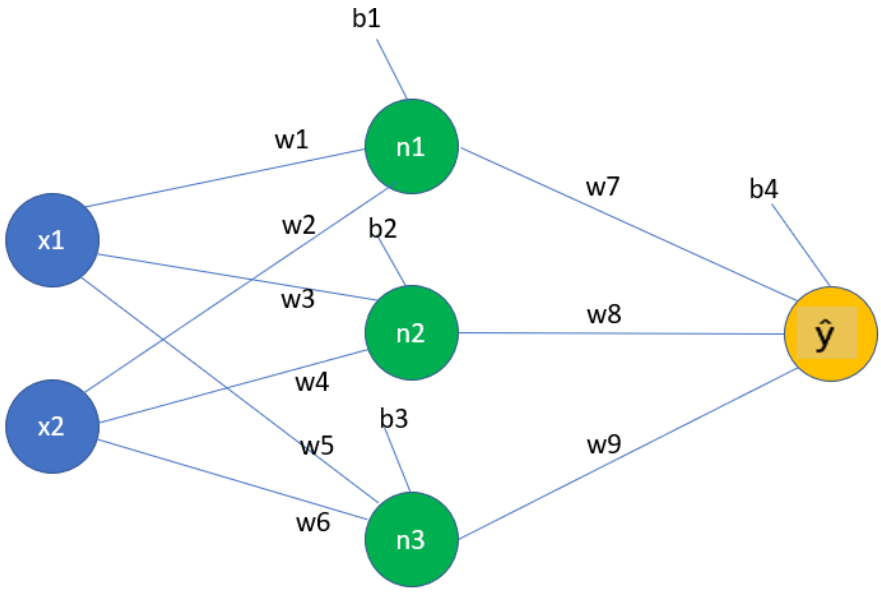
DATA 255 | Spring 2025 | HW – 1
Deadline – 11.59 PM – 2/23/2025
20 Points

Problem 1 (6 pts): The figure below shows a 2-layer, feed-forward neural network with one input layer with two features, one hidden-layer with three neurons and one output layer with one node. x1 and x2 are the two inputs. For the following questions, assume the learning rate is α = 0.1; activation function $ReLU$ for hidden units and sigmoid for output node; loss function, MSE = $\frac{1}{2}(y - \hat{y})^2$; ground truth $y = 1$; For instance, the output of $n1$ equal $ReLU(w1 * x1 + w2 * x2 + b1)$. Compute one step of the backpropagation.

a. Assume x1 = 0, x2 = 1; all weights and biases equal 1. Compute the updated weights for w7, w8, and w9. Compare their updates and do you see any symmetry breaking issue. Show all steps in your calculations. (**3 pts**)

b. Assume x1 = 1, x2 = 1; w1 = w3 = w5 = - 0.5; w2 = w4 = w6 = 0.25; w7= 1, w8 = -1, w9 =0 and biases weights equal 0.1. Compute the updated weights: if your student id's last digit is even then update weights: w3, w8; if odd update: w7, w1. You must show all steps in your calculations. (**3 pts**)



CODING!

Problem 2-4: We will develop Artificial Neural Networks using MNIST digit data, you can directly download the data using https://keras.io/api/datasets/mnist/. The dataset contains 10 classes where each of the image sizes is (28×28). Train for minimum number of $epoch = 100$; You should split the training data into training and validation sets, and for training the model use these datasets. Test data should be kept separated and used only for evaluation purpose. Your ANN must contain minimum of 2 hidden layers. Apply early stopping criteria based on validation loss with patience 3 and with restoring best weights = true. You may use any regularizes to avoid overfitting.

Problem 2 (8 pts): You should select last two digits of your student ID – meaning that if your student id is 006000104, then you should select 0 and 4 for developing the binary classification model. If both last digit is identical, then select first and last digit. For this task, you must prepare (filter) your data at first to convert the multiclass classification into a binary classification system.

A. Build an ANN for binary classification. Evaluate your model on the test data. Construct a confusion matrix. Present learning curve (showing training loss and validation loss against number of epochs) and include some examples of your prediction. **(3pts)**
B. Build ANNs for binary classification using combinations of **weight initializers (Normal, He, and Xavier) and activation functions (ReLU, Sigmoid, and tanh)**. You may use early stopping callback function. Construct confusion matrices and show learning curves for each combination. Create a table, showing combinations, and accuracy. Now, from your experimental results – write a comparative analysis on – impact of different combinations of initializers and activation functions in terms of performance and learning curves. **(5 pts)**

Problem 3 (2 pts): Build an ANN for multi-class classification considering all the classes (10 classes) in the MNIST digit dataset. Finally, present classification report, including class-wise precision, recall, f1-score, and discuss your result.

Problem 4 (4 pts): Build ANNs for multi-class classification considering all the classes (10 classes) in the MNIST digit dataset with **combinations of batch sizes and learning rates**. Consider batch sizes: 4, 16, 32, and 64; and learning rate 0.01, 0.001, 0.0001, and 0.00001. Finally, create a plot of test accuracy vs. ratio of batch size to learning rate**,** and discuss your findings.

You are required to follow:

1. Submit **one** MS/PDF/Scanned document:

- Include all the steps of your calculations.
- Include the summary of the model.
- Attach screenshots of your code.
- Attach screenshots – showing first few epochs of model training.
- Attach screenshots of the important code outputs such as confusion matrices, learning curves, and classification reports.

2. Source code:

a. Python (Jupyter Notebook)

b. Ensure it is well-organized with comments and proper indentation.

- **Failure to submit the source code will result in a deduction of full/partial points.**
- Format your filenames as follows: "your_last_name_HW1.pdf" for the document and "your_last_name_HW1_source_code.ipynb" for the source code.
- Before submitting the source code, please double-check that it runs without any errors.
- Must submit the files separately.
- Do not compress into a zip file.
- HW submitted more than 24 hours late will not be accepted for credit.