CS 240: Lab 5
Hidden Layers

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, 13 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 sigmoid activation functions on all neurons except the input neurons on the MNIST data set as well as the datasets for those functions in the last lab out of (XOR, palindrome detection, majority function, and the even parity function) whose training loss does not converge to 0 without hidden layers.

1. **Preprocessed MNIST dataset:**
   MNIST is a widely used dataset of handwritten digits, containing 60,000 training and 10,000 testing grayscale images of size 28×28 pixels. Each image represents a single digit (0–9). The preprocessed data (flattening and normalization) is given in the template file.

   Input: Each sample (image) has $28 \times 28 = 784$ float values between 0 and 1 (after normalization)
   Output: 10 float values, each corresponding to the probability of the input being in a class (used to find the predicted class from 0-9)

2. **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).

3. **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 5 inputs)
   Output: A single binary value (1 if the sequence is a palindrome, 0 otherwise).

4. **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 5 inputs)
   Output: A single binary value (1 if more than half the bits are 1, otherwise 0).

5. **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 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 - For this lab, use the datasets for the following functions: XOR, Palindrome Detection, Majority, and Even Parity, from your previous lab submission on Moodle. Select only those functions whose training loss does not converge without any hidden layer. Additionally, use the MNIST dataset, where the training and testing datasets are provided separately in the template code.

Task 2 - Implement the Neural Network:

- Define the architecture of a feedforward neural network with hidden layers and initialize the weights and biases randomly.
- Use the sigmoid activation function for all neurons except the input neurons. Train the network using both loss functions: Total Sum Squared (TSS) Loss and Cross-Entropy (CE) Loss.
- Choose the number of hidden layers and neurons per layer such that the testing accuracy (percentage of test images correctly classified) exceeds 80% for the MNIST dataset and the training loss converges to 0 for the other datasets.

Task 3 - Implement forward propagation and backpropagation, and update the weights and biases of the neural network using gradient descent.

Task 4 - Train separate neural networks for each dataset.

Task 5 - To visualize the training progress, plot both the Total Sum Squared Error (TSS) loss and Cross-Entropy (CE) loss over the epochs. This will help in understanding 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.