

START YOUR MACHINE LEARNING JOURNEY FROM TENSORFLOW PLAYGROUND

BIYONG SUN

FEB 18 2020



PUBLIC



SECURE CONNECTIONS
FOR A SMARTER WORLD

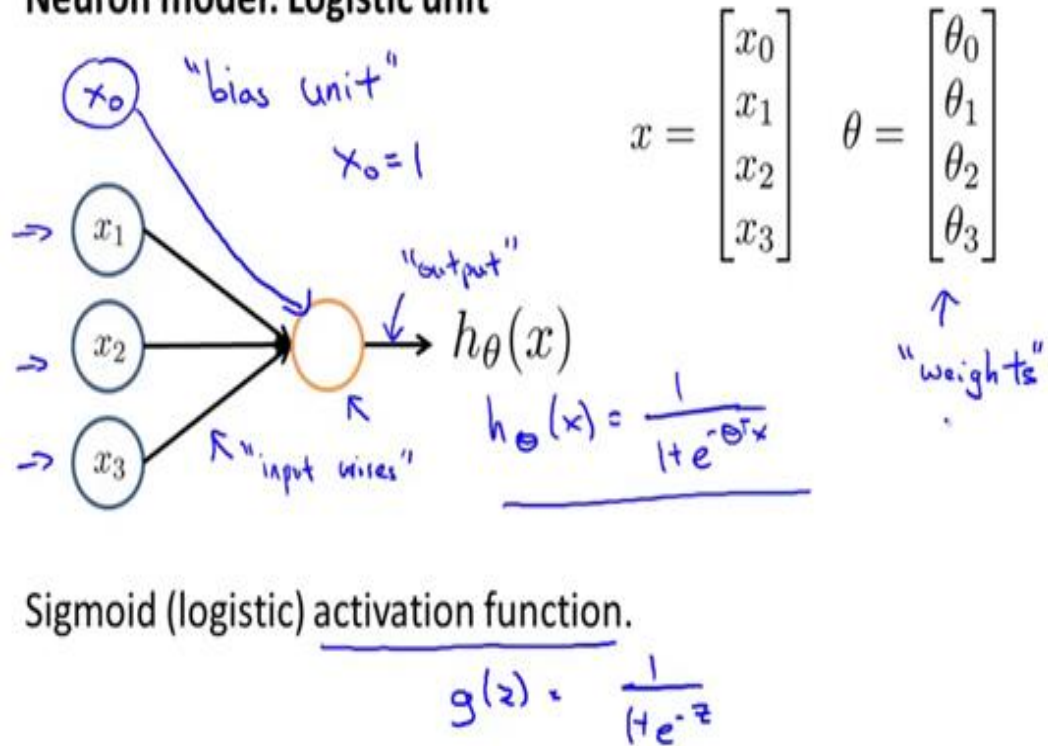
SCOPE

This presentation is to introduce a way to start your study of machine learning.
Purpose here is to start by seeing the Neural Network not the mathematics.

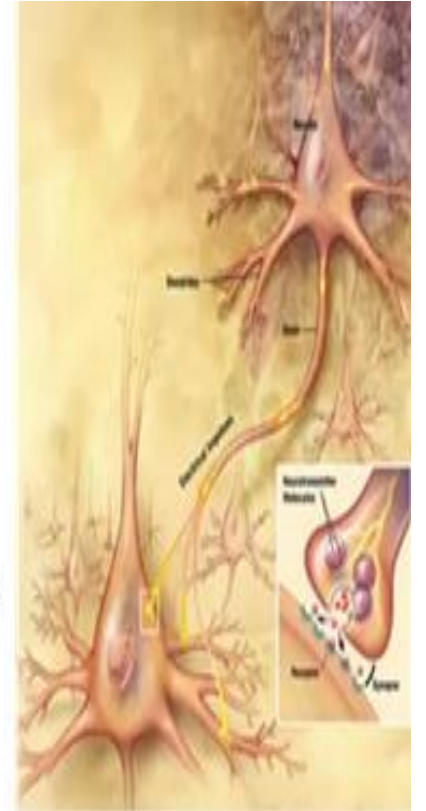
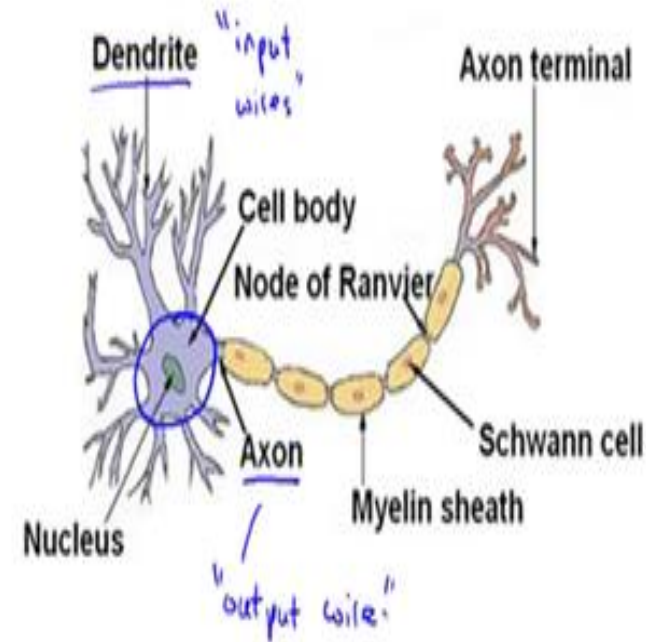
Introduce the architecture, visual view of Neural Network.
No details like Activation ReLU, tanh, sigmoid in this presentation, Regularization L1, L2.
Actually, hope you explore those details by yourselves.

Neuron model:logistic unit

Neuron model: Logistic unit



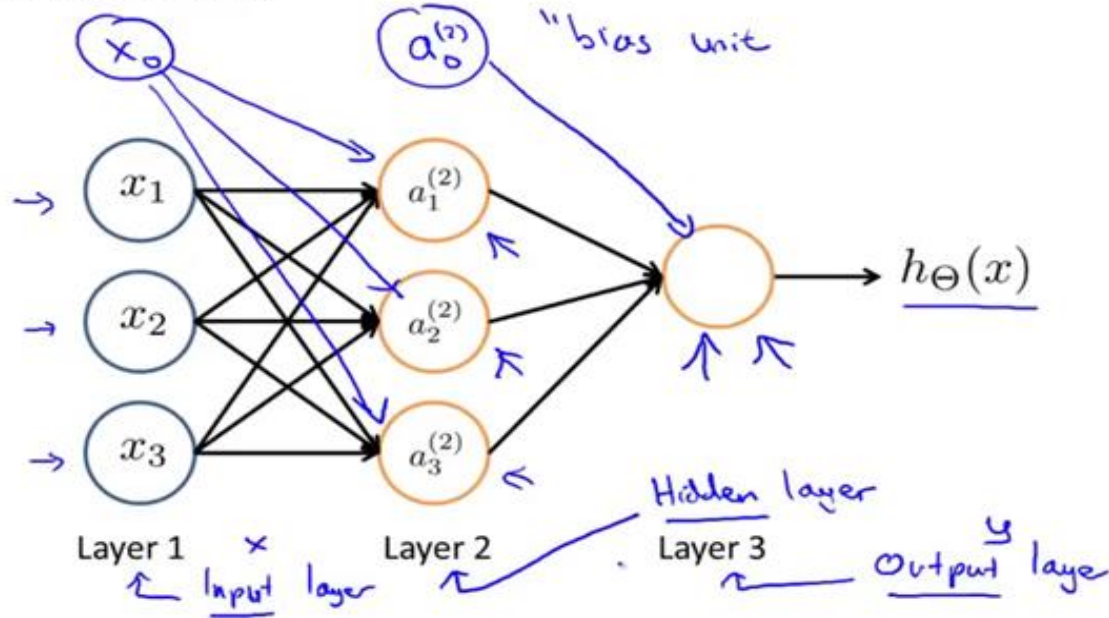
Neuron in the brain



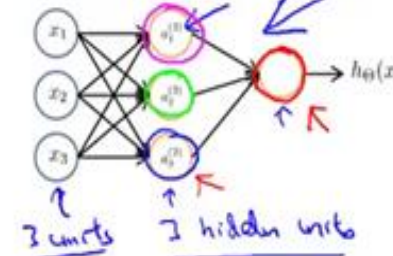
Note: Neuron in the brain activation is non-linear

Neural Network

Neural Network



Neural Network



$\rightarrow a_i^{(j)}$ = "activation" of unit i in layer j
 $\rightarrow \Theta^{(j)}$ = matrix of weights controlling function mapping from layer j to layer $j+1$

$\Theta^{(1)} \in \mathbb{R}^{3 \times 4}$ $h_{\Theta}(x)$

$\rightarrow a_1^{(2)} = g(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3)$

$\rightarrow a_2^{(2)} = g(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3)$

$\rightarrow a_3^{(2)} = g(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3)$

$\rightarrow h_{\Theta}(x) = a_1^{(3)} = g(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)} + \Theta_{13}^{(2)}a_3^{(2)})$

\rightarrow If network has s_j units in layer j , s_{j+1} units in layer $j+1$, then $\Theta^{(j)}$ will be of dimension $\underline{s_{j+1}} \times \underline{(s_j + 1)}$. $s_{j+1} \times (s_j + 1)$

Note: Connections have weights
Neurons are in hidden layer

Tensorflow Playground

<http://playground.tensorflow.org/>



Epoch
000,000

Learning rate
0.03

Activation
Tanh

Regularization
None


Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?

Exclusive or

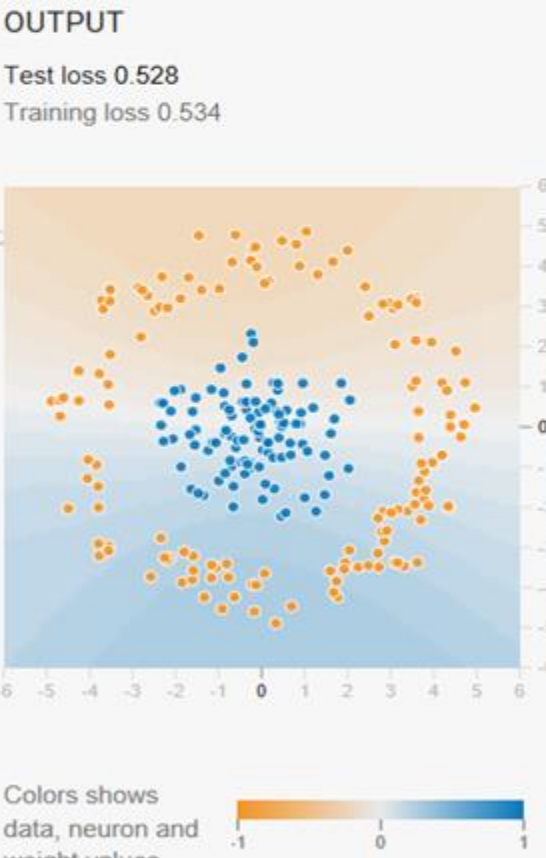
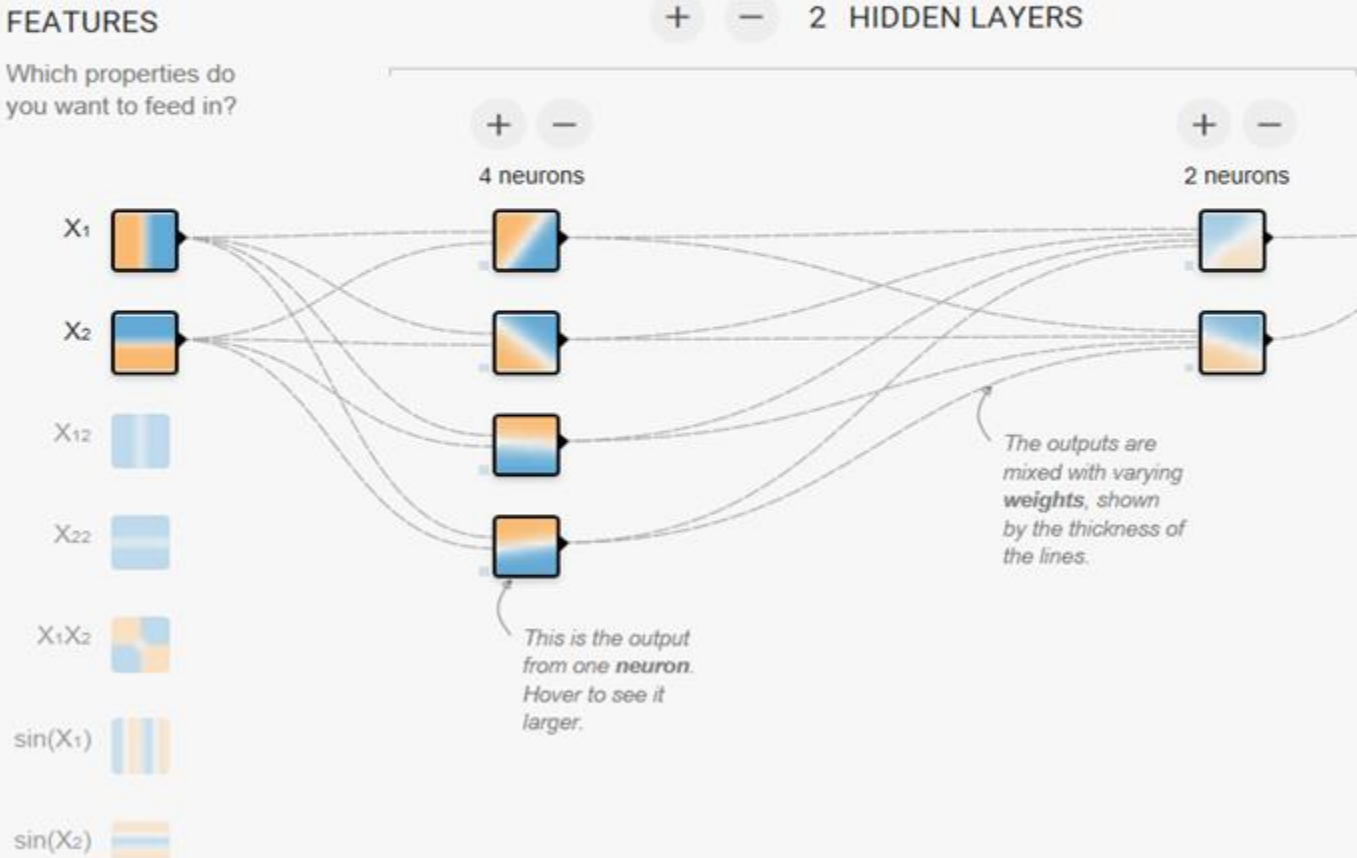


Ratio of training to test data: 50%

Noise: 0

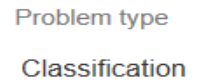
Batch size: 10

REGENERATE



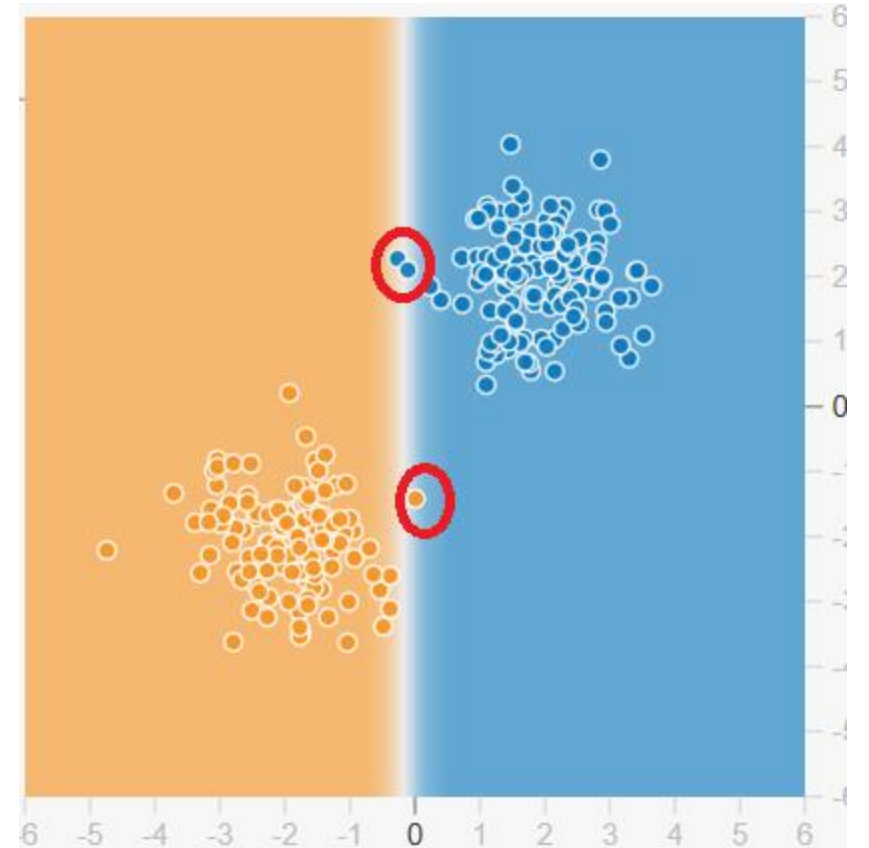
Only X1 for features, no neurons in hidden layer

Only X1 for features, no neurons in hidden layer

☐ Discretize output

1st Game: Gaussian, x1, Linear, Classification(Cont.)

2 blue points and 1 orange point are incorrect.



2nd Game: Gaussian, X1, X2, Linear, Classification (Cont.)

FEATURES

Which properties do you want to feed in?

+

-

0 HIDDEN LAYERS

X₁



X₂



X₁₂



X₂₂



X₁X₂

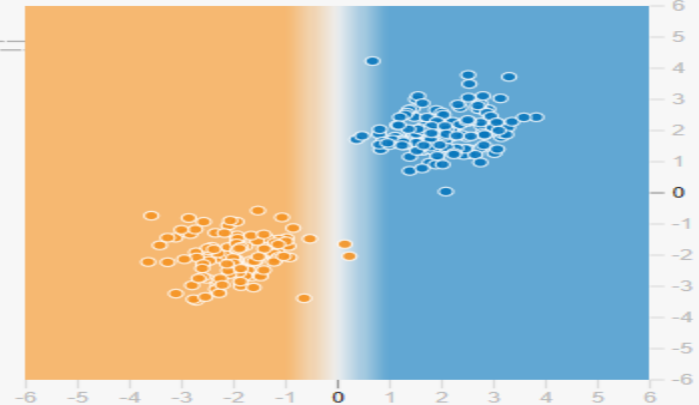


X₁, X₂ work together

OUTPUT

Test loss 0.968

Training loss 0.923



FEATURES

Which properties do you want to feed in?

X₁



X₂



X₁₂



X₂₂



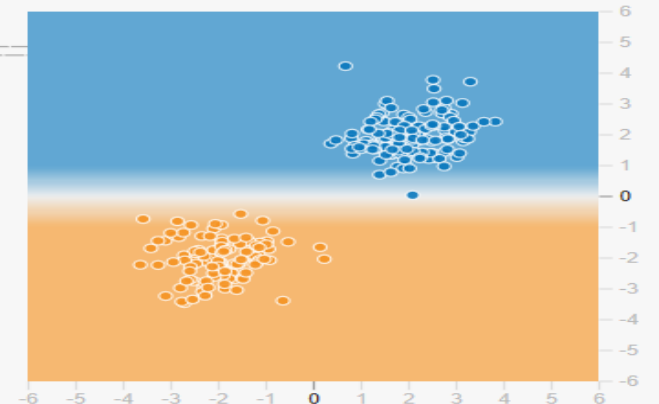
X₁X₂



OUTPUT

Test loss 0.799

Training loss 0.777



2nd Game: Gaussian, X1, X2, Linear, Classification

X1, X2 for features, no neurons in hidden layer

↺

▶

▶

Epoch
000,134

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%

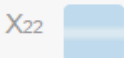
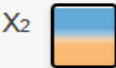
Noise: 0

Batch size: 10

REGENERATE

FEATURES

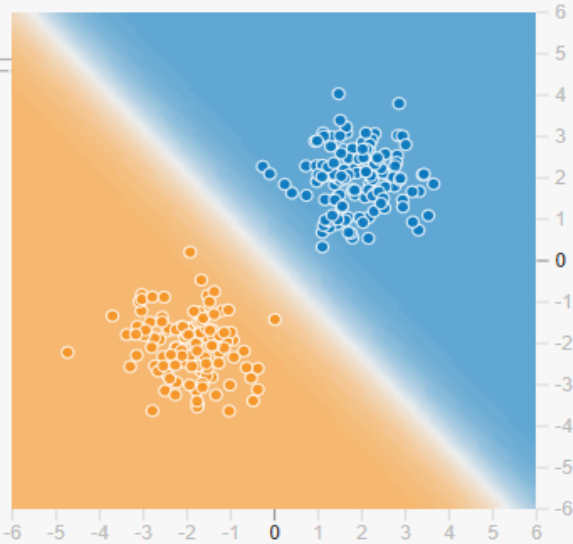
Which properties do you want to feed in?



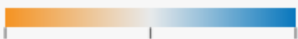
+ - 0 HIDDEN LAYERS

OUTPUT

Test loss 0.000
Training loss 0.000

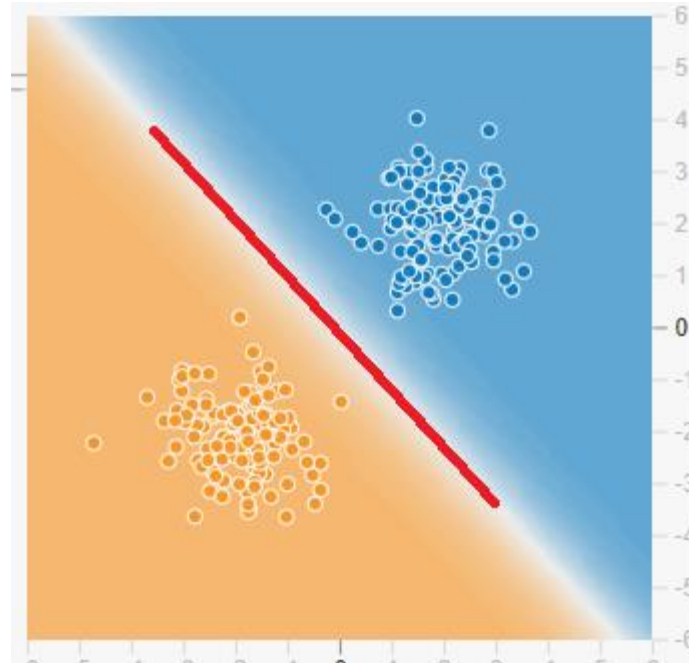


Colors shows
data neuron and



2nd Game: Gaussian, X1, X2, Linear, Classification (Cont.)

X1, X2 for features, the classification could be 100% correct.



3rd Game: Circle, 2 neurons(tanh), Classification



Epoch
000,139

Learning rate
0.03

Activation
Tanh

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

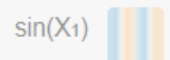
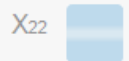
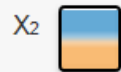
Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?



+ - 1 HIDDEN LAYER

+ -

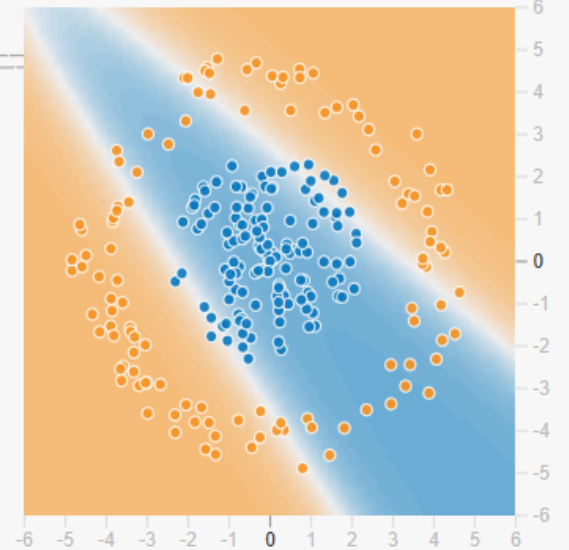
2 neurons



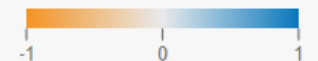
This is the output from one **neuron**.
Hover to see it larger.

OUTPUT

Test loss 0.319
Training loss 0.215



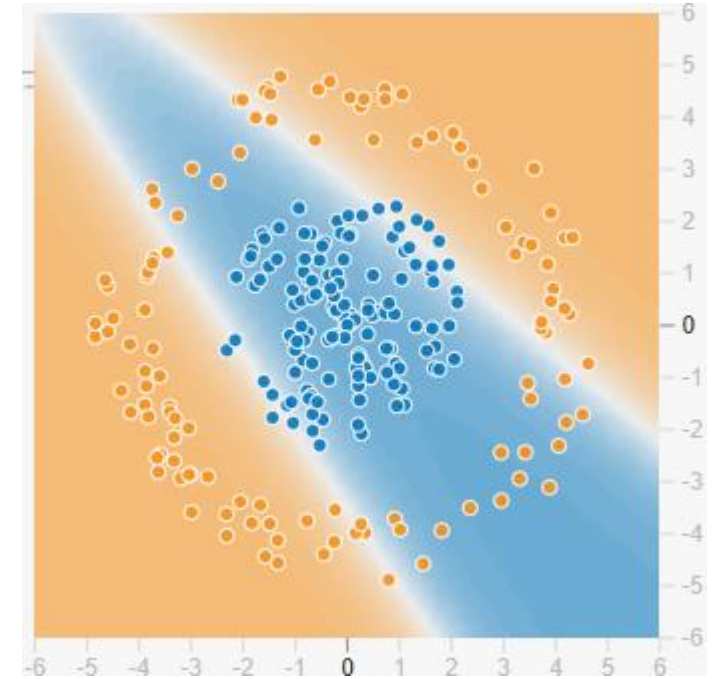
Colors shows
data, neuron and
weight values



3rd Game: Circle, 2 neurons(tanh), Classification(Cont.)

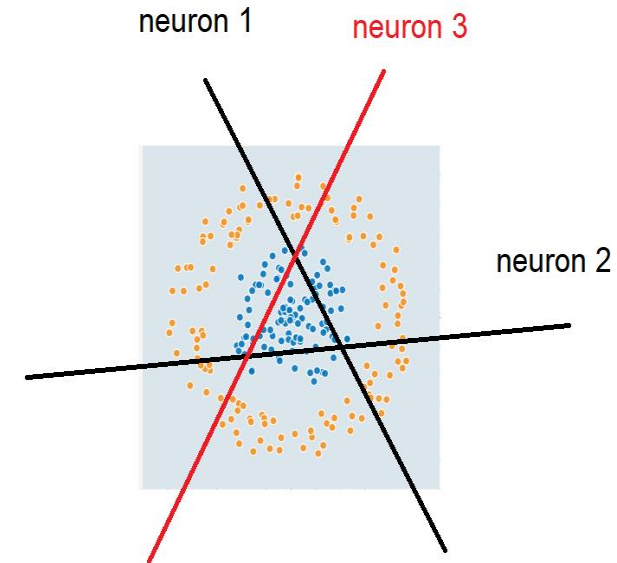
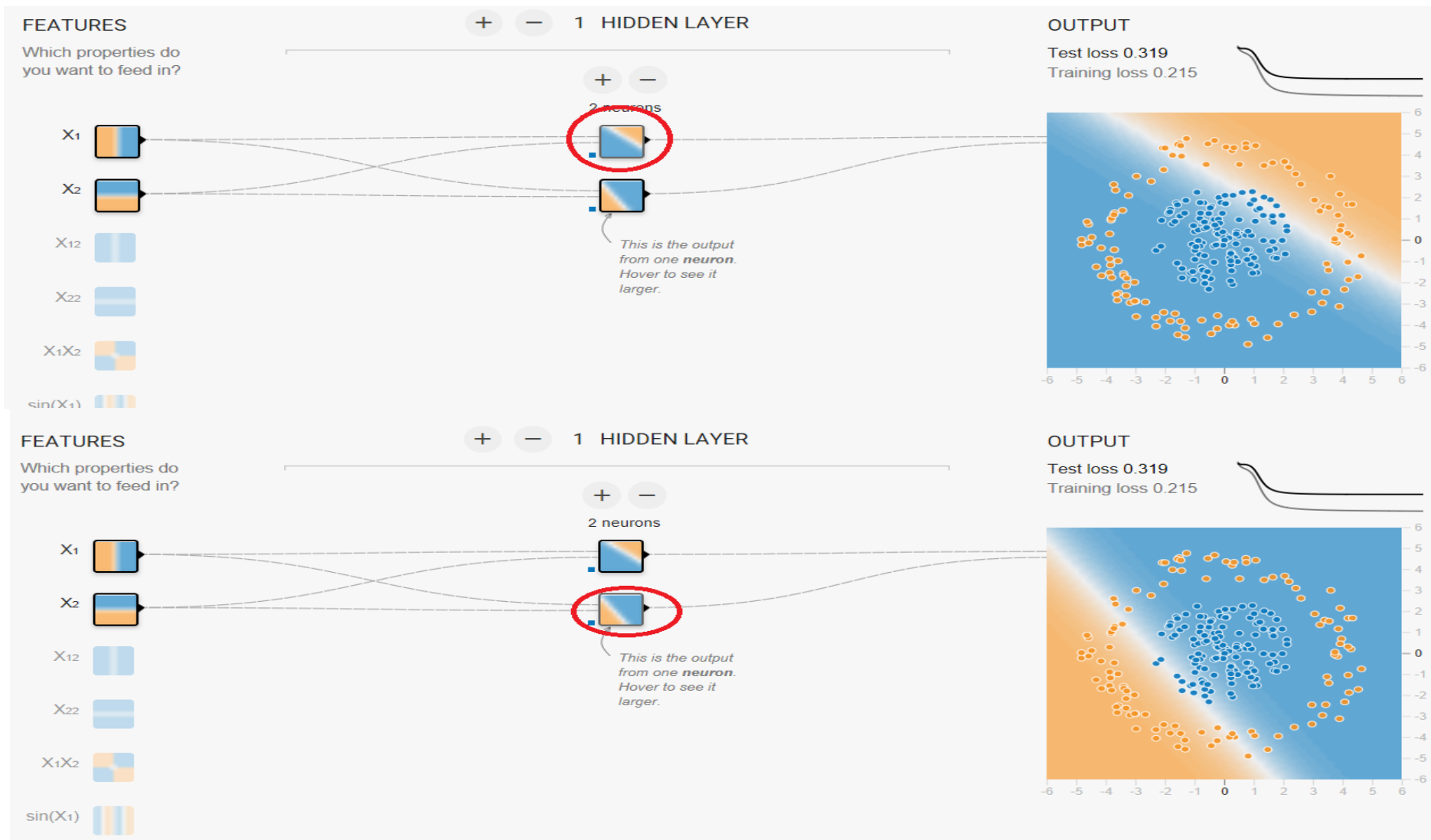
It is very clear. The classification is not good.
We have put 2 neurons(tanh), still cannot get satisfied result.

Why? Let's check each of the neuron.



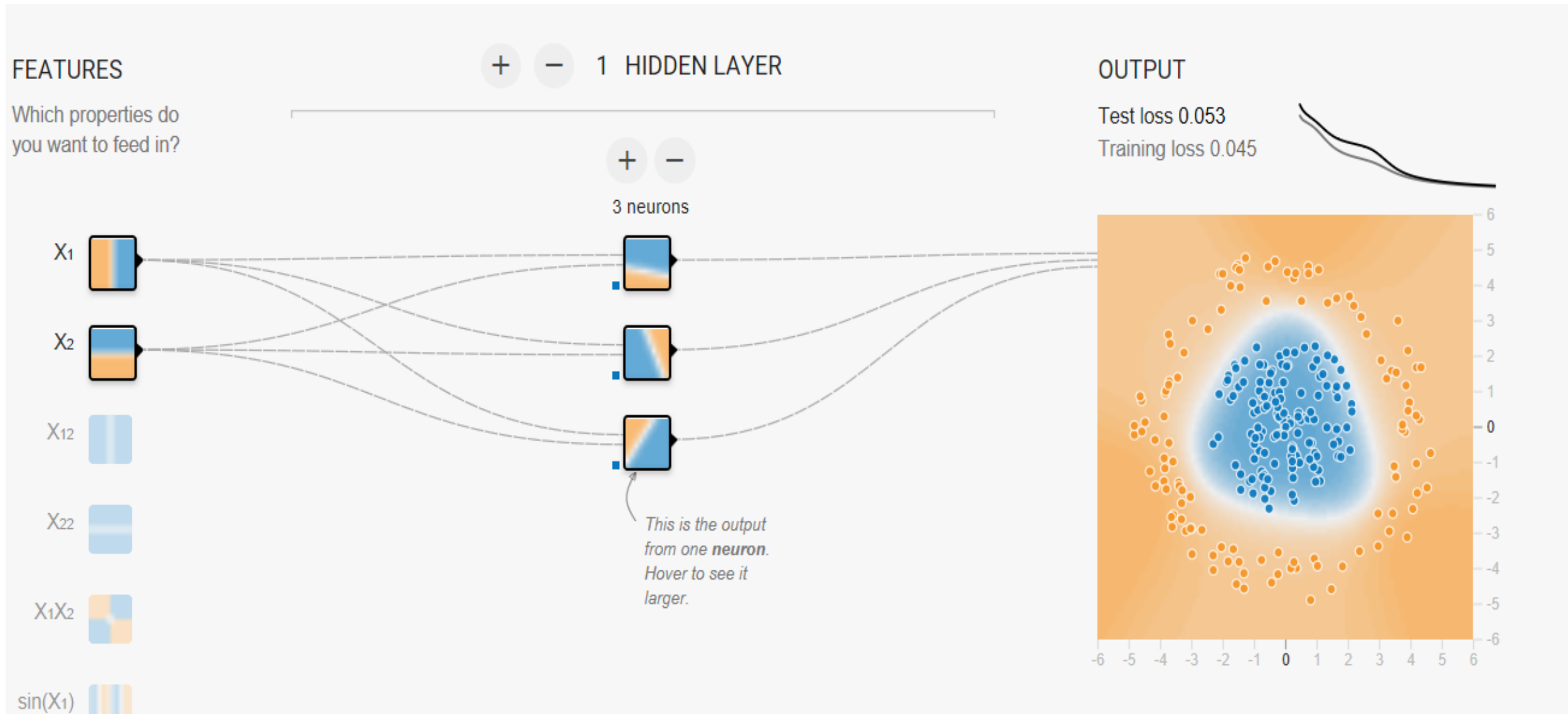
3rd Game: Circle, 2 neurons(tanh), Classification(Cont.)

By checking the each one of the neuron. 2 neurons are not enough. Let's add one more neuron to do again.

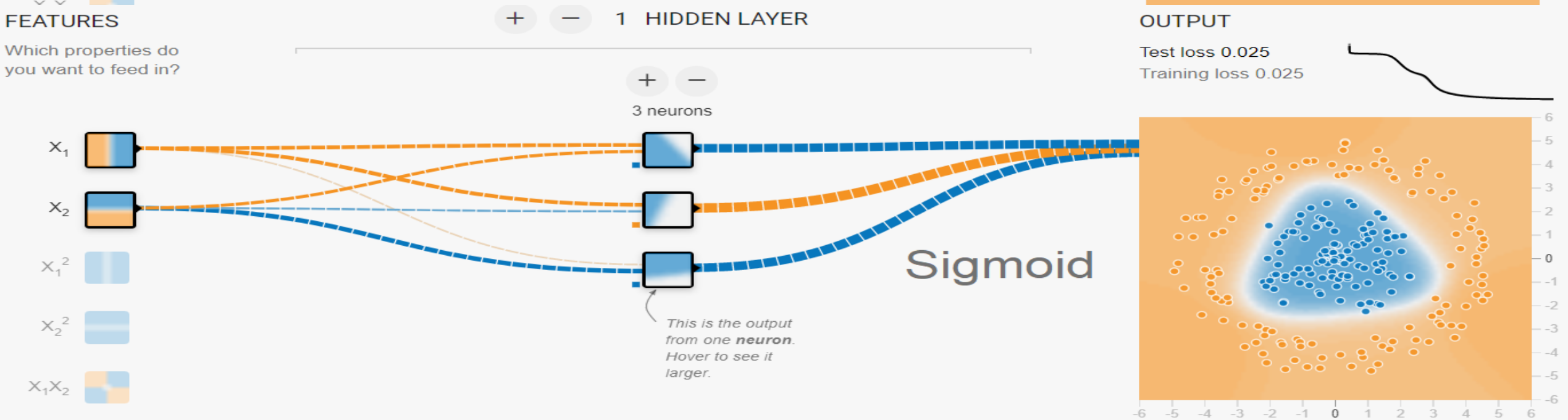
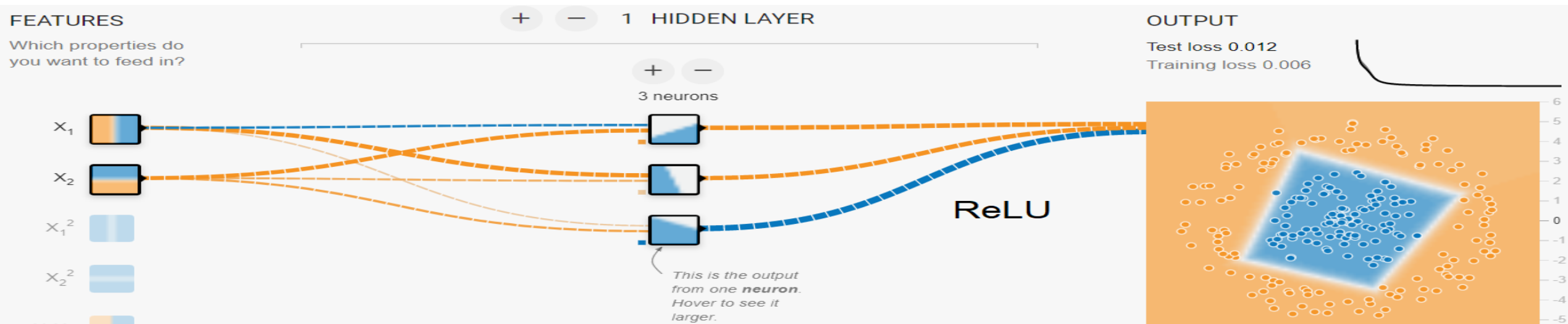


3rd Game: Circle, 2 neurons(tanh), Classification(Cont.)

3 neurons working together.



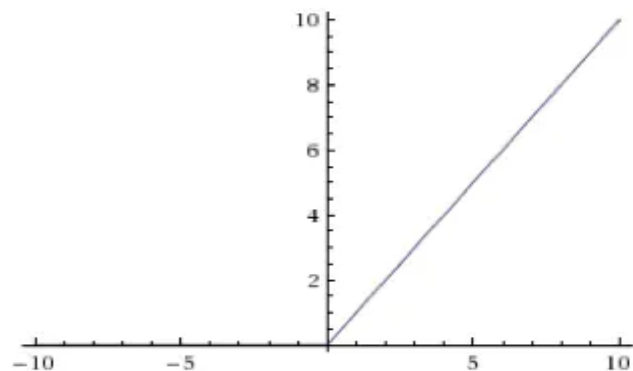
4rd Game: ReLu vs Sigmoid



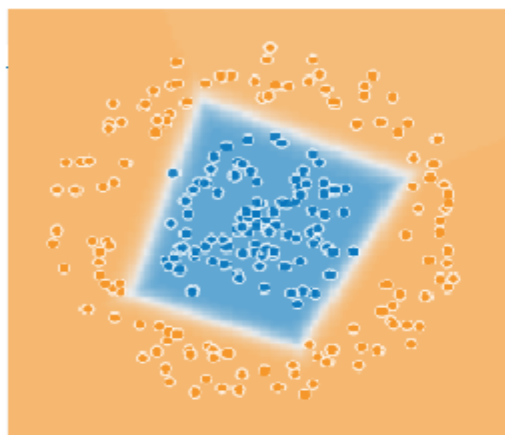
4rd Game: ReLu vs Sigmoid (Cont.)

$$f(x) = \max(0, x)$$

ReLU

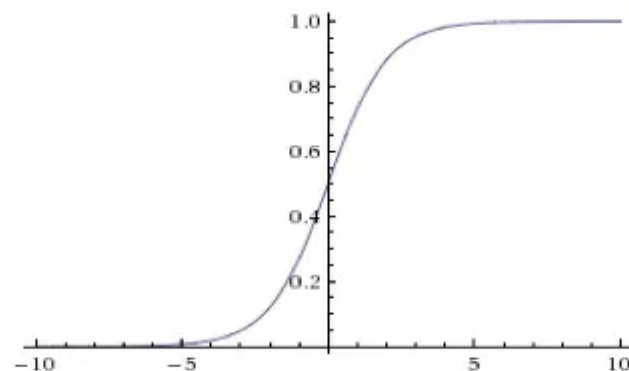


Output with ReLU

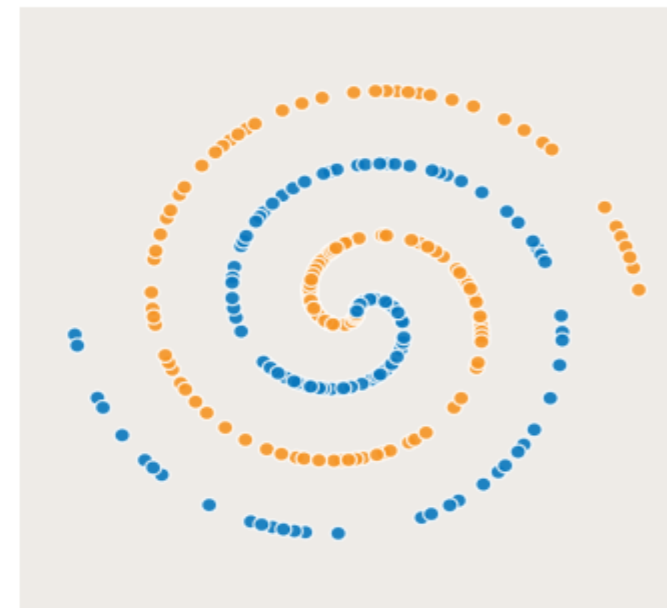
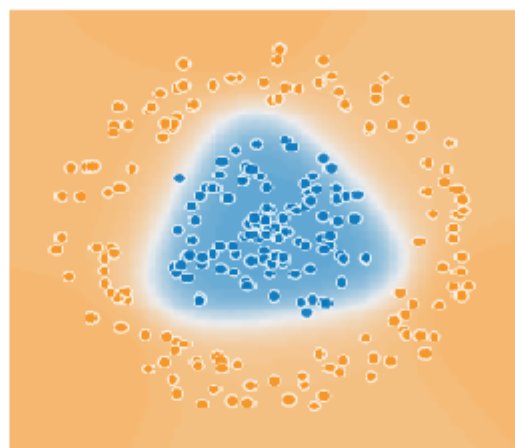


$$\sigma(x) = 1/(1 + e^{-x})$$

Sigmoid



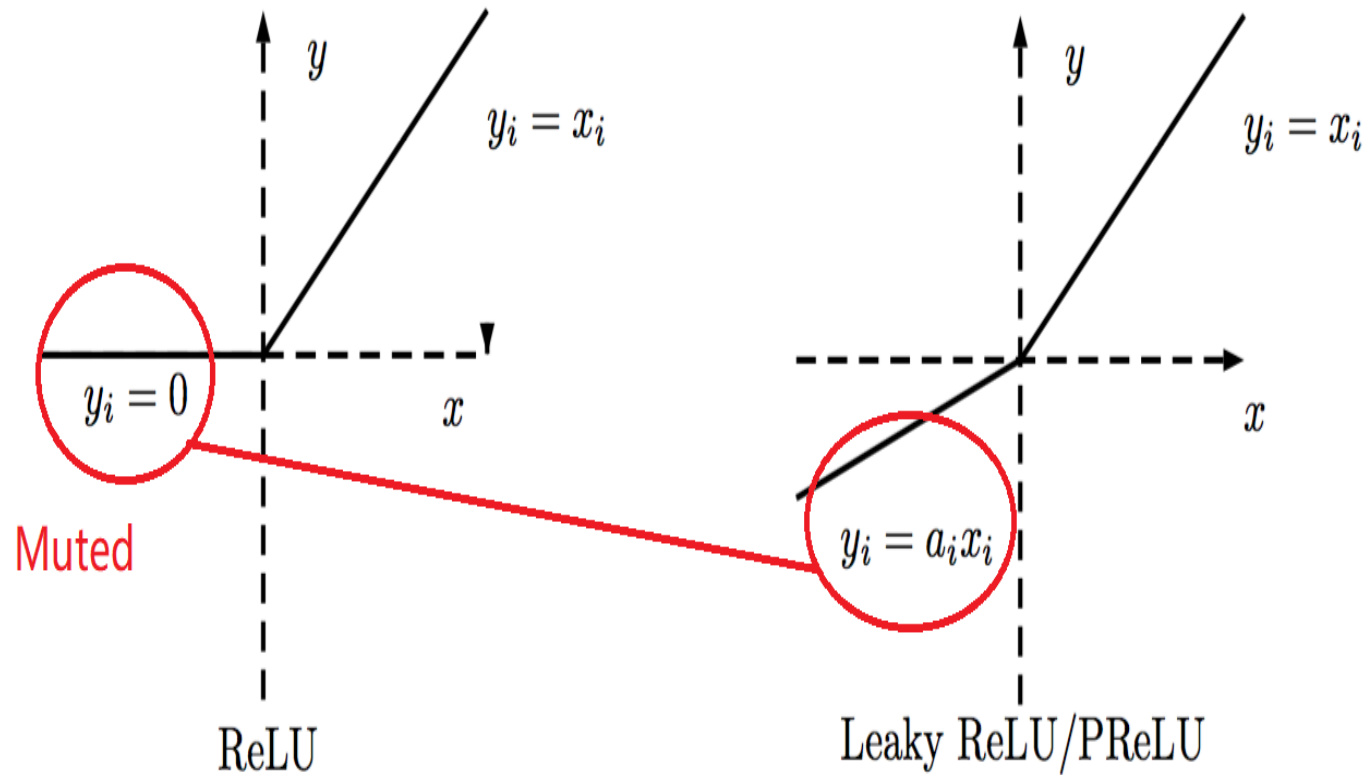
Output with Sigmoid



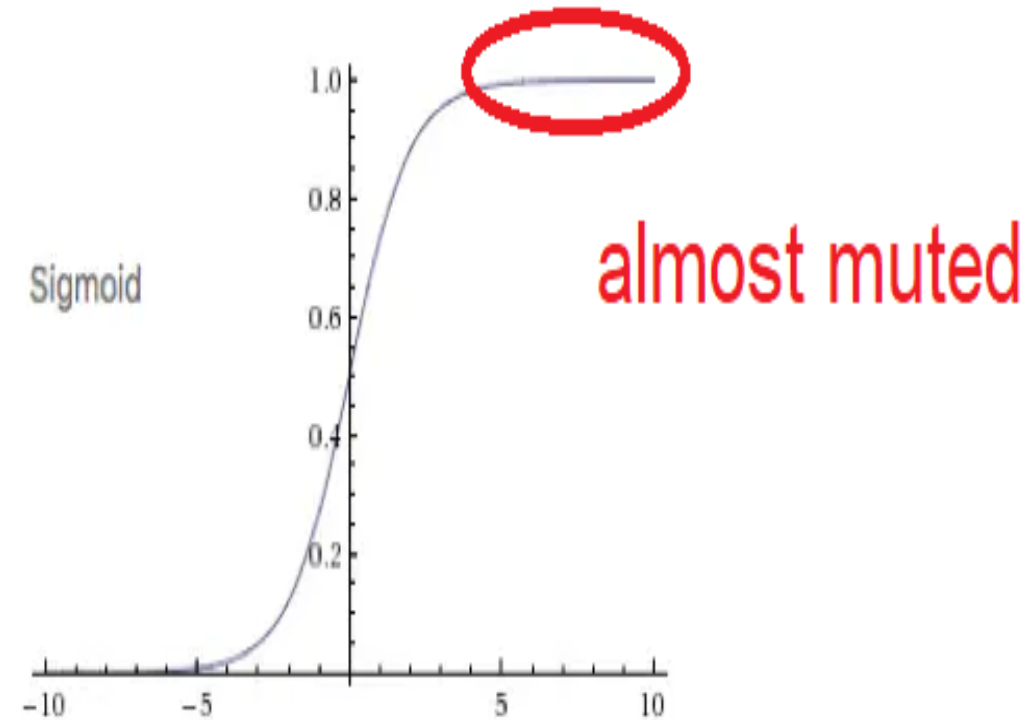
Try to think about **Spiral**

Please pay attention to the **Shape**

4rd Game: ReLu vs Sigmoid (Cont.)

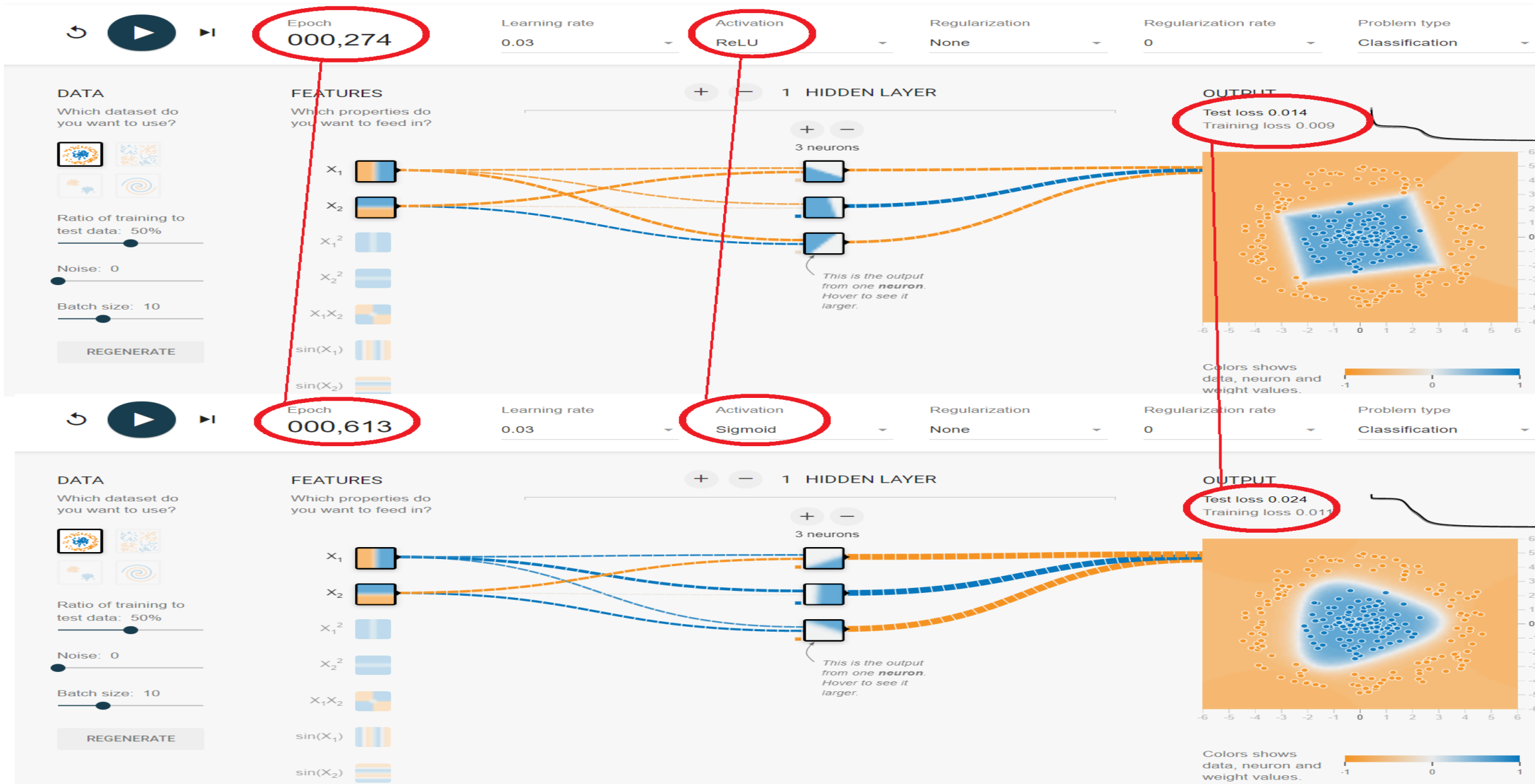


Leaky ReLU($a = 0.01$)



Like to say muted ranther than dead neuron

5th Game: ReLu vs Sigmoid Speed



6th Game: Watching ReLU, tanh, Sigmoid in neurons



Epoch
000,000

Learning rate
0.03

Activation

Linear

ReLU

Tanh

Sigmoid

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

FEATURES

Which properties do you want to feed in?

X_1 X_2 X_1^2 X_2^2 X_1X_2

3 neurons



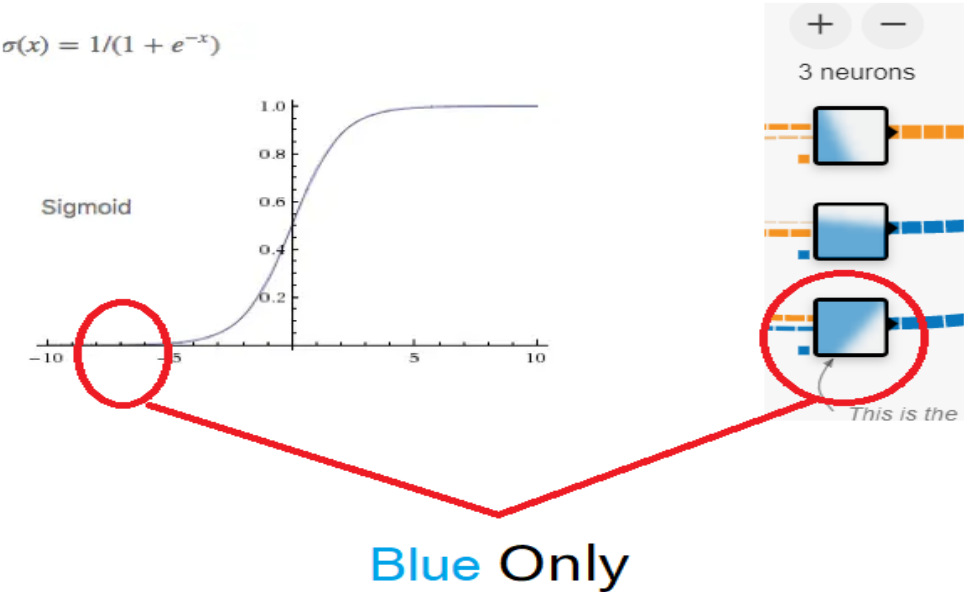
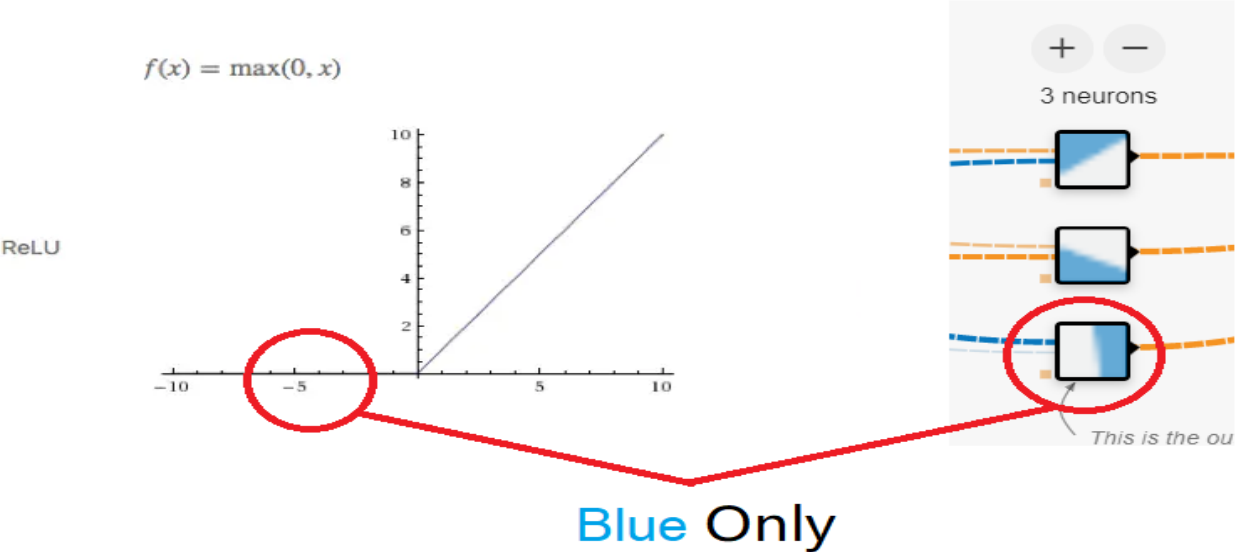
This is the output from one **neuron**.
Hover to see it larger.

OUTPUT

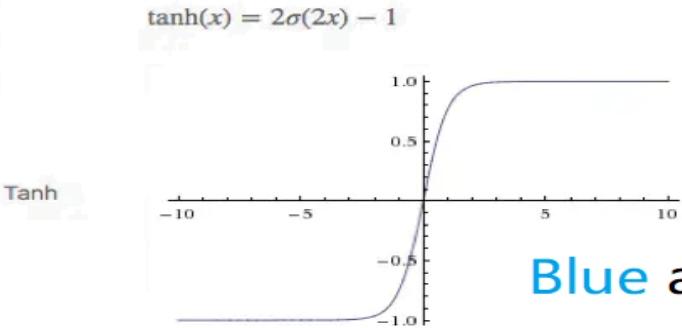
Test loss 0.656
Training loss 0.692



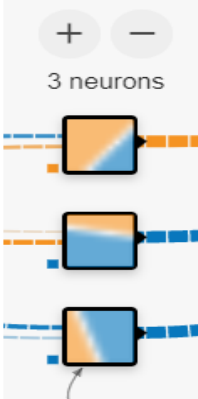
6th Game: Watching ReLU, tanh, Sigmoid in neurons(Cont.)



Linear 50% Blue
50% Orange



Blue and Orange
Not 50% to 50%



7th Game: Spiral



Epoch
000,000

Learning rate

0.03

Activation

Tanh

Regularization

L2

Regularization rate

0.03

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?

X_1



X_2



X_1^2



X_2^2



$X_1 X_2$



$\sin(X_1)$



$\sin(X_2)$



+ - 2 HIDDEN LAYERS

+ -

7 neurons

+ -

2 neurons

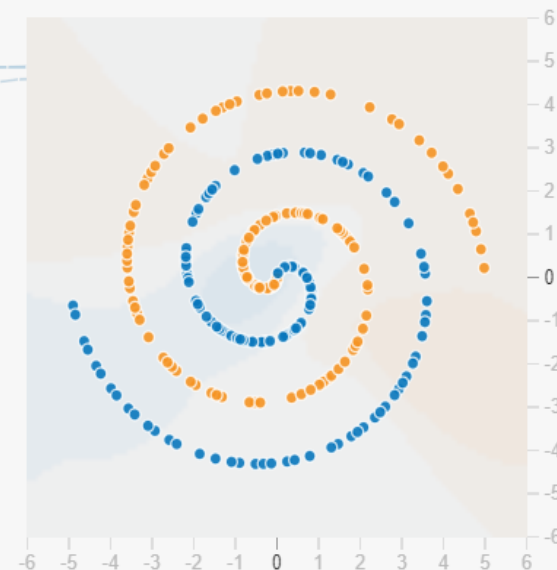
Click anywhere to edit.
Weight is 0.43.

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

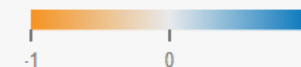
OUTPUT

Test loss 0.503

Training loss 0.506

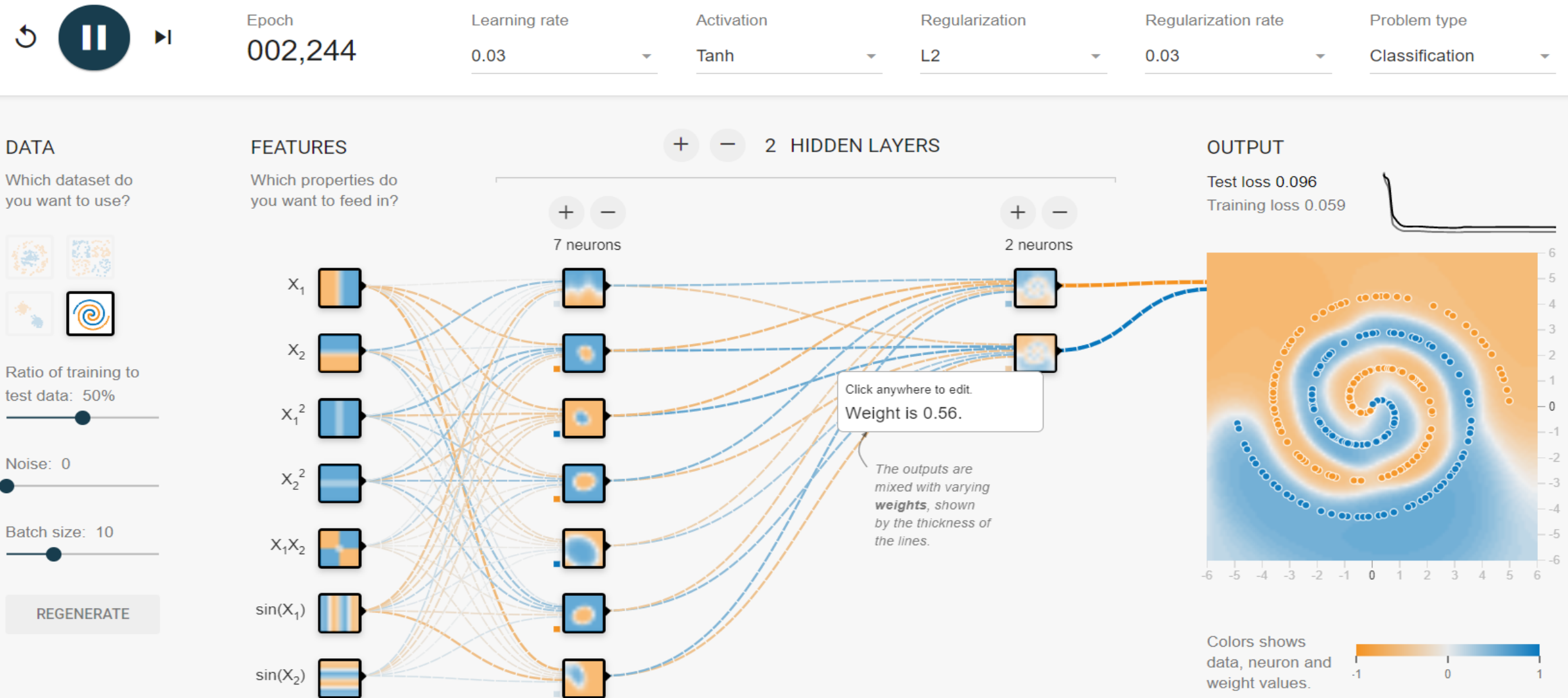


Colors shows data, neuron and weight values.

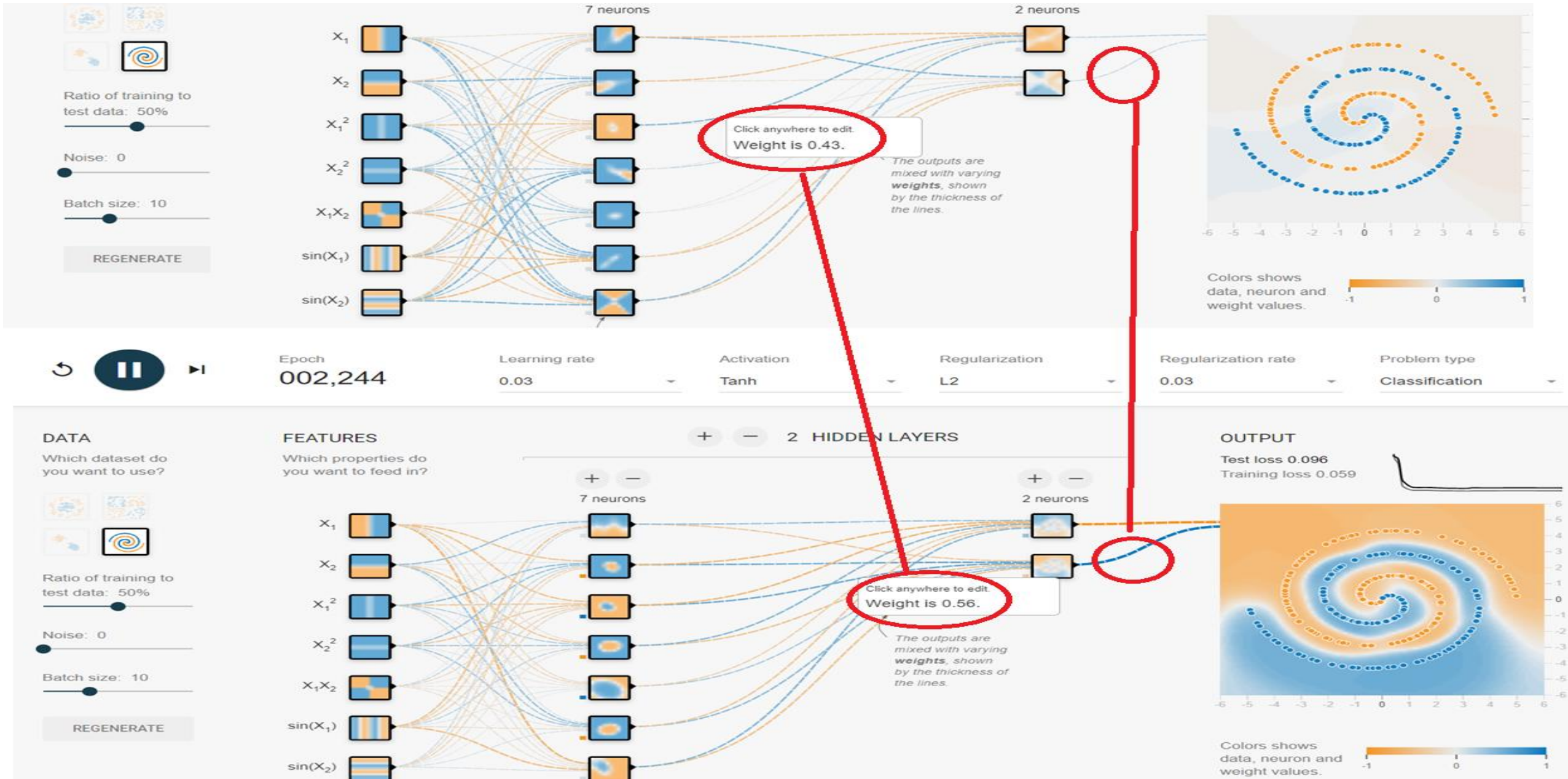


7th Game: Spiral (Cont.)

With the training, the weight changing.



7th Game: Spiral (Cont.)



8th Game: Overfitting & Regularization



Epoch
000,000

Learning rate
0.03

Activation
ReLU

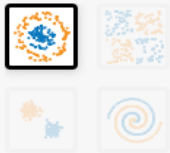
Regularization
None
L1
L2

Regularization rate
0.1

Problem type
Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%



Noise: 50



Batch size Max Noise 50



REGENERATE

FEATURES

Which properties do you want to feed in?

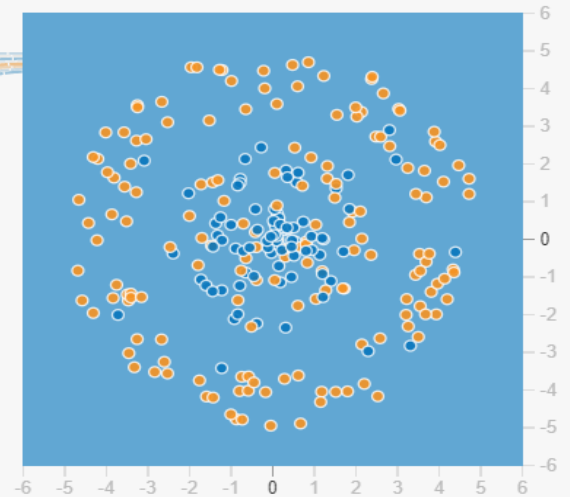
X_1
 X_2
 X_1^2
 X_2^2
 $X_1 X_2$
 $\sin(X_1)$
 $\sin(X_2)$

+ - 1 HIDDEN LAYER

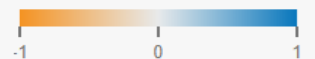
+ -
8 neurons

OUTPUT

Test loss 1.177
Training loss 1.145



Colors shows data, neuron and weight values

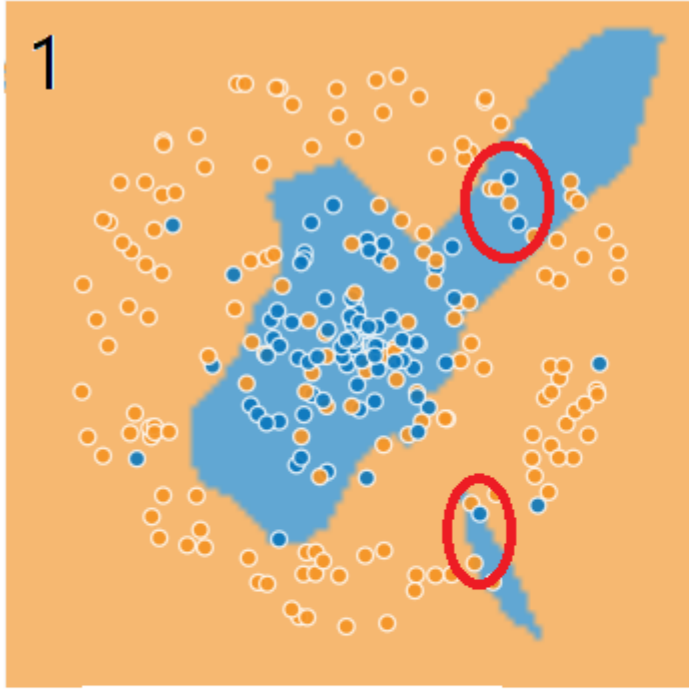


Use Logistic Regression to see it clear

☒ Discretize output

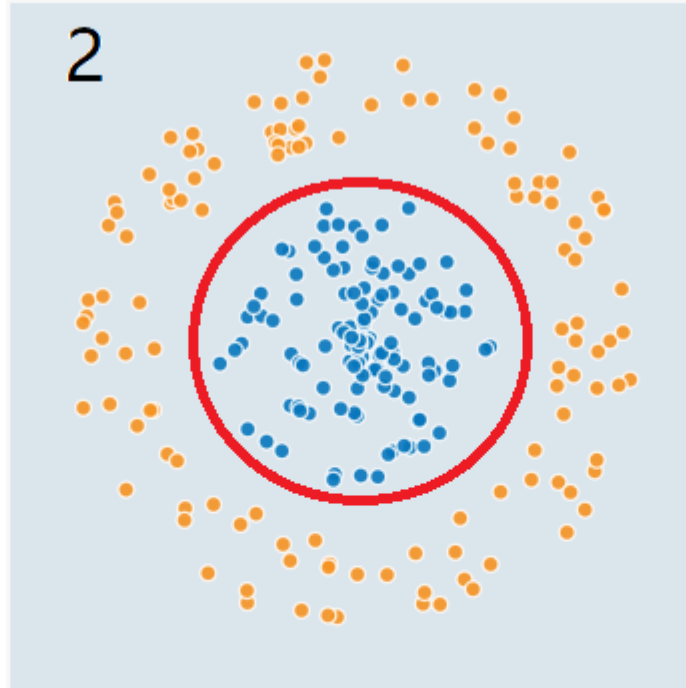
8th Game: Overfitting & Regularization (Cont.)

Regularization: None



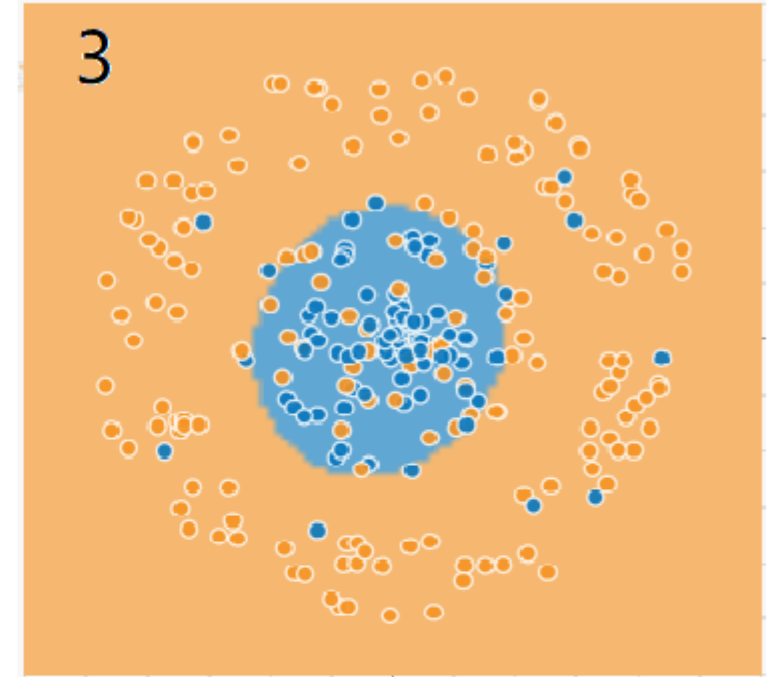
Overfitting
Too "restricted"
Want to get every
Blue Point

no noise



Ideal result
Our wish

Regularization: L2, Rate: 0.1



"loose"
#3 result is close to #2
#3 is better than #1



SECURE CONNECTIONS
FOR A SMARTER WORLD