CS 240: Lab 4
Sigmoid Training

TAs: Jaladhi Joshi, Thomas Biju

Instructions

- This lab will be **graded**.
- Please read the problem statement and submission guidelines carefully.
- For any doubts or questions, please contact either the TA assigned to your lab group or one of the two TAs involved in making the lab.
- The deadline for this lab is **Thursday, 6 February, 5 PM** but solutions till 5:30 PM will be accepted. No submissions will be accepted after 5:30 PM.
- The submissions will be checked for plagiarism, and any form of cheating will be penalized.

Problem Statement

The objective of this lab is to build and train a neural network using a sigmoid activation function to solve four different problems: XOR, palindrome detection, majority function, and the even parity function.

1. **XOR (Odd Parity) function:**
   XOR takes two binary inputs and outputs 1 if odd number of inputs are 1s, and 0 otherwise.

   Input: Two binary inputs
   Output: A single binary value (1 if the number of 1s is odd, otherwise 0).

2. **Palindrome Detection on Binary Sequences**
   A palindrome is a sequence that reads the same forward and backward.

   Input: A binary sequence of fixed length (You have to implement for 4 and 5 inputs)
   Output: A single binary value (1 if the sequence is a palindrome, 0 otherwise).

3. **Majority Function**
   The majority function takes a binary sequence as input and outputs 1 if the number of 1s is more than number of 0s, and 0 otherwise.

   Input: A binary sequence of fixed length (You have to implement for 4 and 5 inputs)
   Output: A single binary value (1 if more than half the bits are 1, otherwise 0).

4. **Even Parity**
   This function takes an input binary sequence and outputs 1 if the number of 1s is even, else 0.

   Input: A binary sequence of fixed length (You have to implement for 4 and 5 inputs)
   Output: A single binary value (1 if the number of 1s is even, otherwise 0).

Tasks to be Completed

Complete the following tasks in the provided Jupyter Notebook file:

Task 1 - Create the datasets for all the four functions - XOR, Palindrome detection, Majority, Even Parity. The dataset for XOR should be for two inputs and for each of the other three functions, two datasets have to be created, corresponding to 4 and 5 inputs.

Task 2 - Implement the Neural Network

- Define the architecture of a feedforward neural network with an input layer (with 2, 4 or 5 neurons), no hidden layer, and an output layer (with 1 neuron). Initialize the weights and biases of the neurons randomly.
- Use a sigmoid activation function for all neurons other than the input neurons (that is, only the output neuron). Take the loss function as MSE loss.

Task 3 - Set up forward propagation and back propagation and update the weights and biases of the neural network using gradient descent.

Task 4 - For each of the 7 datasets, separately train the appropriate neural networks. Think about whether adding a hidden layer is necessary or not.

Task 5 - To visualize the training progress of the neural network, plot the Mean Squared Error (MSE) loss over the epochs. This will help to understand how the model improves over time.

Submission

- Submissions should be made on Moodle. Submit the Jupyter Notebook file renamed as `rollnumber1_rollnumber2.ipynb` (the ”b” in roll number should be in small case).
- The hard deadline for submission is 5:30 pm. No submission after that will be evaluated.
- Only one person per team should submit their solution.