

DR. ALVIN'S PUBLICATIONS

LEARNING TENSORFLOW PLAYGROUND

DR. ALVIN ANG

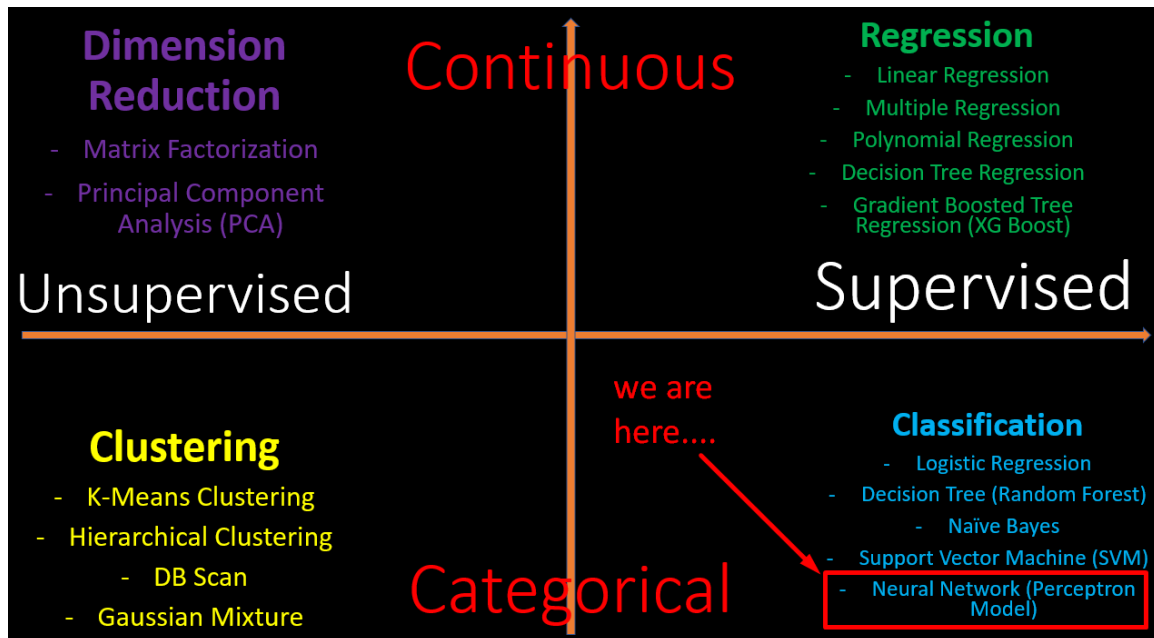


1 | PAGE

COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG

CONTENTS

I. Deep Learning = Neural Network (NN)	3
II. Learning Tensorflow Playground	4
A. Warm Up I	4
B. Warm Up II	6
III. Mini Challenge	8
A. Initial Settings.....	9
B. Settings 1.....	10
C. Settings 2.....	11
D. Settings 3.....	12
E. Settings 4.....	14
F. Settings 5.....	16
About Dr. Alvin Ang	17

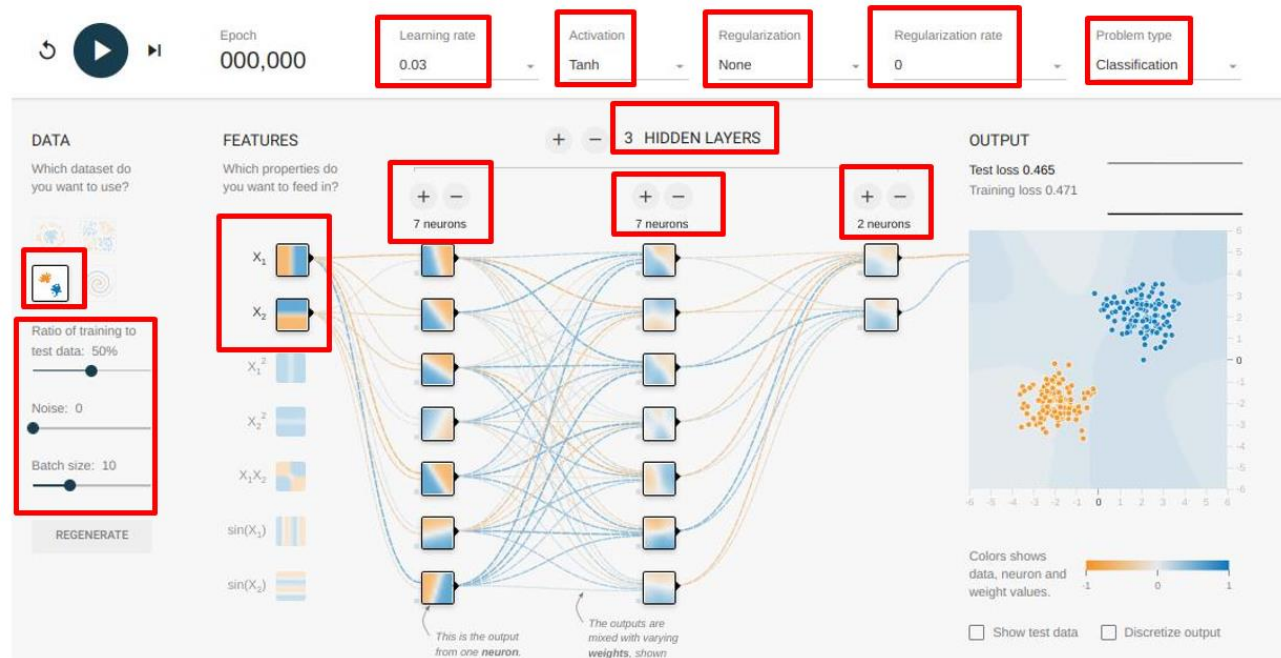


- Above is a table categorizing the different Machine Learning algorithms.
- Objective of Neural Network is to predict a CATEGORY.
- (actually, it can also be used to predict Regression...but most literature use it for classifying images like cats vs dogs....so we mainly use it for Classification...)
- Most of the ANN “how it works” has already been explained here: <https://www.alvinang.sg/s/Artificial-Neural-Network-ANN-How-It-Works-by-Dr-Alvin-Ang.pdf>
- we will not be going thru them again.

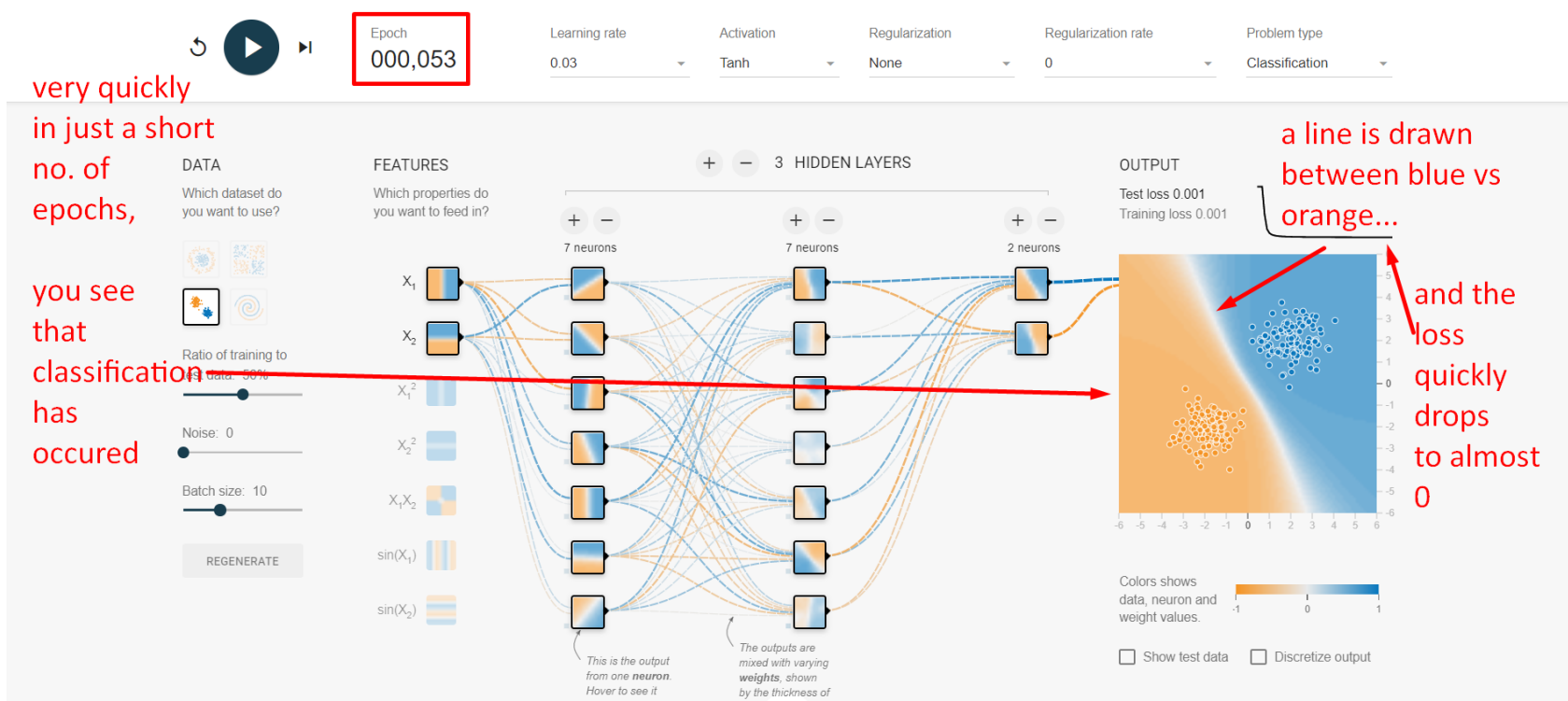
II. LEARNING TENSORFLOW PLAYGROUND

Go here: <https://playground.tensorflow.org/>

A. WARM UP I



Set the parameters to above and click play



B. WARM UP II

The screenshot shows a neural network training interface. At the top, there is a play button and a refresh button. The current epoch is 000,000. A red box highlights the top configuration bar with the following settings: Learning rate: 0.03, Activation: ReLU, Regularization: None, Regularization rate: 0, and Problem type: Classification.

The interface is divided into several sections:

- DATA:** A dropdown menu is set to a circular dataset. Below it, there are sliders for "Ratio of training to test data: 50%", "Noise: 0", and "Batch size: 10". A "REGENERATE" button is at the bottom.
- FEATURES:** A list of input features: X_1 , X_2 , X_1^2 , X_2^2 , X_1X_2 , $\sin(X_1)$, and $\sin(X_2)$.
- 3 HIDDEN LAYERS:** A red box highlights the layer configuration: 7 neurons, 7 neurons, and 2 neurons.
- OUTPUT:** A scatter plot showing two classes of data points (blue and orange) in a 2D space. The plot shows "Test loss 0.501" and "Training loss 0.507". A color scale legend indicates values from -1 to 1. There are checkboxes for "Show test data" and "Discretize output".

Annotations at the bottom of the network diagram include: "This is the output from one neuron." and "The outputs are mixed with varying weights, shown by the thickness of the lines."

Set the above parameters and click play



Epoch
000,079

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

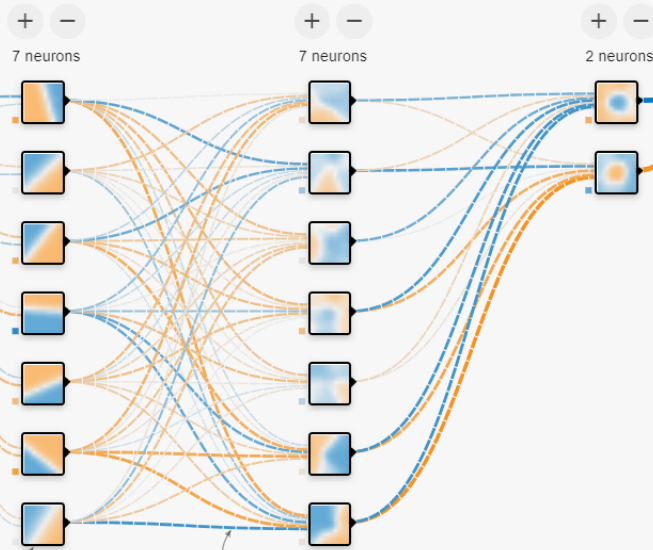
REGENERATE

FEATURES

Which properties do you want to feed in?

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

3 HIDDEN LAYERS

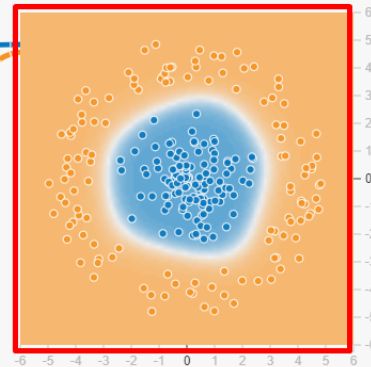


This is the output from one neuron. Hover to see it

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

OUTPUT

Test loss 0.004
Training loss 0.003



Colors shows data, neuron and weight values.

Show test data Discretize output

III. MINI CHALLENGE

1. Choose the most challenging dataset (spiral) and train the model
2. Change the architecture of the model by adding new hidden layers, neurons and changing activation functions
3. Tune the model hyperparameters such as learning rate and regularization to achieve better model performance.
4. Perform feature engineering to improve model performance



A. INITIAL SETTINGS

The screenshot displays a neural network training interface. At the top, the 'INITIAL SETTINGS' section includes a play button, 'Epoch 000,141', and a table of parameters: Learning rate (0.03), Activation (Tanh), Regularization (None), Regularization rate (0), and Problem type (Classification). Below this, the 'DATA' section shows a dataset selection icon and sliders for 'Ratio of training to test data: 50%', 'Noise: 0', and 'Batch size: 10'. The 'FEATURES' section lists input variables: X_1 , X_2 , X_1^2 , X_2^2 , $X_1 X_2$, $\sin(X_1)$, and $\sin(X_2)$. The 'HIDDEN LAYERS' section shows a network with 7 neurons in the first two hidden layers and 2 neurons in the output layer. The 'OUTPUT' section displays 'Test loss 0.508' and 'Training loss 0.454', along with a scatter plot of data points forming a spiral. A color scale at the bottom indicates values from -1 to 1. Red annotations highlight the high loss and the difficulty of the classification task.

if i used these parameter settings

u see that i'm unable to converge

i can't classify between orange vs blue because its the most difficult problem

the loss is still very high

it can't seem to classify properly

B. SETTINGS 1

now i changed to ReLU

and after 757 runs you see it performing better...

however still can't completely classify

and the loss is still quite high

The screenshot shows a neural network simulator interface. At the top, there are settings for Epoch (000,757), Learning rate (0.03), Activation (ReLU), Regularization (None), Regularization rate (0), and Problem type (Classification). Below the settings, there is a 'DATA' section with a 'Which dataset do you want to use?' dropdown and a 'Ratio of training to test data: 50%' slider. The 'FEATURES' section has a 'Which properties do you want to feed in?' dropdown. The main part of the interface is a neural network diagram with 3 hidden layers: 7 neurons, 7 neurons, and 2 neurons. The output is a 2D plot showing a spiral dataset with a color scale from -1 to 1. The test loss is 0.294 and the training loss is 0.221. A small graph shows the loss over time. Red arrows point from the text annotations to the ReLU activation function, the epoch number, the spiral plot, and the loss values.

C. SETTINGS 2

The screenshot shows a neural network training interface with the following components and annotations:

- Epoch Counter:** A red box highlights the value "Epoch 000,692". A red arrow points to it with the text "took quite a huge number of epochs".
- Settings:** Learning rate is 0.03, Activation is ReLU, Regularization is None, Regularization rate is 0, and Problem type is Classification.
- Neural Network Structure:** A red box highlights the "5 HIDDEN LAYERS" section, which includes 7 neurons, three layers of 8 neurons each, and 2 neurons in the output layer. A red arrow points to this structure with the text "and extra layers and neurons to learn this classification".
- Loss Graph:** A line graph shows training and test loss. A red box highlights a sharp peak in the training loss, with a red arrow and the text "there was a spike here". Another red box highlights a flat region of the training loss, with a red arrow and the text "stabilized here".
- Output Visualization:** A heatmap shows the output of the network, displaying a spiral pattern. A red box highlights this visualization, with a red arrow pointing to it from the "stabilized here" annotation.
- Annotations:** On the left side, red text says "which is not bad!".

D. SETTINGS 3

Epoch: 000,000
Learning rate: 0.03
Activation: ReLU
Regularization: None
Regularization rate: 0
Problem type: Classification

now we are going to leave all previous settings as is

but we are going to make the problem much harder by increasing the noise to 25

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

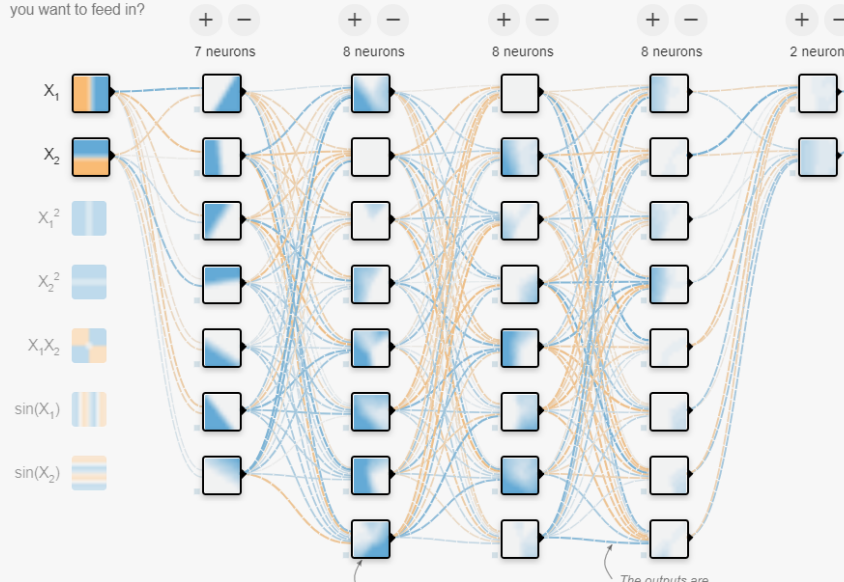
Noise: 25

Batch size: 10

REGENERATE

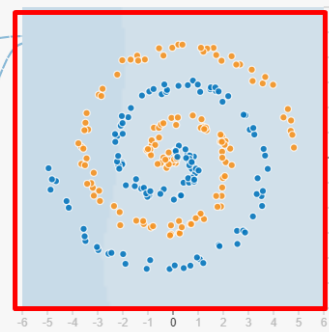
FEATURES

Which properties do you want to feed in?



OUTPUT

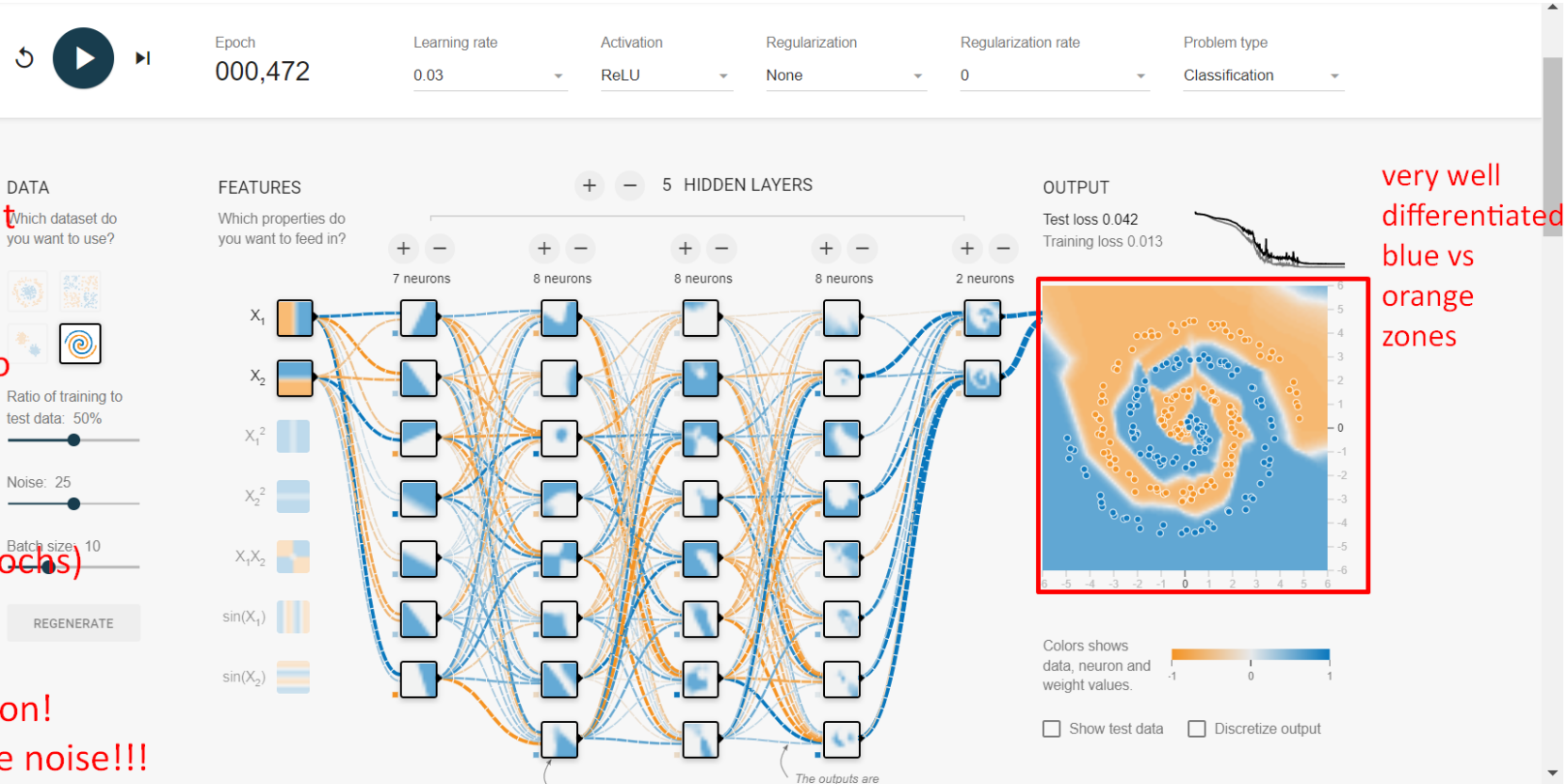
Test loss 0.518
Training loss 0.523



and u see now that the orange dots are twined with the blue dots its much harder to do classification!

not bad!
even at
our current
settings
we are
still able to
produce
a loss
of 0.013
(at 472 epochs)

and great
classification!
given more noise!!!



very well
differentiated
blue vs
orange
zones

E. SETTINGS 4

now we create a very very hard problem

max noise with the hardest dataset to classify...

Epoch 000,000


Learning rate 0.03

Activation ReLU

Regularization None

Regularization rate 0

Problem type Classification

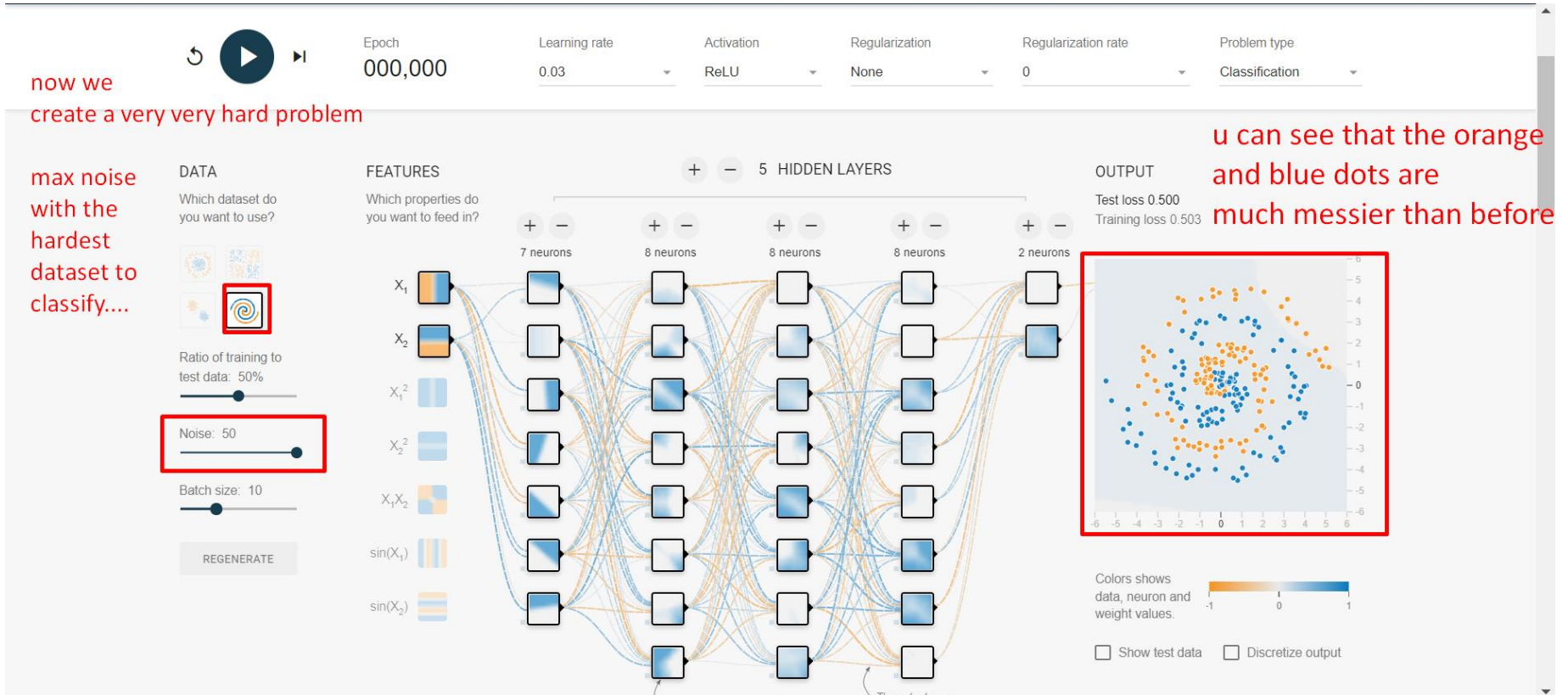
DATA
Which dataset do you want to use?

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

FEATURES
Which properties do you want to feed in?
 X_1
 X_2
 X_1^2
 X_2^2
 $X_1 X_2$
 $\sin(X_1)$
 $\sin(X_2)$

5 HIDDEN LAYERS
7 neurons, 8 neurons, 8 neurons, 8 neurons, 2 neurons

OUTPUT
Test loss 0.500
Training loss 0.503

u can see that the orange and blue dots are much messier than before



Colors shows data, neuron and weight values.
-1 0 1

Show test data Discretize output

you can see that with the current settings

at around 700 plus epochs

it still managed to classify

and get a low error of 0.058...



Epoch
000,745

Learning rate	Activation	Regularization	Regularization rate	Problem type
0.03	ReLU	None	0	Classification

DATA
Which dataset do you want to use?



Ratio of training to test data: 50%

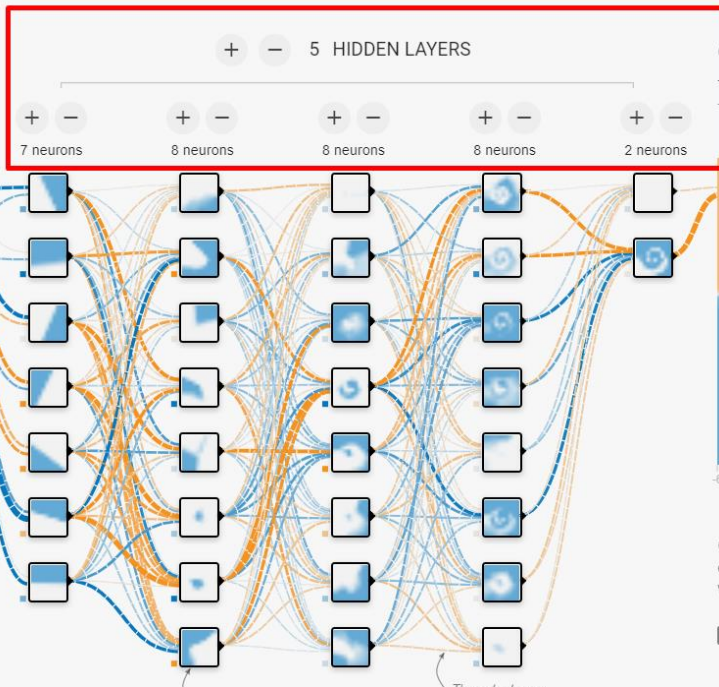
Noise: 50

Batch size: 10

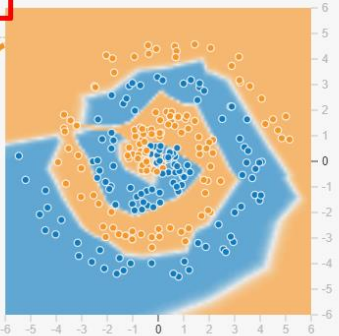
REGENERATE

FEATURES
Which properties do you want to feed in?

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



OUTPUT
test loss 0.117
training loss 0.058



Colors shows data, neuron and weight values.
 Show test data Discretize output

F. SETTINGS 5

now given
the same settings
as we did previously

on the hardest
dataset to
classify
with highest
noise....

we managed
to cut epoch size
by half to around 320
and got 0.052 loss
and classification done!



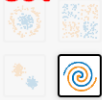
Epoch
000,320

Learning rate: 0.03
Activation: ReLU
Regularization: None
Regularization rate: 0
Problem type: Classification

reason for the faster classification was because we input all features
which made the NN learn faster...

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 50

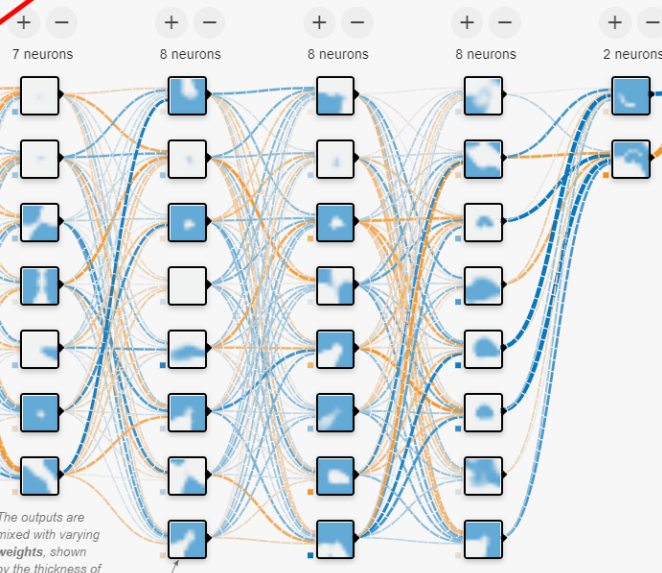
Batch size: 10
REGENERATE

FEATURES

Which properties do you want to feed in?

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

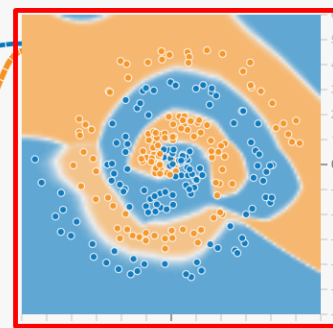
5 HIDDEN LAYERS



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

OUTPUT

Test loss 0.113
Training loss 0.052



Colors shows data, neuron and weight values.

Show test data Discretize output

ABOUT DR. ALVIN ANG



Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.