperparameters used, and substantiate your choices. Show the heatmaps generated.
  - Training and testing errors have to be reported. Remember to report means with the corresponding standard deviations, and report how many independent runs have been performed.
  - To compare individual training algorithms, plot their training and testing loss values over multiple epochs.

# 4 Marking and general guidelines

The primary deliverable that will guide the assessment is the video presentation. Other submitted deliverables should complement the presentation and serve as evidence of work authenticity.

For this assignment, you have to submit a *zip* file containing both a **video presentation** where you discuss your findings, and a Jupyter notebook containing all of your code. Your presentation may not exceed **50MB** in size, and may not be longer than **10 minutes**. You must include a video of yourself talking (head and shoulders) in the upper right corner of the presentation.

To aid with marking consistency, a powerpoint template for the presentation is available on ClickUP. You may adapt it as you see fit. Your presentation must cover the following aspects (each aspect can go over multiple slides if necessary, as long as the overall video is not longer than 10 minutes):

1. **Experimental Setup**

   Mention all tools and libraries used by you to accomplish the task. Specify the hardware that you have used to run the experiments. Explain how statistics were gathered (i.e., how many independent runs you have managed to execute per experiment).

2. **Dataset Preparation**

   Explain how you have gone about preparing the dataset. Talk about scaling used, outlier / missing values treatment, columns retained, training / testing split, class imbalance handling, etc.

3. **Hyperparameter Tuning**

   This is the section where you report your results for hyperparameter tuning. Explain how you have picked hyperparameter values, and which were tuned experimentally. Show the heatmaps for the grid searches that you performed. Conclude the discussion with a table listing final chosen hyperparameter values.

4. **Sign-Based Adaptive Learning Rate**

   Discuss how you have implemented the sign-based adaptive learning rate. Show experimental results, and discuss how adaptive learning rate performed compared to a static counterpart. Reason about the results that you observe.

5. **Conclusions**

   Summarise the main take-aways from your experiments, highlight the biggest roadblocks, reflect on what you would have done differently if you had to do the same task again.

6. **Bibliography**

   Provide a list of academic and other resources that you have used. No need to talk through this slide, just include it for completeness.

Please **remember** to include the Jupyter notebook containing all of your code in your zip submission! Presentations not accompanied by a Jupyter notebook will have their total mark halved.

## 4.1 Marking

The following general breakdown will be used during the assessment of this assignment:

| **Category** | **Mark Allocation** |
|---|---|
| Format | 5 marks |
| Experimental Setup | 10 marks |
| Data Preparation | 20 marks |
| Hyperparameter Optimisation | 30 marks |
| Sign-Based Adaptive Learning Rate | 30 marks |
| Conclusions | 5 marks |
| Penalty for not including a Jupyter notebook | -50% of total marks |
| **TOTAL** | 100 marks |

Upload the ZIP file to the appropriate assignment slot on ClickUp. Multiple uploads are allowed, but only the last one will be marked. The deadline is **3 October 2025, at 23h30**.