Lab D Report
Yanzhi Li

Observation when the learning rate is 0.3:
The converge behavior of loss when learning rate is 0.3 is very weird. The outcome is unpredictable no matter how I change the epochs. When the epochs are small, say less than 100000, it is very unlikely for the loss to converge to 0(falls below 0.1). However, it sometimes did converge to 0.01 even when the epochs are 1000-100000. As I increased the epochs, there were three noticeable things. First, the loss could converge to a large value such as 0.477 or 0.346(This happened many times) for example:

```
Current loss is: 0.3470960891031589 current epochs: 1400000
Current loss is: 0.346849350336236 current epochs: 1300000
Current loss is: 0.34676314108068146 current epochs: 1200000
Current loss is: 0.34671880448456494 current epochs: 1100000
Current loss is: 0.3466916593707199 current epochs: 1000000
Current loss is: 0.3466732753316874 current epochs: 900000
Current loss is: 0.3466599735334934 current epochs: 800000
Current loss is: 0.34664988790028595 current epochs: 700000
Current loss is: 0.3466419692255098 current epochs: 600000
Current loss is: 0.3466355814859893 current epochs: 500000.
```

Second, sometimes RuntimeWarning: overflow encountered in exp return 1/(1 + np.exp(-x)) would occur, this will guarantee that the loss will be reduced to below 0.01. Third, if the initial loss(first reported loss) is larger than 1, the loss is guaranteed to converge to 0.01. Fourth, loss could suddenly be reduced sharply,
for example:

```
Current loss is: 0.7207640368665067 current epochs: 1000000
Current loss is: 0.0009869818633656486 current epochs: 900000
Current loss is: 0.0004911593731700591 current epochs: 800000
```

In conclusion, the behavior is very unpredictable. It is hard to tell if there is a value of epochs that will guarantee to converge the loss to 0.01. The largest epochs I experiment with is 2,500,000.

Observation when the learning rate is 0.6:
Same as 0.3. It sometimes converged to a large value.

Observation when the learning rate is 0.1:
Same thing happened when the rate was 0.1. However, it is more likely for the loss to converge to 0.01 than the other learning rate.

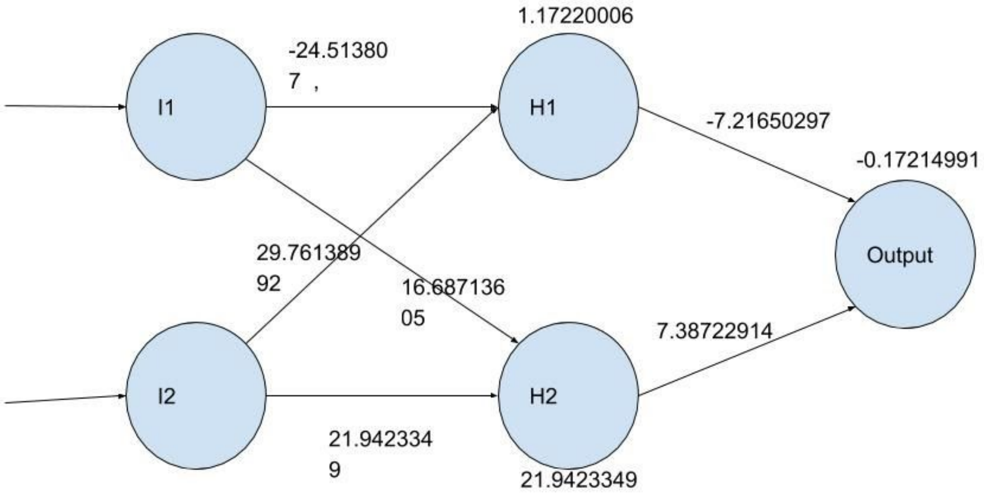
Observation when the learning rate is 0.001:

The loss when the learning rate is 0.001could still get stuck above 0.1. Besides, even if the loss was reduced steadily for 0.001 learning rate, it decreased at a very slow rate, which made it hard to fall below 0.1. This is a situation where the loss is unlikely to converge to 0.01 when the epochs are small.

Neural network draw:

```
act(3000000,"xor.txt",2,2,1,0.001)
```

```
{'hidden_weights': array([[-24.513807  ,  29.76138992],
       [ 16.68713605,  21.9423349 ]]), 'hidden_biases': array([[  1.17220006],
       [-15.2600507 ]]), 'output_weights': array([[-7.21650297,  7.38722914]]), 'output_biases': array([[-0.17214991]])}
```



This is one instance of my network. It is hard to tell whether it seems correct. Because the best prediction I made from my code is array([[0.00147924, 0.99893276, 0.9979738 , 0.00147929]]), which is supper close to [0110], while the worst could be ([[4.15343194e-04, 6.66439498e-01, 6.66439344e-01, 6.66842184e-01]]), where the middle two values are unreasonable.
In conclusion, the network sometimes works and sometimes fails.