Deep Learning Assignment One

Due October 20, 2023. This assignment must be completed in groups of 3 or 4.

Problem

Develop an autodifferentiation package based on Numpy. Do not use an existing deep learning or autodifferentiation package. Your package should support:

- Multi-layer fully-connected networks with multiple inputs
- ReLU nonlinearities
- Regression and binary classification

You will show that your package produces the correct gradients for an example network. The network should have two inputs, two hidden layers with ten neurons each, and a single output. The hidden-layer neurons should have ReLU nonlinearities. The output neuron should not have a nonlinearity. The loss should be $(\hat{y} - y)^2/2$. Use the weights, biases, inputs, and targets in the file *assignment-one-test-parameters.pkl*, which is on Learn.

Print out the gradients of the first-layer weights and biases of the untrained network, for the first (input, target) pair in the training dataset.

Next, train the network for five epochs. In each epoch, calculate the average gradient with respect to **all** parameters over **all** (input, output) pairs in the dataset. Perform one parameter update for each epoch, by subtracting $1/100^{th}$ of the gradient from the parameters. Plot a training curve with average loss over the dataset before any updates (with initial parameters) and after each update.

Submission Checklist

Submit the following:

- A PDF file with:
    - An explanation of your design choices and how the code works (up to two pages including code listings and diagrams)
    - The results of your tests, including the requested gradients and the requested plot.
- Your Python source code, including your package and your test code.

Grading

Marks will be assigned as follows:

- Completeness with respect to the requirements (3 marks)
- Correctness on the demonstration data (3 marks)
- Reasonableness of design choices (2 marks)
- Clarity of the code (1 marks)
- Clarity of the written explanation (1 mark)