SEP769 Cyber Physical Systems
Assignment No. 1

Date: July 7th, 2025 Due Date: July 17th, 2025

Instructions (Please read carefully):

This assignment covers core concepts while giving hands-on experience with fundamental implementations. The difficulty progresses from single perceptrons to full backpropagation networks. The assignment contains two theory-based question and 4 programming/implementation problems covering these topics: deep learning fundamentals, ANNs, perceptrons, backpropagation, and NumPy/Matplotlib implementation.

- Attempt all questions and submit your solutions in assignment drop box on Avenue before the submission deadline.
- Submit a Jupyter notebook with solutions.
- Theory answers should be in markdown cells with proper formatting.
- Code must be well-commented.
- In the coding problems, you can use the python libraries Numpy and Matplotlib. All coding can be done using standard python constructs (for/while loops, if-else blocks, vectorization).
- All plots should have labels and legends (if required).
- Mathematical answers/notations should use LaTeX in markdown.
- You are encouraged to discuss the assignments, but duplicates / copies of the solutions will lead to zero credit. Make sure that you submit YOUR OWN SOLUTION. If two assignments are found to be copies of each other, they BOTH receive 0.
- If you have a question about this assignment, you are encouraged to post your question on the Avenue discussion board, or contact the instructor or the TA.

Evaluation Criteria and Rubrics:

Theory Questions (40 pts)

Criteria	Excellent (90-100%)	Good (70-89%)	Fair (50-69%)	Poor (<50%)
Accuracy	All concepts correct with precise mathematical notation	Minor inaccuracies with mostly correct notation	Several conceptual errors	Major misunderstandings
Depth	Covers all required aspects with insightful additions	Meets all requirements	Missing 1-2 key aspects	Superficial treatment
Clarity	Exceptionally well-organized with logical flow	Clear organization	Somewhat disorganized	Difficult to follow

Programming Problems (60 pts)

Criteria	Excellent (9-10 pts)	Good (7-8 pts)	Fair (5-6 pts)	Poor (<5 pts)
Correctness	Implements all requirements flawlessly	Minor bugs that don't affect core functionality	Several functional issues	Major implementation errors
Efficiency	Optimal vectorized NumPy operations	Mostly vectorized with few loops	Some inefficient implementations	Poor computational practices
Visualization	Professional plots with labels/legends	Good visuals missing 1-2 elements	Basic plots lacking details	Missing or incorrect visuals
Documentation	Excellent comments and function docstrings	Adequate comments	Minimal commenting	No documentation
Edge Cases	Handles all edge cases gracefully	Handles most common cases	Fails on some edge cases	No edge case consideration

Bonus Points (Up to +5)

- Exceptional code organization

- Additional insightful visualizations
- Creative extensions beyond requirements
- Superior theoretical explanations

Penalties (-1 to -5)

- Late submission
- Code plagiarism
- Missing essential components
- Poor formatting

Theory based question

The questions in this part are based on this recent research article entitled "A comprehensive survey of loss functions and metrics in deep learning by Juan Terven et. al., Artificial Intelligence Review (2025) 58:195, https://doi.org/10.1007/s10462-025-11198-7 has been uploaded with this assignment. Read the relevant sections of this paper and answer the following questions:

Q. No. 1 (20 points). For regression problems, explain / describe in your own words:

1. Mean Squared Error (MSE) Loss
2. Root Mean Squared Error (RMSE)
3. Mean Absolute Error (MAE) Loss
4. Mean Absolute Percentage Error (MAPE)
5. Huber Loss
6. Log-Cosh Loss
7. Quantile Loss
8. Poisson Loss
9. Coefficient of Determination R^2
10. Adjusted R^2

Q. No. 2 (20 points). In the context of classification problem solving in deep learning, explain / describe in your own words the following:

1. Binary Cross-Entropy (BCE) Loss
2. Categorical Cross-Entropy (CCE) Loss
3. Sparse Categorical Cross-Entrpoy (Sparce CCE) Loss
4. Weighted Cross-Entropy (WCE) Loss
5. Polyloss
6. Hinge Loss
7. Confusion Matrix (binary and multi-class)
8. Accuracy, Precision and Recall
9. F1-Score and Precision-Recall Curve
10. AUC-ROC (binary and multi-class)

Programming / Implementation Based Questions

Note: All implementations should use NumPy and visualization of results with Matplotlib.

Q. No. 3 **(15 points)** Implement a single-layer perceptron and a single layer Adaline that learn the AND, OR, and XOR logic gates.

a) Train separate perceptrons for each gate
b) Plot the decision boundary for AND and OR cases
c) Explain why XOR fails (include in comments)

Q. No. 4 **(20 points)** Implement full backpropagation for a 2-layer network. Run and compare the results of your full backpropagation code for (i) Linear, (ii) Binary sigmoid, (iii) Bipolar Sigmoid, (iv) ReLU, (v) Leaky ReLU activation functions for the hidden layer and Linear activation function for the output layer with the following:

a) Use synthetic data (e.g., sklearn.make_moons) for a binary classifcation problem
b) Include learning rate and epochs as parameters
c) Visualize the loss and accuracy curves, with the final decision boundary

Q. No. 5 **(15 points)** Use the backpropagation program developed in Q. No. 4 above, with a 3-layer neural network having one input neuron, and one output neuron and variable number of hidden neurons to model the following functon (for the input values such that $-1.0 \leq x \leq 2.0$):

$$f(x) = \frac{0.8}{((x-0.1)^2 + 0.01)} + \frac{1}{((x-0.5)^2 + 0.01)} + \frac{0.6}{((x-0.8)^2 + 0.01)}$$

Generate 100 equidistant points in the given input range. Use 80 of these points for training and 20 as testing points. Compare results by plotting the training and testing errors by changing the number of neurons from 1 to 20 hidden neurons. Discuss your results.

Q. No. 6 **(10 points)** Visualize learned features:

a) Train a backpropagation network on MNIST (using tensorflow)
b) Plot the weights of first hidden layer as 28×28 images
c) Briefly interpret what patterns the network learned