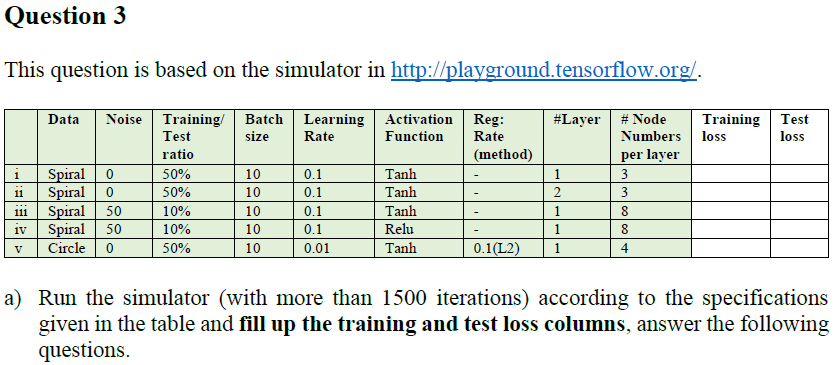




In the above equation, the differential values of the parameters Weight and Bias are proportional to the difference between the predicted value and the actual value (sigma (z) -y). Therefore, it is possible to obtain a result of updating more for an input value with a larger error and less updating for an input value with a smaller error.

Also, since sigma '(z) is not included in the derivative, there is no slowdown problem caused by the characteristic of the sigmoid function that occurs when the cost function is defined using the MSE function.

Therefore, it can be seen that the cross-entropy cost function is more suitable for training neural networks than the MSE cost function.

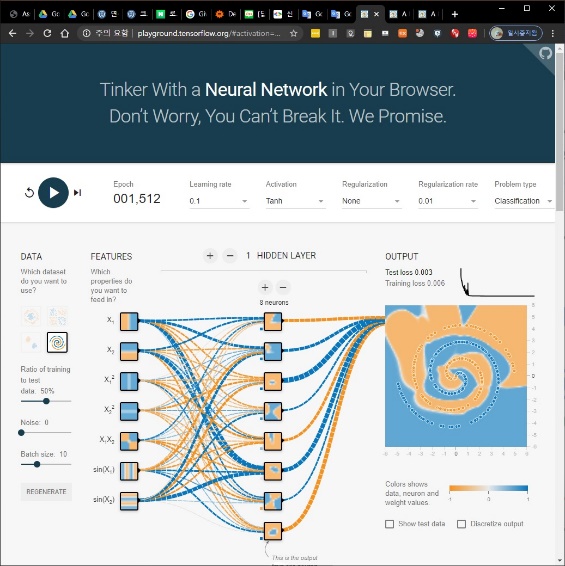


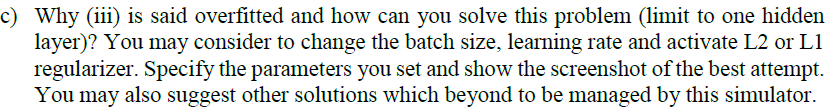
|  |  |  |
| --- | --- | --- |
|  | Training Loss | Test Loss |
| 1 | 0.323 | 0.52 |
| 2 | 0.341 | 0.459 |
| 3 | 0.154 | 0.672 |
| 4 | 0.382 | 0.883 |
| 5 | 0.25 | 0.261 |



is underfitted Because Training Loss is Big

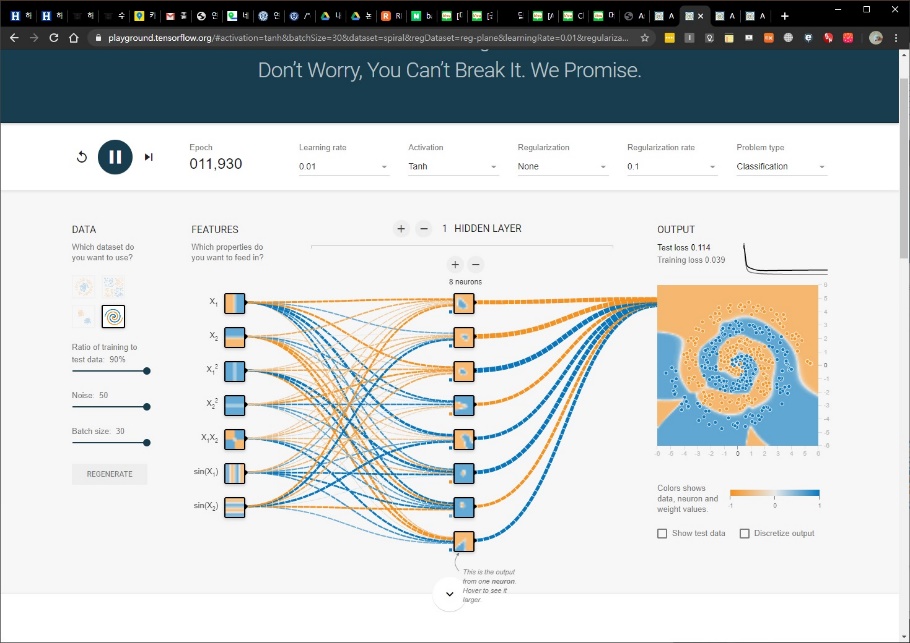
we can solve this problem by using variety of features.





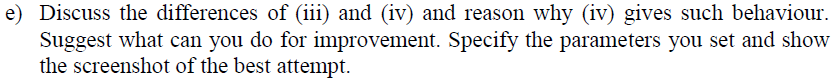
It has small Training Loss but has large Test Loss, So It is overfitted

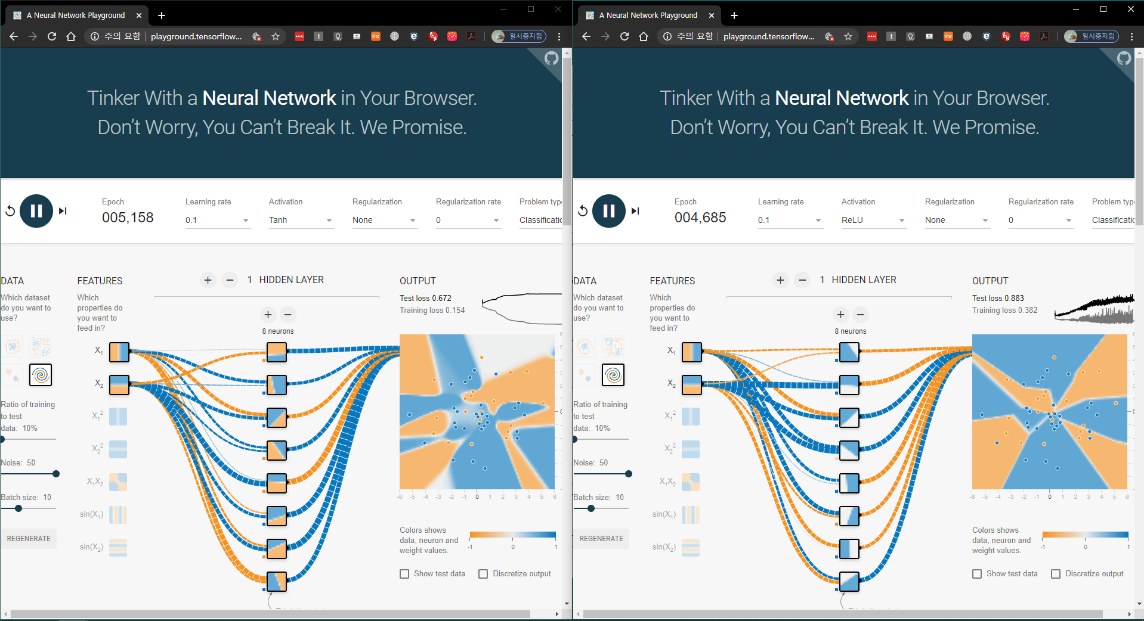
You should increase the batch size, Reduce the learning ratio



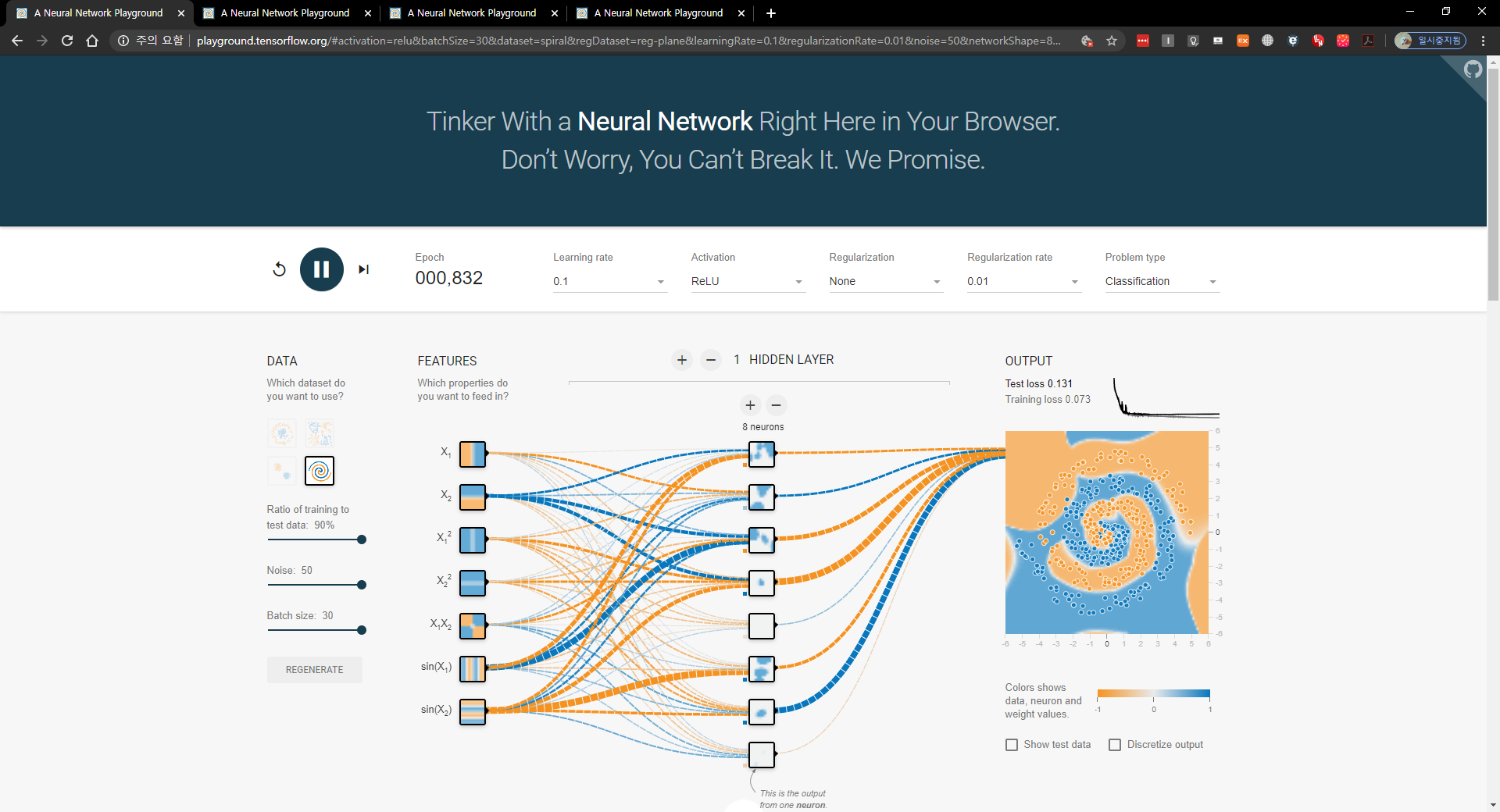


As the L2 coefficient increases, the underfitting becomes more severe, and accordingly, a lot of epoch training is possible.



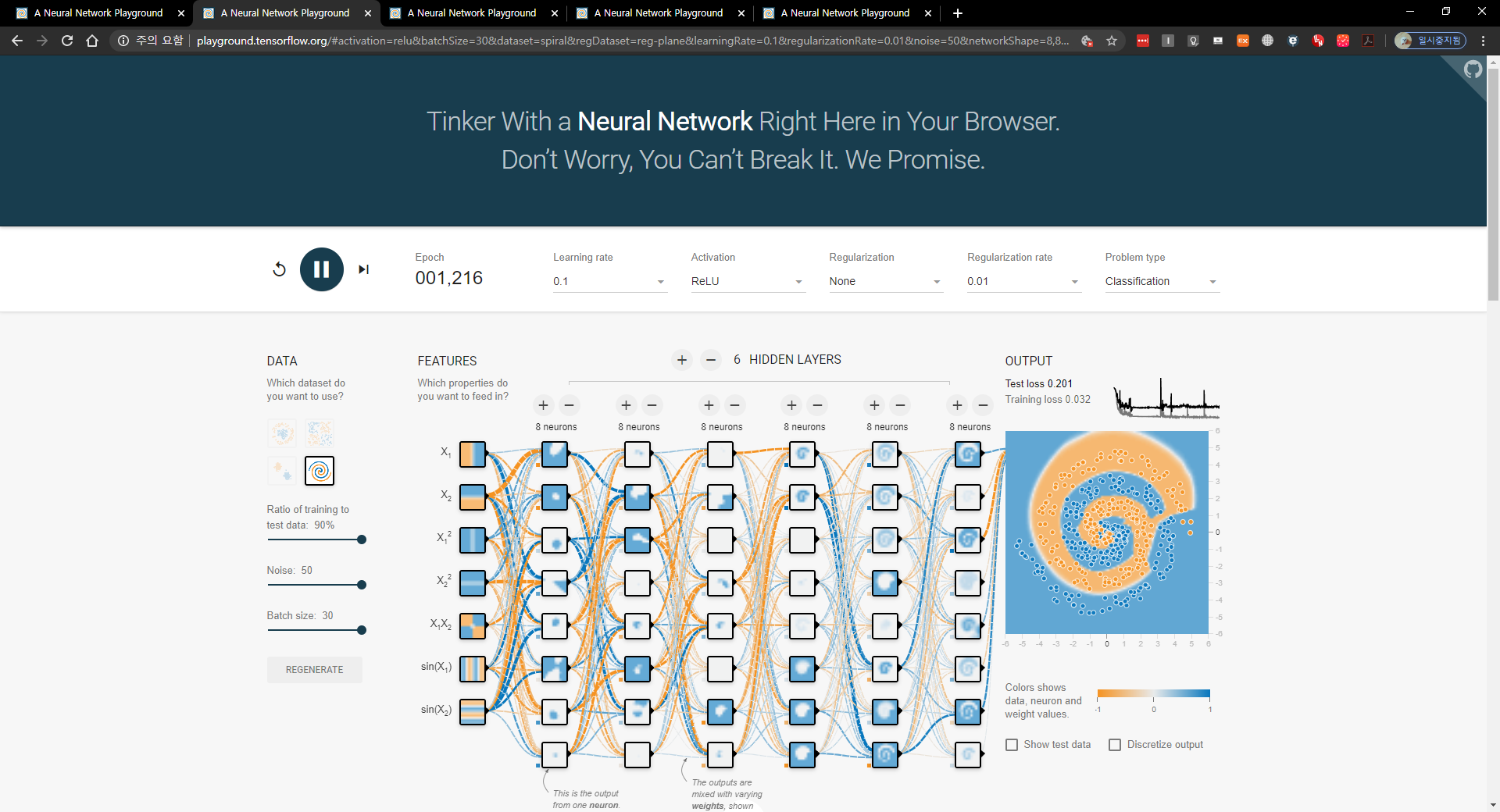


They both have high Test Loss but it has high Training Loss at ReLu

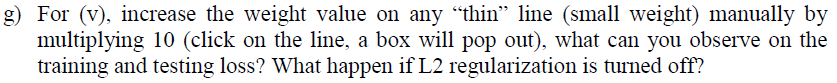


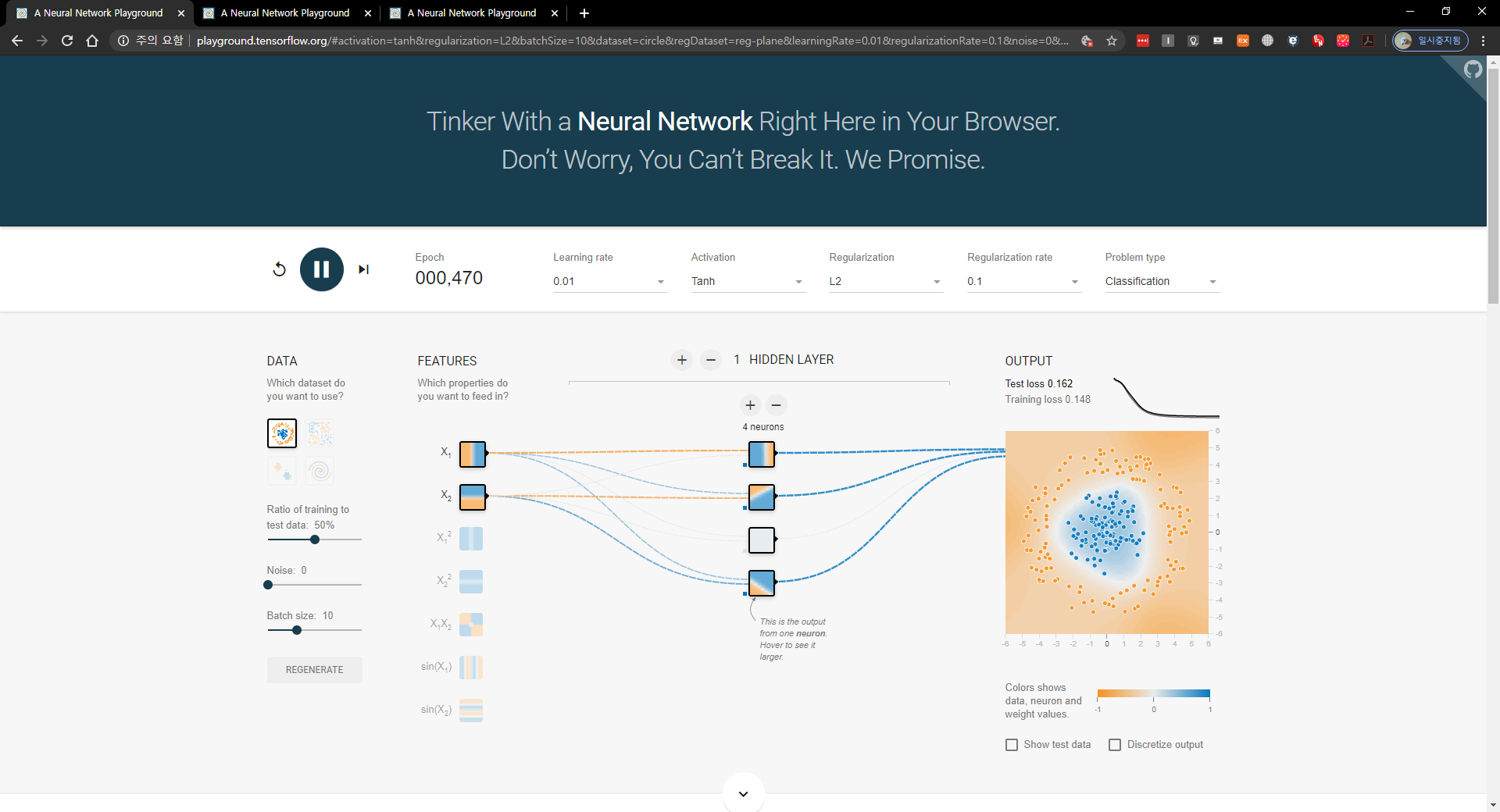
We can get better solution if we choose high Ratio of Training to test data and Batch Size

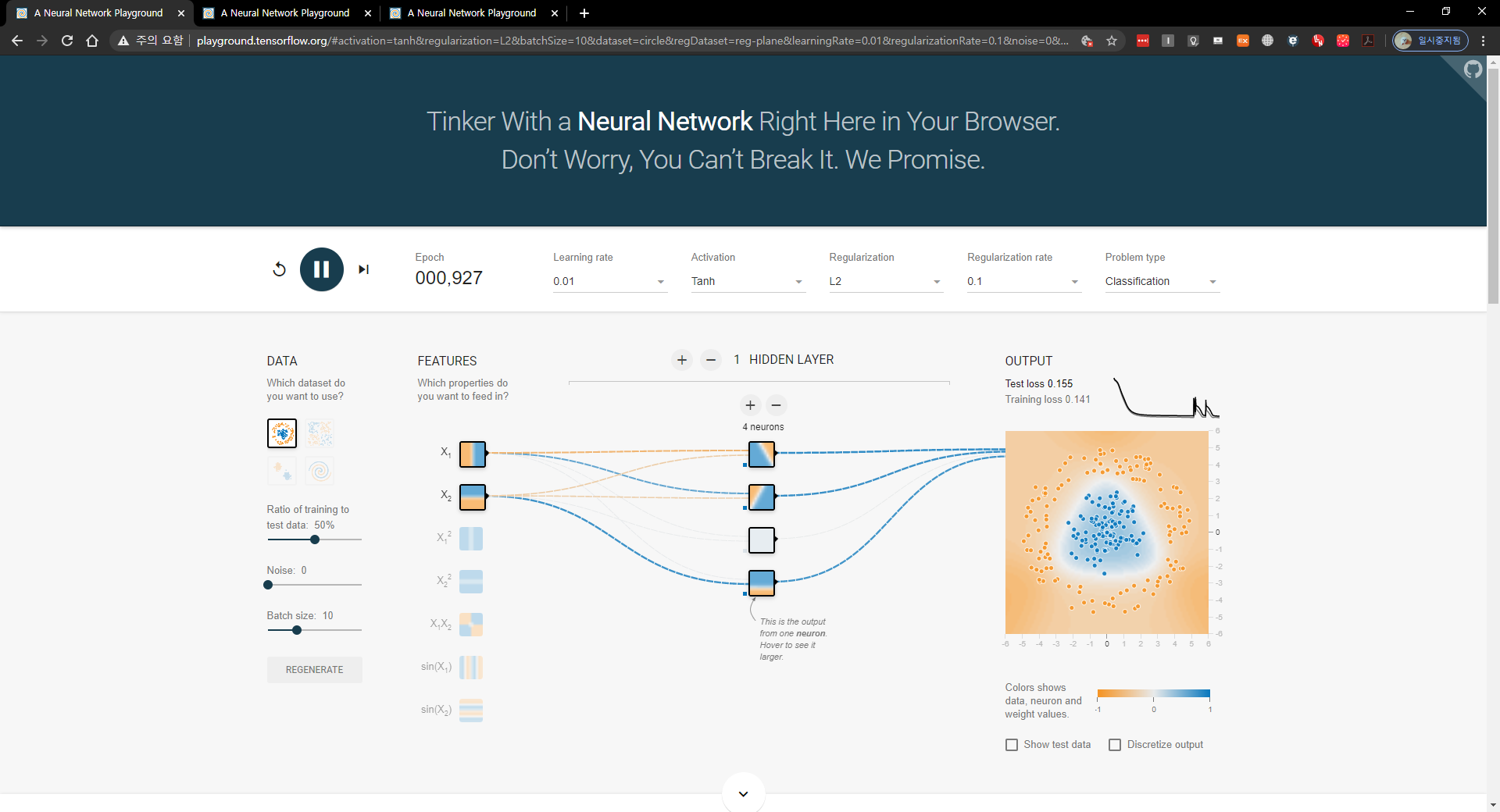




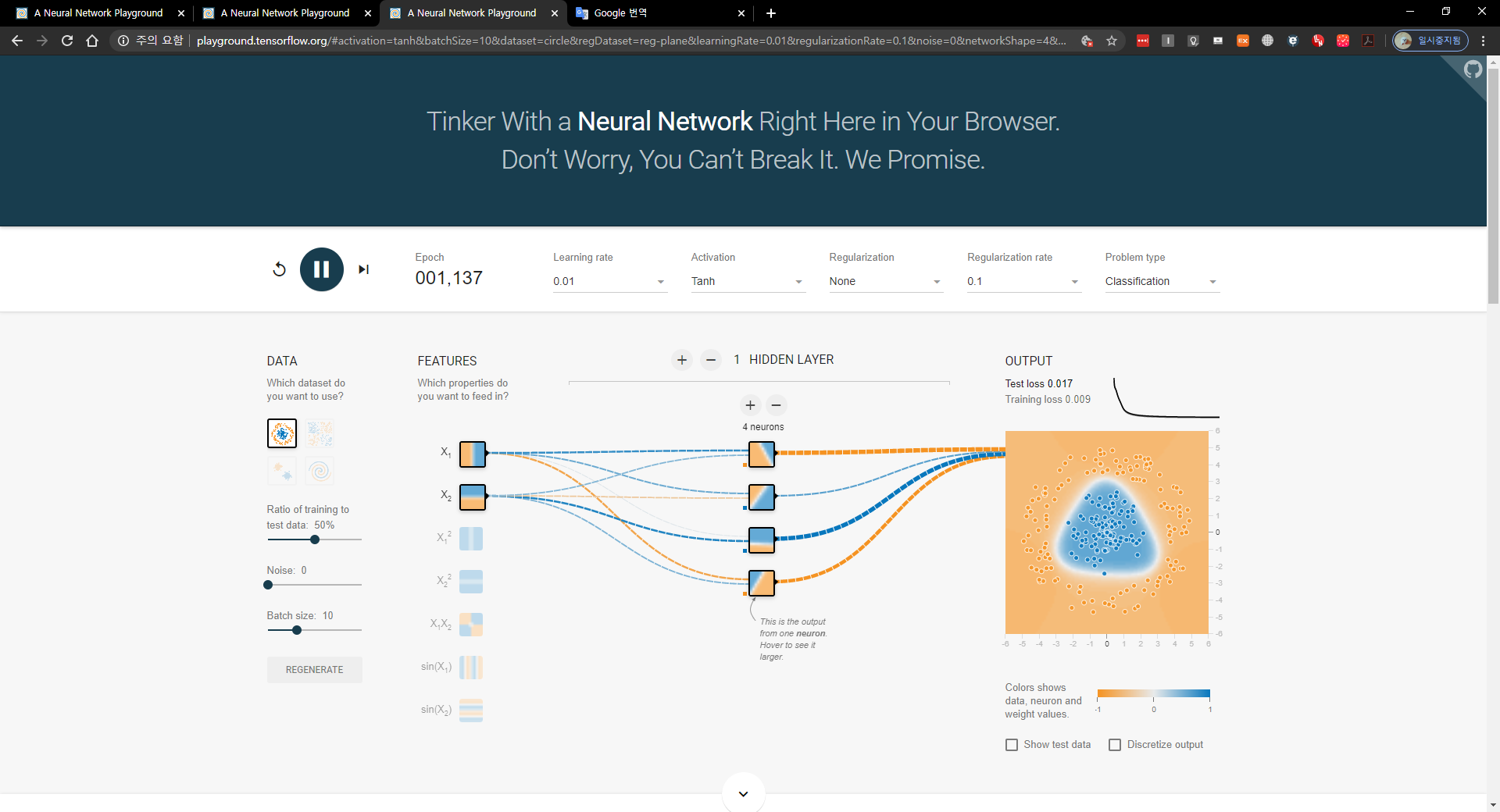
We have better result because of using more hidden layer as back-propagation





 weight changed

As you can see an output, Test loss and Training loss changed higher momentarily. But they Falls quickly.

 L2 turned off

Their boundary becomes more clear and Test loss and Training loss are gets lower.