COMP47790 Assignment 2

Deadline: Sunday, Nov 30, 2025. Submissions after this date will incur late submission penalties.

Instructions

Answer all questions. Submit your assignment as one Jupyter notebook file (not a DOC/ DOCX/ODT/ZIP/PDF file) via the module Brightspace page. Please ensure that together with the code, **you provide sufficient details of your solutions in markdown cells**. In particular, please explain what your decision variable means, what is your objective function, any analysis you have done etc.

This assessment should be completed individually. Any evidence of plagiarism will be reported to the CS plagiarism committee and it can result in a Fail grade. In particular, the use of ChatGPT, Gemini, Claude, Chegg or other such tools for this assignment is strictly prohibited.

Question 1 (30 points)

You are tasked with determining the optimal relationship between a single predictor variable, house_size, and a target variable, price, using a linear model of the form price $= a *$ house_size $+ b$. Recognising that the traditional Least Squares method is vulnerable to data anomalies, you opt for a more robust regression technique: Least Absolute Deviations regression. Your goal is to find the parameters a and b that minimise the sum of the absolute errors across all N data points, specifically minimising the metric:

House Size (sq. meter)	Price (In '000 Euros)
200	300
280	320
120	260
205	395
595	495
685	540
625	580
810	610
975	700
895	545

$\sum_{i=1}^{N} |\text{price}_i - a * \text{house\_size}_i - b|$. Formulate this Least Absolute Deviations problem as a standard Linear Programming problem. Solve it using Julia/JuMP for the above data. Carefully explain all the decision variables, objective function and the constraints involved. Explain how you converted the apparent non-linear objective function into a linear one.

Question 2 (35 points)

Your task is to minimise the following multivariate function using the convex optimisation techniques that you learnt in the module.

$$f(x, y) = 0.1x^2 + 10y^2$$

Use the starting point (-2.5, 2) and compare the performance of Gradient Descent, Momentum, Nesterov Momentum and Adam optimiser for this task. Use 50 steps with a step factor of 0.1 for all of these techniques. For techniques that require a decay term, use 0.9 and for Adam, you can use epsilon to be 10^{-8}. Plot the contour lines and show the iterative progress made by the different optimisation techniques (just as you did in your tutorial). How do these techniques compare empirically? Do you observe any benefit of Adam over other techniques for these functions? Analyse and explain the observed differences in the trajectories, drawing upon the theoretical mechanisms of each technique.

Now change the function to $f(x, y) = x^2 + y^2$. Run the different optimisation techniques with the same parameters for this function and plot the contour lines showing the iterative progress of these techniques. Explain why, for this specific function, Gradient Descent performs as effectively as the ADAM optimiser. Justify your answer by referencing the characteristics of the loss function and the underlying mechanisms of Gradient Descent versus ADAM.

Question 3 (35 points)

Consider a neural network for learning the XOR function with mean square error (MSE) loss. You want to learn a feed-forward neural network with sigmoid activation function that takes in 2 bits of input, has one hidden layer with 5 neurons and outputs 1 bit of output. Ignore the bias terms (treat their values as zero) and just focus on learning the weights for all the links in the neural network. Compare the performance of Gradient descent, momentum, Adagrad and Adam optimiser with a fixed step size (learning rate) of 0.1 and 15000 steps. Plot the loss curves for different optimisers. You should determine the best values of the remaining parameters empirically.

Please fill in the details in the code structure provided in the separate Jupyter notebook.