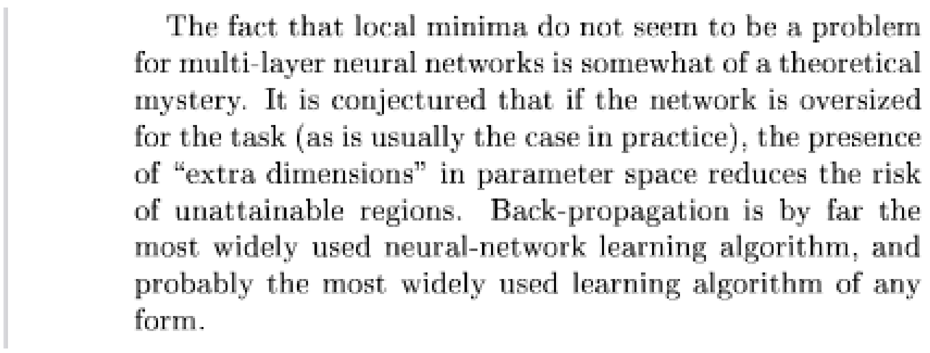
Experiments on Neural Network Playground

Last amended: 24th Jan, 2021
My folder: C:\Users\Administrator\OneDrive\Documents\neural_network
Ref: https://playground.tensorflow.org
Aurelien Book: Pages 327/328

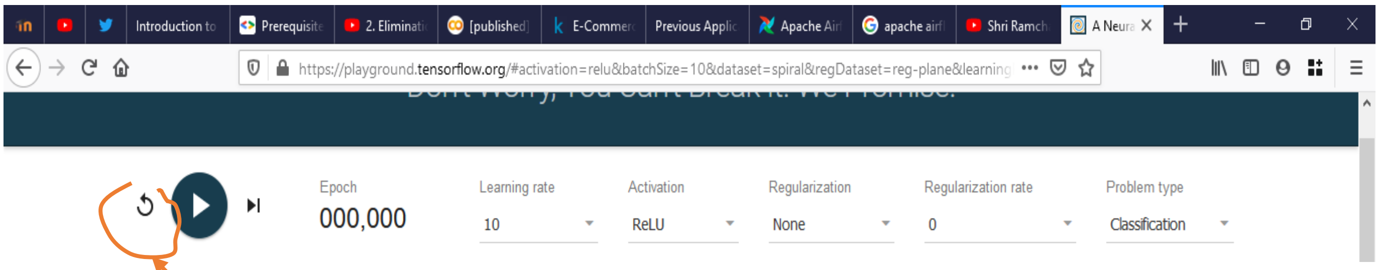
Contents

Use Reset switch

1. Before any experiment, after setting the configurations, click **Reset** switch to reset weights and biases.
2. The moment you hit ***Regenerate*** button, new data will be generated and you may get very different results. **SO WHEN YOU EXPERIMENT, IN BETWEEN DO NOT GENERATE FRESH DATA**. Decide what kind of data you want and then experiment.
3. Hovering over any neuron, your will see the kind of pattern it is able to distinguish. Generally, neurons in the Ist hidden layer, all have linear distinguishing patterns.
4. Read these lines from Page 4 of this article:



> The fact that local minima do not seem to be a problem for multi-layer neural networks is somewhat of a theoretical mystery. It is conjectured that if the network is oversized for the task (as is usually the case in practice), the presence of "extra dimensions" in parameter space reduces the risk of unattainable regions. Back-propagation is by far the most widely used neural-network learning algorithm, and probably the most widely used learning algorithm of any form.


https://playground.tensorflow.org/#activation=relu&batchSize=10&dataset=spiral&regDataset=reg-plane&learning

Epoch 000,000

Learning rate: 10

Activation: ReLU

Regularization: None

Regularization rate: 0

Problem type: Classification

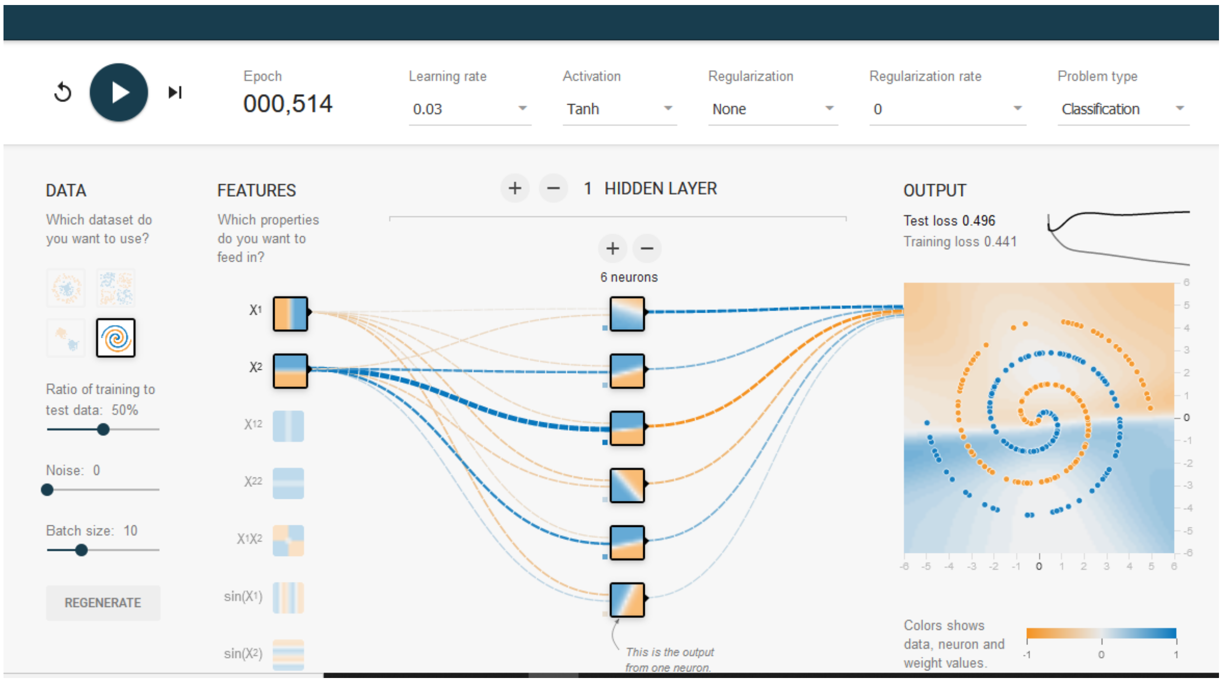
*Figure 1: Always click **Reset** before starting another experiment. AND before you start experiments, click REGENERATE button just once and DO not touch it again. We do not want data to change between experiments.*

Experiment 1:

Change Activations & see effects

|

Configuration		Results	
Nos of hidden layers:	1	Test loss:	0.**496**
No of neurons in hidden layer:	6	Training loss:	0.441
Learning rate	0.01		
Activation:	**'tanh'**		
Epochs:	500		


Epoch 000,514
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

FEATURES
Which properties do you want to feed in?

1 HIDDEN LAYER
6 neurons

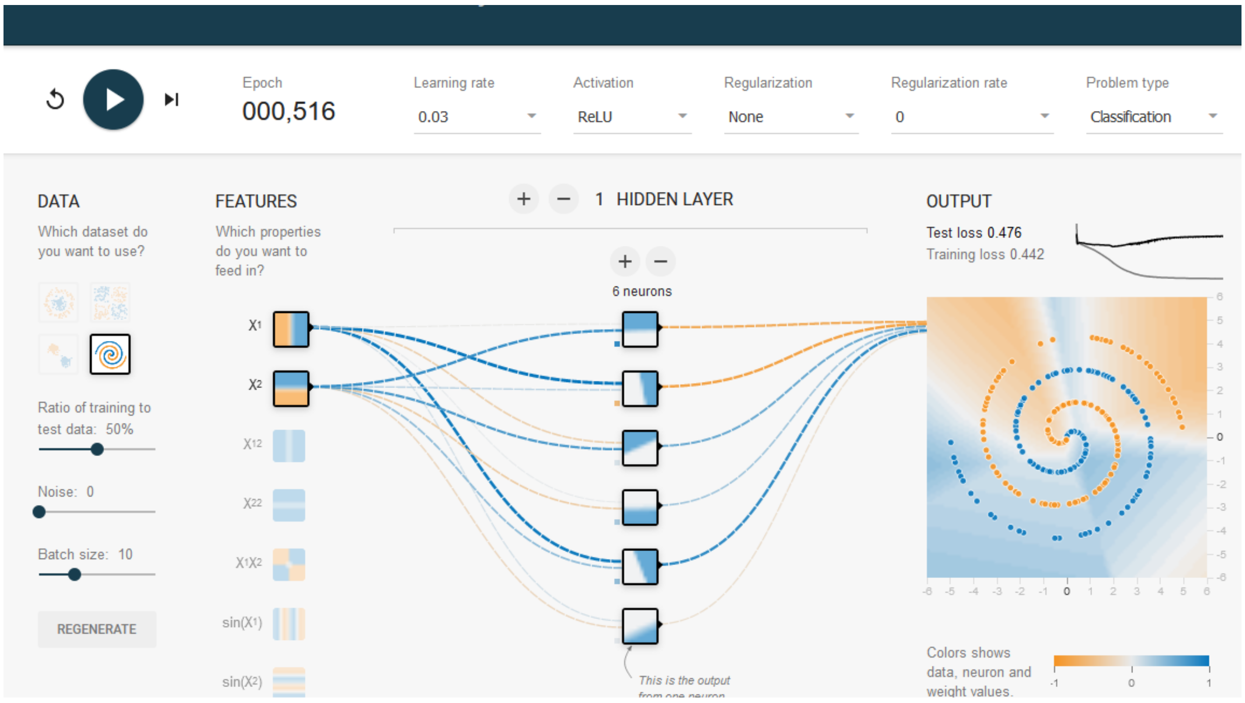
OUTPUT
Test loss 0.496
Training loss 0.441

REGENERATE
X1
X2
X1²
X2²
X1X2
sin(X1)
sin(X2)
This is the output from one neuron.
Colors shows data, neuron and weight values.
-1
0
1

Figure 2: Activation is tanh

II

Configuration		Results	
Nos of hidden layers:	1	Test loss:	0.**476**
No of neurons in hidden layer:	6	Training loss:	0.442
Learning rate	0.01		
Activation:	**'relu'**		
Epochs:	500		


Epoch 000,516
Learning rate 0.03
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: 0
Batch size: 10
REGENERATE

FEATURES
Which properties do you want to feed in?

1 HIDDEN LAYER
6 neurons

OUTPUT
Test loss 0.476
Training loss 0.442

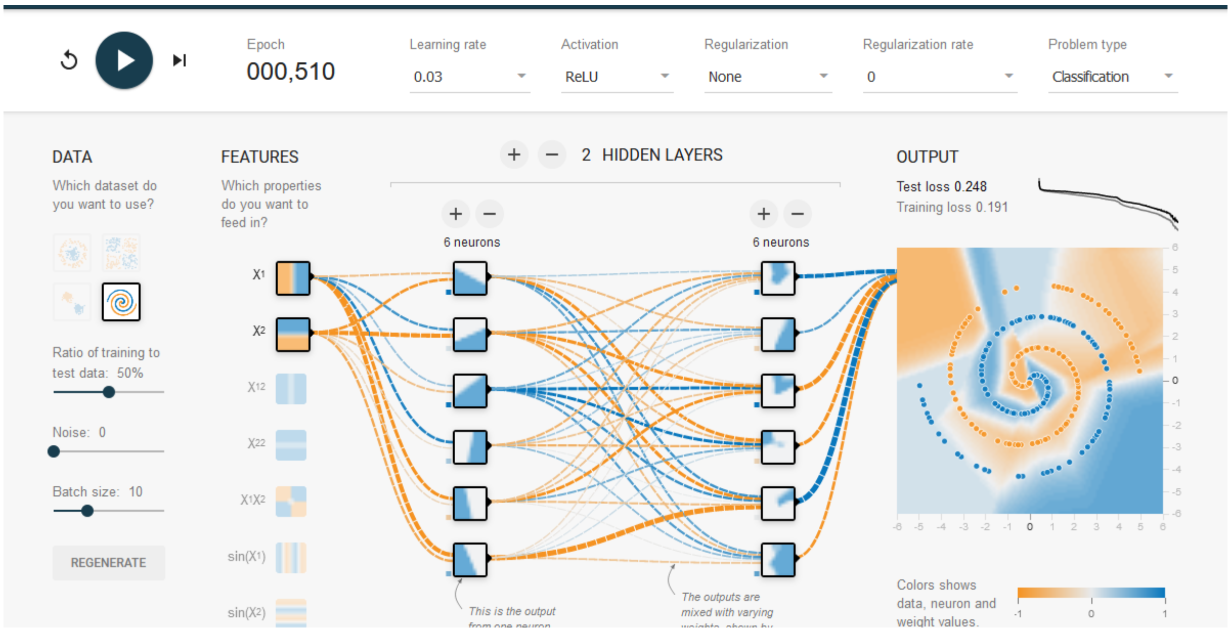
Colors shows data, neuron and weight values.

Figure 3: Activation is relu

Experiment 2

Change no of hidden layers & see effects

Configuration		Results	
Nos of hidden layers:	**2**	Test loss:	0.248
No of neurons in hidden layer:	6	Training loss:	0.191
Learning rate	0.01		
Activation:	'relu'		
Epochs:	500		



Epoch 000,510

Learning rate 0.03

Activation ReLU

Regularization None

Regularization rate 0

Problem type Classification

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

2 HIDDEN LAYERS

6 neurons

6 neurons

Test loss 0.248

Training loss 0.191

DATA
Which dataset do you want to use?
REGENERATE
FEATURES
Which properties do you want to feed in?
X1
X2
X12
X22
X1X2
sin(X1)
sin(X2)
This is the output from one neuron
The outputs are mixed with varying weights, shown by
OUTPUT
Colors shows data, neuron and weight values.
-1
0
1

Figure 4: Two hidden layers with relu. Compare losses with those in Figure 3.

Experiment 3

Change no of hidden layers & see effects

Configuration		Results	
Nos of hidden layers:	**3**	Test loss:	0.022
No of neurons in hidden layer:	6	Training loss:	0.004
Learning rate:	0.01		
Activation:	'relu'		
Epochs:	500		
Batch Size	10		

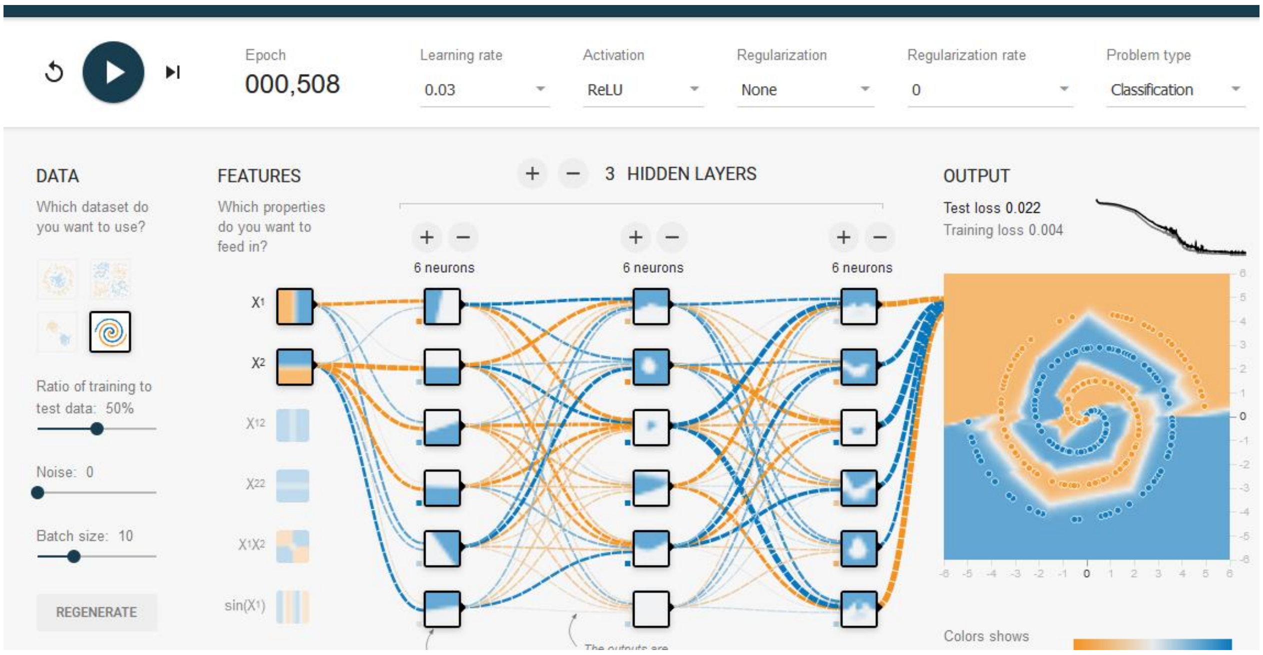
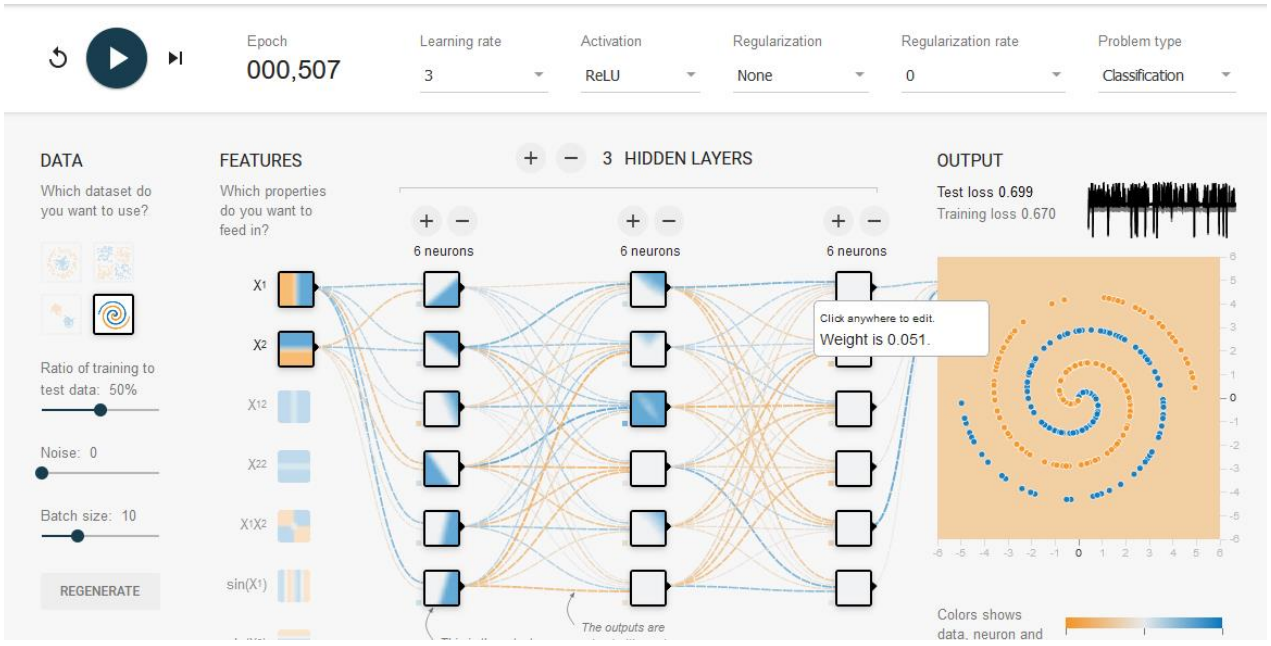

Epoch 000,508
Learning rate 0.03
Activation ReLU
Regularization None
Regularization rate 0
Problem type Classification
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
3 HIDDEN LAYERS
Test loss 0.022
Training loss 0.004

Figure 5: Hidden layers are 3. Loss is still lower. Compare with Figure 3 & Figure 4.

Experiment 4

Change learning rate & see effects

Configuration		Results	
Nos of hidden layers:	**3**	Test loss:	0.022
No of neurons in hidden layer:	6	Training loss:	0.004
Learning rate:	**3**		
Activation:	'relu'		
Epochs:	500		



Epoch 000,507

Learning rate 3

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: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

3 HIDDEN LAYERS

6 neurons

6 neurons

6 neurons

Click anywhere to edit.

Weight is 0.051.

The outputs are

OUTPUT

Test loss 0.699

Training loss 0.670

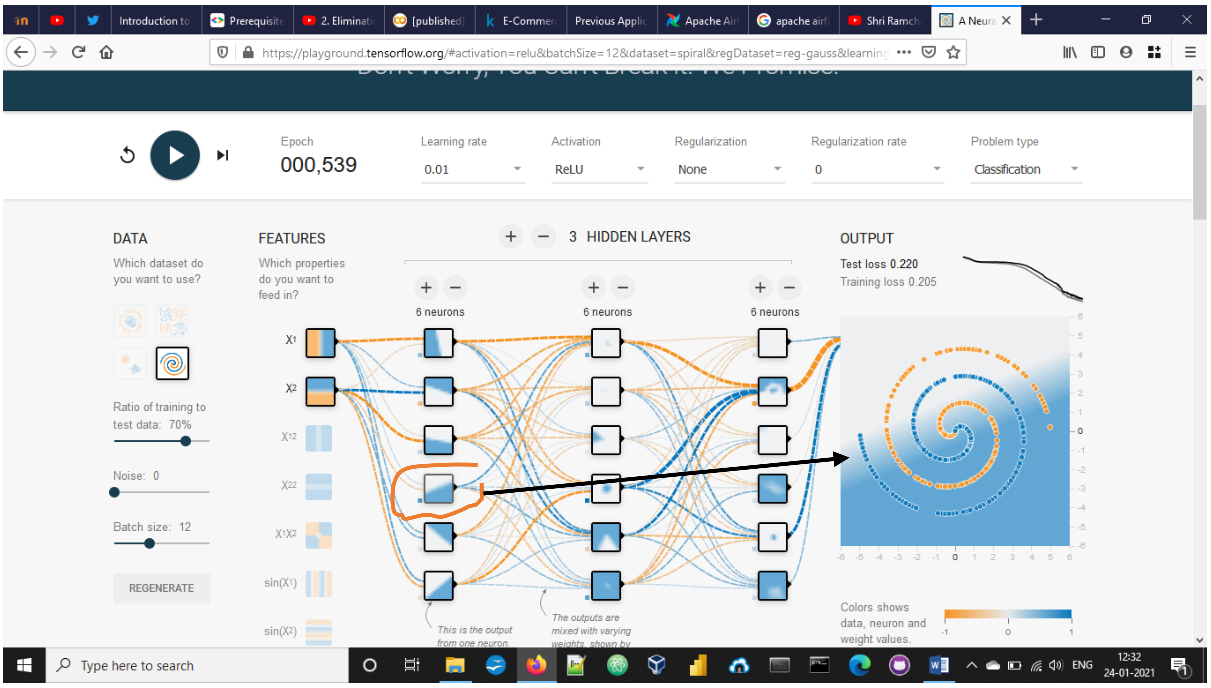
Colors shows data, neuron and

Figure 6: With learning rate high (3), loss is oscillating between high (0.8) and low (0.65) and not converging

Experiment 5

What kind of pattern are neurons learning?

Hover your cursor, over any neuron. And the Image of pattern that it has learnt to distinguish appears on the right. Neurons in the Ist hidden layers learn simple linear features. In the IInd hidden layer, neurons learn features that are more complex. Further, neurons in the IIIrd layer learn features that are still more complex.



Epoch 000,539 · Learning rate 0.01 · Activation ReLU · Regularization None · Regularization rate 0 · Problem type Classification

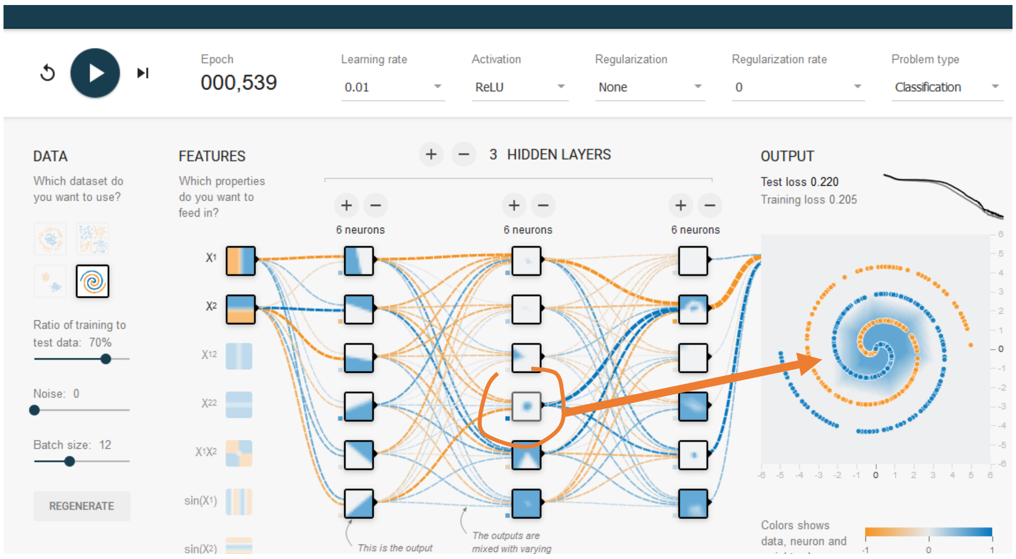
Which dataset do you want to use? Ratio of training to test data: 70% · Noise: 0 · Batch size: 12

Features: X1, X2 · 3 HIDDEN LAYERS: 6 neurons, 6 neurons, 6 neurons

Test loss 0.220
Training loss 0.205

Introduction to
Prerequisite
2. Eliminati
[published]
E-Commer
Previous Applic
Apache Air
apache airf
Shri Ramch
A Neura
https://playground.tensorflow.org/#activation=relu&batchSize=12&dataset=spiral&regDataset=reg-gauss&learning
DATA
FEATURES
Which properties do you want to feed in?
REGENERATE
OUTPUT
This is the output from one neuron.
The outputs are mixed with varying weights, shown by
Colors shows data, neuron and weight values.
Type here to search
ENG
12:32
24-01-2021

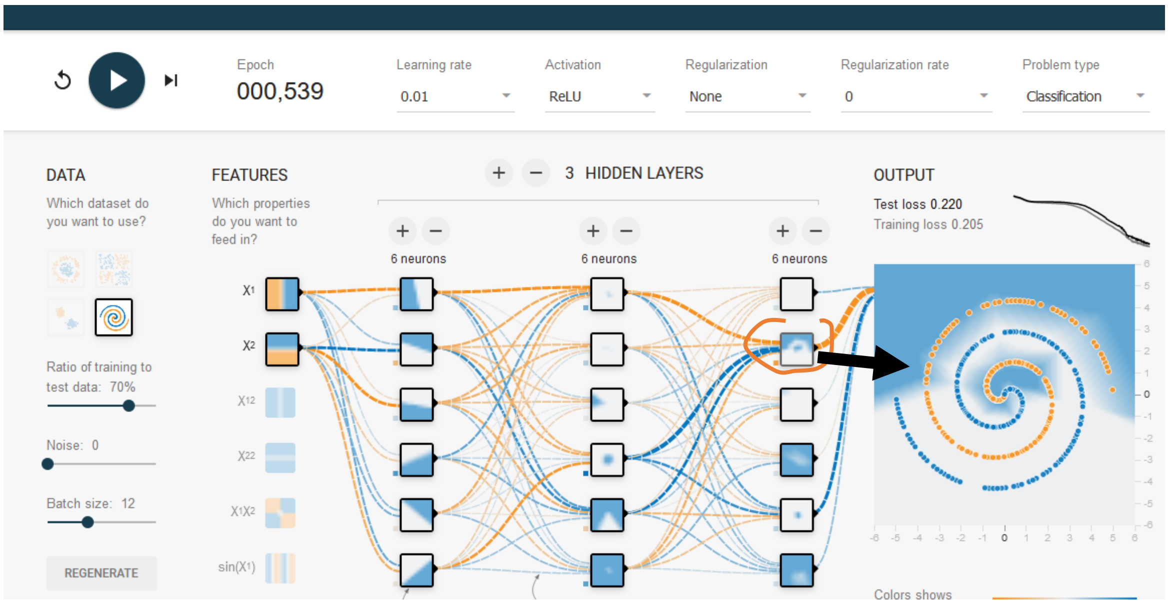
Figure 7: Hovering over a neuron, gives the type of pattern the neuron has learnt to distinguish.



Epoch 000,539 · Learning rate 0.01 · Activation ReLU · Regularization None · Regularization rate 0 · Problem type Classification

Test loss 0.220
Training loss 0.205

Figure 8: In the first layer, neurons have learnt linear features of various slants. In the IInd layer, features are more rounded.


Epoch 000,539

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: 70%

Noise: 0

Batch size: 12

FEATURES

Which properties do you want to feed in?

3 HIDDEN LAYERS

6 neurons

6 neurons

6 neurons

OUTPUT

Test loss 0.220

Training loss 0.205

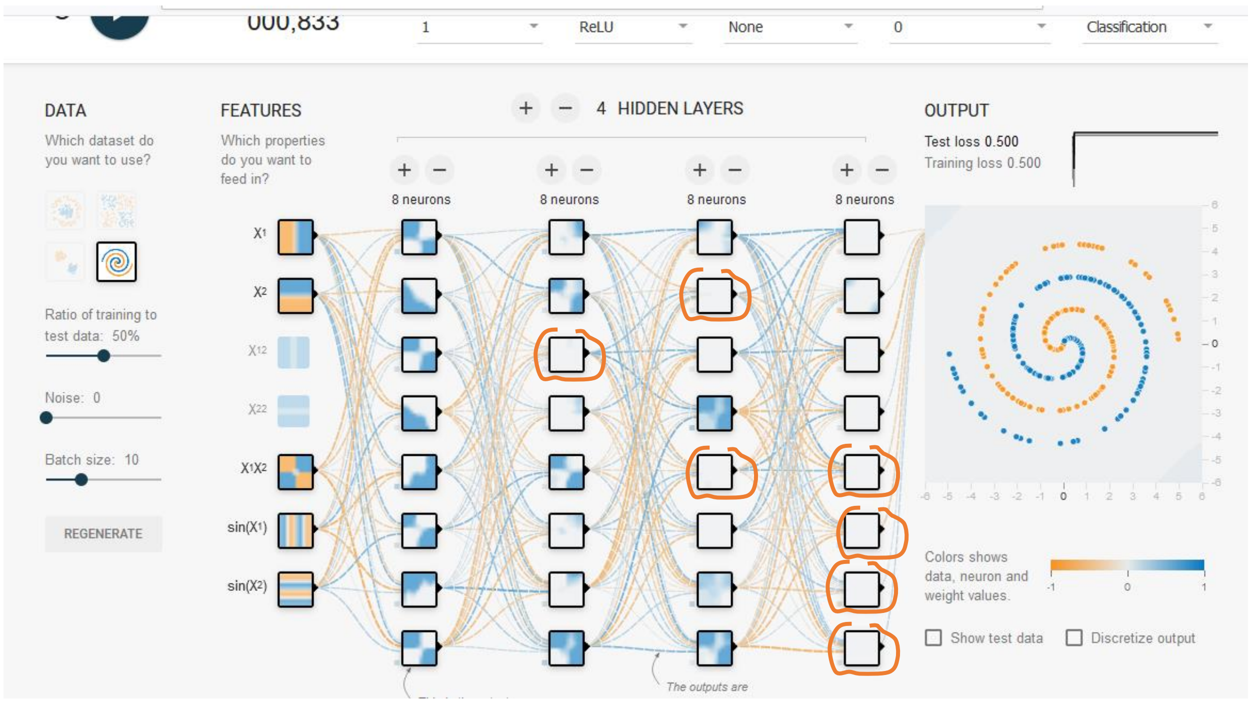
Figure 9: IIIrd layer learns more complex features.

So why deep learning?

Thus, even though, a single hidden layer can approximate any function, a series of successive hidden layers makes the task of learning easy, as layers learn progressively more abstract features <u>in a short time</u>.

Relu leads to dead neurons

When activation function is ReLu and when sum of input weights is negative, ReLu neuron becomes dead and does not output anything. Once dead, always dead. See figure below:


000,833

1

ReLU

None

0

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?

4 HIDDEN LAYERS

8 neurons

8 neurons

8 neurons

8 neurons

OUTPUT

Test loss 0.500

Training loss 0.500

Colors shows data, neuron and weight values.

Figure 10: Dead neurons. Encircled neurons are dead. Blue lines indicate –ve weights and orange lines indicate +ve weights. Hover over a weight to know weight value.

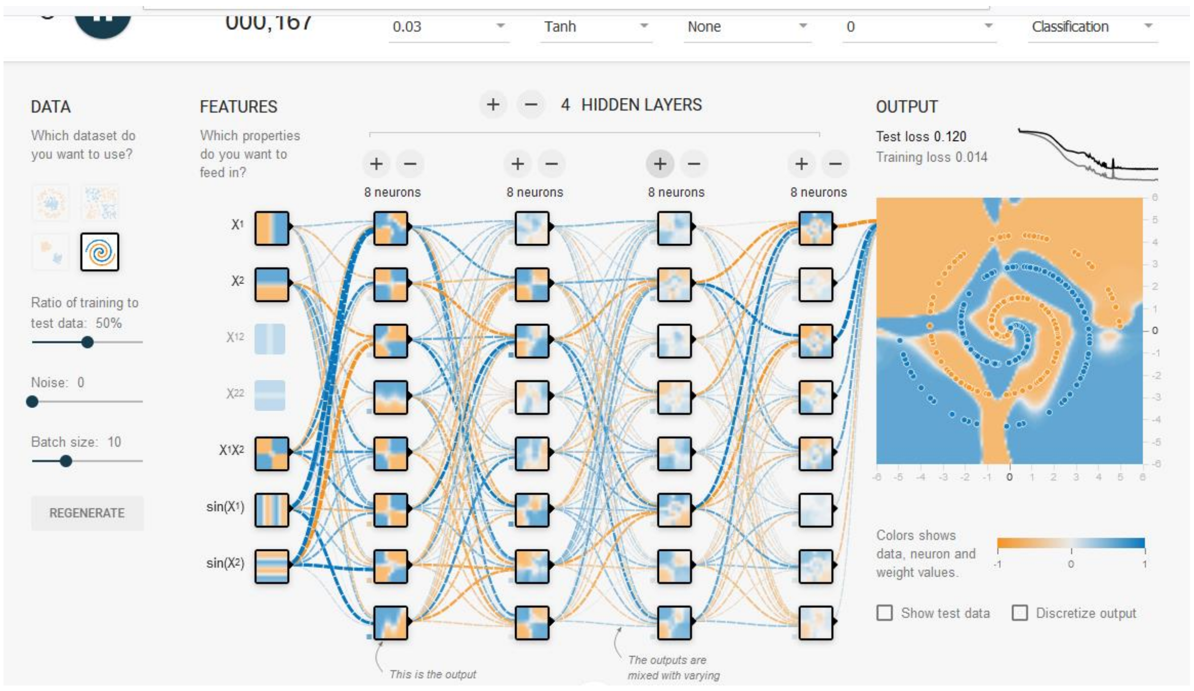


Figure 11: No death with 'tanh'

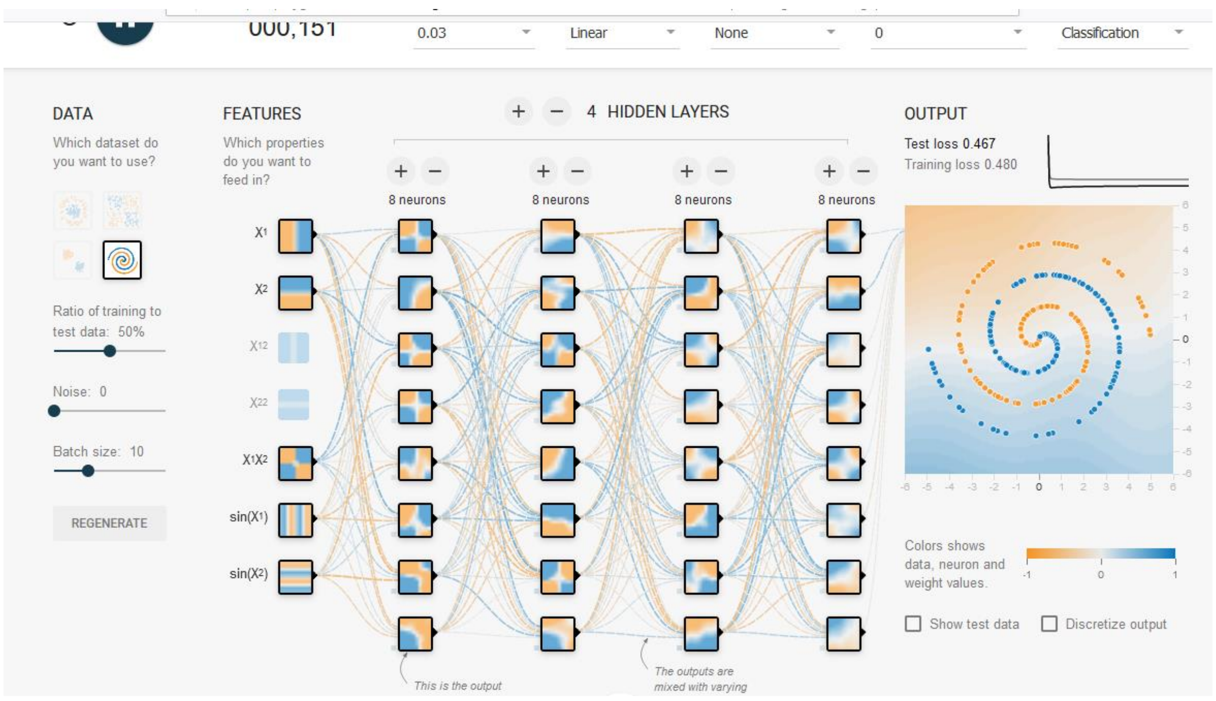


Figure 12: No death with 'linear' activation.

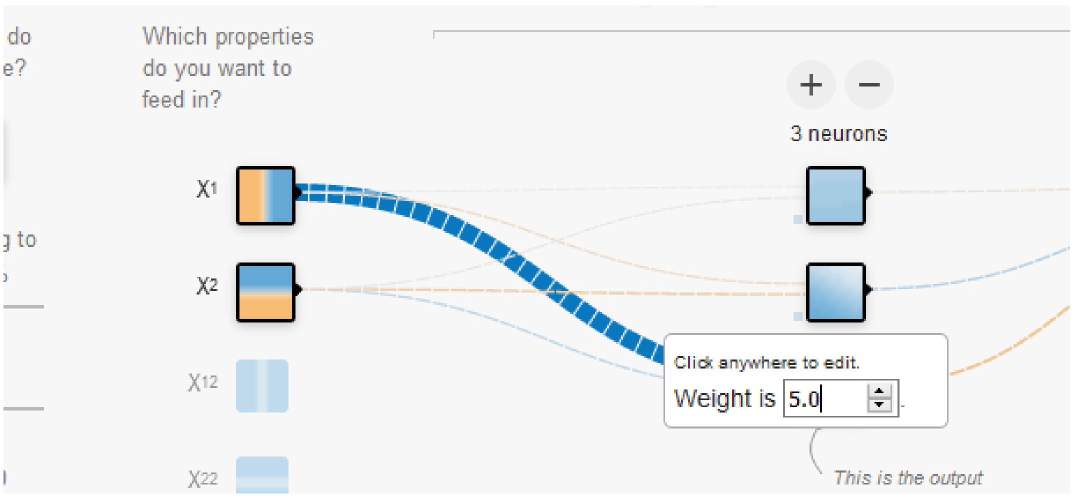
Learning-rate and saturation

Large learning rates may lead to saturation and slowdown of learning. Note that the weight adjustment formula is:

$$W_{new} = W_{old} + \lambda * \Delta w$$

where λ is learning rate

So if λ is large, we may have a learning slowdown with no improvement in *Test loss* even after many epochs. You can experiment with the following sigmoid network. Experiment by adjusting starting weights to very high value. And, when you start training by clicking '*Play*' button, no improvement in test loss happens.

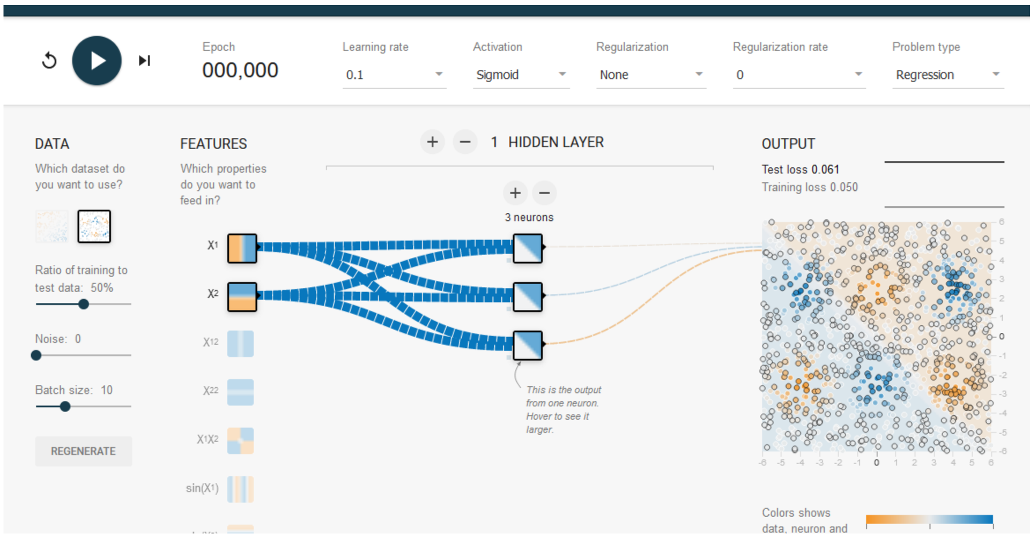


Click anywhere to edit.
Weight is 5.0

Figure 13: A close-up of weight initialization

Proceed like this:

Set problem type to *Regression*. Design neural network with one hidden layer and three neurons, as in the figure below. Initialize this network by clicking on '*Reset*' button. After, resetting weights, manipulate all incoming weights manually to a value of '*+5*'—do this only for inputs to hidden layer as in the figure below.

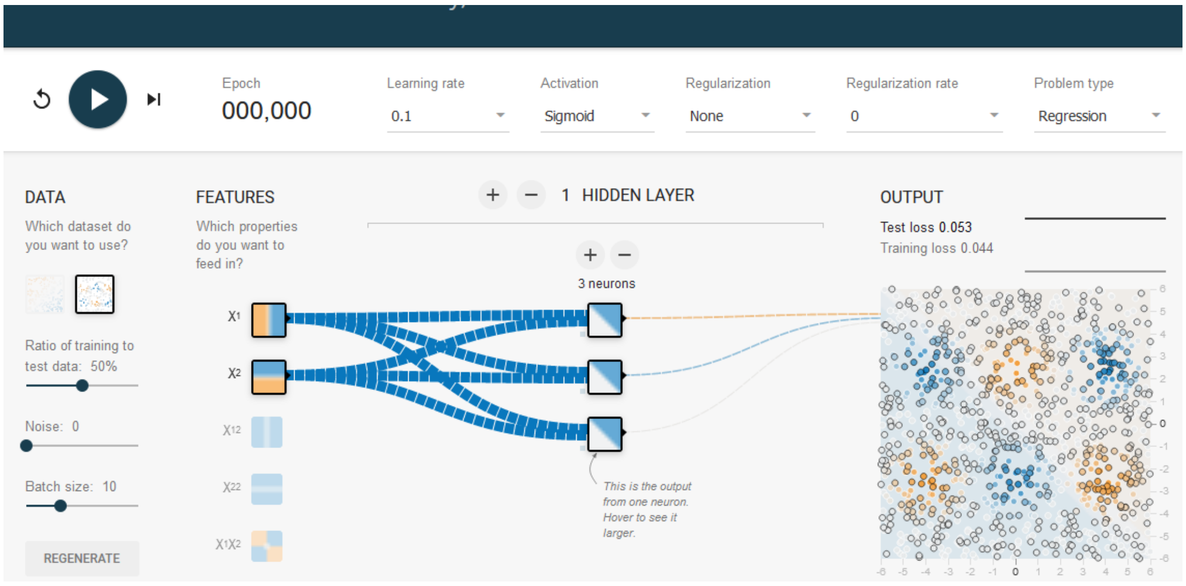

DATA
Which dataset do you want to use?
REGENERATE
FEATURES
Which properties do you want to feed in?
X1
X2
X12
X22
X1X2
sin(X1)
1 HIDDEN LAYER
3 neurons
This is the output from one neuron. Hover to see it larger.
OUTPUT
Colors shows data, neuron and

Epoch 000,000
Learning rate 0.1
Activation Sigmoid
Regularization None
Regularization rate 0
Problem type Regression

Ratio of training to test data: 50%
Noise: 0
Batch size: 10

Test loss 0.061
Training loss 0.050

Figure 14: Adjust weights BEFORE hidden layer and NOT after hidden layer.


Epoch 000,000

Learning rate 0.1

Activation Sigmoid

Regularization None

Regularization rate 0

Problem type Regression

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?

1 HIDDEN LAYER

3 neurons

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

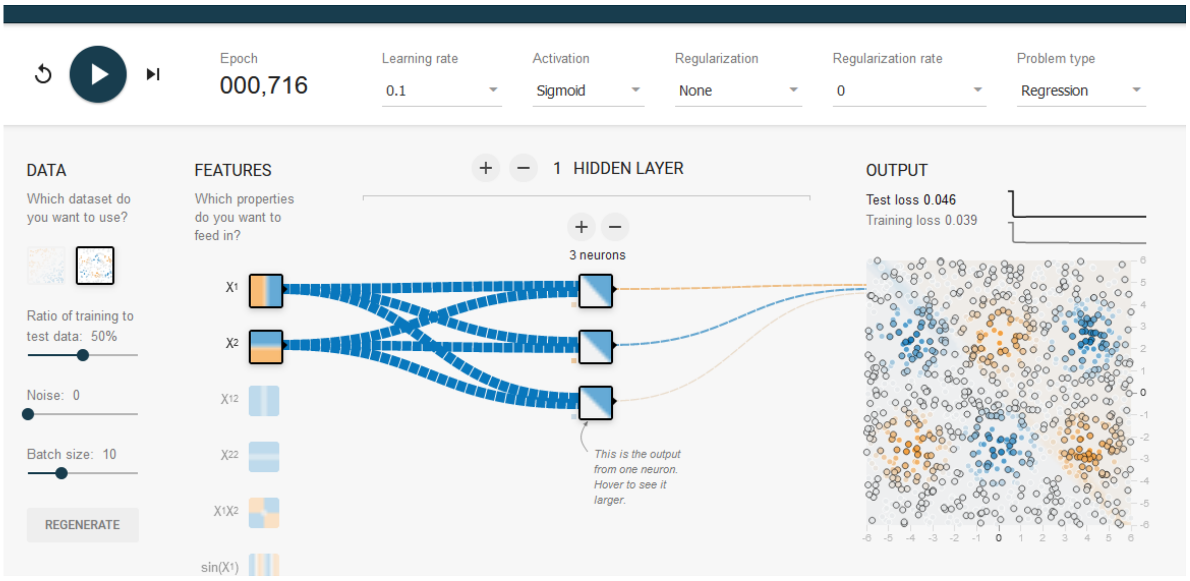
OUTPUT

Test loss 0.053

Training loss 0.044

Figure 15: This is our network. We will now click 'Play' button and observe 'Test Loss' (see below) figure. Starting Test loss is 0.053.

Now, click the *Play* button. Test loss after initial decrease (because of adjustment of weights between hidden layer and output neuron) soon freezes and shows no improvements.


Epoch 000,716

Learning rate 0.1

Activation Sigmoid

Regularization None

Regularization rate 0

Problem type Regression

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?

1 HIDDEN LAYER

3 neurons

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

OUTPUT

Test loss 0.046

Training loss 0.039

Figure 16: Test loss after it reaches 0.046, becomes stationary and does not reduce any further.

####################