# Force majeure 1

Name:

Morteza Rashidpour

## Exercise 1: Linear and non-linear neural network

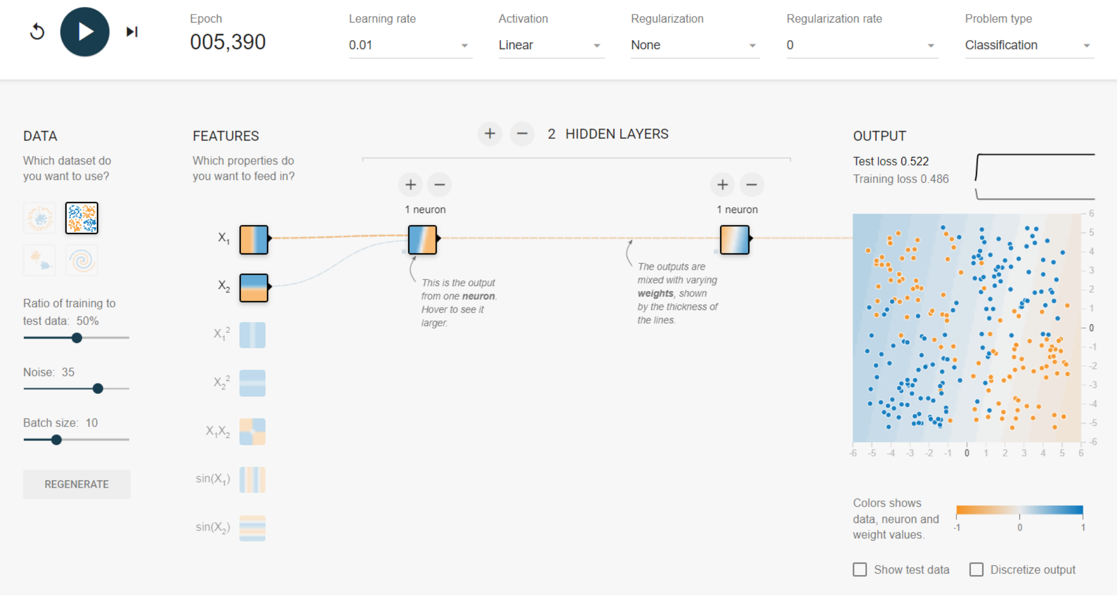
### Task 1:

This model will not learn non-linearity because its activation functions are linear.

After 5390 epochs it didn’t learn anything effectively.

Test loss: 0.522

Training loss: 0.486



Epoch 005,390

Learning rate 0.01

Activation Linear

Regularization None

Regularization rate 0

Problem type Classification

Test loss 0.522

Training loss 0.486

Ratio of training to test data: 50%

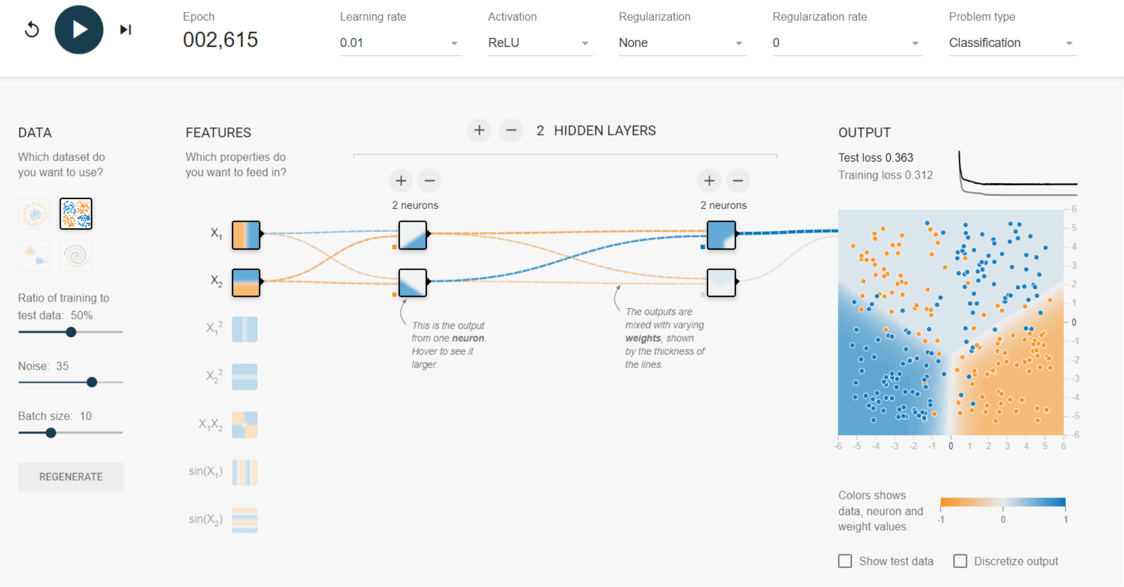
Noise: 35

Batch size: 10

DATA
Which dataset do you want to use?
REGENERATE
FEATURES
Which properties do you want to feed in?
$X_1$
$X_2$
$X_1^2$
$X_2^2$
$X_1X_2$
$\sin(X_1)$
$\sin(X_2)$
+ − 2 HIDDEN LAYERS
+ −
1 neuron
+ −
1 neuron
This is the output from one neuron. Hover to see it larger.
The outputs are mixed with varying weights, shown by the thickness of the lines.
OUTPUT
6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6
-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6
Colors shows data, neuron and weight values.
-1
0
1
□ Show test data
□ Discretize output

### Task 2:

Yes, by increasing the number of neurons in the hidden layers and use non-linearity activation functions like Relu it can potentially model relatively simple nonlinear relationships in the data and it can’t model the data effectively.



Epoch 002,615

Learning rate 0.01

Activation ReLU

Regularization None

Regularization rate 0

Problem type Classification

Test loss 0.363

Training loss 0.312

Ratio of training to test data: 50%

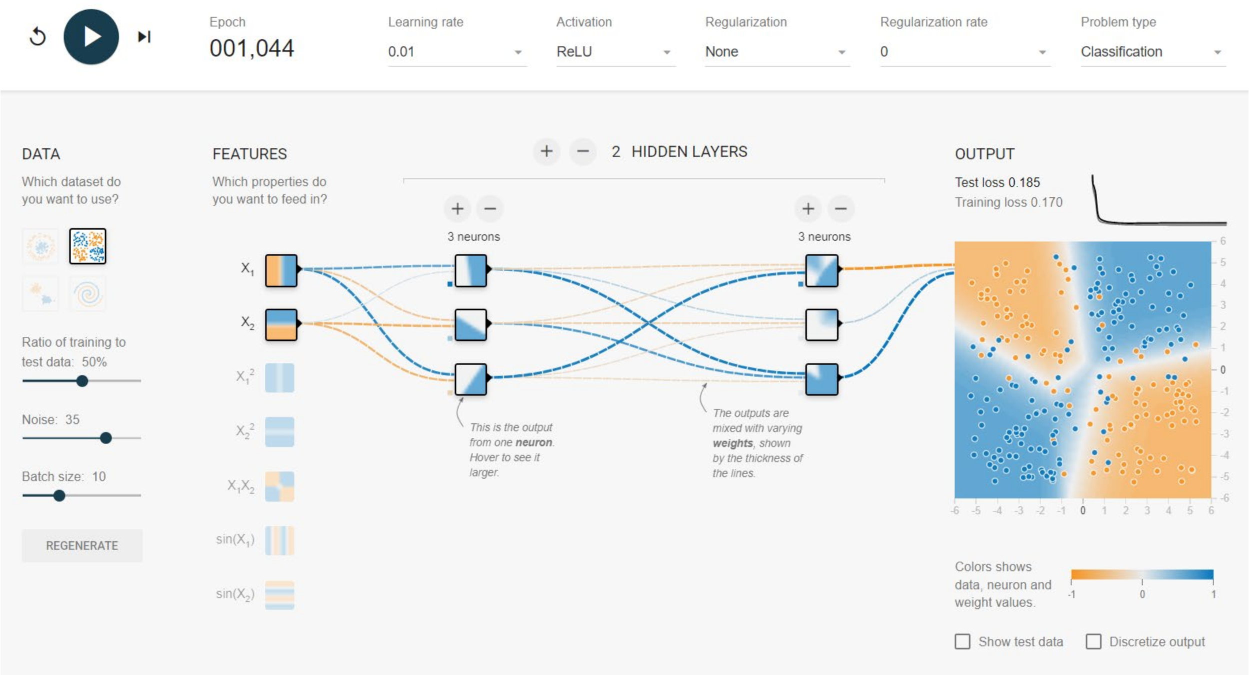
Noise: 35

Batch size: 10

DATA
Which dataset do you want to use?
REGENERATE
FEATURES
Which properties do you want to feed in?
$X_1$
$X_2$
$X_1^2$
$X_2^2$
$X_1X_2$
$\sin(X_1)$
$\sin(X_2)$
+ − 2 HIDDEN LAYERS
+ −
2 neurons
+ −
2 neurons
This is the output from one neuron. Hover to see it larger.
The outputs are mixed with varying weights, shown by the thickness of the lines.
OUTPUT
6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6
-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6
Colors shows data, neuron and weight values.
-1
0
1
□ Show test data
□ Discretize output

**Task 3:**

Yes it model the data effectively. Model quality increase from run to run.



Epoch 001,044

Learning rate 0.01

Activation ReLU

Regularization None

Regularization rate 0

Problem type Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 35

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

2 HIDDEN LAYERS

3 neurons

3 neurons

This is the output from one **neuron**. Hover to see it larger.

The outputs are mixed with varying **weights**, shown by the thickness of the lines.

OUTPUT

Test loss 0.185

Training loss 0.170

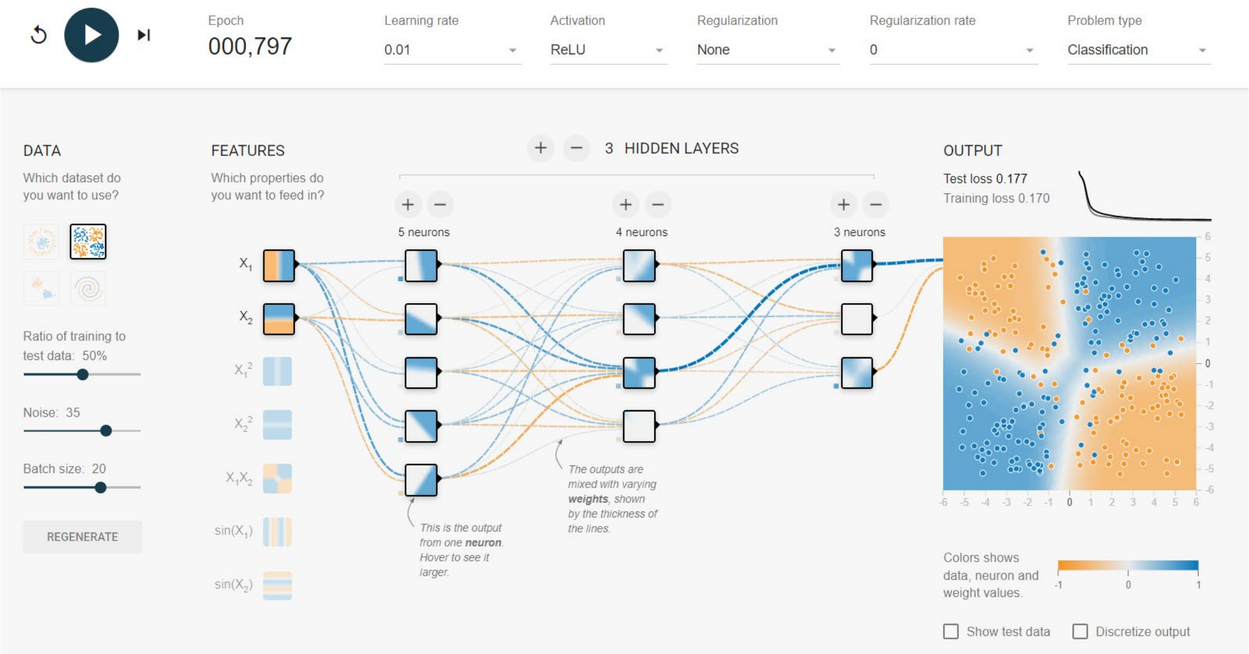
Colors shows data, neuron and weight values.

□ Show test data □ Discretize output

**Task 4:**

With this architecture we give test loss of 0.177:

First hidden layer with 5 neurons, Second hidden layer with 4 neurons, Third hidden layer with 3 neurons, Batch size = 20, activation function = relu.



Epoch 000,797

Learning rate 0.01

Activation ReLU

Regularization None

Regularization rate 0

Problem type Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 35

Batch size: 20

REGENERATE

FEATURES

Which properties do you want to feed in?

3 HIDDEN LAYERS

5 neurons

4 neurons

3 neurons

This is the output from one **neuron**. Hover to see it larger.

The outputs are mixed with varying **weights**, shown by the thickness of the lines.

OUTPUT

Test loss 0.177

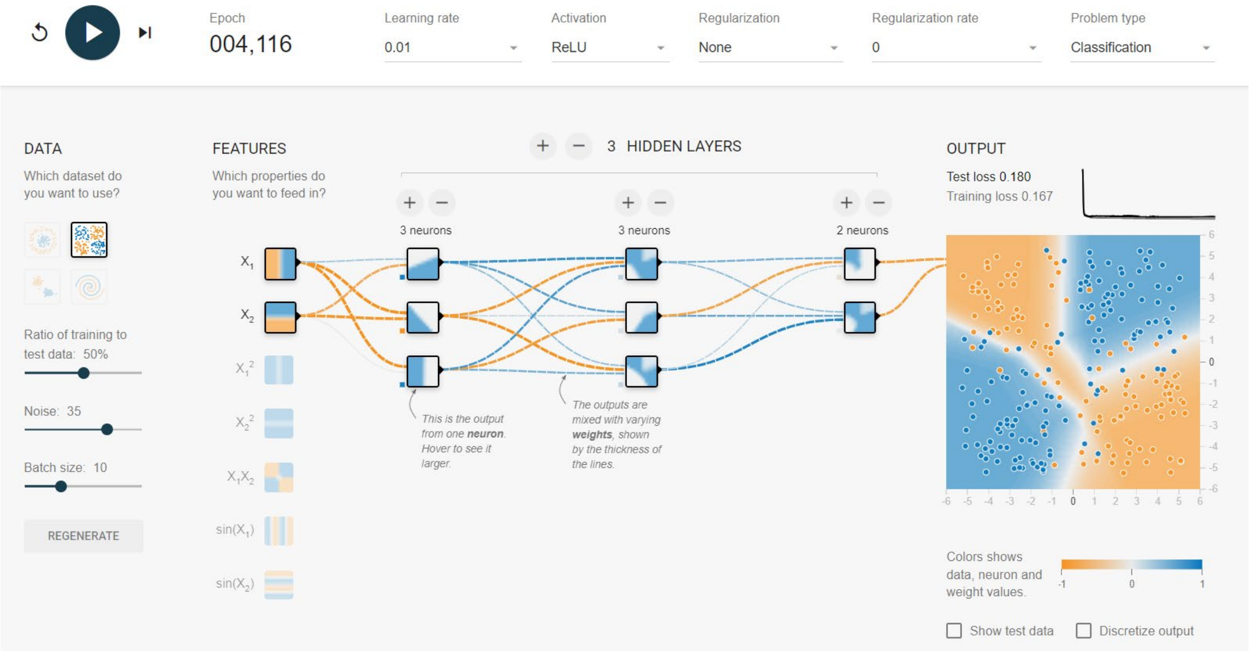
Training loss 0.170

Colors shows data, neuron and weight values.

□ Show test data □ Discretize output

### Task 5:

Increasing the model size can impact both fit and convergence in machine learning, but it's not a straightforward yes or no answer. It can improve fit but also lead to overfitting and slower convergence.



Epoch 004,116

Learning rate 0.01

Activation ReLU

Regularization None

Regularization rate 0

Problem type Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 35

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

3 HIDDEN LAYERS

3 neurons

3 neurons

2 neurons

*This is the output from one **neuron**. Hover to see it larger.*

*The outputs are mixed with varying **weights**, shown by the thickness of the lines.*

OUTPUT

Test loss 0.180

Training loss 0.167

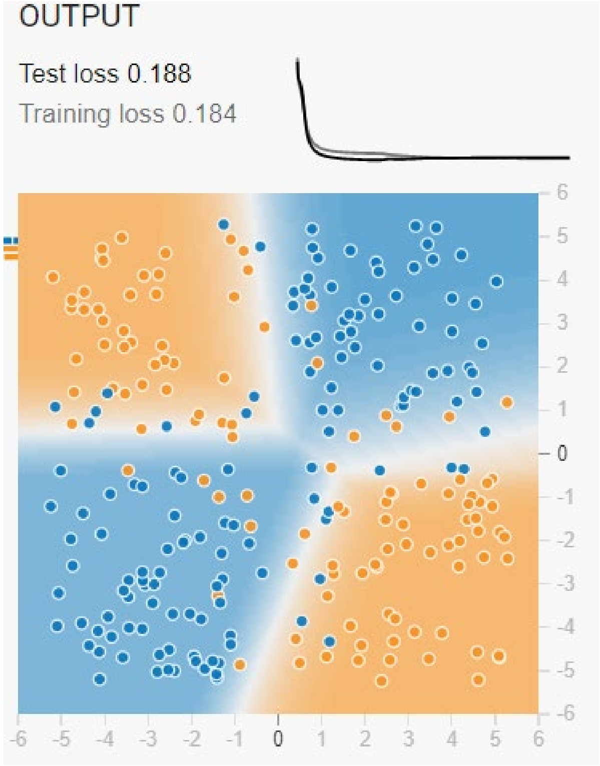
Colors shows data, neuron and weight values.

□ Show test data □ Discretize output

**Exercise 2: Neural Net Initialization**

**Task1:**

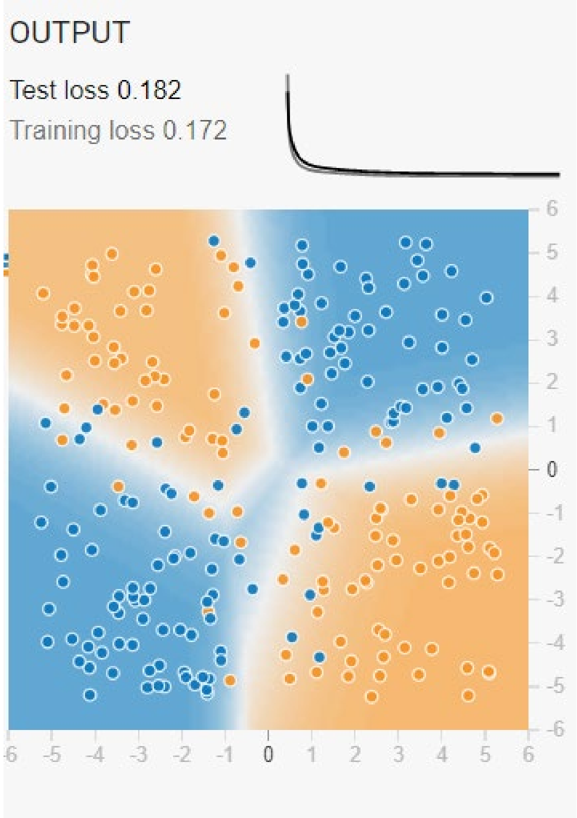
Initialization plays a crucial role in nonconvex optimization problems, as these problems often have multiple local optima, and the performance of optimization algorithms can heavily depend on the starting point.



OUTPUT

Test loss 0.188

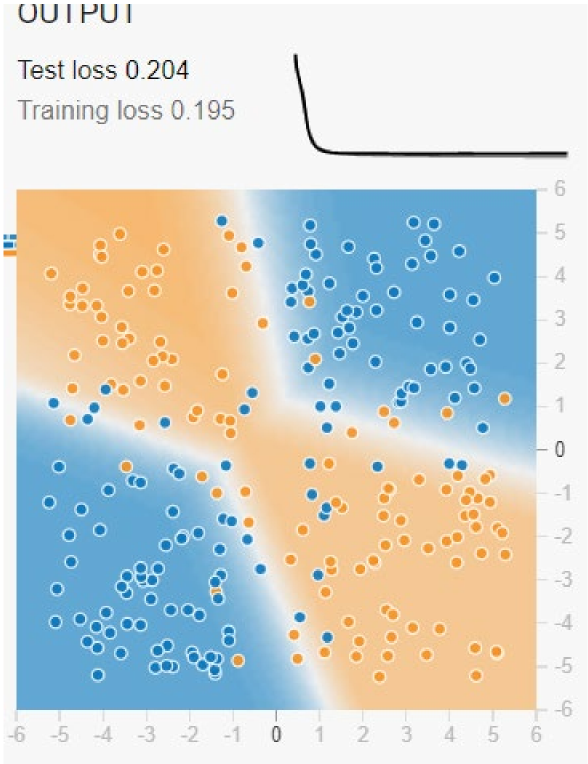
Training loss 0.184



OUTPUT

Test loss 0.182

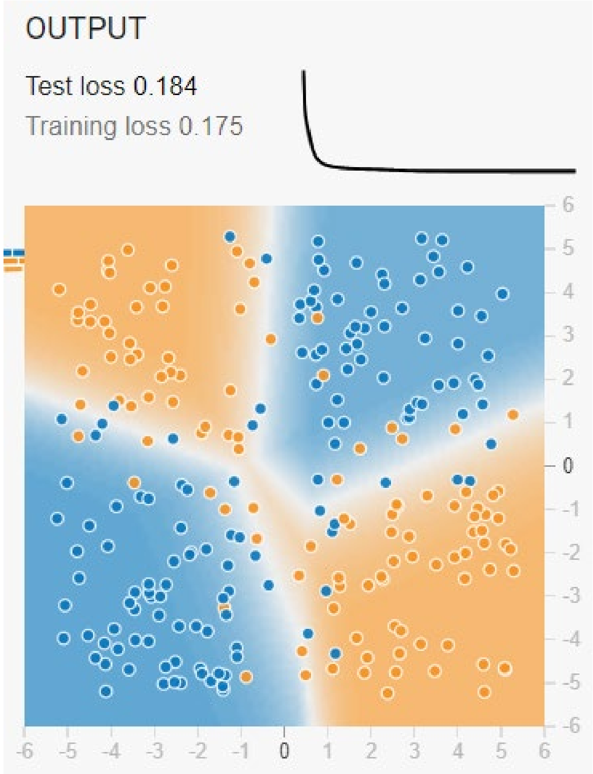
Training loss 0.172



OUTPUT

Test loss 0.204

Training loss 0.195



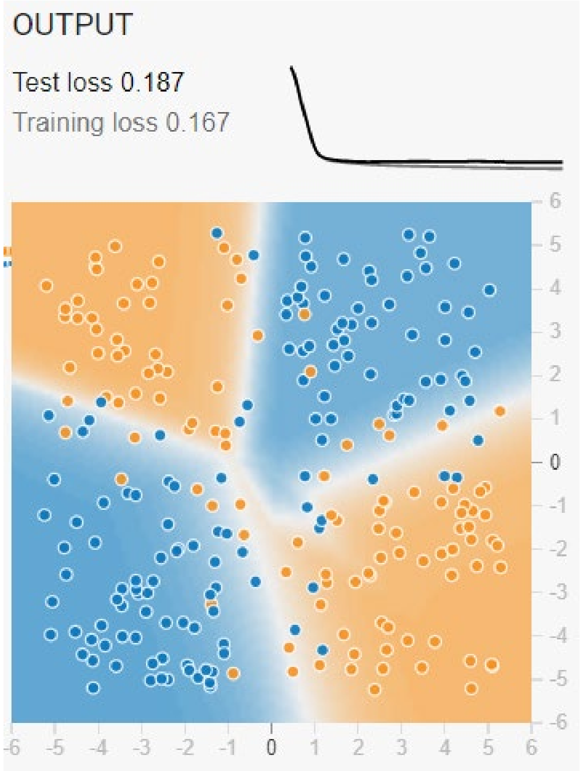
OUTPUT

Test loss 0.184

Training loss 0.175

Task 2:

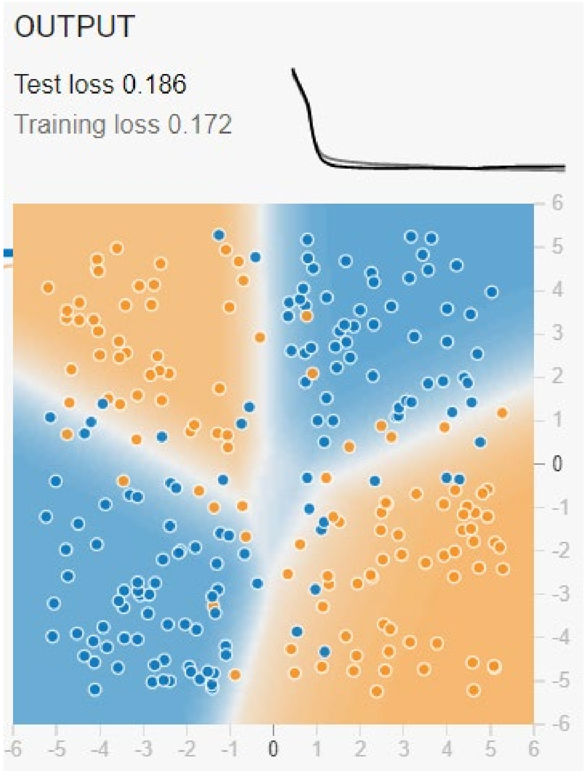
increasing model layers and nodes can potentially improve result stability.



OUTPUT

Test loss 0.187

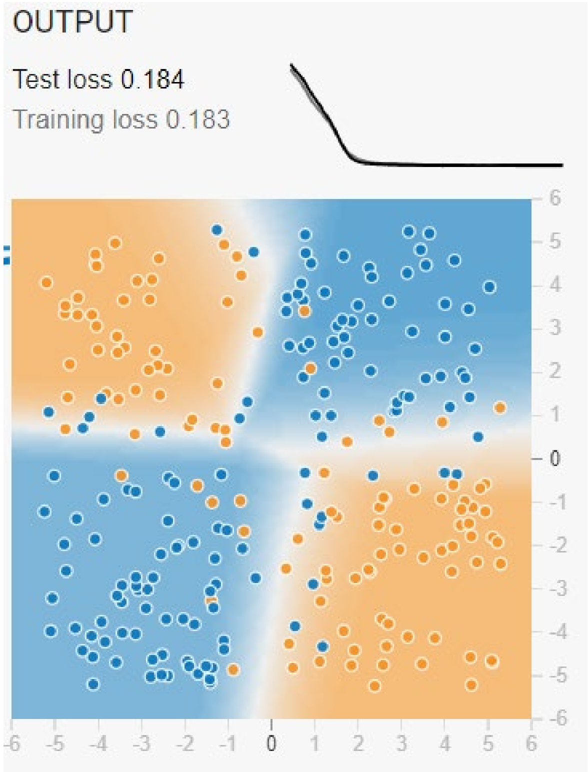
Training loss 0.167



OUTPUT

Test loss 0.186

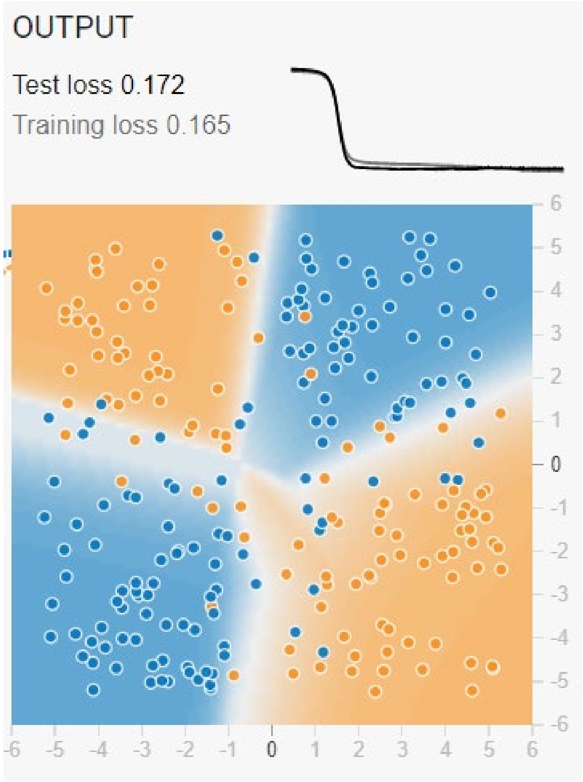
Training loss 0.172



OUTPUT

Test loss 0.184

Training loss 0.183



OUTPUT

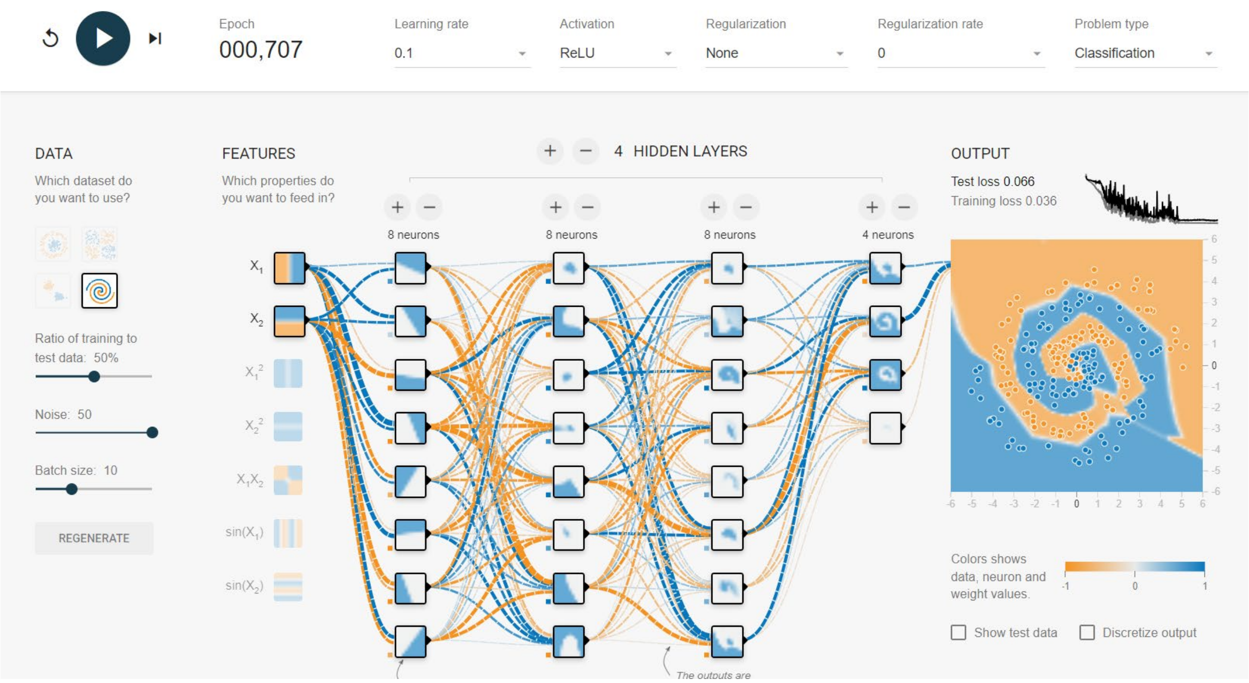
Test loss 0.172

Training loss 0.165

## Exercise 3: Complex dataset (Spiral)

### Task 1:

I get 0.066 test loss however output surface is a little smooth.



Epoch 000,707

Learning rate 0.1

Activation ReLU

Regularization None

Regularization rate 0

Problem type Classification

Ratio of training to test data: 50%

Noise: 50

Batch size: 10

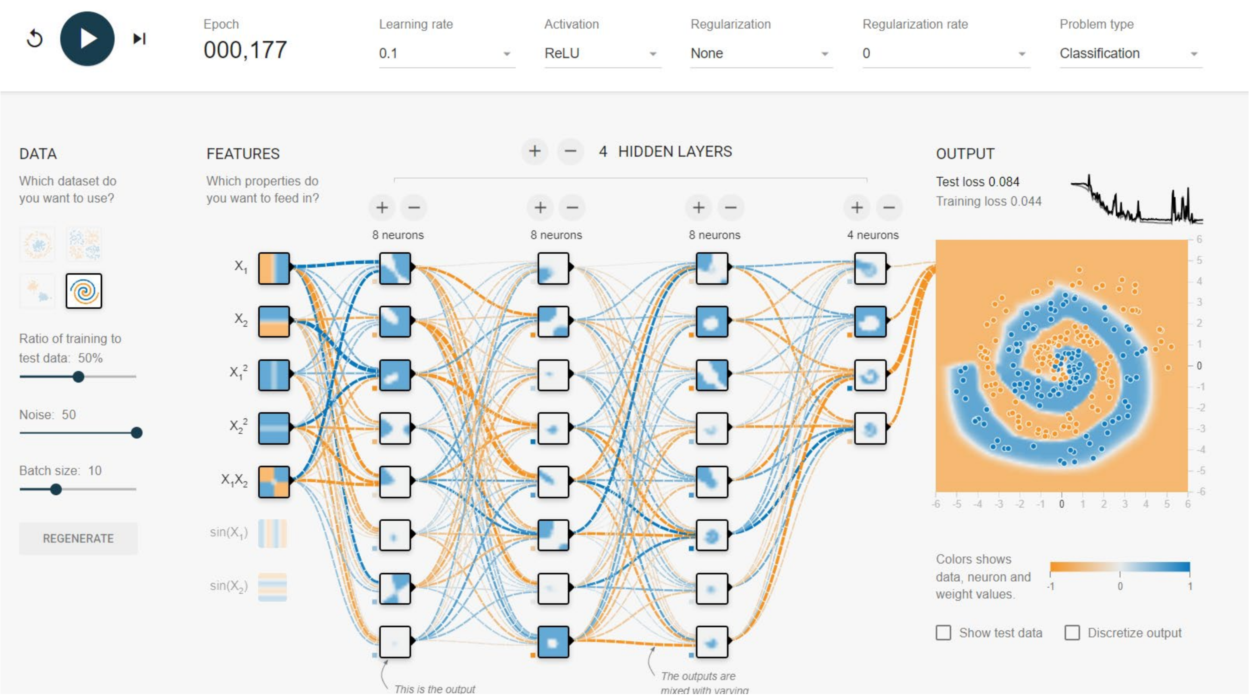
4 HIDDEN LAYERS

Test loss 0.066

Training loss 0.036

### Task 2:

After adding additional cross product and polynomial features, model get better and smoother output surface.



Epoch 000,177

Learning rate 0.1

Activation ReLU

Regularization None

Regularization rate 0

Problem type Classification

Ratio of training to test data: 50%

Noise: 50

Batch size: 10

4 HIDDEN LAYERS

Test loss 0.084

Training loss 0.044