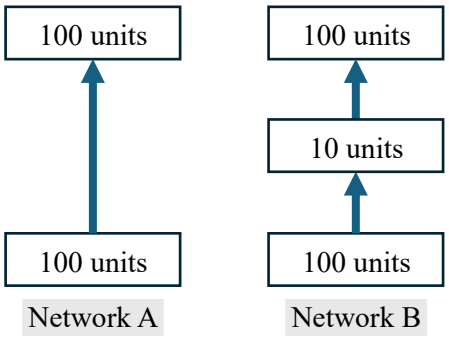
EE708: Fundamentals of Data Science and Machine Intelligence

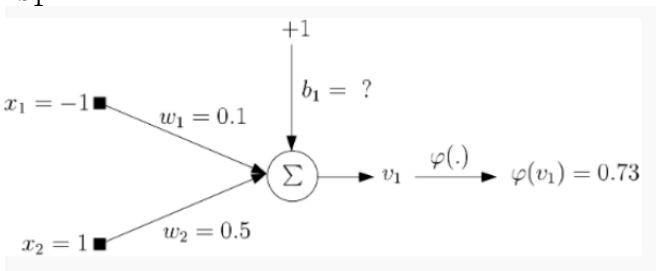
Assignment 5

Based on Module 6: Artificial Neural Networks

1. A neural network consists of N input neurons, H hidden neurons, and C output neurons. How many total weights are there in the network (excluding biases)?
   a. If the connections between input and hidden layers, and between hidden and output layers.
   b. If the connections between input and hidden layers, between hidden and output layers, and input and output layers.
2. Consider the following two multilayer perceptrons, where all layers use linear activation functions.
   a. Give one advantage of Network A over Network B.
   b. Give one advantage of Network B over Network A.



3. Consider a two-input neuron with a logistic activation function with slope parameter a = 2. Let the inputs be [-1, 1] and the weights are [0.1, 0.5] respectively. The output of the neuron is 0.73. What is the value of the bias b_1?



4. Show the perceptron that calculates:
   a. NOT of its input.
   b. NAND of its two inputs.
5. Show the perceptron that calculates the Parity of its three inputs. The parity of a binary number refers to whether the number of 1-bits in its binary representation is even or odd.
6. Derive the weight update equations for an MLP that uses ReLU in its hidden units. Assume MLP with one hidden layer of H units and one output trained for regression.

Programming Questions:

7. Perceptron Learning Algorithm for Binary Classification: For the dataset in *A5_P1.csv*, implement a Perceptron using gradient descent to classify the data points.
   - Activation function: step
   - Weight initialization: random (using *numpy* package)
   - Learning rate: 0.01
   - Number of weight update iterations: 20

   Visualize the decision boundary by plotting the dataset points and the separating line. (Mark data points for different classes with different colors.)
   Rerun the code with different random weight initializing to understand the importance of initialization.
8. Neural Network from Scratch: Using the Iris dataset, implement a feedforward neural network with one hidden layer (with five neurons) and train it using Stochastic Gradient Descent (SGD) to classify the species correctly.
   - Load the Iris dataset from *sklearn.datasets.load_iris()*.
   - Split the dataset into training (80%) and testing (20%) subsets.
   - Standardize the features to have zero mean and unit variance.

- Activation functions: tanh for the hidden layer and softmax for the output layer.
- Loss function: Mean Squared Error (MSE)
- Weight initialization: random
- Learning rate: 0.01
- Number of weight update iterations: 1000

Implement forward and backward propagation.

a. Plot the training and testing loss vs. epochs.
b. Plot the training and testing accuracy over epochs.