Feed-forward neural networks

From the top

What is machine learning? A way to develop a function. We define the *form* of the function and use a whole bunch of data to learn the *parameters*.

What is a neural network? A particular functional *form*.

Building blocks

The input *data* x, a $d \times 1$ vector.

one column

Weights (parameters) w, a $d \times 1$ vector.

An *activation function* $a : \mathbb{R} \to \mathbb{R}$.

the simplest type of neural network:
we have some data
multiply each datum/predictor by the associated weight(dot function

A *perceptron* is a function of the form

$$\text{neuron}(x) = a(w^T x)$$

Neural networks

A multi-layer perceptron is a *neural network*.

$$\mathrm{layer}_i(z_i) = a_i(w_i^T z_i)$$

where z_i is the concatenation of some $\mathrm{layer}_j(z_j)$ ($j < i$).

Each perceptron corresponds to a "neuron".

"Layers"

A set of neurons connected to the same other set(s) of neurons, with the same activation function.

They are often represented in aggregate:

in this situation, f is not a scalar, it's a vector

$$f_i(z_i) = a_i(W_i^T z_i)$$

where W_i^T is a $d_i \times d_{i+1}$ matrix. $a_i()$ operates element-wise.

Training

Gradient descent: walk through the parameter space in the direction that reduces the training error.

Backpropagation is the process of inferring the correct direction. Compute $\frac{\partial E}{\partial w}$, the way the error changes with the weight: this is zero at a local minimum of E.

start somewhere, typically initializes it and randomly choose a set of parameters and then say, which direction should I go to make this better. e.g parameter1 needs to go up, parameter 55 needs to go down, we'll take a little step in that direction and we'll do it again

we're more concerned with can I make a function on the basis of these parameters that does my particular problem efficiently

Notes

- Every neural network is a feed-forward neural network.
- Weights may be "shared" - during training, enforce that $w_i = w_j$.
- In a fully-connected node, all w are trained. Otherwise, pin some w to zero.

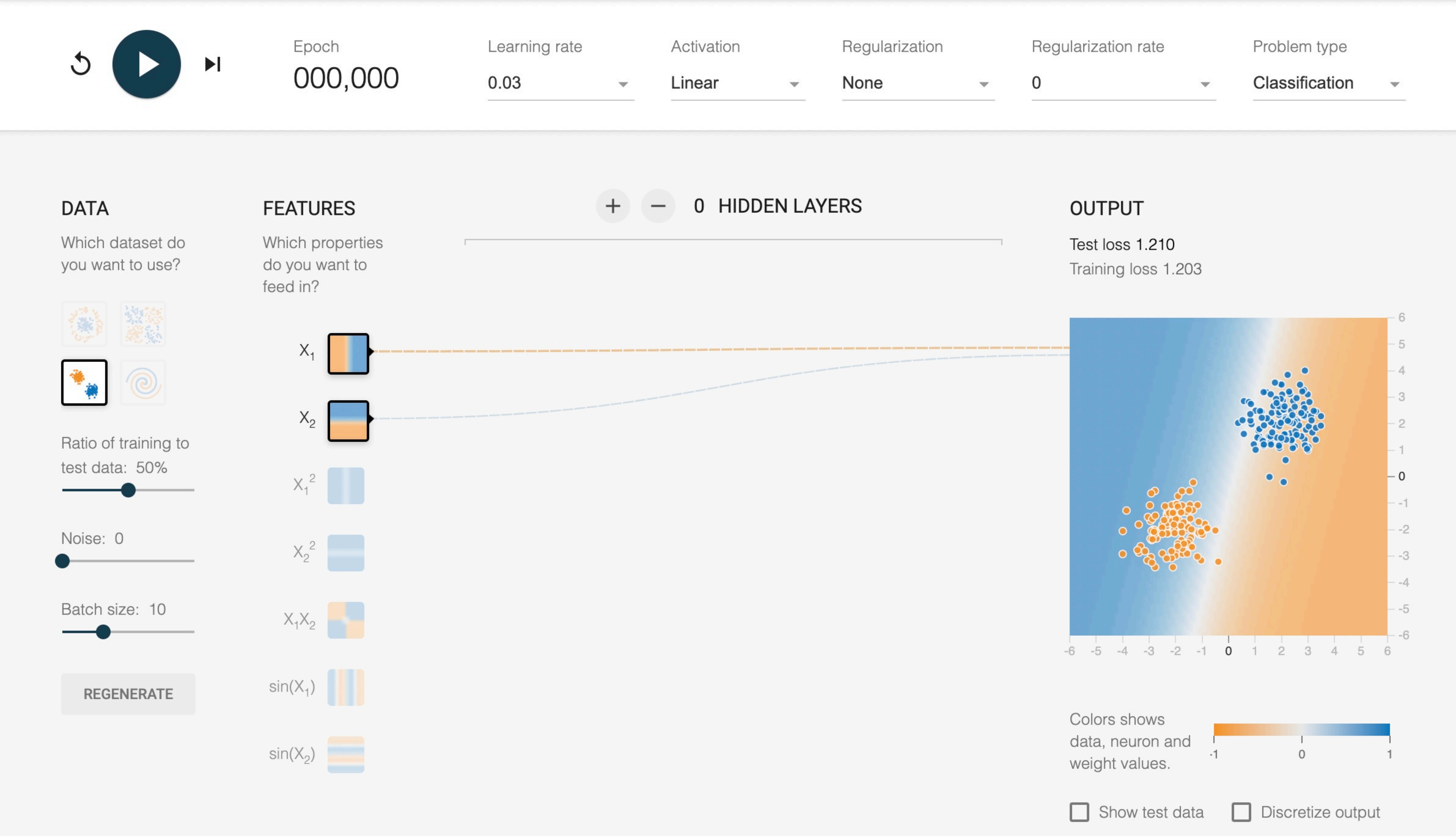
setting a weight to zero ---> just removing that connection entirely

every neuron in each layer is connecter to every neuron in the next layer

every time we can share weights, it means that this parameter space is getting smaller and smaller. The smaller the parameter space is , the easier it is to learn

Playground

https://playground.tensorflow.org/



Epoch
000,000

Learning rate
0.03

Activation
Linear

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

0 HIDDEN LAYERS

OUTPUT

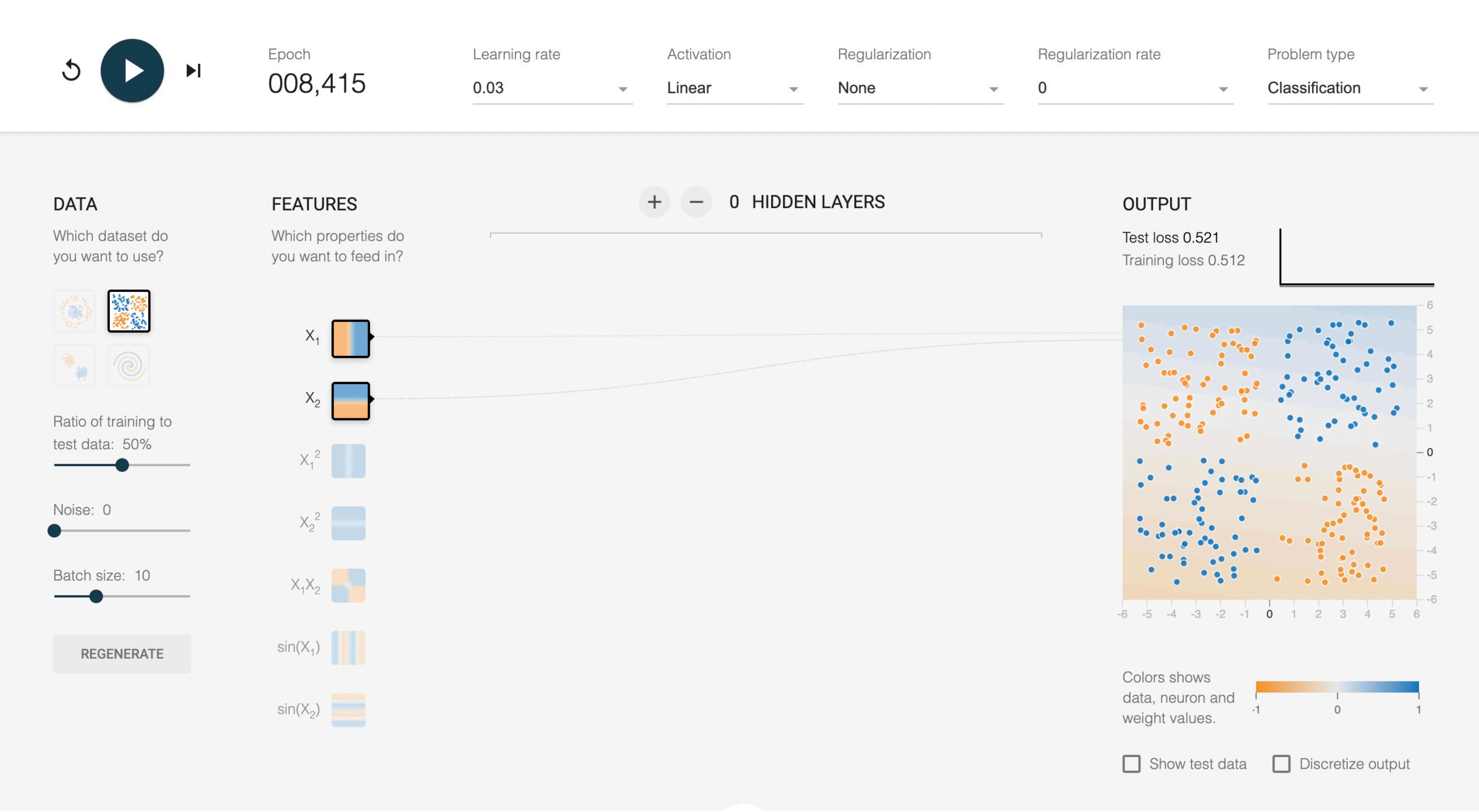
Test loss 1.210

Training loss 1.203

Colors shows data, neuron and weight values.

□ Show test data

□ Discretize output



Epoch
008,415

Learning rate
0.03

Activation
Linear

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

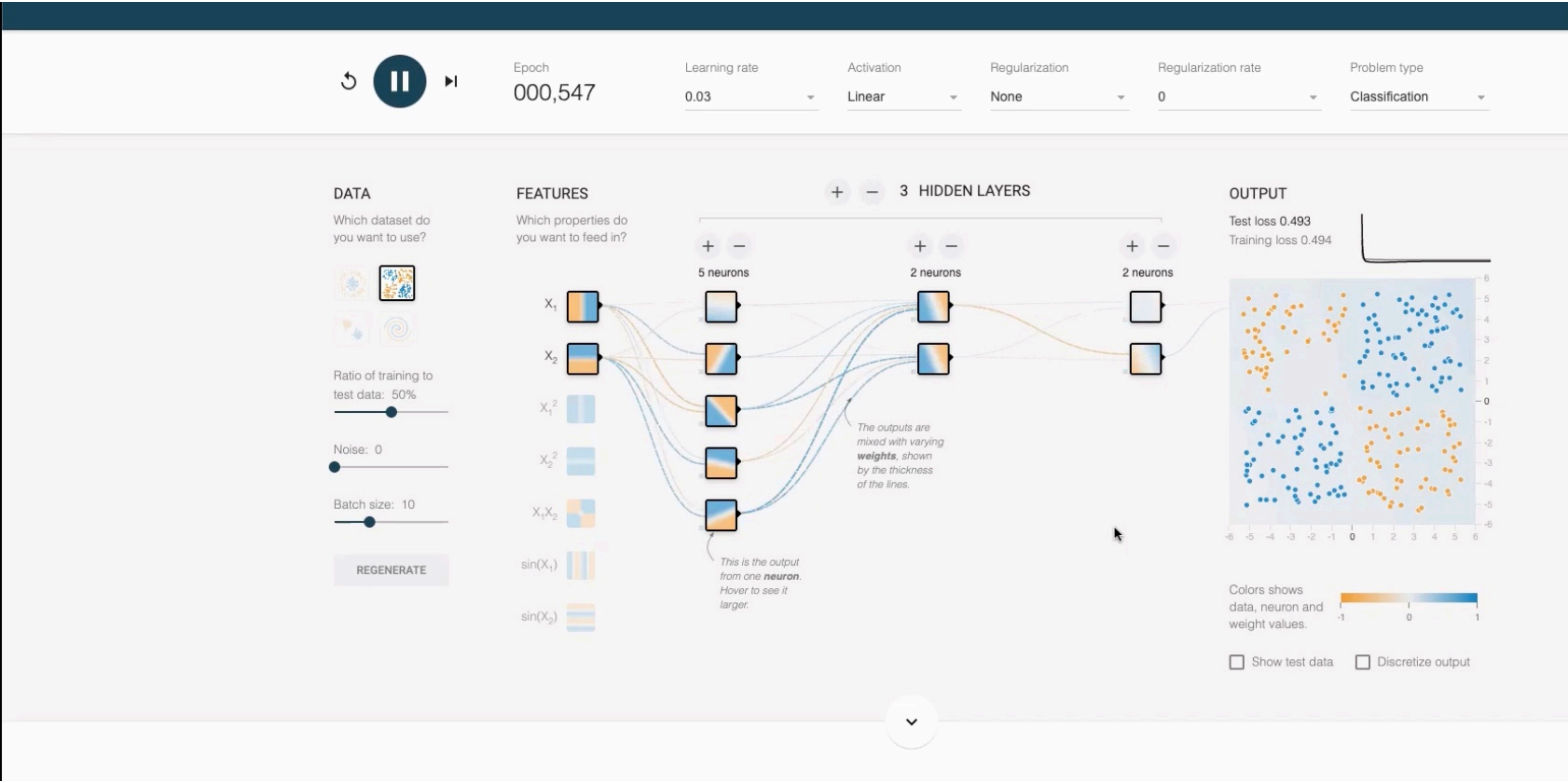
0 HIDDEN LAYERS

OUTPUT

Test loss 0.521
Training loss 0.512

Colors shows data, neuron and weight values.

□ Show test data □ Discretize output



Epoch 000,547

Learning rate 0.03

Activation Linear

Regularization None

Regularization rate 0

Problem type Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

3 HIDDEN LAYERS

5 neurons

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.*

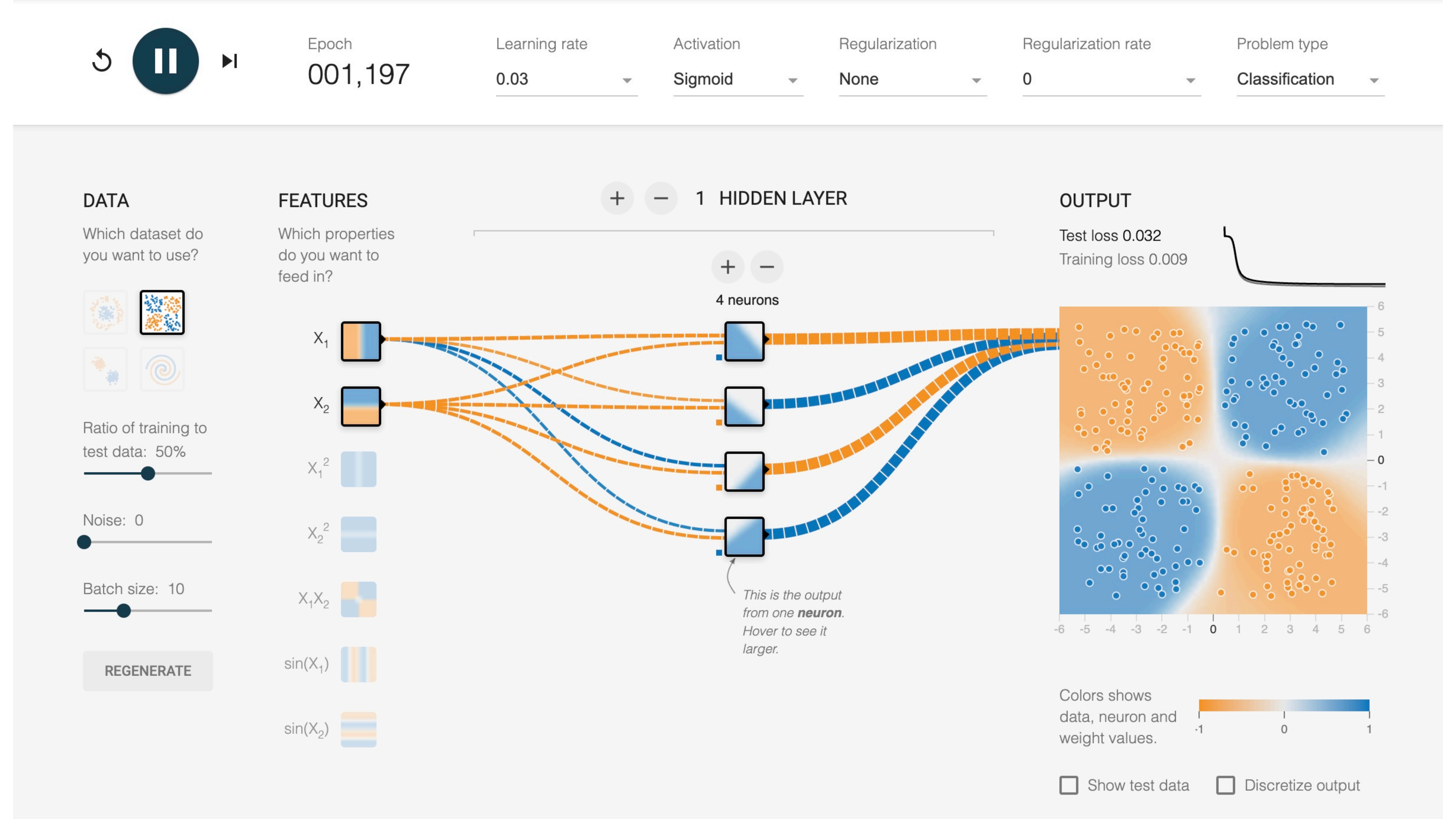
OUTPUT

Test loss 0.493

Training loss 0.494

Colors shows data, neuron and weight values.

□ Show test data □ Discretize output



Epoch 001,197

Learning rate 0.03

Activation Sigmoid

Regularization None

Regularization rate 0

Problem type Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

+ − 1 HIDDEN LAYER

+ −

4 neurons

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

OUTPUT

Test loss 0.032

Training loss 0.009

Colors shows data, neuron and weight values.

□ Show test data □ Discretize output