Due on Thursday, September 5th at 11:59pm
Answer all problems following the instructions. Please submit a PDF report with pictures of the output for each step followed by relevant code snapshots/screenshots that executed the function.

1 Assignment Instructions

Assignments in EEE 598 are due by the date assigned. You must use Latex to typeset your reports. **We will not accept Jupyter notebooks as reports.** Your report should have a section for each problem, and present your approach to solving the problem, any relevant code and figures to explain your method, and your results. Think about this like a job, you need to explain to your supervisor/boss what you did without dumping a whole bunch of code, answering questions in comments (big NO), and using labeled figures with captions. All figures must have axes labeled and an appropriate title.

2 Problems

Problem 1. *Perceptron*

1. Let D be the following dataset consisting of points $x_i = (x_1; x_2) \in \mathbb{R}^2$ with associated label $y_i = \pm 1$, written in form $[(x_1; x_2); y]$. Let

   $D = ([(1, 1), +1], [(2, -2), -1], [(-1, -1.5), -1], [(-2, -1), -1], [(-2, 1), +1], [(1.5, -0.5), +1])$

   Perform all steps of the perceptron algorithm by hand, going through each point in order (left to right, do not randomize the order). Show all calculations, and plot the hyperplane and all data points (with their labels) for every iteration. Note: you will have to come up with a way to take care of the bias term. Report the final converged solution, and plot it.

2. Now write a code to perform the perceptron algorithm given any set of data points. Show that you can reproduce the result of the dataset given in the step above.

3. Add the minimum number of additional points to make this particular dataset not linearly separable. Show that your code keeps looping and doesn't attain convergence (it suffices to measure the change in w from iteration to iteration, and plot this to show that it never converges to zero).

Problem 2. *Backpropagation*
Complete the attached Python notebook to implement backpropagation for a simple artificial neural network. Please refer to Section 7.4 of the book "Understanding Deep Learning" by Simon Prince (`https://github.com/udlbook/udlbook/releases/download/4.0.4/UnderstandingDeepLearning_08_05_24_C.pdf`) for guidance on this problem.

For your report, show the relevant TODO lines and what you wrote as code, and **also write mathematically what the expression corresponds to, i.e. "I implemented the change in activations as a function of weights, $\partial f / \partial \Omega$, etc.** Also show the successful output that confirms you implemented backpropagation correctly.

Problem 3. *MLP versus KANs*

In this problem, we are going to train a multi-layer perceptron (or ANN) on some data involving house prices. It will be your job to define the machine learning task you want to explore within this dataset. Then you will also implement a Kolmogorov-Arnold Network (KAN) and show how the KAN achieves superior performance (in whatever aspect you choose) on a machine learning task of your own choosing in this data.

1. Use the house dataset (`https://www.kaggle.com/datasets/harlfoxem/housesalesprediction`), you are going to formulate a **supervised learning** task that leverages a MLP. This task can be classification or regression or something else of your own choosing. Separate the dataset following a training, testing and validation split of 70-20-10 respectively. Set up an MLP using whichever values of data you deem appropriate and report the results. Explain your choice of parameters for the MLP. Presentation is very important here, you must convince us that the learning task is interesting and relevant, why did you pick the task you did? Then show that the MLP successfully completed the task, and analyze its performance.

2. Using what you have learned about KANs, construct a KAN using parameters you deem appropriate and report the results. Explain your choice of parameters. You may use the following repository to aid you (`https://github.com/KindXiaoming/pykan`).

3. Demonstrate a scenario in which the KAN outperforms the MLP on the same train-test-val split of data using selected network parameters of your choice. Give reasoning as to why you think the KAN performs better (use evidence from the paper to help you with that claim).

3 Grading and Report

The assignment report is very important for the grading of this assignment. We will return reports that we are not satisfied with the presentation, this is to help you practice improving your written communication skills. Grading breakdown is as follows:

- Problem 1: 30 points
- Problem 2: 20 points
- Problem 3: 30 points
- Presentation (are answers clearly defined and easy to find, are code snippets and figures utilized well for the report, overall readability): 20 points
- **Total:** 100 points

Please only upload the PDF of the final report (we do not want any code). Also acknowledge any web sources, classmate help that you used in this assignment in an acknowledgements section.