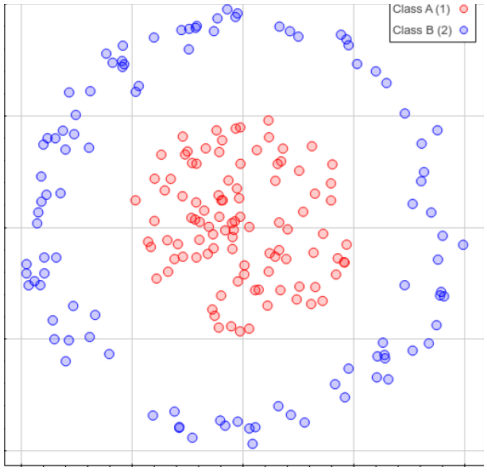
Deep Learning: Assignment # 2

This assignment consists of the following tasks:

1. Download the Jupyter notebook, NNExampleTorch, from the course page. This notebook contains code for training a two-layer neural network with ReLU activation units. The code uses autograd feature to compute gradients. The input is a two-dimensional feature vector and the desired output is coded as a one-hot vector.

    - First play with a few learning rates over the range of 0.1 to 0.1e-8. Then perform an experiment by selecting three learning rates and generate plots showing how loss goes down with training.
    - Pick the best learning rate found in the previous step and change the number of hidden layer neurons to 20, 30, 40, and 50. Plot the accuracy vs hidden layer size.
    - Next, replace the ReLU function with sigmoidal units and repeat the above steps.

2. Next, you will generate 150 examples each from two classes as per the picture shown below, and train a two layer sigmoidal network to obtain the best accuracy that you can achieve with fewer hidden units.

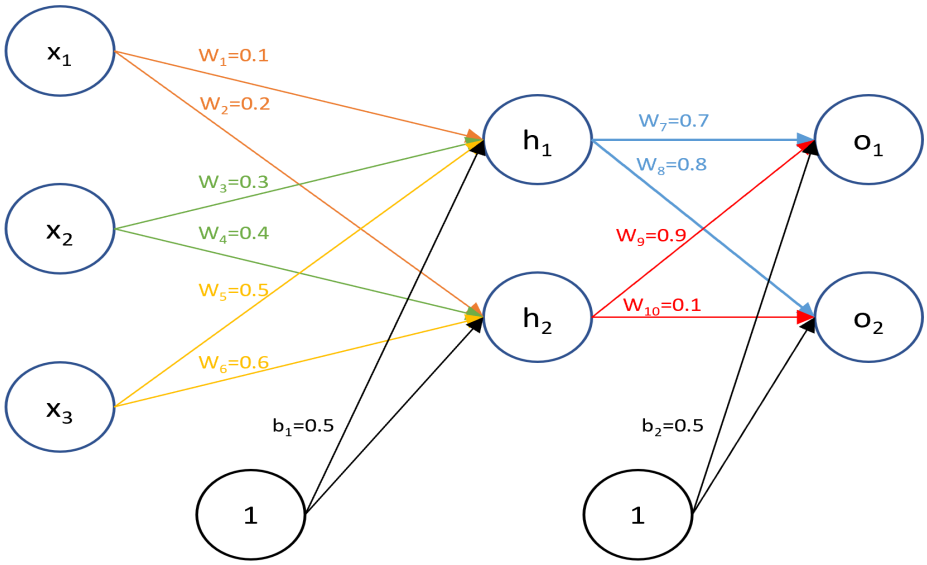


3. This exercise is meant for gaining an understanding of the backpropagation algorithm. You are given a simple neural net as shown in the figure with initial weights as indicated. The neurons are all sigmoidal neurons.

The initial and target values of the sample we'll be working with are:

$$x_1 = 1, x_2 = 4, x_3 = 5, t_1 = 0.1, t_2 = 0.05$$

Your task is to calculate the updated weight values for all the weights after presenting the input shown above. Assume learning rate as 0.01.



IMP. All submissions should be in one PDF/html file.