EE 105 Fall 2025

Homework 5 – upload to grade scope

(Due November 25, before class, late submission will incur 20% points/day penalty)

Instructions: Perform the following tasks based on the circuits and concepts discussed in class. Be sure to show all work where applicable.

---

1) Perceptron Model Calculation (Hand Calculation)

- Given a simple perceptron with weights **w1 = 2**, **w2 = -3**, **w3 = 1.5** and input values **x1 = 1**, **x2 = 0**, **x3 = 1**, calculate the output assuming the threshold is **0**. Use the perceptron rule:
    - **Output** = 1 if the weighted sum is greater than the threshold, otherwise **0**.

2) Activation Functions (Hand Calculation)

- For a neuron with input **x = 1.0** and weight **w = 2.5**, calculate the output using:
    - **Sigmoid Activation Function**:
    - **ReLU Activation Function**:
    - **Tanh Activation Function**:

3) Design a Neural Network Topology (Conceptual)

- Design a neural network to classify whether a person should attend a music festival. The inputs are:
    - Weather forecast (good or bad)
    - Availability of tickets (yes or no)
    - Friends attending (yes or no)
- Specify the number of inputs, weights, activation function, and output for this simple model.

4) Smallest Neural Network for MNIST Classification (Python Implementation)

- Your task is to design the smallest possible neural network with a single hidden layer that can classify the MNIST handwritten digits with at least **85% accuracy** for the MNIST TEST dataset.
- Calculate the number of floating point operations (meaning number of multiplications and additions) your neural networks requires to generate a single output. We have gone over this in class, so use the same approach we did there.
- **Requirements**:
    - Use python based on the code developed in the labs to implement your model.
    - The network should contain a **single hidden layer** and output predictions for **10 classes** (digits 0-9).
    - The network should be trained on the **MNIST** dataset.
    - Report the architecture you used, including the size of the hidden layer, what activation functions, and the achieved accuracy.