Assignment #1: due week 5 just before class at 1759 hours

Given a fully connected Neural Network as follows:

1. Input (x1,x2): 2 nodes
2. First hidden layer: 10 nodes, with weights (w) and bias (b), sigmoid activation function
3. Second hidden layer: 10 nodes, with weights (w) and bias (b), sigmoid activation function
4. Output (predict): 1 node

(1) Implement this neural network in pytorch
(2) Generate the input date (x1,x2) \in [0,1] drawn from a uniform random distribution
(3) Generate the labels y = (x1*x1+x2*x2)/2
(4) Implement a loss function L = (predict-y)^2
(5) Use batch size of 1, that means feed data one point at a time into network and compute the loss. Do one time forward propagation with one data point.
(6) Compute the gradients using pytorch autograd:
a. dL/dw, dL/db
b. Print these values into a text file: torch_autograd.dat
(7) Implement the forward propagation and backpropagation algorithm from scratch, without using pytorch autograd, compute the gradients using your implementation
a. dL/dw, dL/db
b. Print these values into a text file: my_autograd.dat
(8) Compare the two files torch_autograd.dat and my_autograd.dat and show that they give the same values up to numerical precision errors

Submission:

Submit your code and report (max 2 pages):

1. One copy in NTULearn.
2. One copy in your course GitHub project