Experiment Report

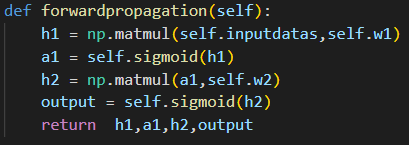
1. **Introduction**

We will do the backpropagation and forward propagation by Numpy (Python lib) in this lab. Before starting to do this lab, I go to understand the derivative of loss function and chain rule which are used to do backpropagation. And then I use a class to package all the function we need, and then build two models which are Linear model and Xor model. Finally, we can get two high accuracy models after we train those.

**2. Experiment setups**

1. Sigmoid functions

Because we are doing binary classification and sigmoid function can map the value into a range in [0,1], that can help us easily to classify the input data. I will use it after a linear layer.

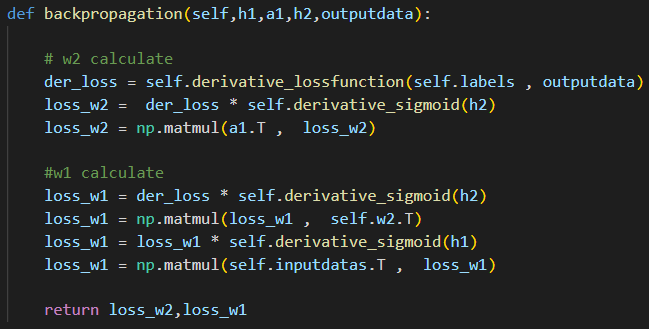


1. Neural network

The above picture is my network architecture. I will use two hidden layers consist of a linear function and sigmoid function. The important thing is I save the output of every layer in order to do backpropagation. In this network, I only use two weights because the sigmoid function is one-to-one function, we don’t need to give it a weight.

1. Backpropagation

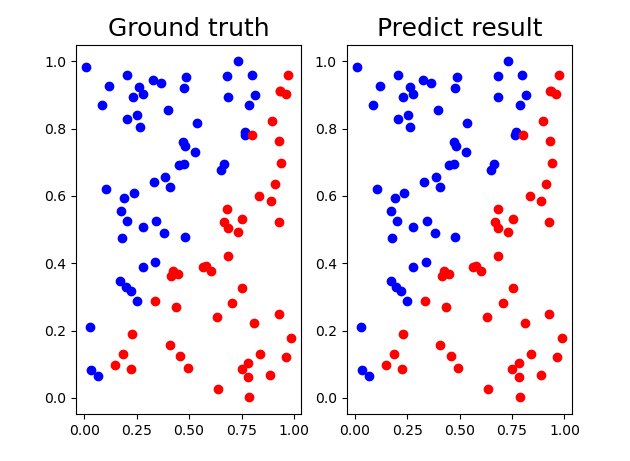
The below picture is the code of my backpropagation. I use chain rule to calculate what I need to multiply.



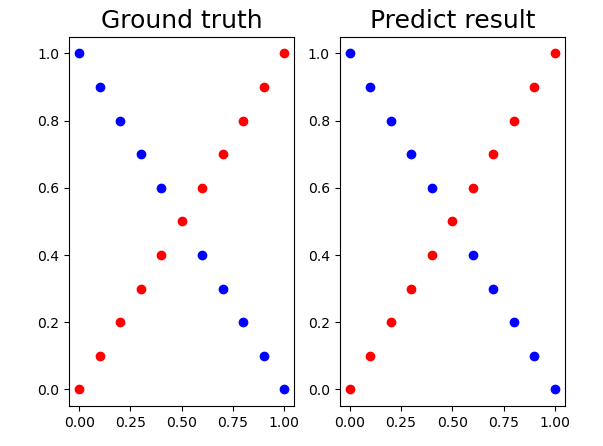
**3. Results of your testing**

1. Screenshot and comparison figure

Linear :



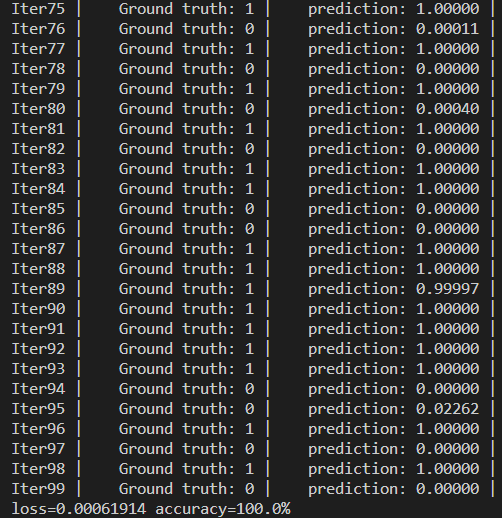
Xor:



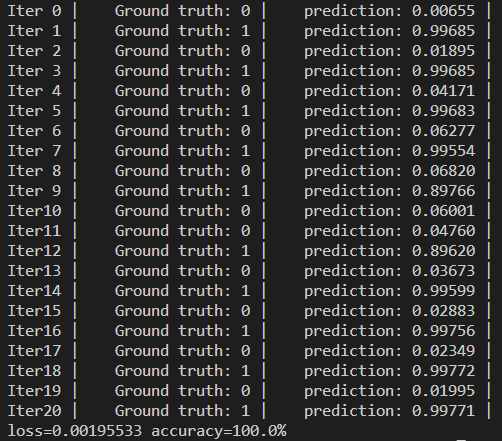
1. Show the accuracy of your prediction

Linear:

Since the Iterator’s results are too much that I can’t screenshot all.

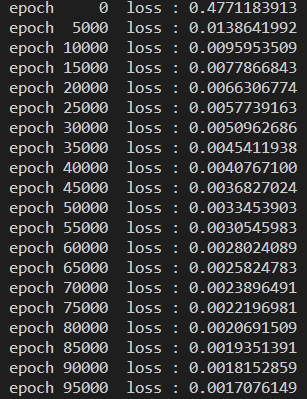


Xor:

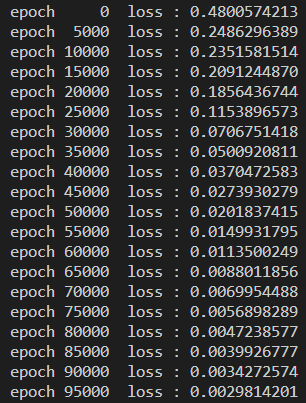


1. Learning curve (loss, epoch curve)

Linear:



Xor:



1. anything you want to present

If you use more hidden layers, you should be aware of the multiply the weights. Because we choose 2x10, 10x10, 10x1. The middle layers will be aware of multiplying the weights, you can’t transpose the wrong weights.

**4. Discussion**

1. Try different learning rates

The loss will decrease faster if I use higher learning rates.

1. Try different numbers of hidden units

At the beginning, Loss is more higher if I use the less hidden units. If we want to get the same loss, the more hidden units use more time to get the same loss.

1. Try without activation functions

I try to remove the first hidden layer’s sigmoid function and keep second layer’s sigmoid function because we need to output the value in range 0 to 1. The result show that the Linear can learn it and get 100% accuracy, but Xor can’t. The loss in Xor model didn’t decrease well, so that the accuracy just 52%.

1. Anything you want to share

I totally understand the backpropagation in this Lab01. I just can say this lab is so good!!