
冨嶽三十六景 神奈川沖浪裏

Lee Hwee Kuan

Head of Research Division
Bioinformatics Institute
A*STAR

leehk@bii.a-star.edu.sg

Why are you taking this course?

How our world is going to change?

How our world is going to change?

Old	New
Driving	Work from home
Manufacturing	Shop from home
Clerical work	Self healthcare
Retail	Self education
Mass education	AI work
Maids & cleaners	

How our world is going to change?

Storm 2011

Boston Dynamics 2017

http://www.mrbrown.com/blog/2011/03/maid-for-the-army-2.html

What’s Deep Learning for?

Deep Learning related work in

Robotics

Check out:
Boston Dynamics

https://www.youtube.com/watch?v=EP_NCB3KkiY
https://www.youtube.com/watch?v=rVlhMGQgDkY
https://www.youtube.com/watch?v=M8YjvHYbZ9w

Deep Learning related work in **Financial Technologies (Fintech)**

“. . .fintech is on the verge of a truly revolutionary moment with the integration of artificial intelligence and deep learning into financial services.

http://fintechnews.sg/8840/fintech/deep-learning-finance-summit-comes-singapore-discuss-emerging-trends-opportunities/

Using machine learning for insurance pricing optimization

https://cloud.google.com/blog/big-data/2017/03/using-machine-learning-for-insurance-pricing-optimization

Much trading is done by machines especially high frequency trading

Deep Learning related work in

Retail

Amazon Go

no cashier, no queues shopping

https://venturebeat.com/2016/12/05/amazon-launches-amazon-go-a-brick-and-mortar-grocery-store-that-does-away-with-checkouts/

check this out!

https://www.amazon.com/b?node=16008589011

Deep Learning related work in **Legal**

“Legal Robot uses machine learning techniques like deep learning to understand legal language, . . . “
https://www.legalrobot.com/

“. . . the UK-based news resource, LegalFutures, predicts that technologies automating the work of associates, herald the collapse of law in less than 15 years

A recent study by McKinsey & Co estimates that 23% of lawyer time is automatable. Similar research by Frank Levy at MIT and Dana Remus at University of North Carolina School of Law concludes that just 13% of lawyer time can be performed by computers. . .”

https://blogs.thomsonreuters.com/answerson/artificial-intelligence-legal-practice/

Deep Learning related work in **Manufacturing**

Foxconn replaces '60,000 factory workers with robots'

http://www.bbc.com/news/technology-36376966

see robotic cook

Deep Learning related work in **Farming**

Farmers are getting help from researchers and scientists who have turned the keen eye of AI toward agriculture, using deep learning applications to not only predict crop outputs but also to monitor water levels around the world and help detect crop diseases before one spreads.

https://dataskeptic.com/blog/news/2017/deep-learning-is-driving-the-new-agriculture-revolution

Deep Learning related work in **Biomedical**

Deep Learning Drops Error Rate for Breast Cancer Diagnoses by 85%

https://blogs.nvidia.com/blog/2016/09/19/deep-learning-breast-cancer-diagnosis/

Google researchers trained an algorithm to recognize a common form of eye disease as well as many experts can.

https://www.technologyreview.com/s/602958/an-ai-ophthalmologist-shows-how-machine-learning-may-transform-medicine/

Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis

G. Litjens et al Scientific Reports 6, 26286 (2016)

Dermatologist-level classification of skin cancer with deep neural networks

A. Esteva Nature 542, 115-118 (2017)

Deep Learning related work in **Pharmaceutical**

The Next Era: Deep Learning in Pharmaceutical Research.

S. Ekins

Pharm Res. 2016 Nov;33(11):2594-603. doi: 10.1007/s11095-016-2029-7. Epub 2016 Sep 6.

Many other Deep Learning related work

Smart City
Transportation
Aviation
Telecommunications
Construction
Education

. . . think of anything XXX, then search for

“deep learning XXX”

Historical notes on neural networks and deep learning

Historical notes

1943 **Warren McCulloch (neurophysiologist), Walter Pitts (mathematician)**
Mathematical model of the brain
McCulloch, Warren; Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". Bulletin of Mathematical Biophysics. ***5*** *(4): 115–133.*

1949 **Donald O. Hebb** Strengthening of connection between neurons
Hebb, D. O. (1949). The Organization of Behavior: A Neuropsychological Theory. New York: Wiley and Sons.

1959 **Bernard Widrow, Marcian Hoff** Single layer and multilayer neural nets. ADALINE and MADALINE
An adaptive "ADALINE" neuron using chemical "memistors"

1970 **Seppo Linnainmaa** While gradient descend algorithm dates back much earlier, Seppo contributed to the modern idea of back propagation
Linnainmaa, Seppo (1970). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. Master's Thesis (in Finnish), Univ. Helsinki, 6-7.

1989 **George Cybenko** Universal approximation theorem, sigmoid function
Cybenko, G. (1989) "Approximations by superpositions of sigmoidal functions", *Mathematics of Control, Signals, and Systems*, 2 (4), 303-314

1991 **Kurt Hornik** Universal approximation theorem, more general function
Kurt Hornik (1991) "Approximation Capabilities of Multilayer Feedforward Networks", Neural Networks, 4(2), 251–257

https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html

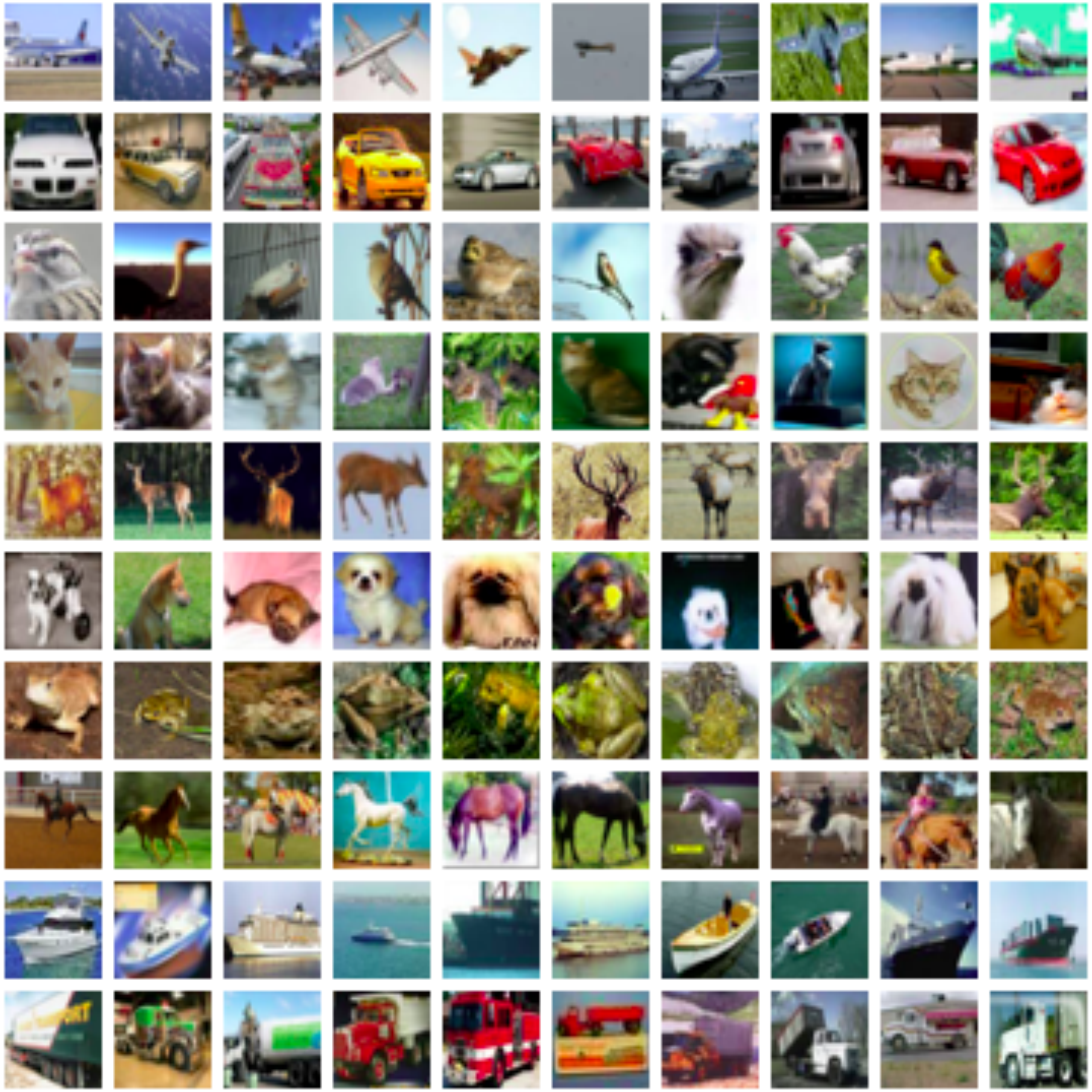
‘Contemporary’ history of neural nets

1974	**Paul Werbos**, Backpropagation
1980	**Kunihiko Fukushima**, Neocogitron which inspired Convolutional Neural Networks
1985	**Hilton & Sejnowski**, Boltzmann Machine
1986	**Paul Smolensky**, Harmonium, later known as Restricted Boltzmann Machine **Michael I. Jordan** Recurrent Neural Network
1990	**Yann LeCun**, LeNet - convolutional neural net
2006	**G. Hinton**, Deep Belieft Net, layer wise pretraining
2009	**Salakhutdinov & Hinton**, Deep Boltzmann Machines
2012	**N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdin**, Dropout
2014	**Ian Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio**, Generative Adversarial Networks
2015	**Kaiming He, *Xiangyu Zhang, Shaoqing Ren, Jian Sun***, Deep Residual Network

Haohan Wang, Bhiksha Raj, On the Origin of Deep Learning arXiv:1702.07800v4

How good is deep learning?

92.45% on CIFAR-10 in Torch

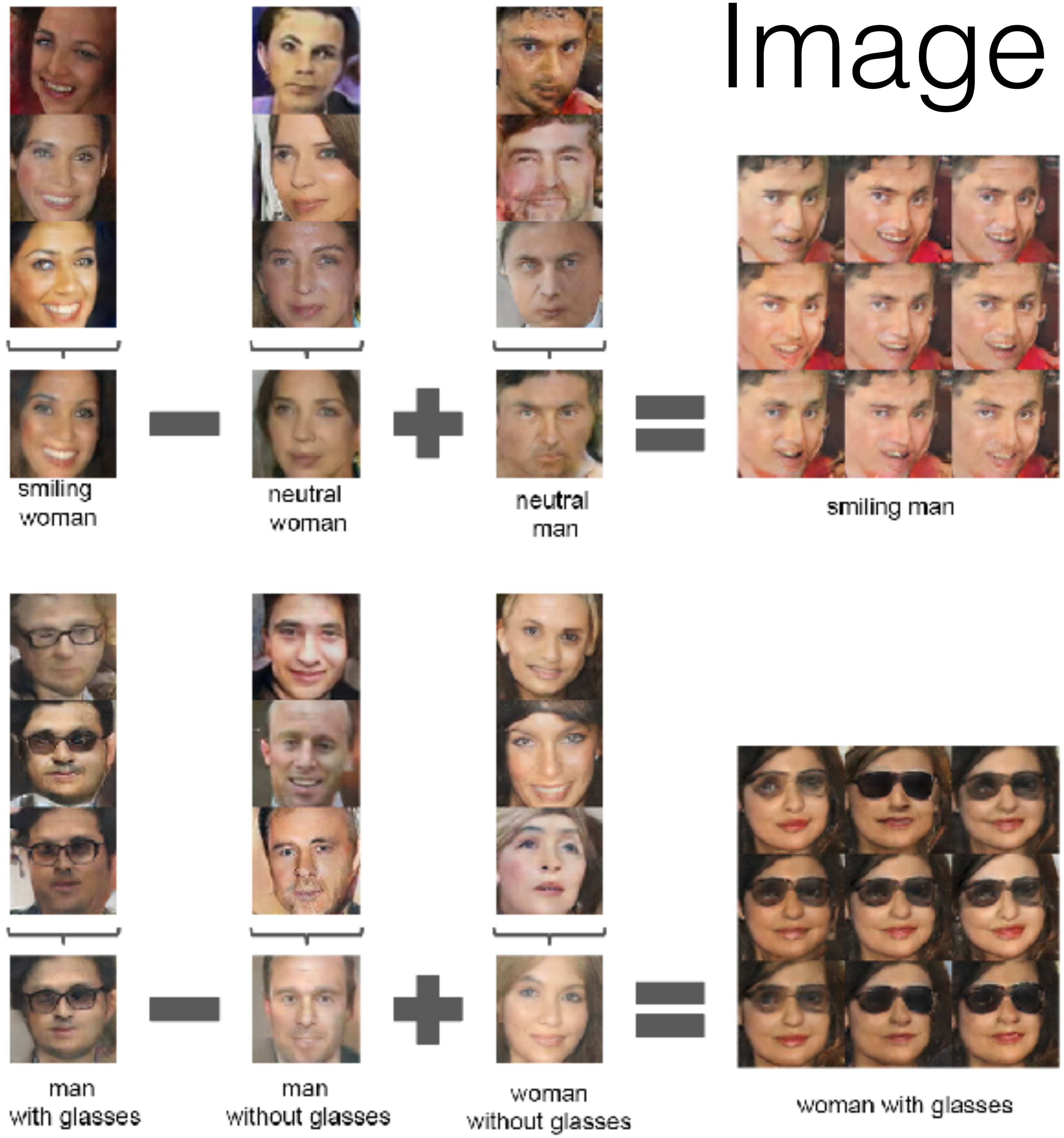


60,000 images, 50K train, 10K test, 10 classes

aeroplane, car, bird, cat, deer, dog, frog, horse, boat, truck

http://torch.ch/blog/2015/07/30/cifar.html

Image Generation

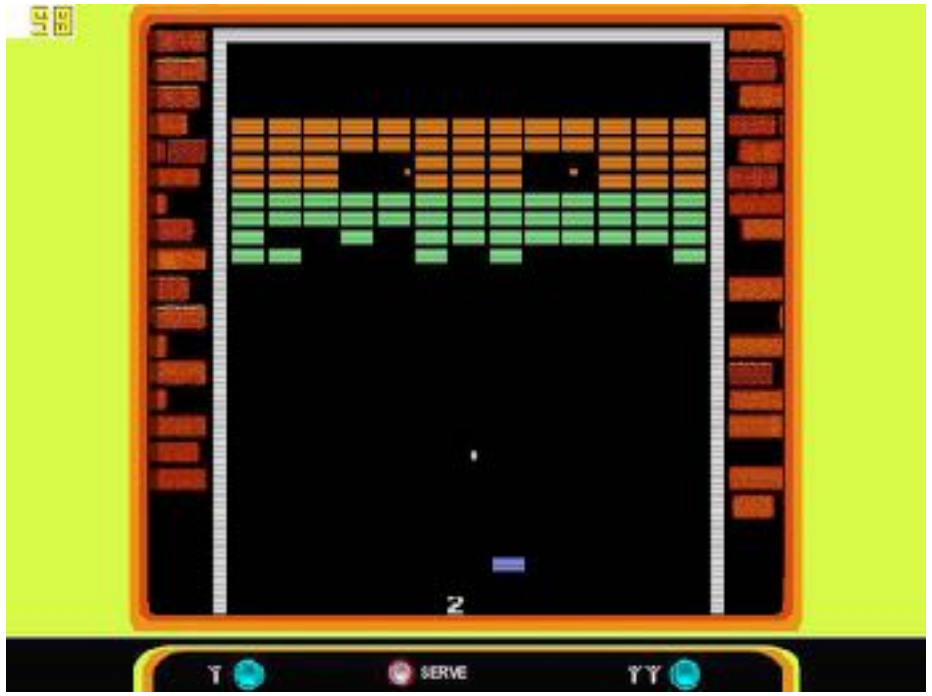


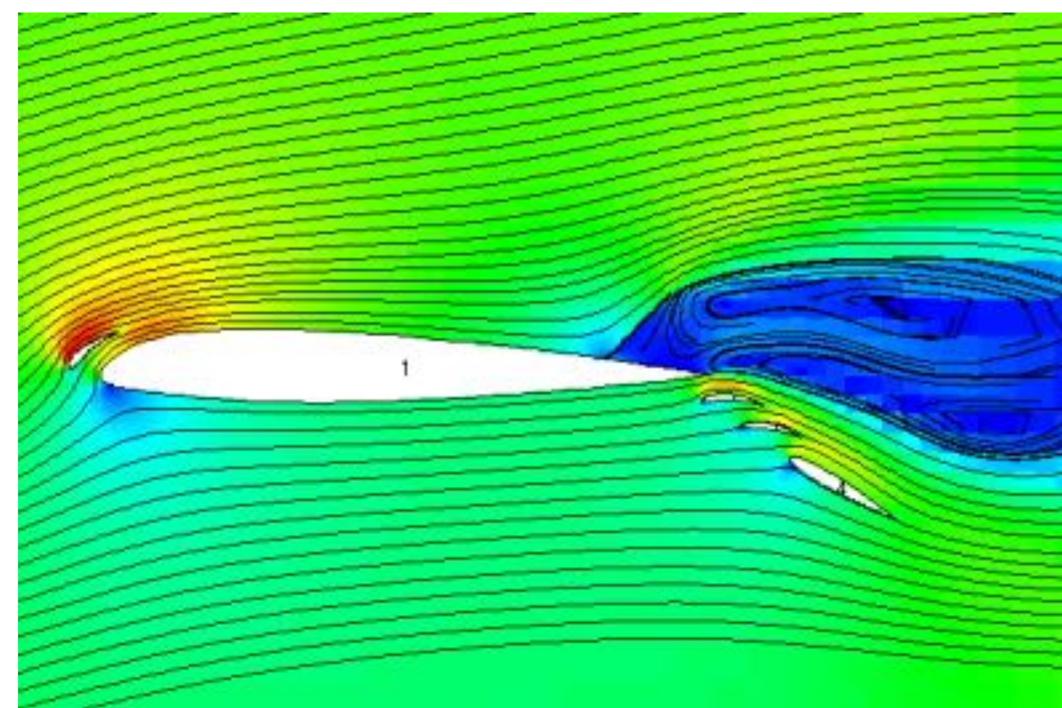
https://www.pinterest.com/pin/205124958008048718/

Alpha-Go

Our world versus computer world

23684184 x 4729472 = 112013685070848





Our world versus computer world

How to learn Deep Learning

The computer always give you

an output

is it correct?

Different levels of understanding Deep Learning usage

There are those who do not know what they are doing. Their computational results are unreasonable

There are those who know how to get some good results but cannot explain them

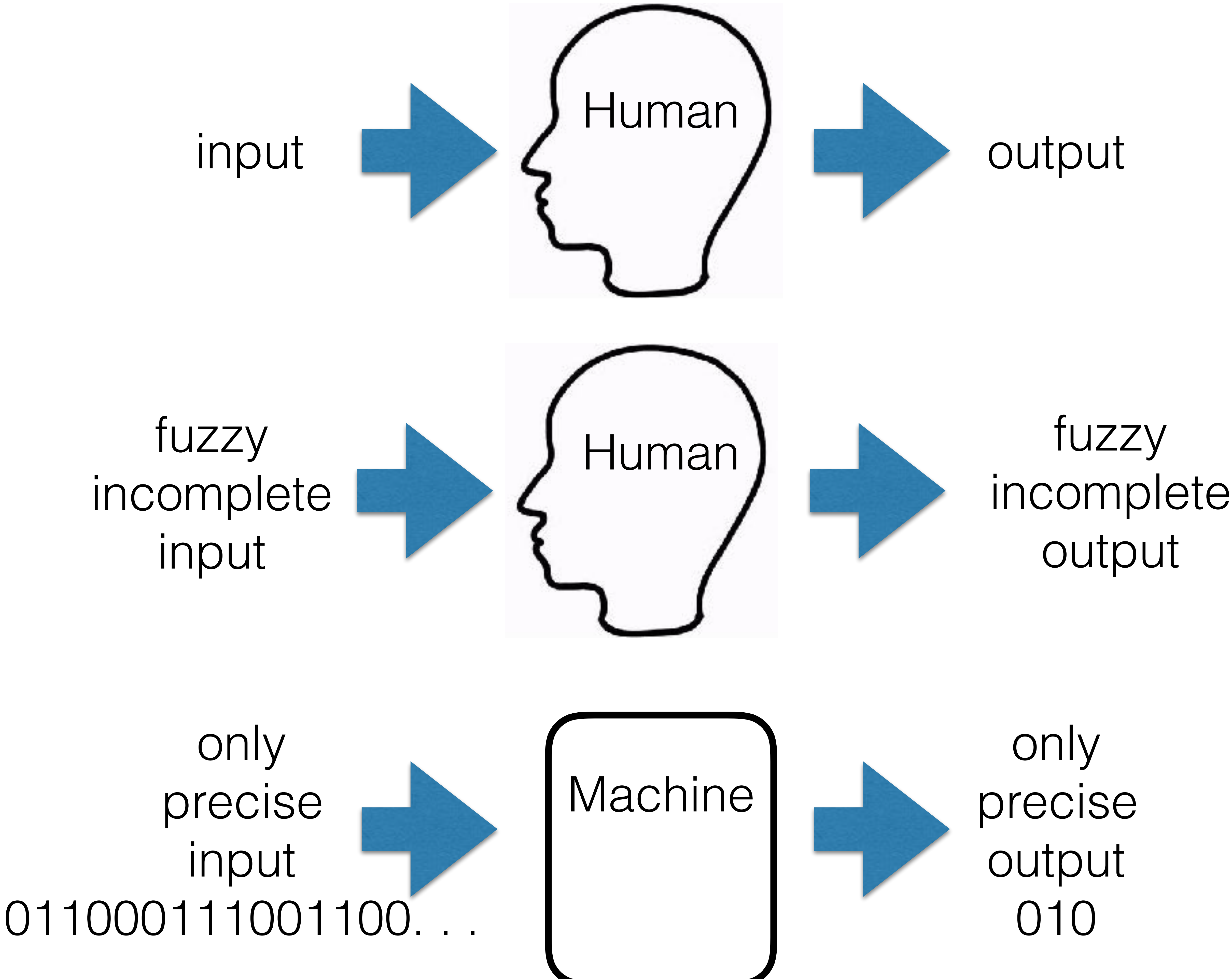
There are those who understand what is going on with their experiments. Able to explain their results

There are those who can combine different methods to create new things in Deep Learning

There are those who can fundamentally change Deep Learning research

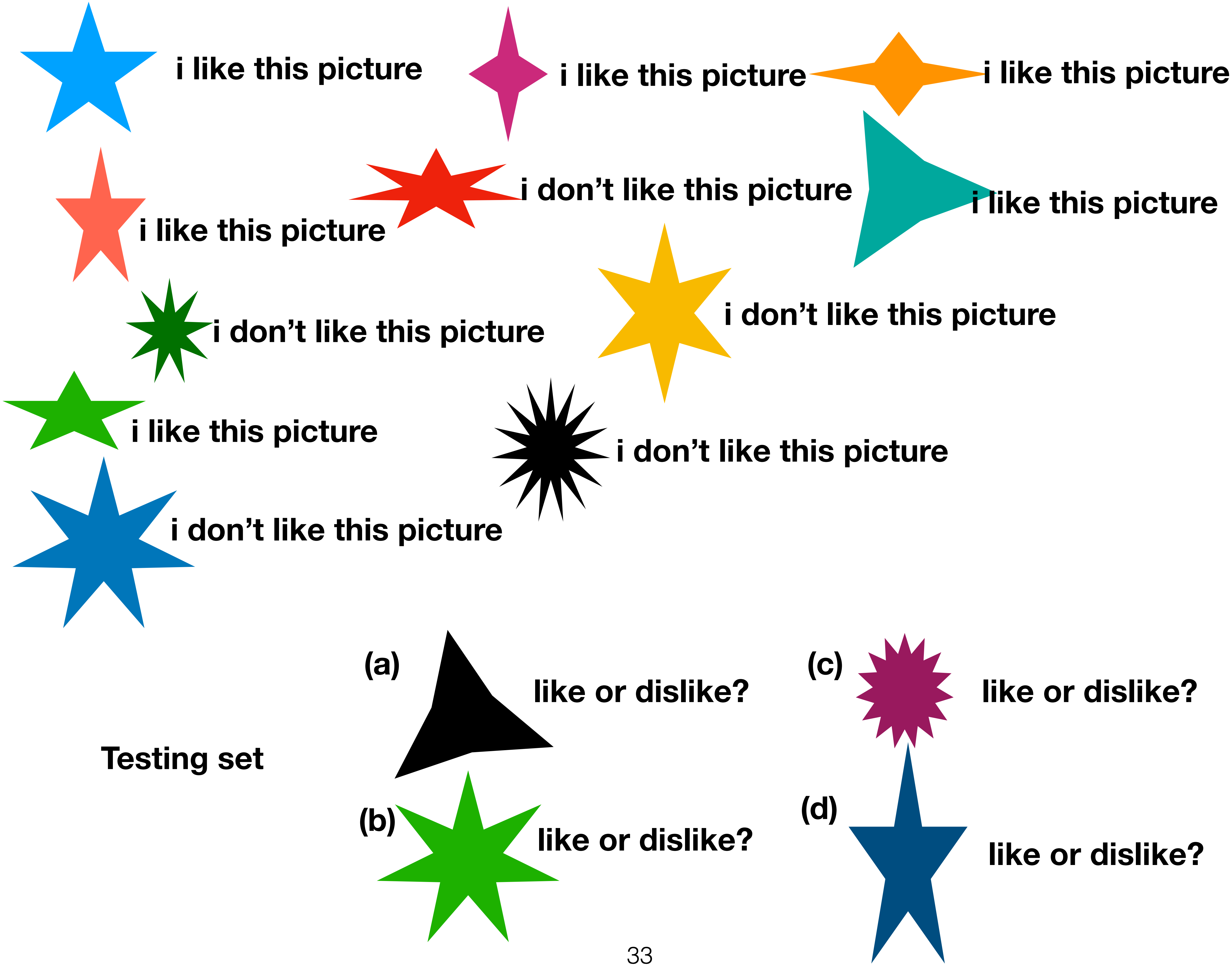
Very basic

lets get everyone on the same level
sorry if this seems too simple to some of you


input
Human
output
fuzzy incomplete input
Human
fuzzy incomplete output
only precise input
01100011100110 0. . .
Machine
only precise output
010

Talk so much . . . what exactly is Neural Networks?

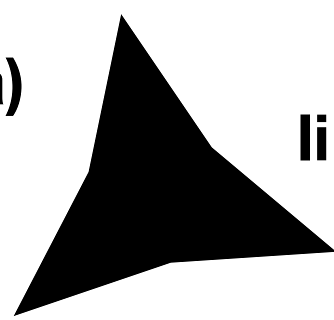
Lets play a game. . .

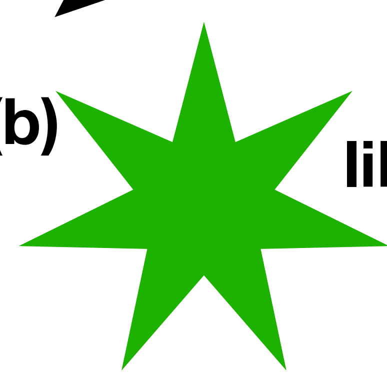

i like this picture
i like this picture
i like this picture
i like this picture
i don’t like this picture
i like this picture
i don’t like this picture
i don’t like this picture
i like this picture
i don’t like this picture
i don’t like this picture
Testing set
(a)
like or dislike?
(b)
like or dislike?
(c)
like or dislike?
(d)
like or dislike?

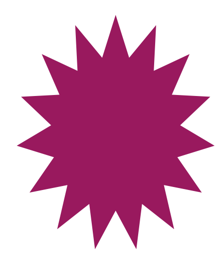
Training set

Like						
Dislike						

Testing set

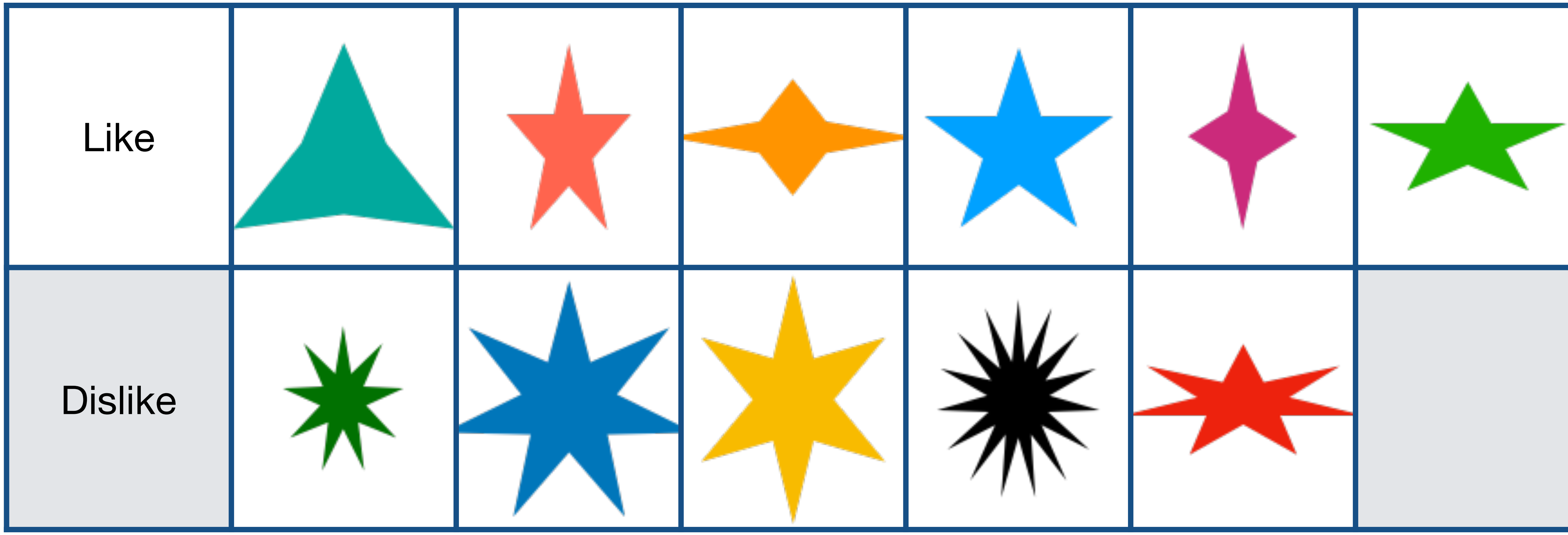
(a)  **like or dislike?**

(b)  **like or dislike?**

(c)  **like or dislike?**

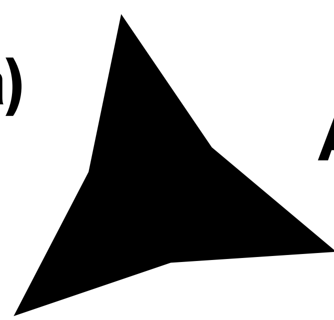
(d)  **like or dislike?**

Training set



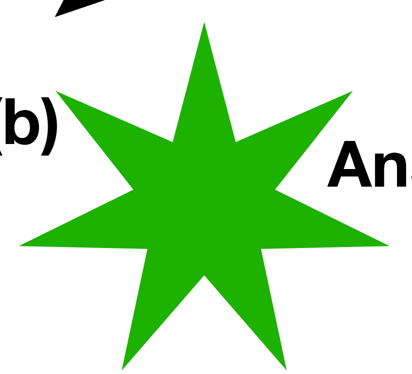
Testing set

(a)



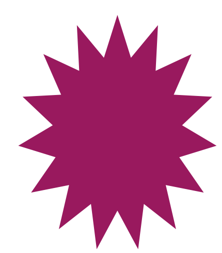
Answer: Like

(b)



Answer: Dislike

(c)



Answer: Dislike

(d)

Answer: Like

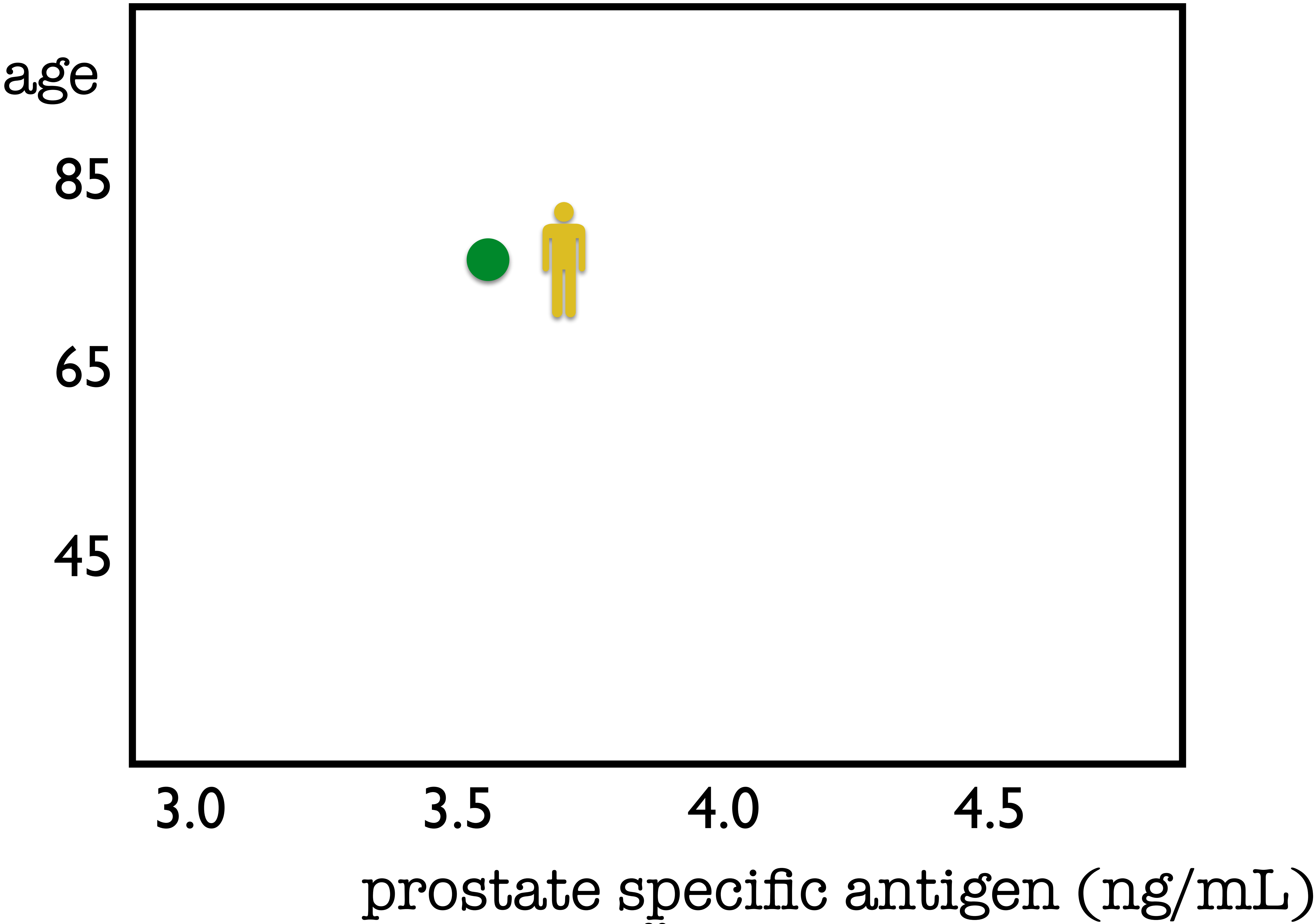
Case of prostate cancer diagnosis

Age	Prostate Specific Antigen (a blood test reading)	Got Prostate Cancer
59	4.9 ng/mL	Yes
72	3.9 ng/mL	Yes
45	6.0 ng/mL	Yes
47	3.2 ng/mL	No
39	3.9 ng/mL	No
89	3.5 ng/mL	Yes
61	5.5 ng/mL	Yes
62	2.1 ng/mL	No
49	3.4 ng/mL	No
95	3.1 ng/mL	Yes
67	4.3 ng/mL	Yes
49	3.8 ng/mL	?
58	4.3 ng/mL	?
88	4.1 ng/mL	?
31	2.1 ng/mL	?

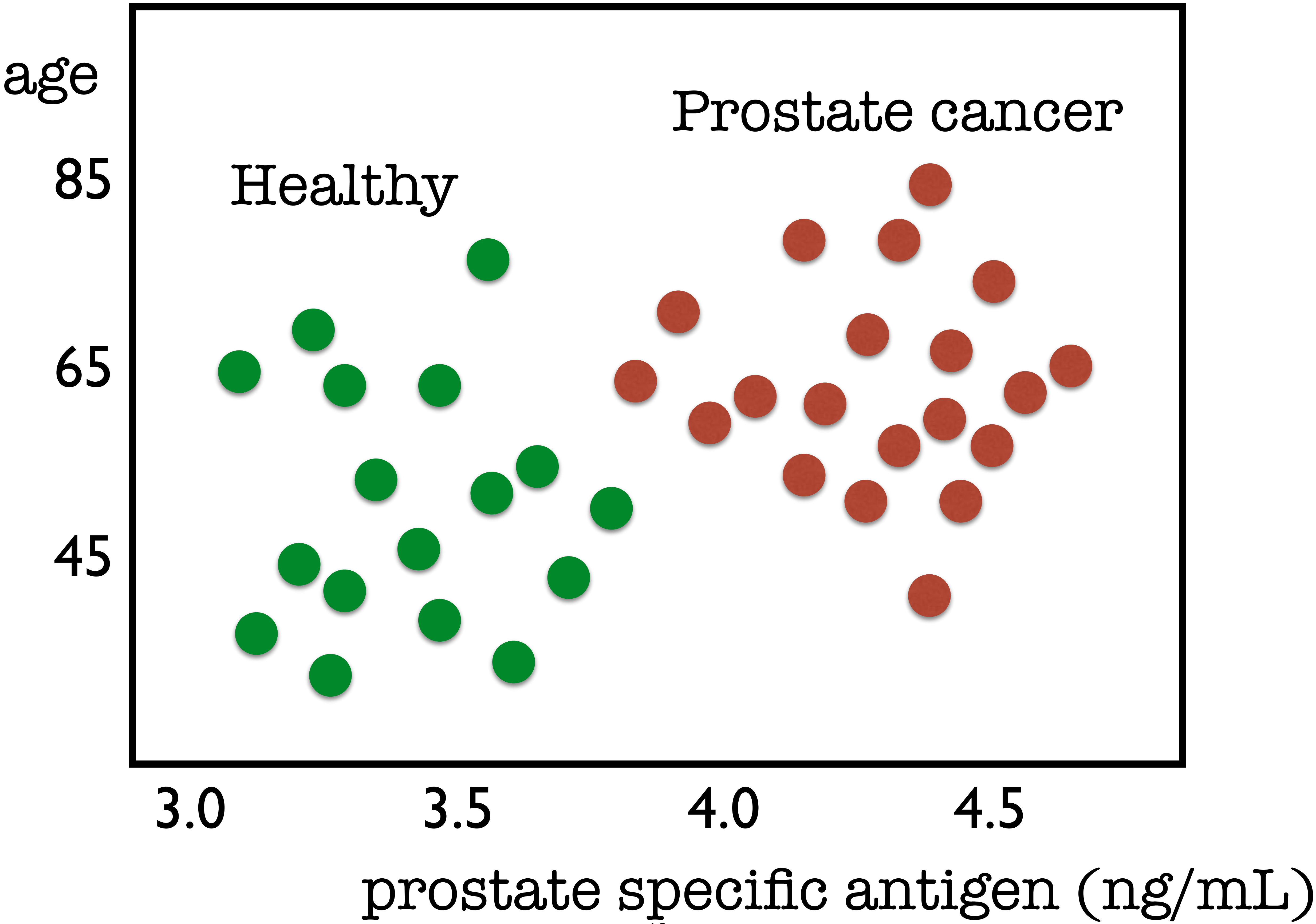
Age	Prostate Specific Antigen (a blood test reading)	Got Prostate Cancer
45	6.0 ng/mL	Yes
61	5.5 ng/mL	Yes
59	4.9 ng/mL	Yes
67	4.3 ng/mL	Yes
72	3.9 ng/mL	Yes
89	3.5 ng/mL	Yes
95	3.1 ng/mL	Yes
39	3.9 ng/mL	No
49	3.4 ng/mL	No
47	3.2 ng/mL	No
62	2.1 ng/mL	No
49	3.8 ng/mL	?
58	4.3 ng/mL	?
88	4.1 ng/mL	?
31	2.1 ng/mL	?

Age	Prostate Specific Antigen (a blood test reading)	Got Prostate Cancer
45	6.0 ng/mL	Yes
61	5.5 ng/mL	Yes
59	4.9 ng/mL	Yes
67	4.3 ng/mL	Yes
72	3.9 ng/mL	Yes
89	3.5 ng/mL	Yes
95	3.1 ng/mL	Yes
39	3.9 ng/mL	No
49	3.4 ng/mL	No
47	3.2 ng/mL	No
62	2.1 ng/mL	No
49	3.8 ng/mL	No?
58	4.3 ng/mL	borderline?
88	4.1 ng/mL	Yes?
31	2.1 ng/mL	No?

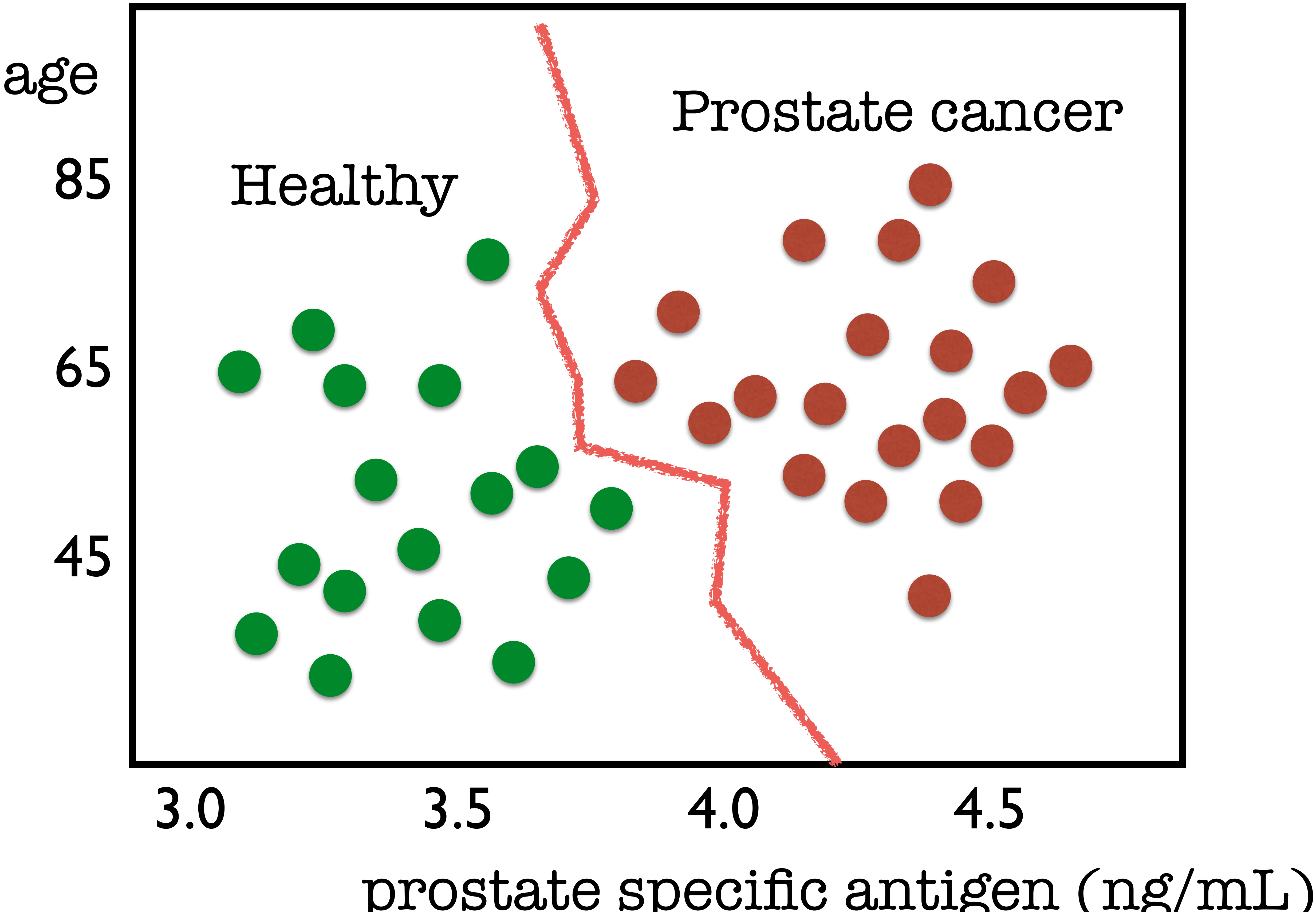
Prostate cancer prediction



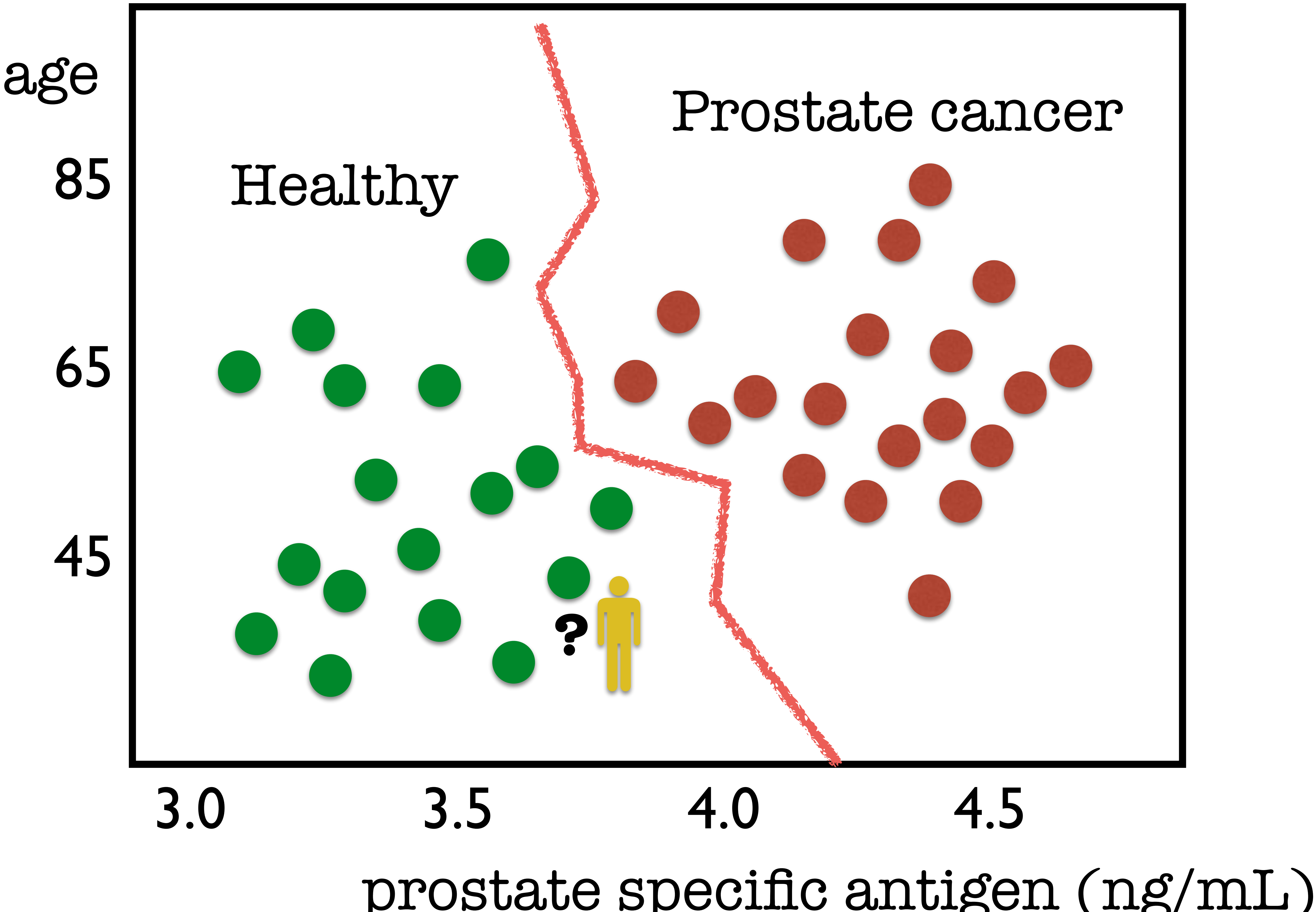
Prostate cancer prediction



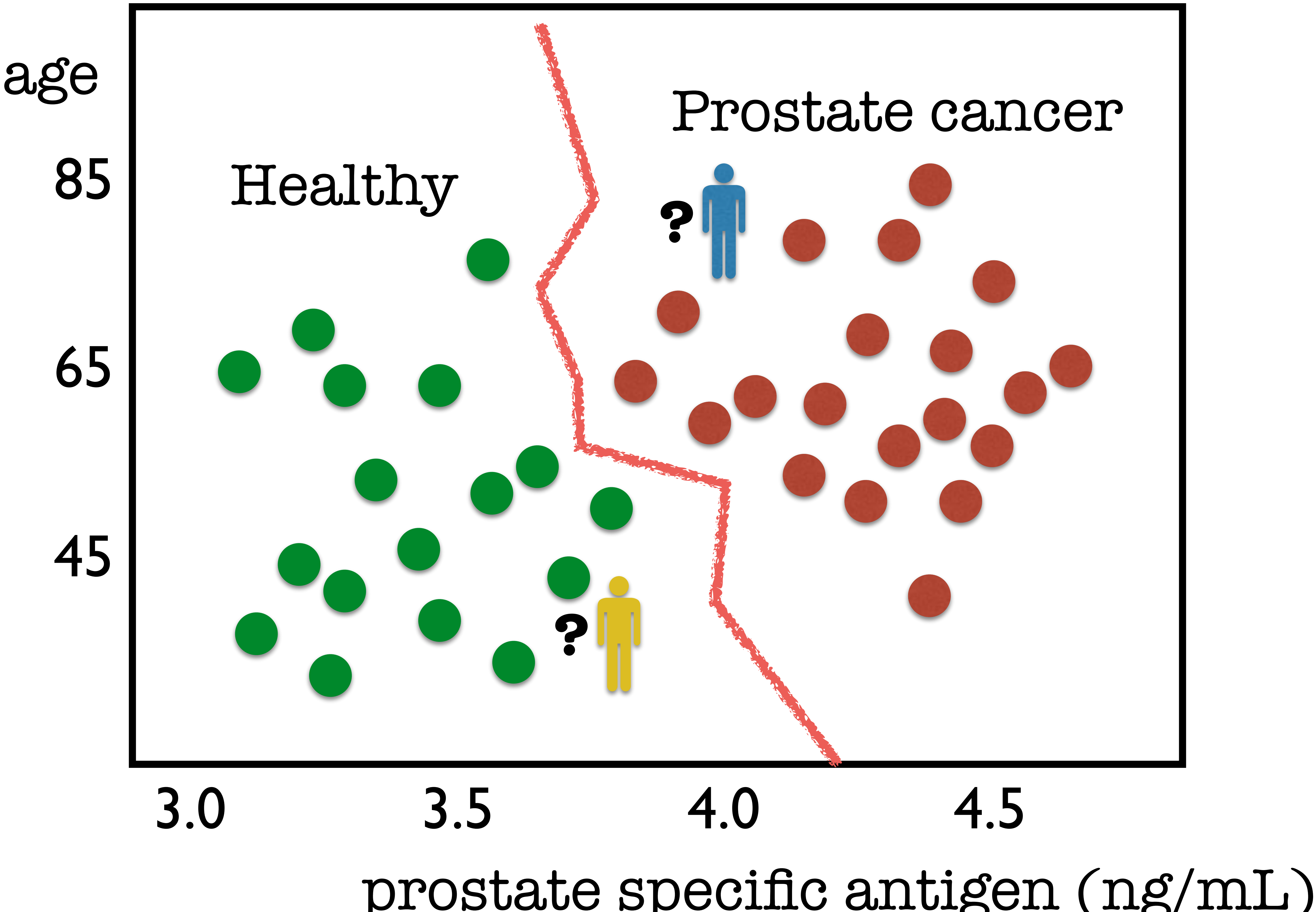
Prostate cancer prediction



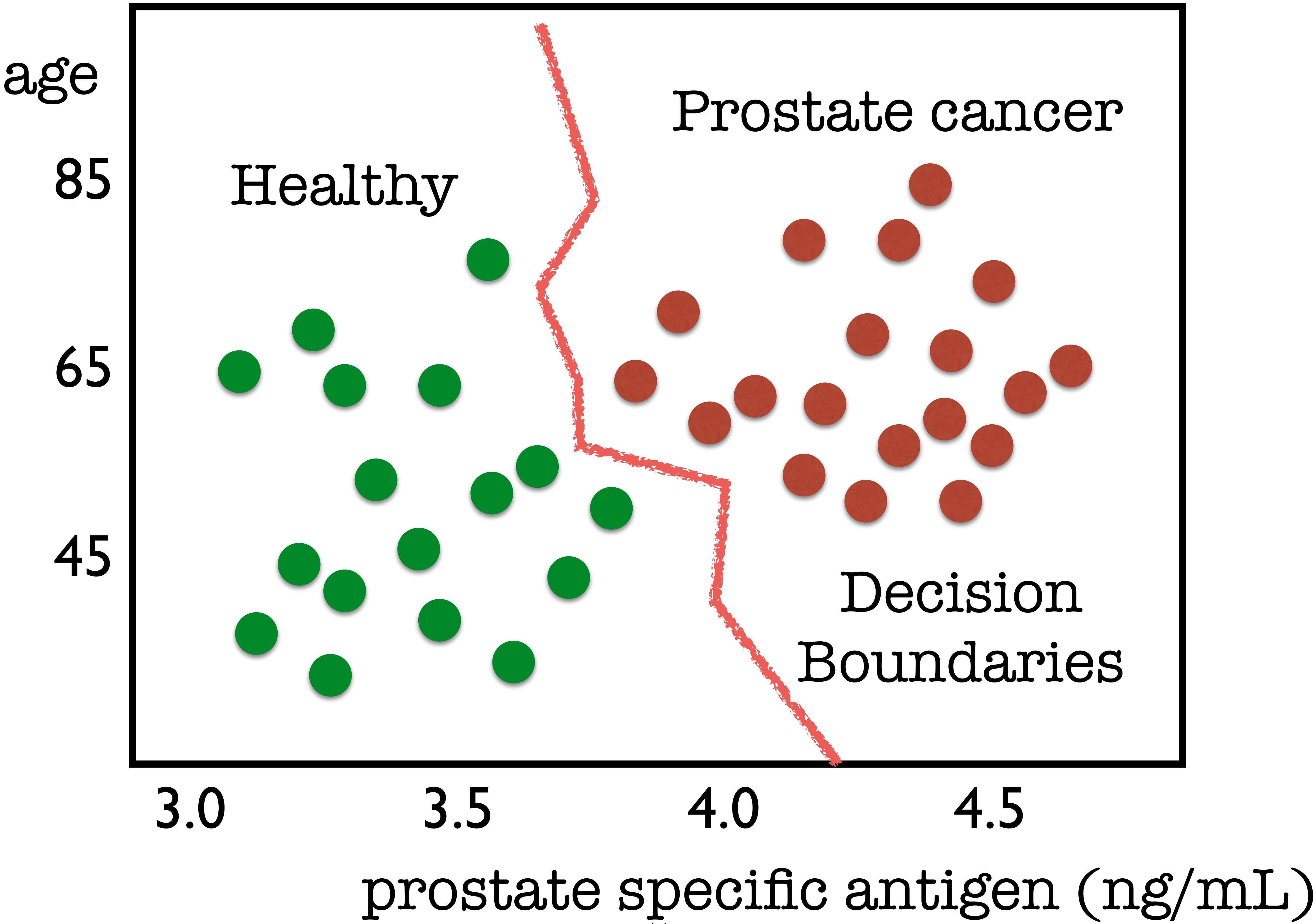
Prostate cancer prediction



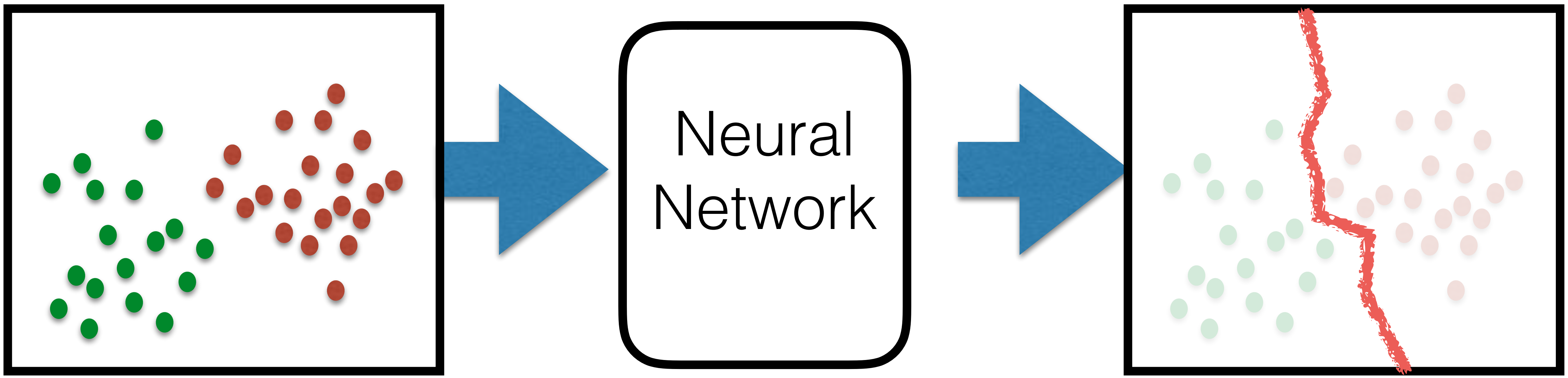
Prostate cancer prediction



Prostate cancer prediction



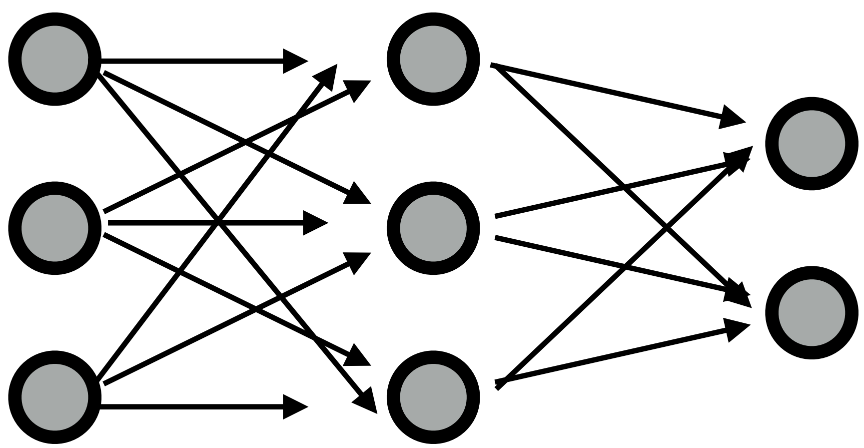
Supervised learning framework



Forward Propagation

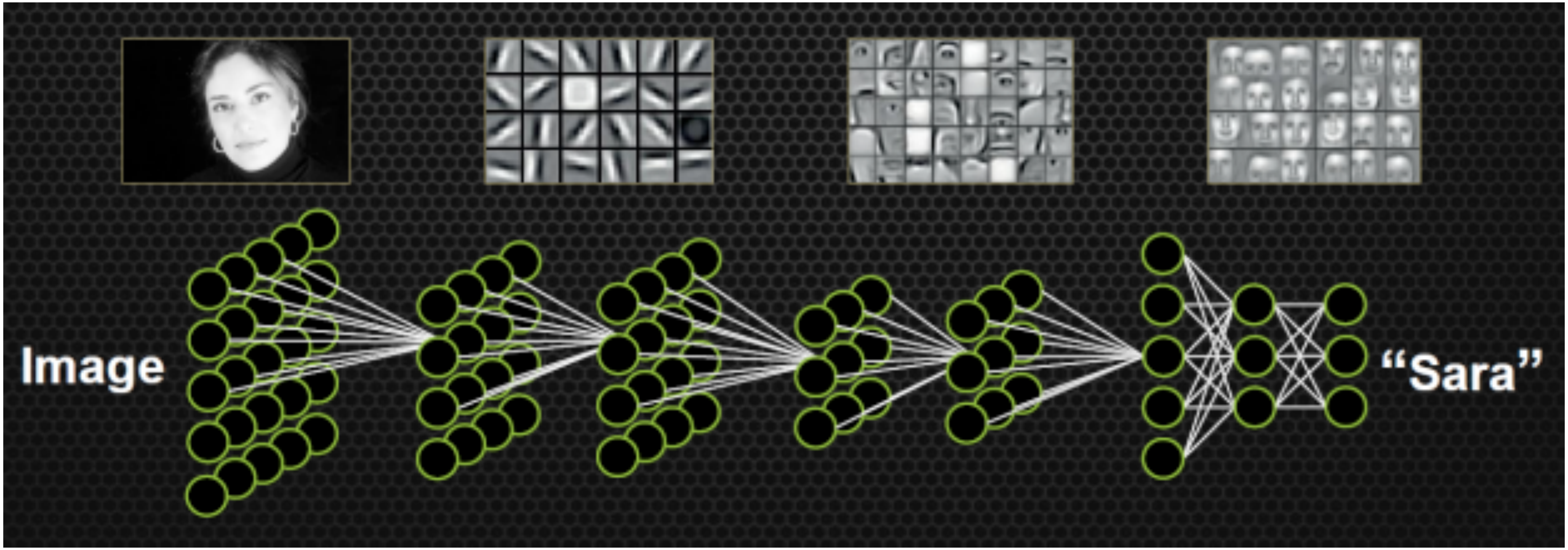
The network and information flow

feed in
data
into
first
layer



final
layer
presents
the
output
of network

There can be millions of connections



Notation

Let $x \in \mathbb{R}^d$ be the input space

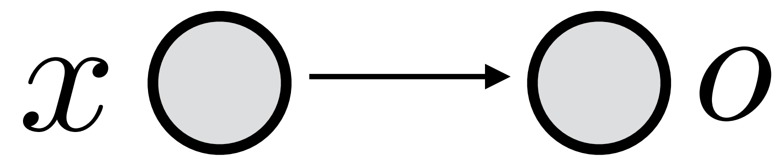
Let $y \in \mathbb{R}$ or $y \in \mathbb{N}$ be the label

Let $o \in \mathbb{R}$ be the output of the neural network

Simplest perceptron - linear activation function

$x \in \mathbb{R}$

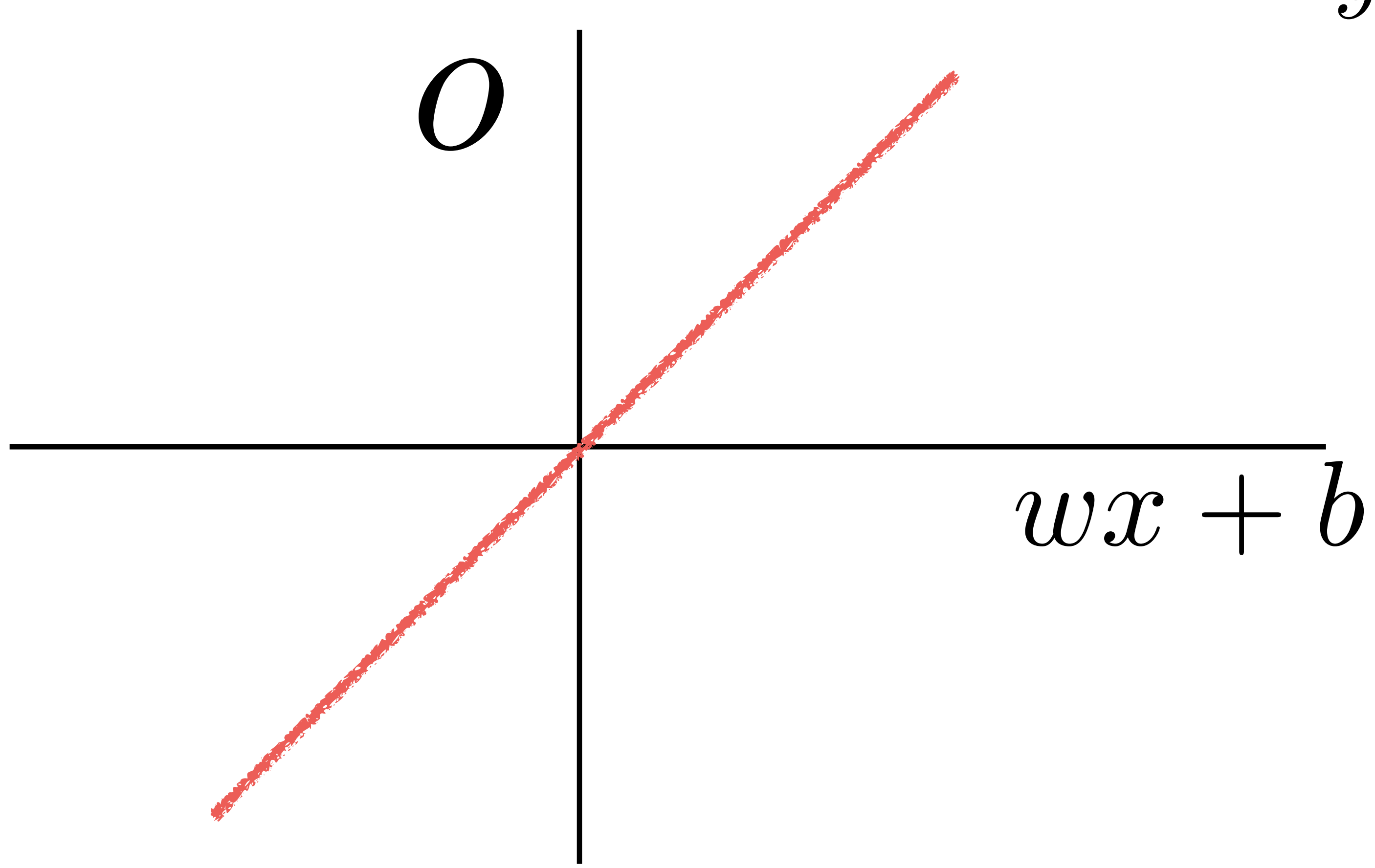
$$y = \begin{cases} 1 & \text{if } x > \tau \\ 0 & \text{otherwise} \end{cases}$$



$$o = wx + b$$

$$y = \begin{cases} 1 & \text{if } wx + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

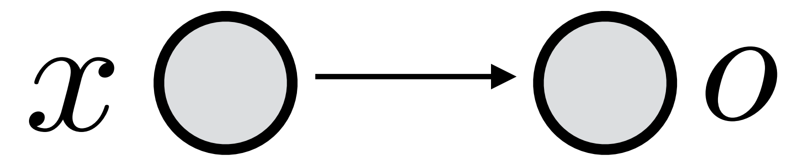
$$w\tau + b = 0$$



Simplest perceptron - rectilinear activation function

$$x \in \mathbb{R}$$

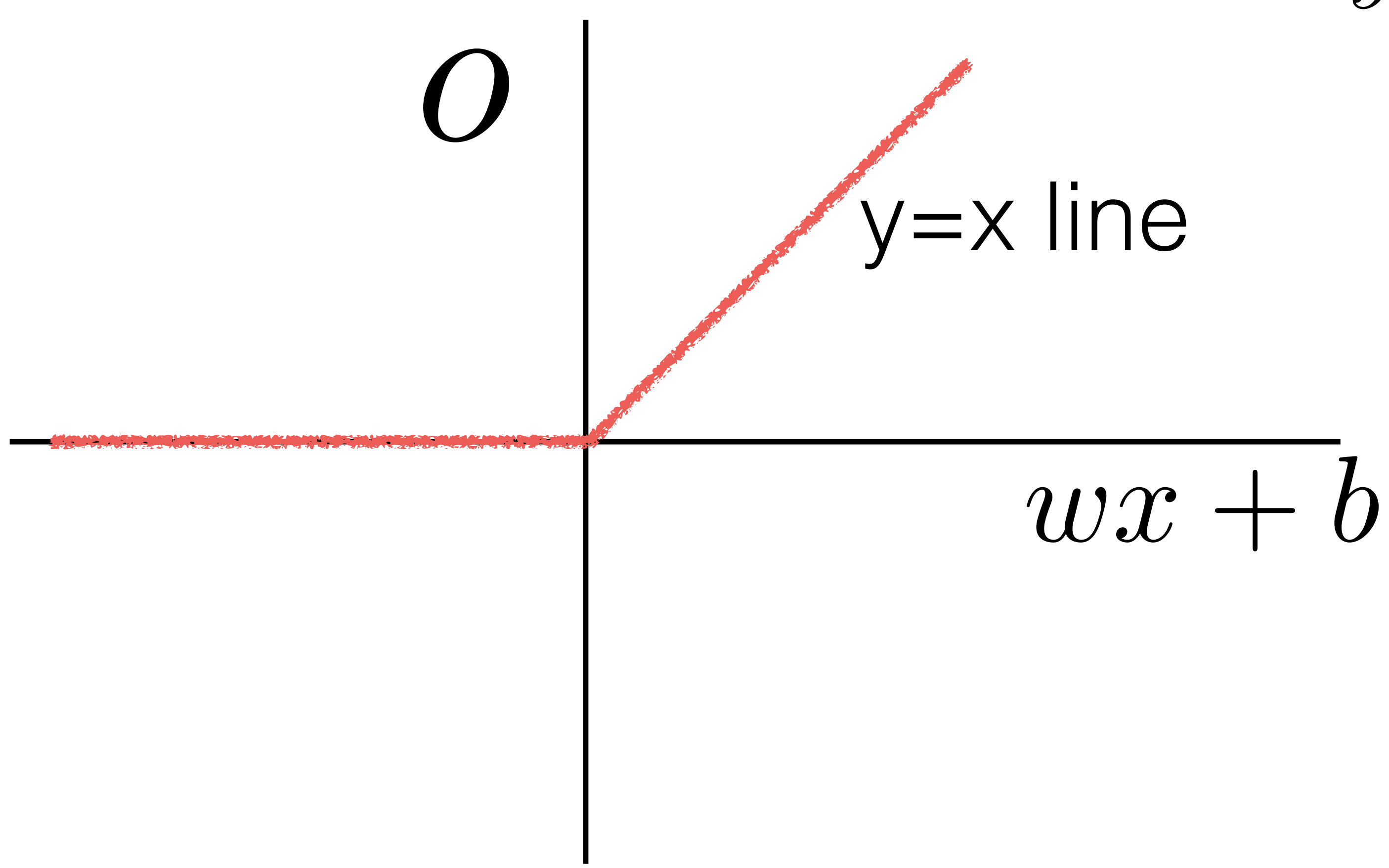
$$y = \begin{cases} 1 & \text{if } x > \tau \\ 0 & \text{otherwise} \end{cases}$$



$$o = ReLu(wx + b)$$

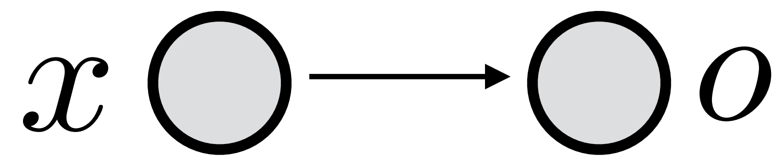
$$y = \begin{cases} 1 & \text{if } wx + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$w\tau + b = 0$$



Simplest perceptron - sigmoid activation function

$x \in \mathbb{R}$

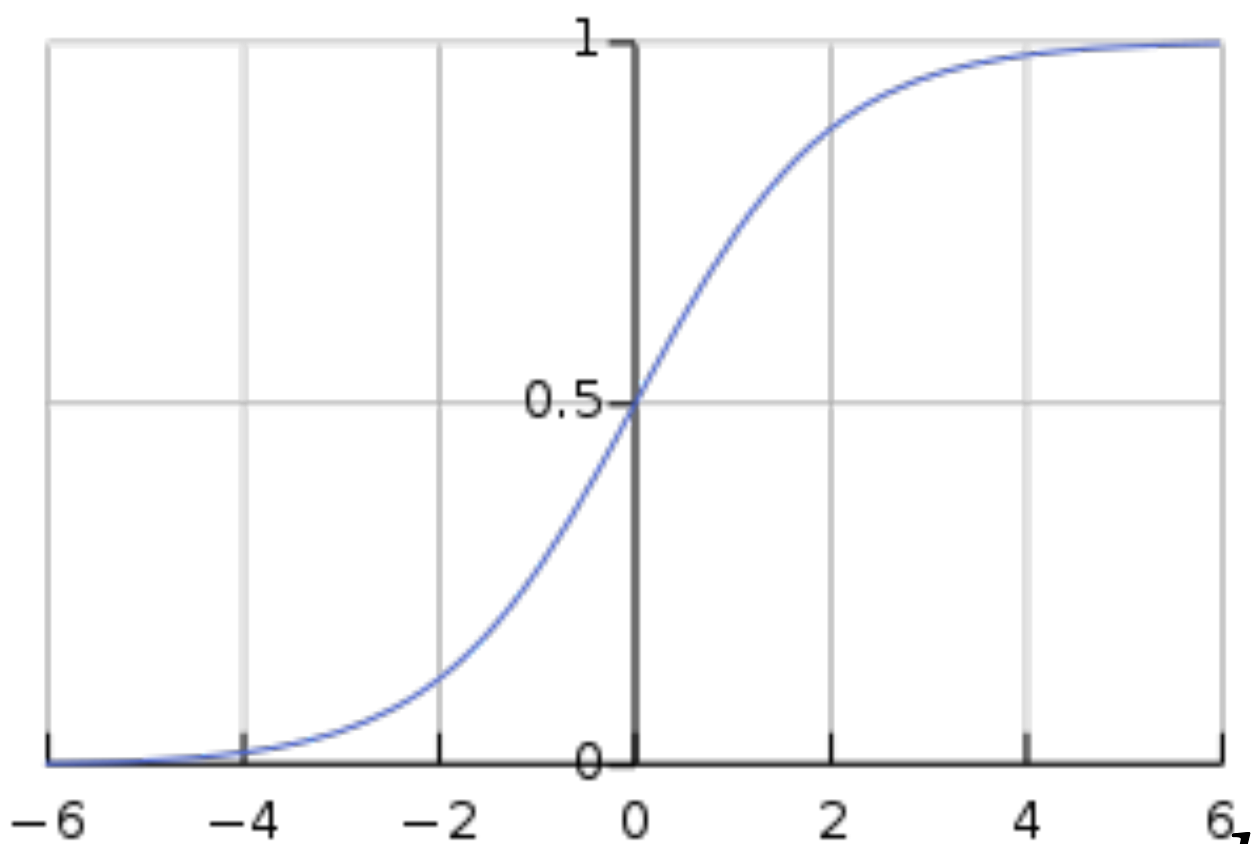


$$y = \begin{cases} 1 & \text{if } x > \tau \\ 0 & \text{otherwise} \end{cases}$$

$$o = \frac{1}{1 + \exp(-wx - b)}$$

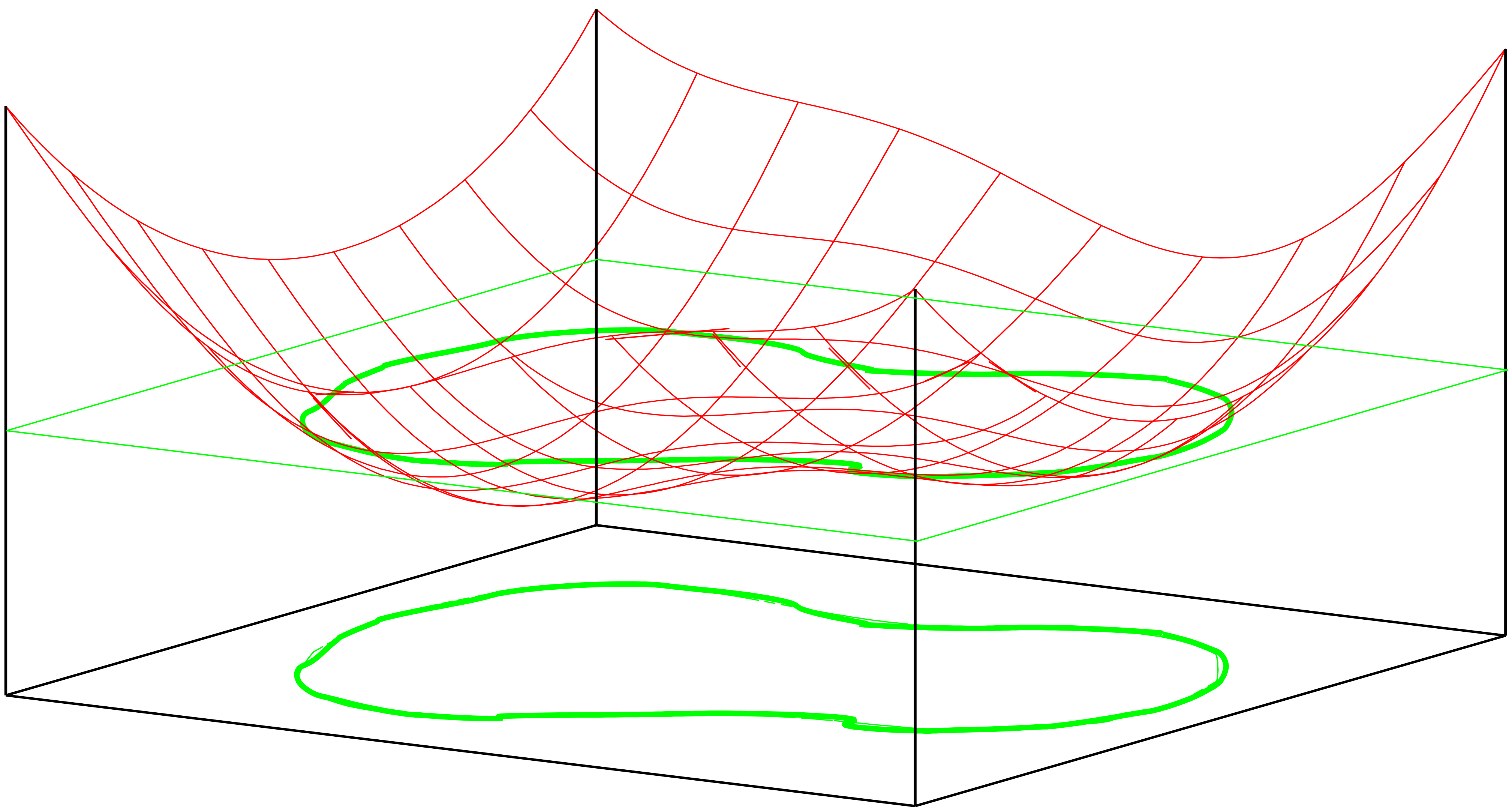
$$y = \begin{cases} 1 & \text{if } wx + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$w\tau + b = 0$$



$wx + b$

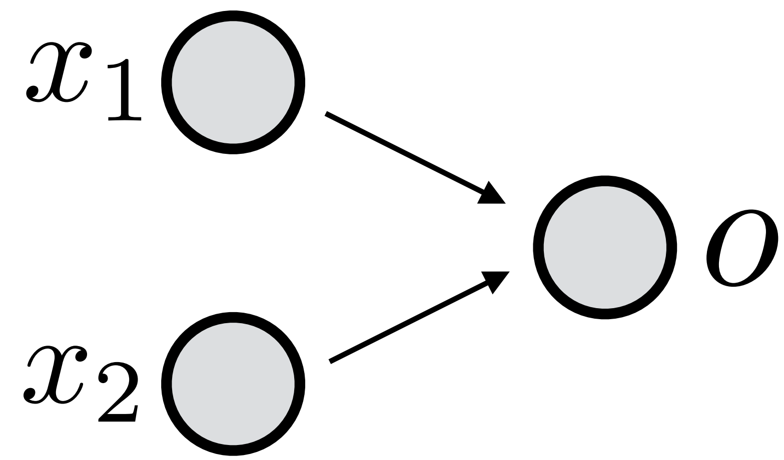
Concept of level sets

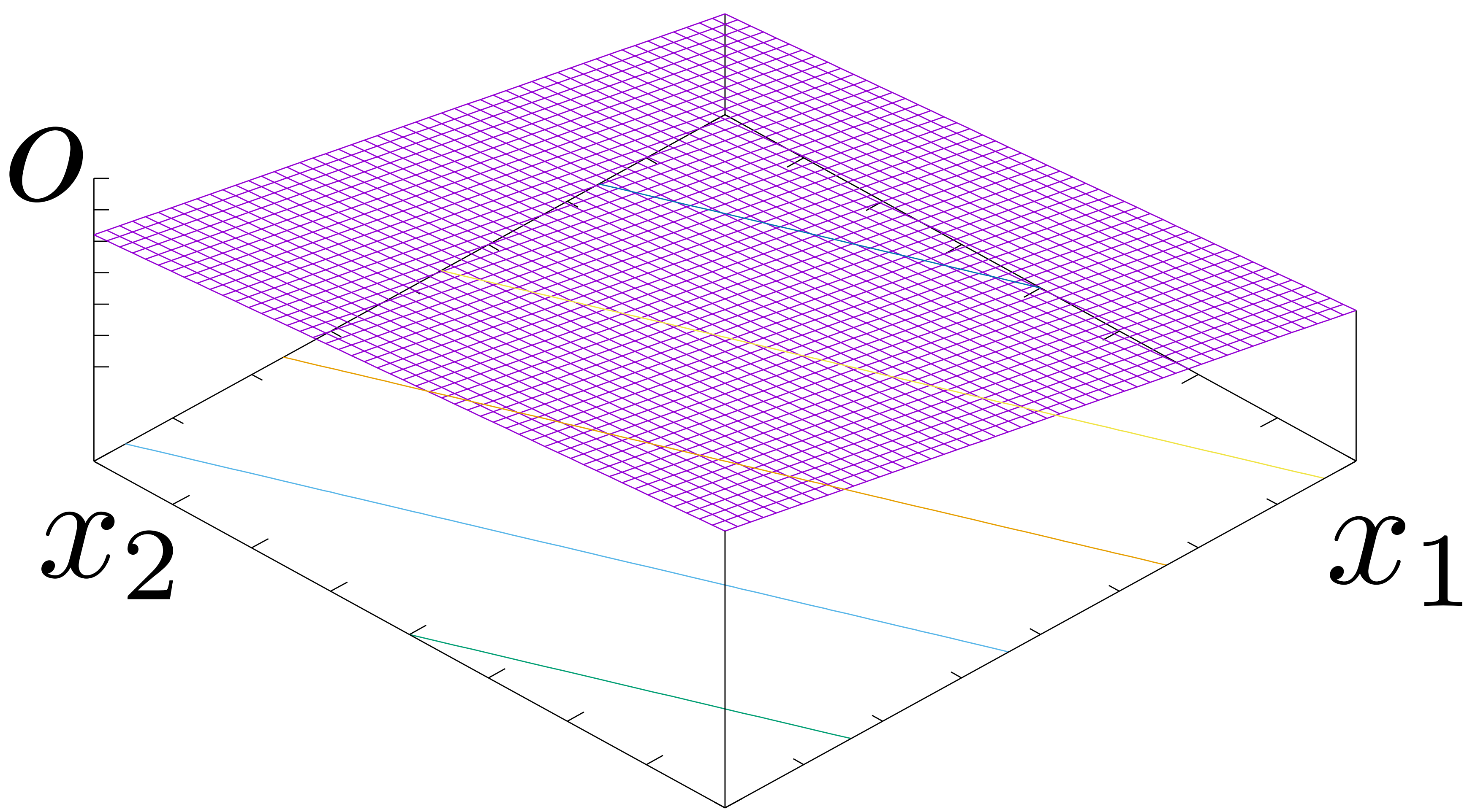


Next to simplest

$$x = (x_1, x_2) \in \mathbb{R}^2$$

$$o = \sigma(w_1 x_1 + w_2 x_2 + b)$$

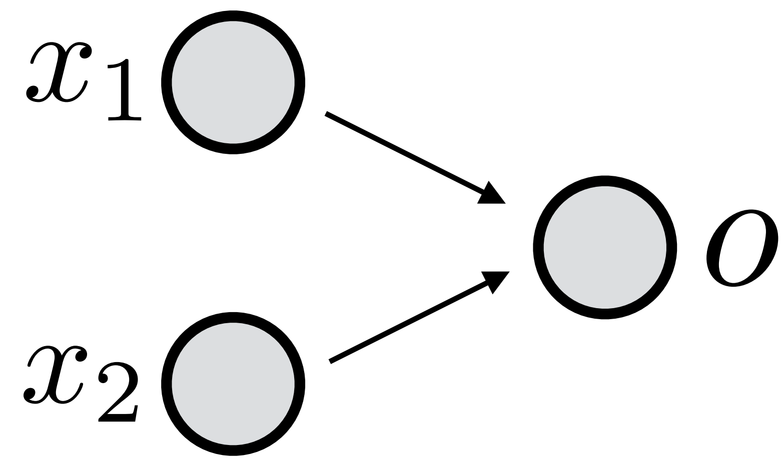


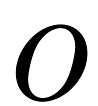


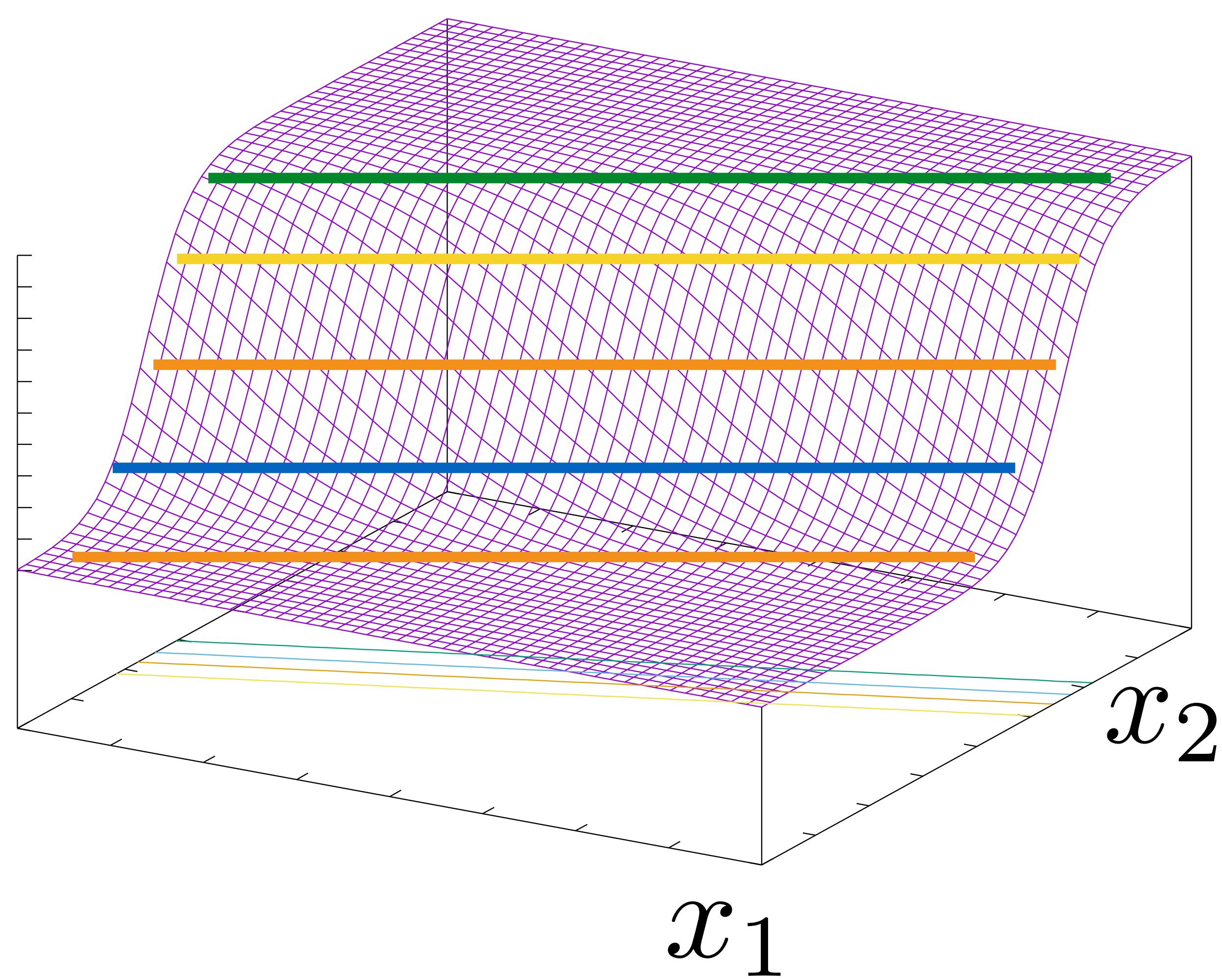
Next to simplest

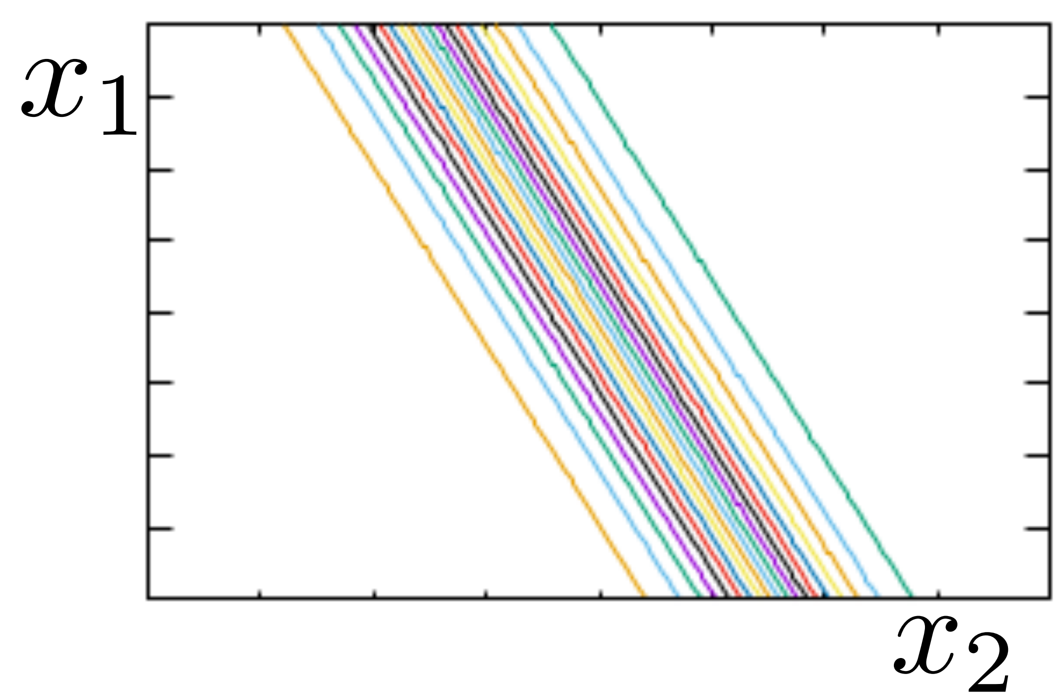
$$x = (x_1, x_2) \in \mathbb{R}^2$$

$$o = \sigma(w_1 x_1 + w_2 x_2 + b)$$

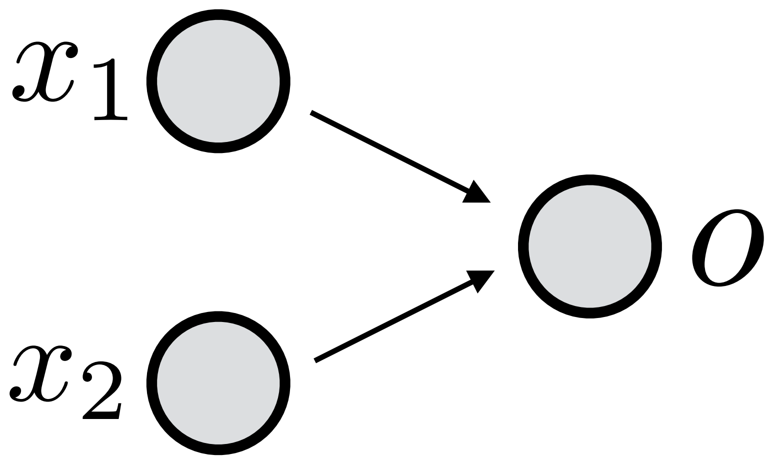






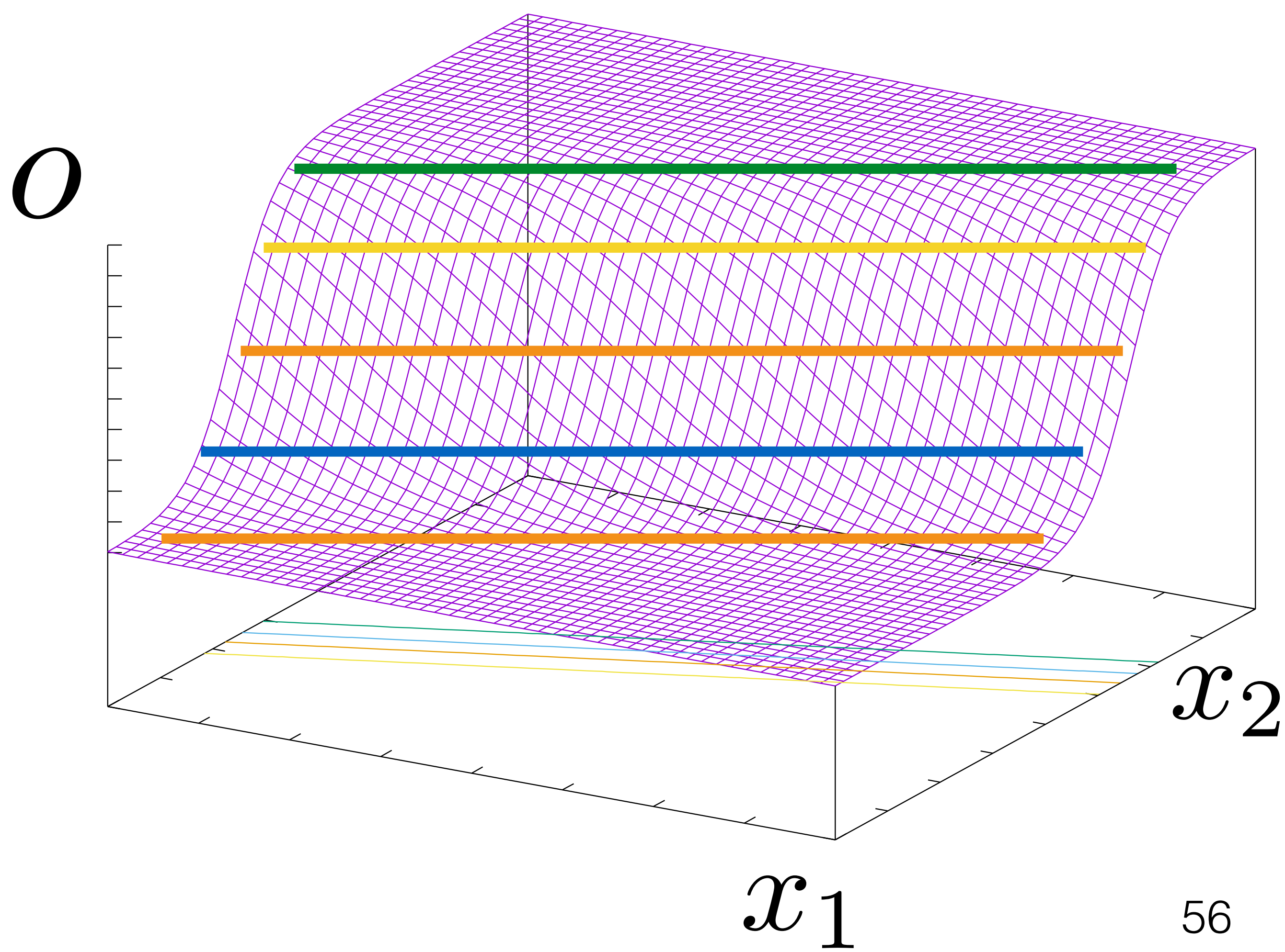


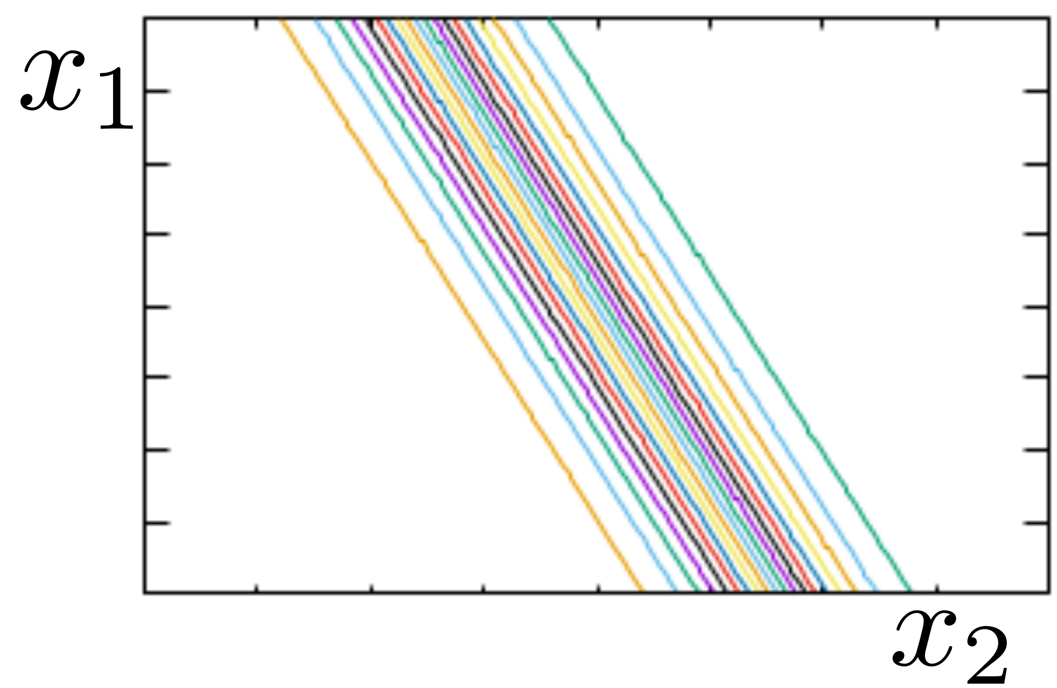
Network view and logistic regression view



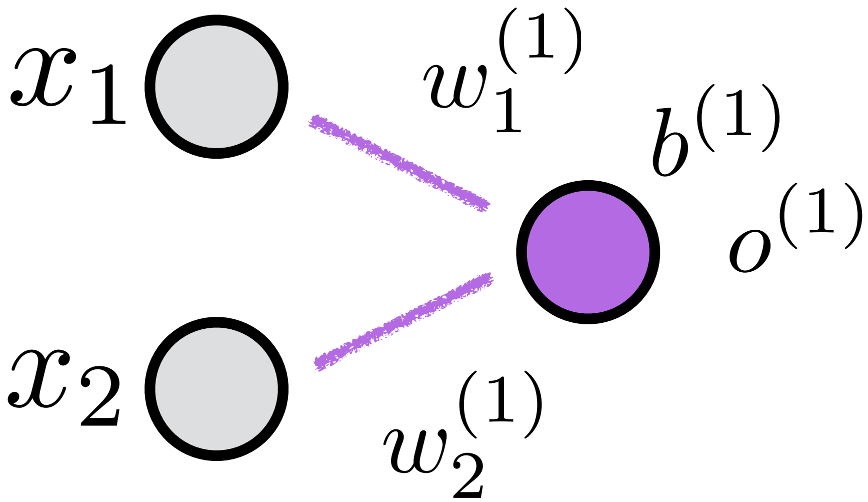
$$o = \frac{1}{1+\exp(-w_1x_1 - w_2x_2 - b)}$$

Logistic regression

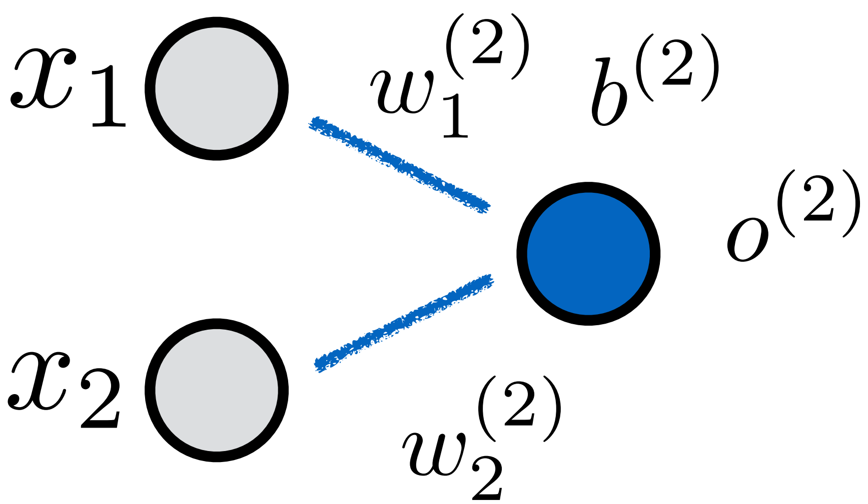




Stacked up

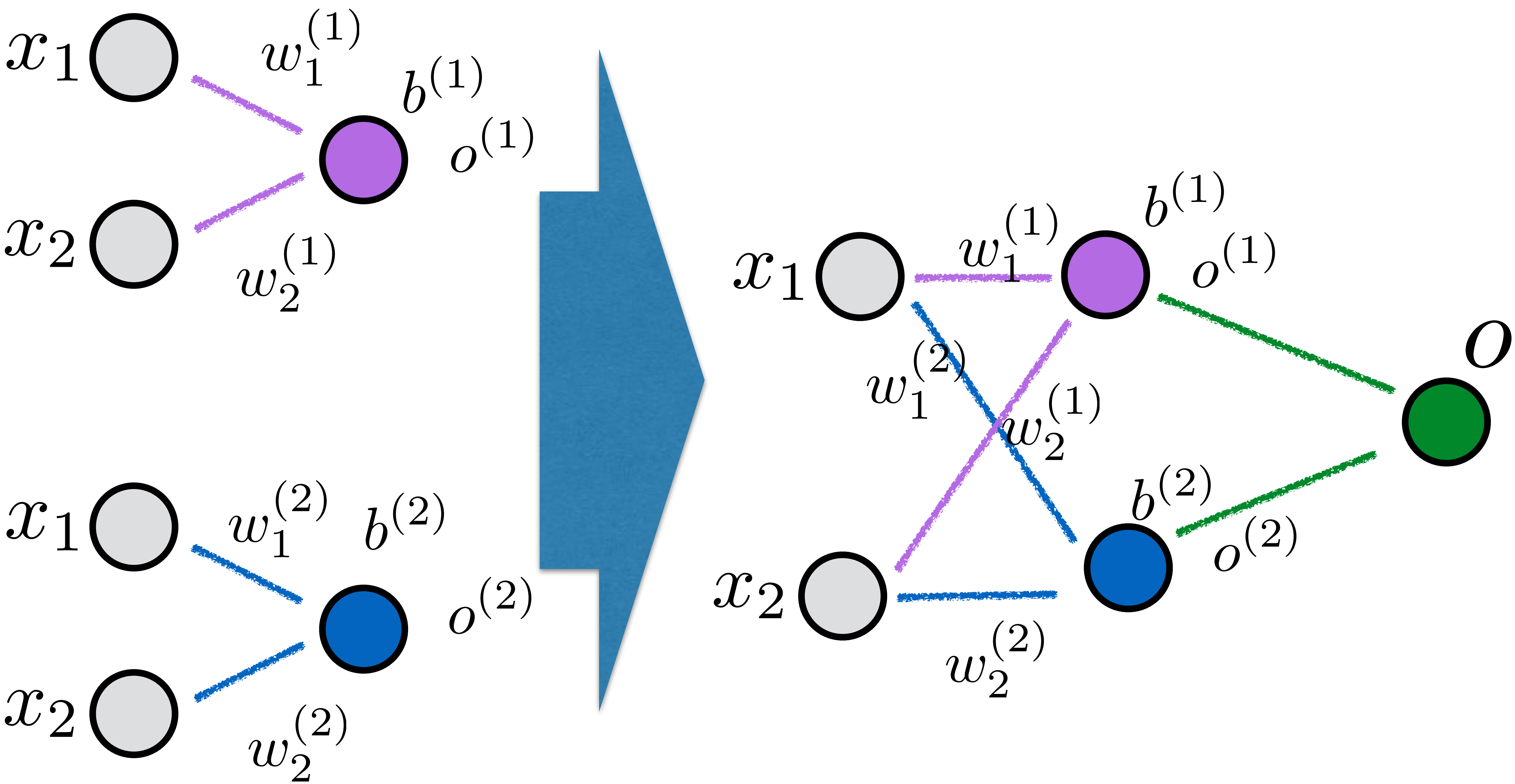


$$o^{(1)} = w_1^{(1)} x_1 + w_2^{(1)} x_2 + b^{(1)}$$

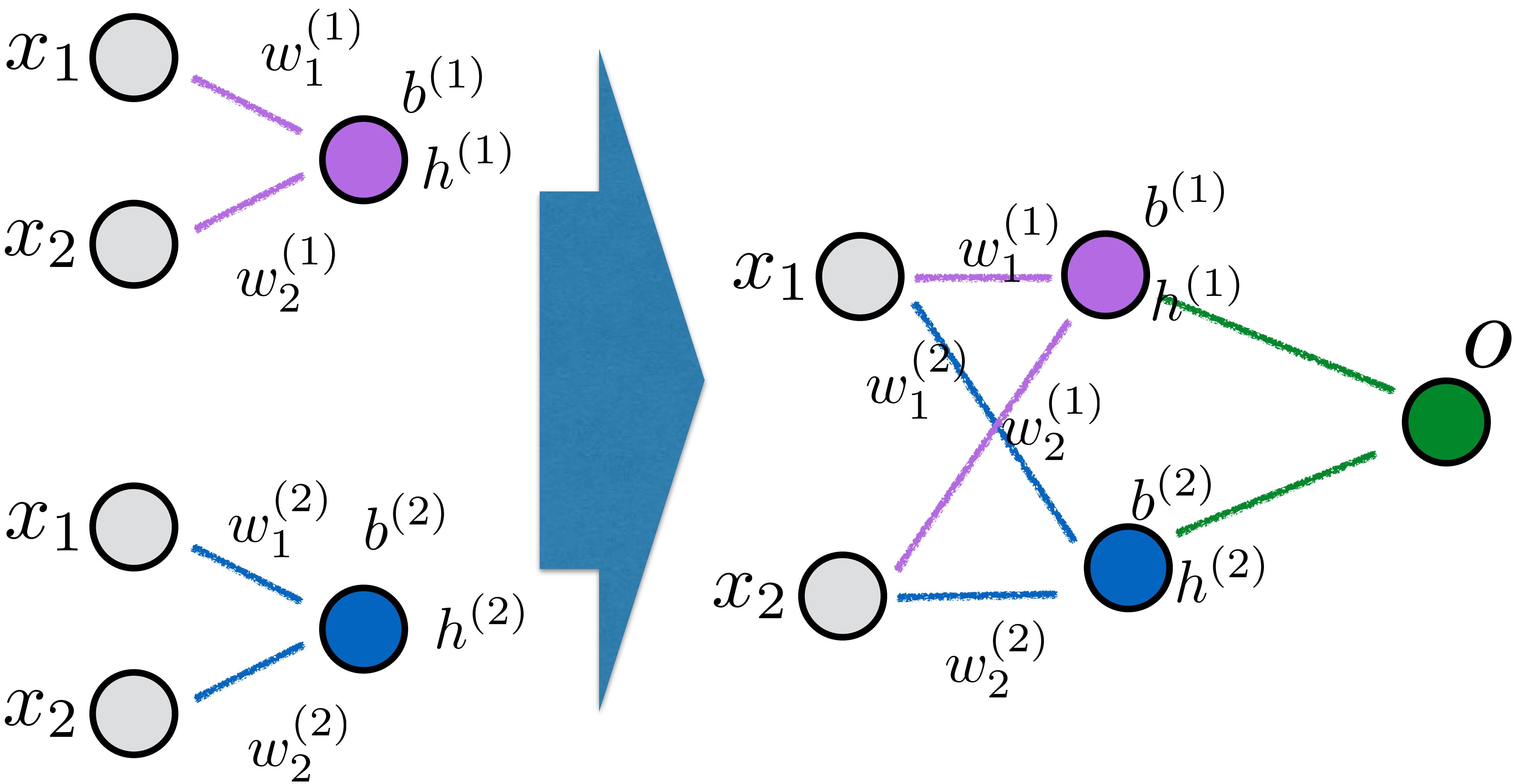


$$o^{(2)} = w_1^{(2)} x_1 + w_2^{(2)} x_2 + b^{(2)}$$

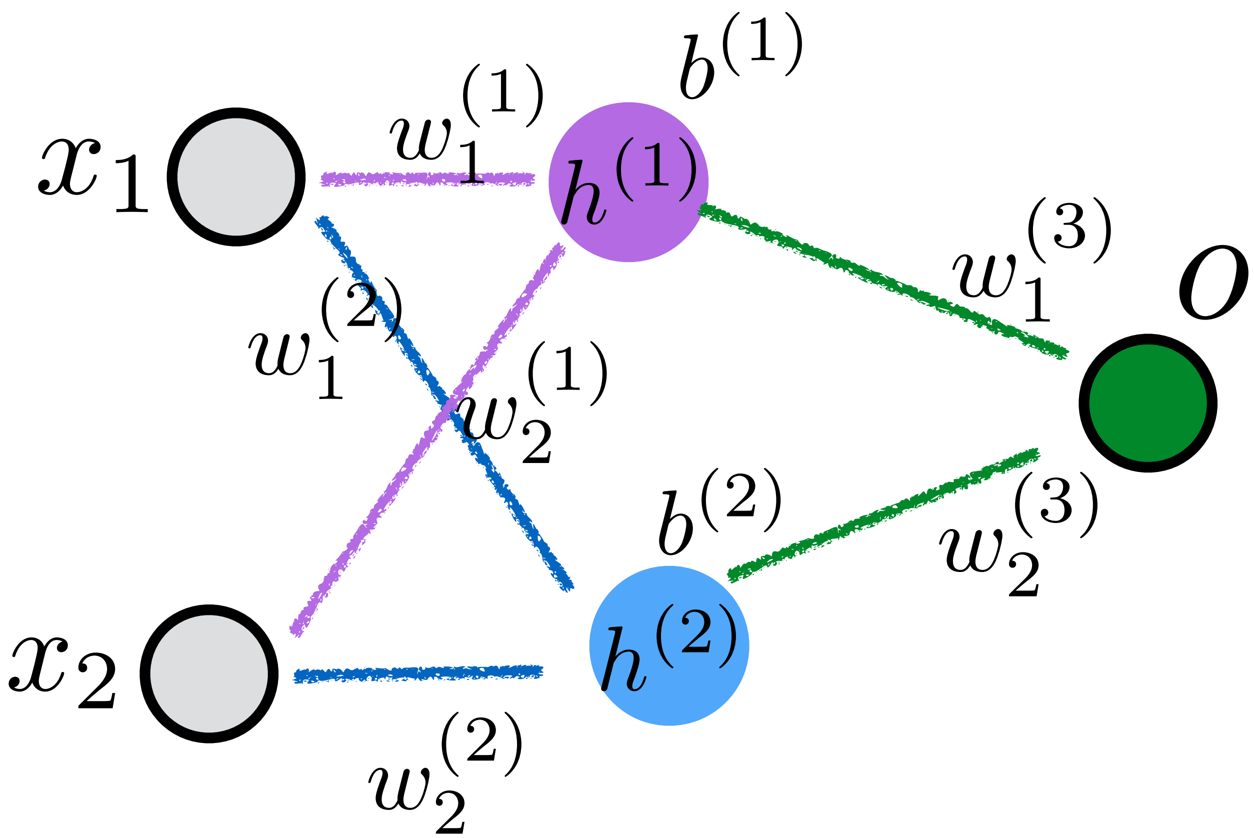
Stacked up


x_1
x_2
$w_1^{(1)}$
$w_2^{(1)}$
$b^{(1)}$
$o^{(1)}$
x_1
x_2
$w_1^{(2)}$
$w_2^{(2)}$
$b^{(2)}$
$o^{(2)}$
x_1
x_2
$w_1^{(1)}$
$w_1^{(2)}$
$w_2^{(1)}$
$w_2^{(2)}$
$b^{(1)}$
$o^{(1)}$
$b^{(2)}$
$o^{(2)}$
o

Stacked up


x_1
$w_1^{(1)}$
$b^{(1)}$
$h^{(1)}$
x_2
$w_2^{(1)}$
x_1
$w_1^{(2)}$
$b^{(2)}$
$h^{(2)}$
x_2
$w_2^{(2)}$
x_1
$w_1^{(1)}$
$b^{(1)}$
$h^{(1)}$
o
$w_1^{(2)}$
$w_2^{(1)}$
$b^{(2)}$
x_2
$h^{(2)}$
$w_2^{(2)}$

Stacked up

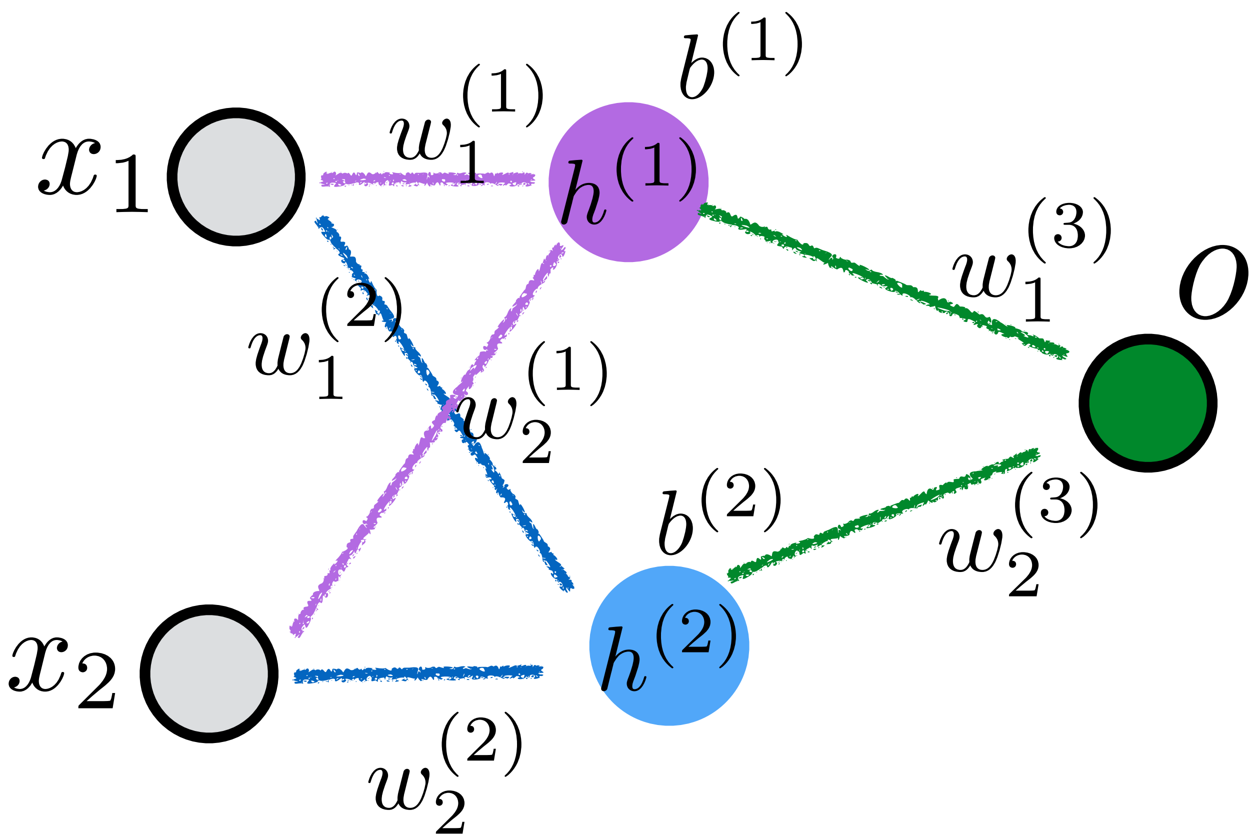


$$h^{(1)} = w_1^{(1)} x_1 + w_2^{(1)} x_2 + b^{(1)}$$

$$h^{(2)} = w_1^{(2)} x_1 + w_2^{(2)} x_2 + b^{(2)}$$

$$o = w_1^{(3)} h^{(1)} + w_2^{(3)} h^{(2)} + b^{(3)}$$

Stacked up with general activation function



$$h^{(1)} = \sigma(w_1^{(1)} x_1 + w_2^{(1)} x_2 + b^{(1)})$$

$$h^{(2)} = \sigma(w_1^{(2)} x_1 + w_2^{(2)} x_2 + b^{(2)})$$

$$o = \sigma(w_1^{(3)} h^{(1)} + w_2^{(3)} h^{(2)} + b^{(3)})$$

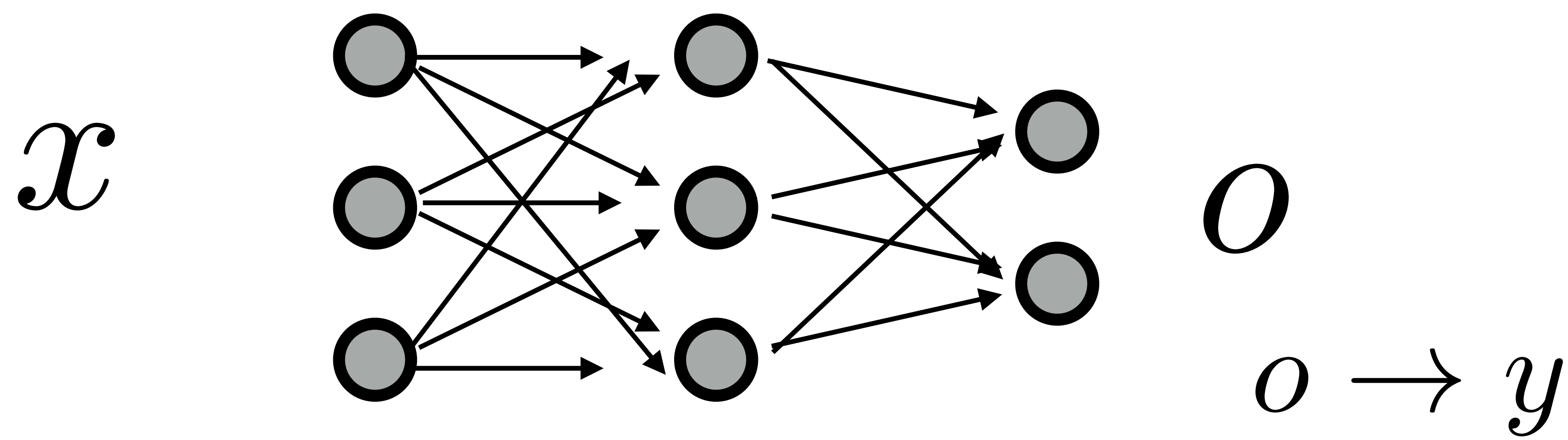
A general objective

Tweak the output of neural network to be as similar to the label as possible.

$$o \approx y$$

How to tweak? Adjust the weights

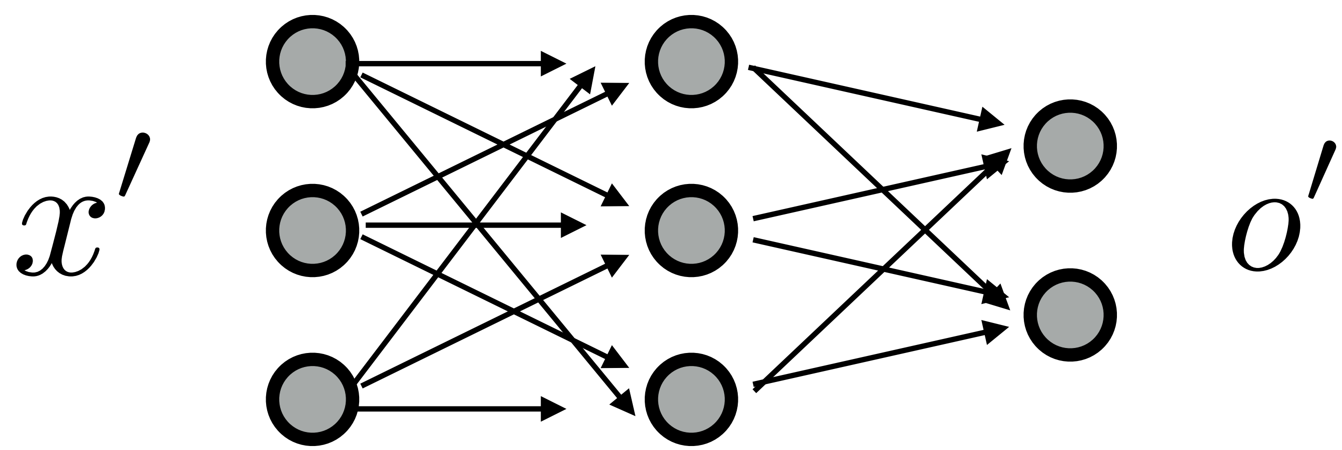
Given an example (x, y)



For this example, the neural network ‘behaves well’

After the first example, give another example

$$(x', ?)$$

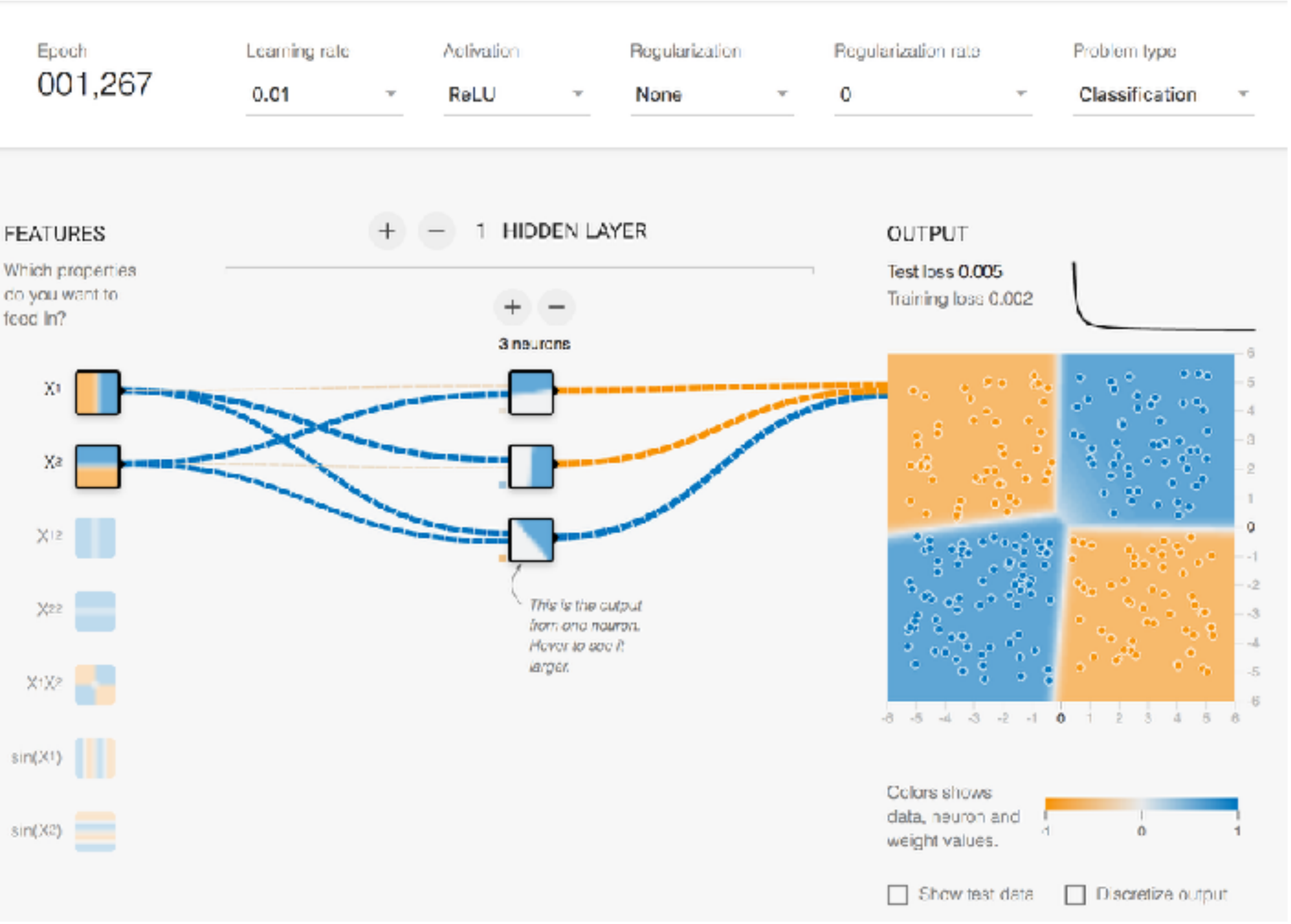


Now we say for a well behaved neural network

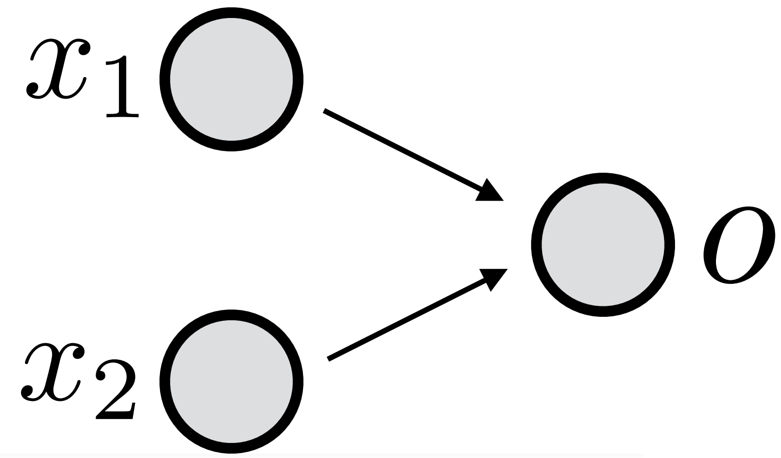
$$o' = y'$$

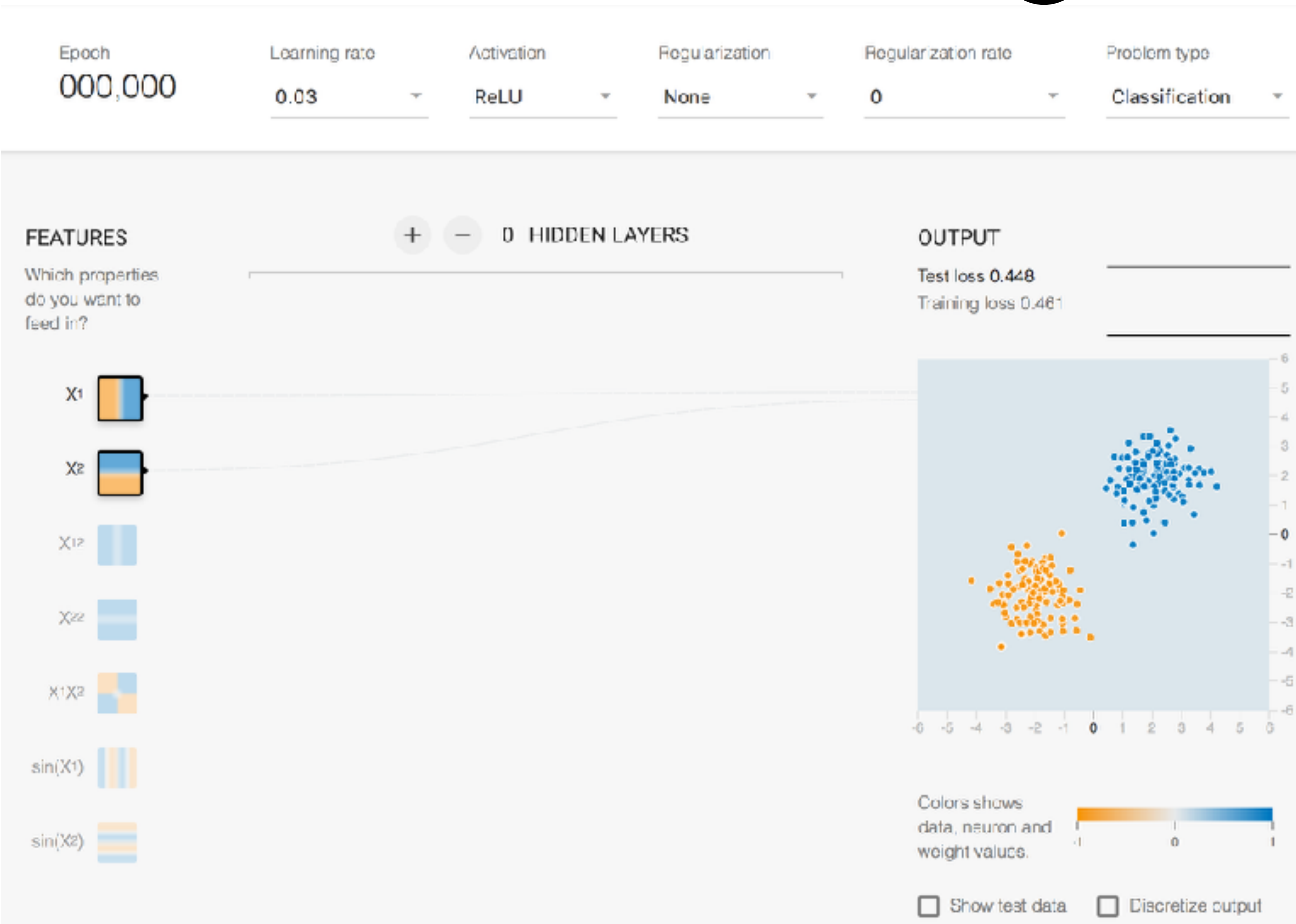
Animation @ playground search for "playground tensorflow"

introduction

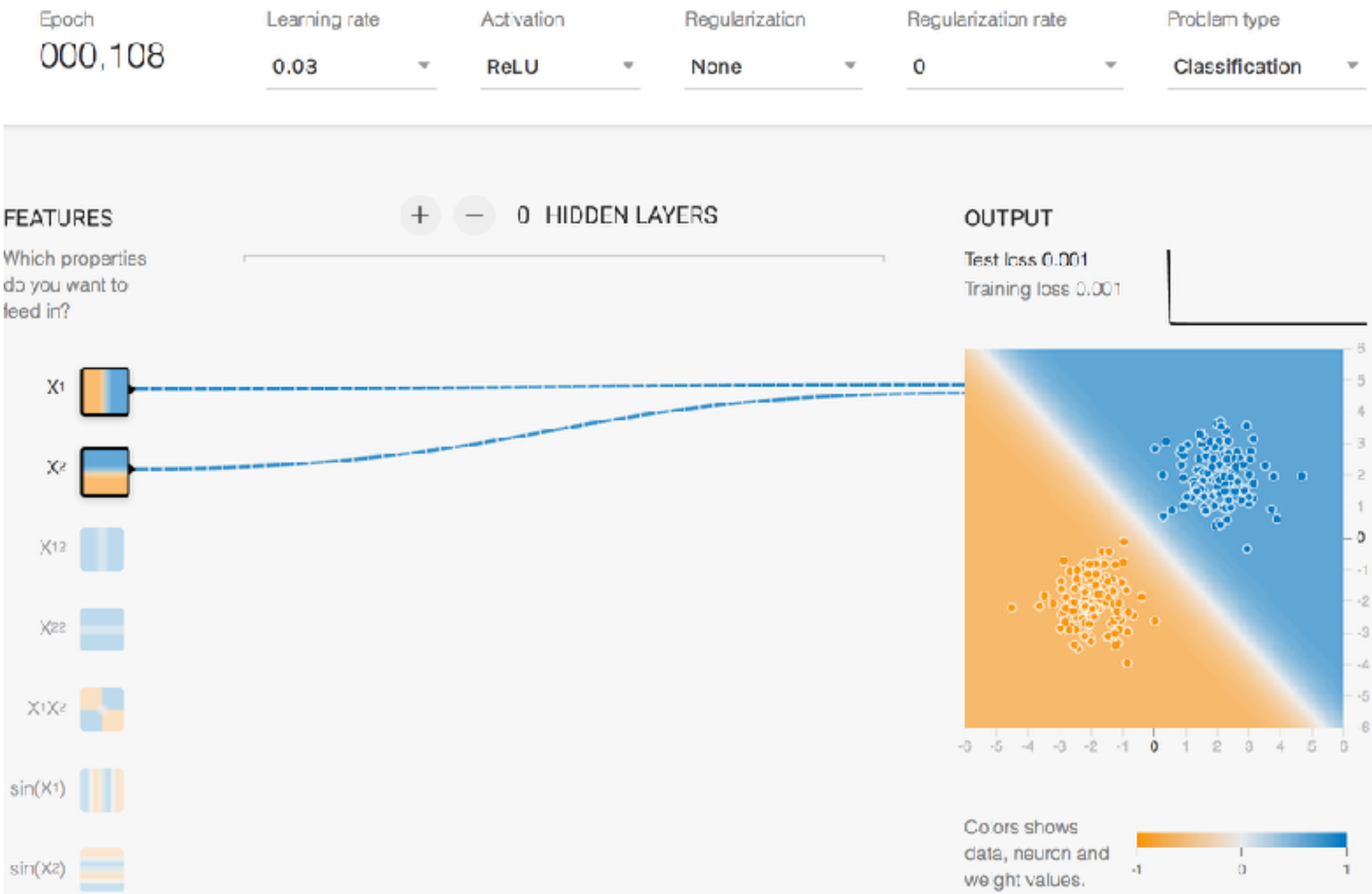


question


x_1
x_2
o

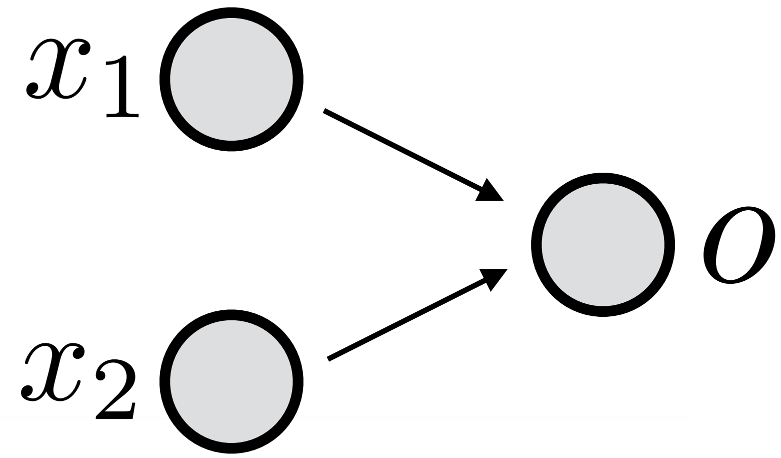

Epoch
000,000
Learning rate
0.03
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
X1
X2
0 HIDDEN LAYERS
OUTPUT
Test loss 0.448
Training loss 0.461
Colors shows data, neuron and weight values.
Show test data
Discretize output

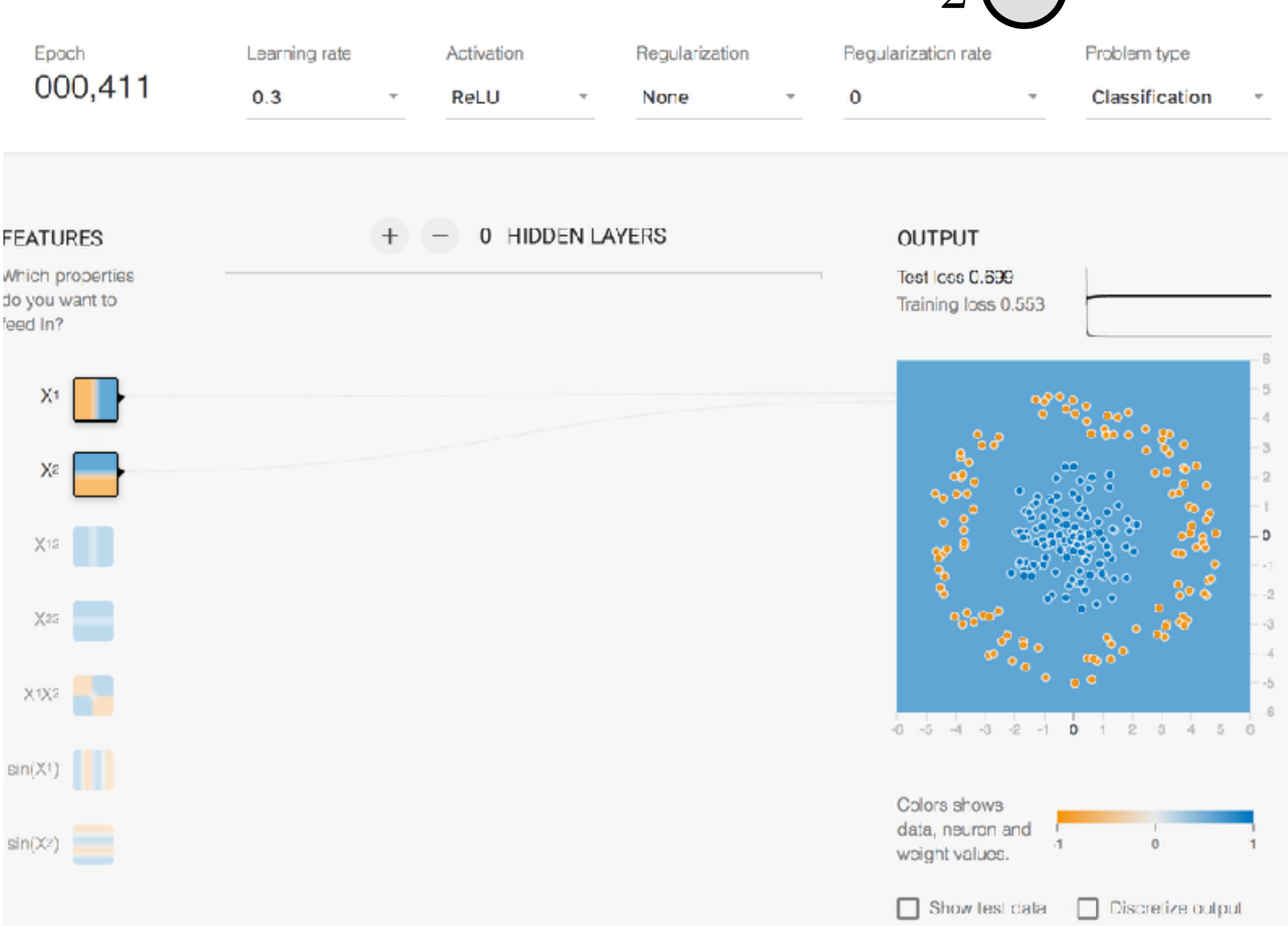
answer



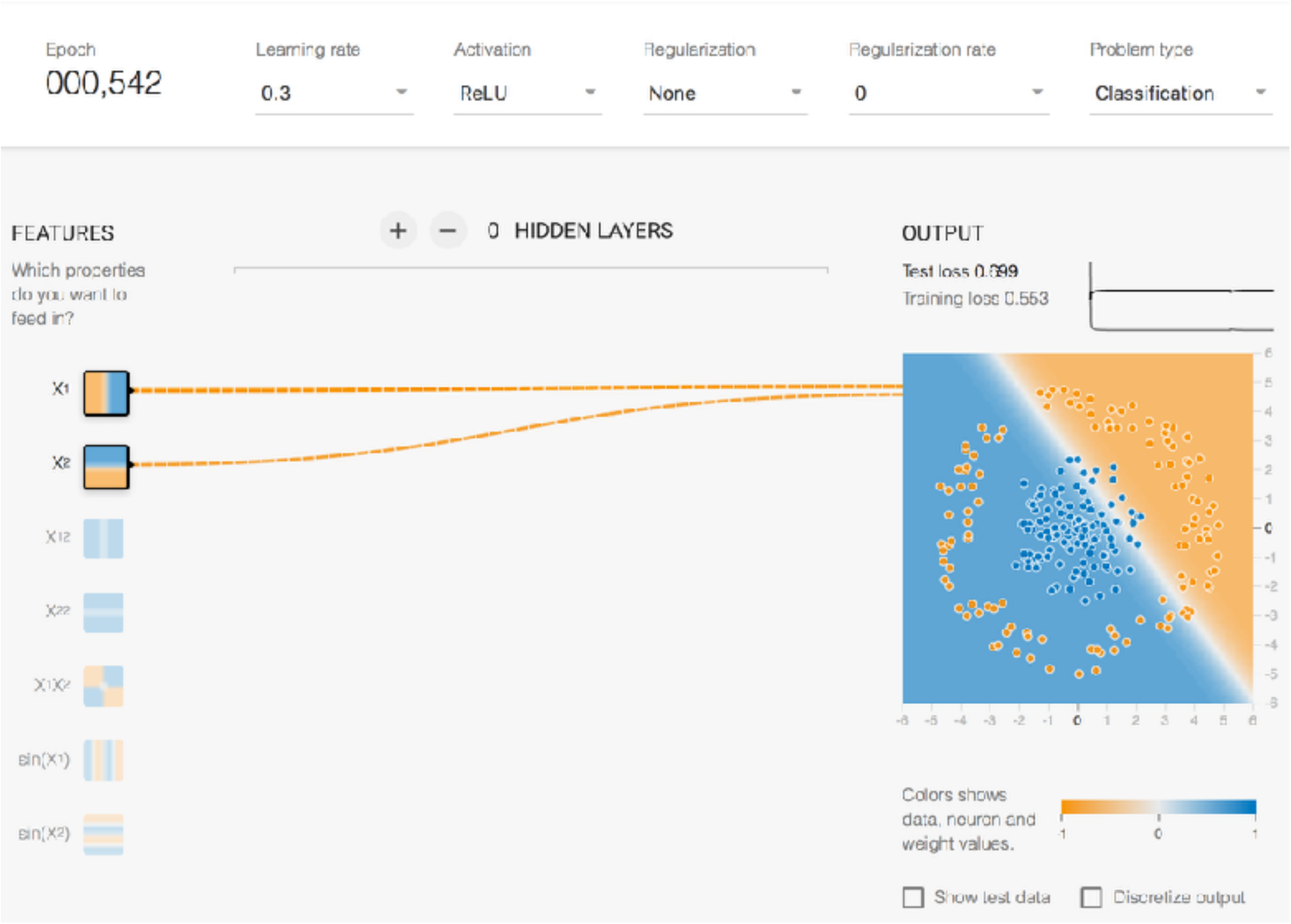
Epoch
000,108
Learning rate
0.03
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
0 HIDDEN LAYERS
OUTPUT
Test loss 0.001
Training loss 0.001
Colors shows data, neuron and weight values.

question

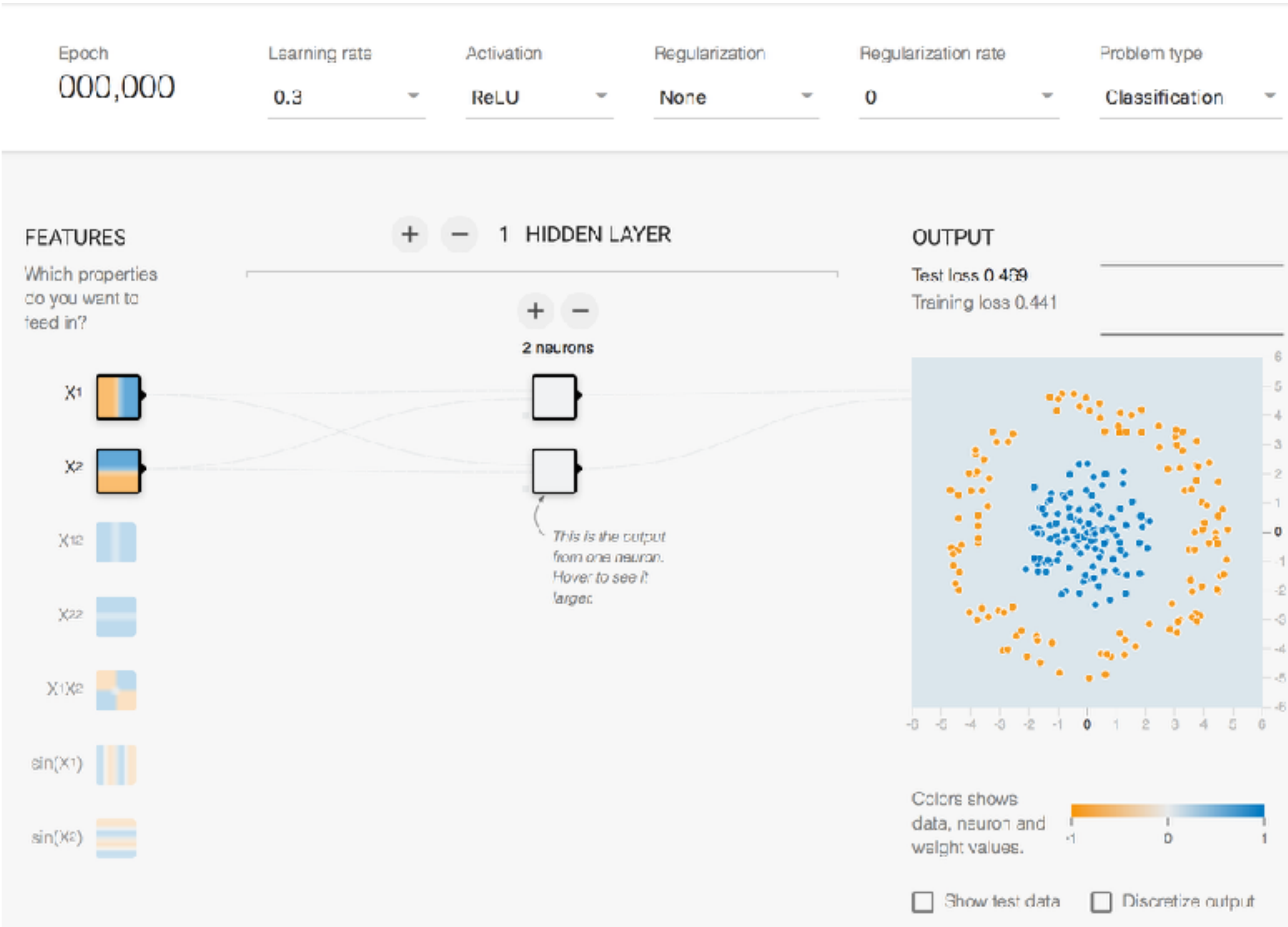

x_1
x_2
o


Epoch
000,411
Learning rate
0.3
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
0 HIDDEN LAYERS
OUTPUT
Training loss 0.553
Colors shows data, neuron and weight values.
Show test data
Discretize output

answer

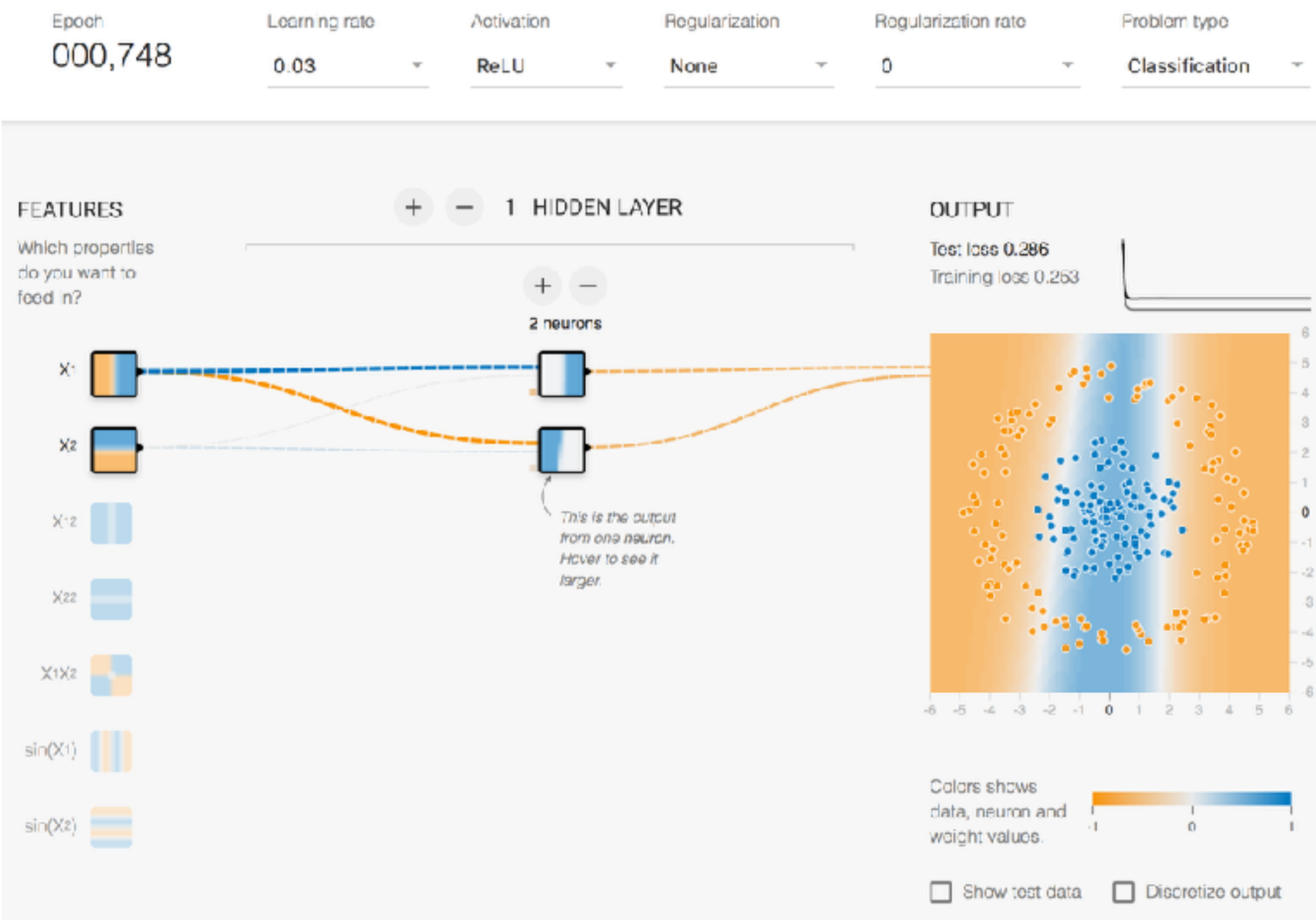

Epoch
000,542
Learning rate
0.3
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
X1
X2
sin(X1)
sin(X2)
0 HIDDEN LAYERS
OUTPUT
Training loss 0.553
Colors shows data, neuron and weight values.
Show test data
Discretize output

question

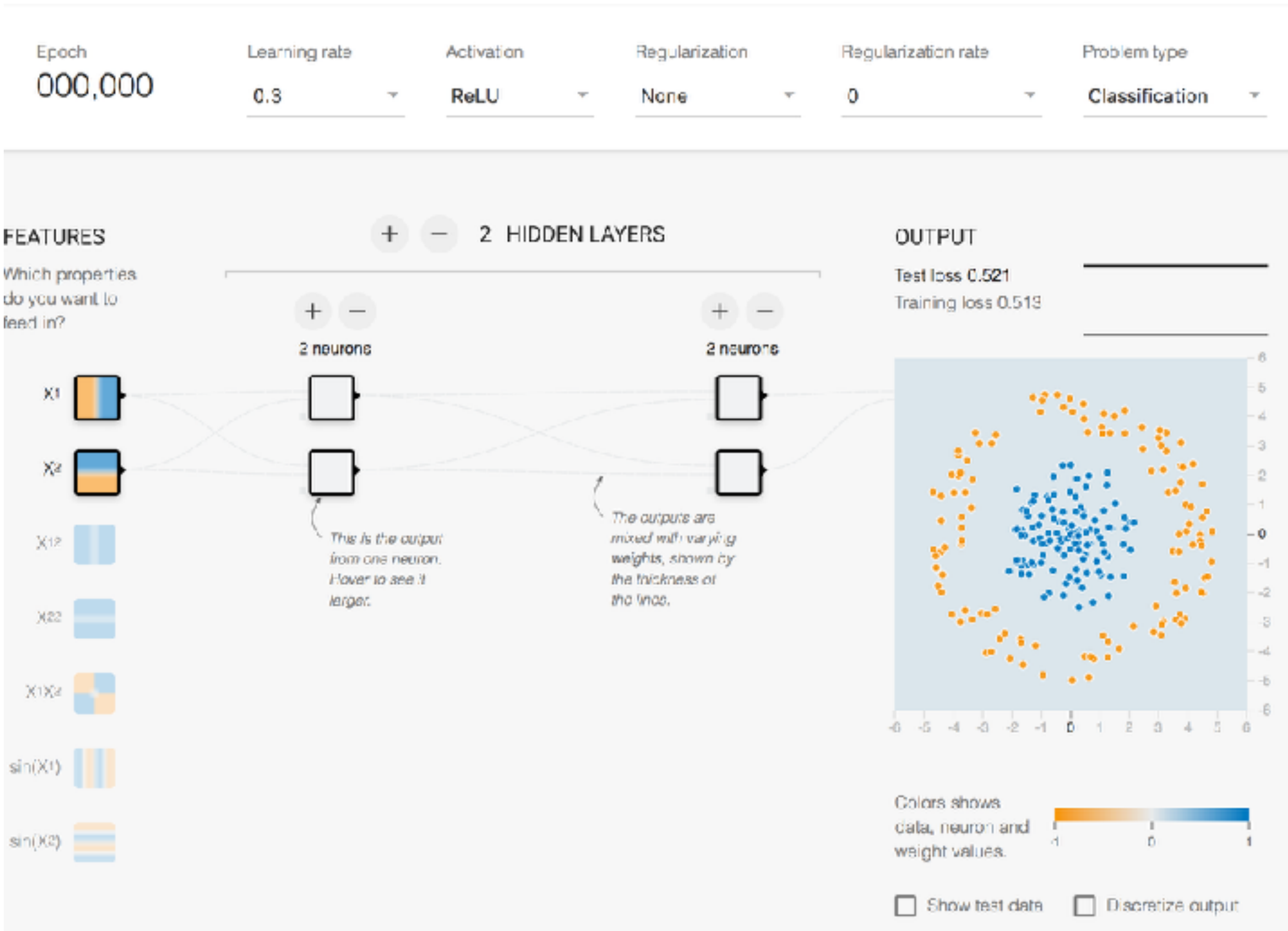


Epoch
000,000
Learning rate
0.3
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
2 neurons
This is the output from one neuron. Hover to see it larger.
OUTPUT
Test loss 0.469
Training loss 0.441
Colors shows data, neuron and weight values.
Show test data
Discretize output

answer

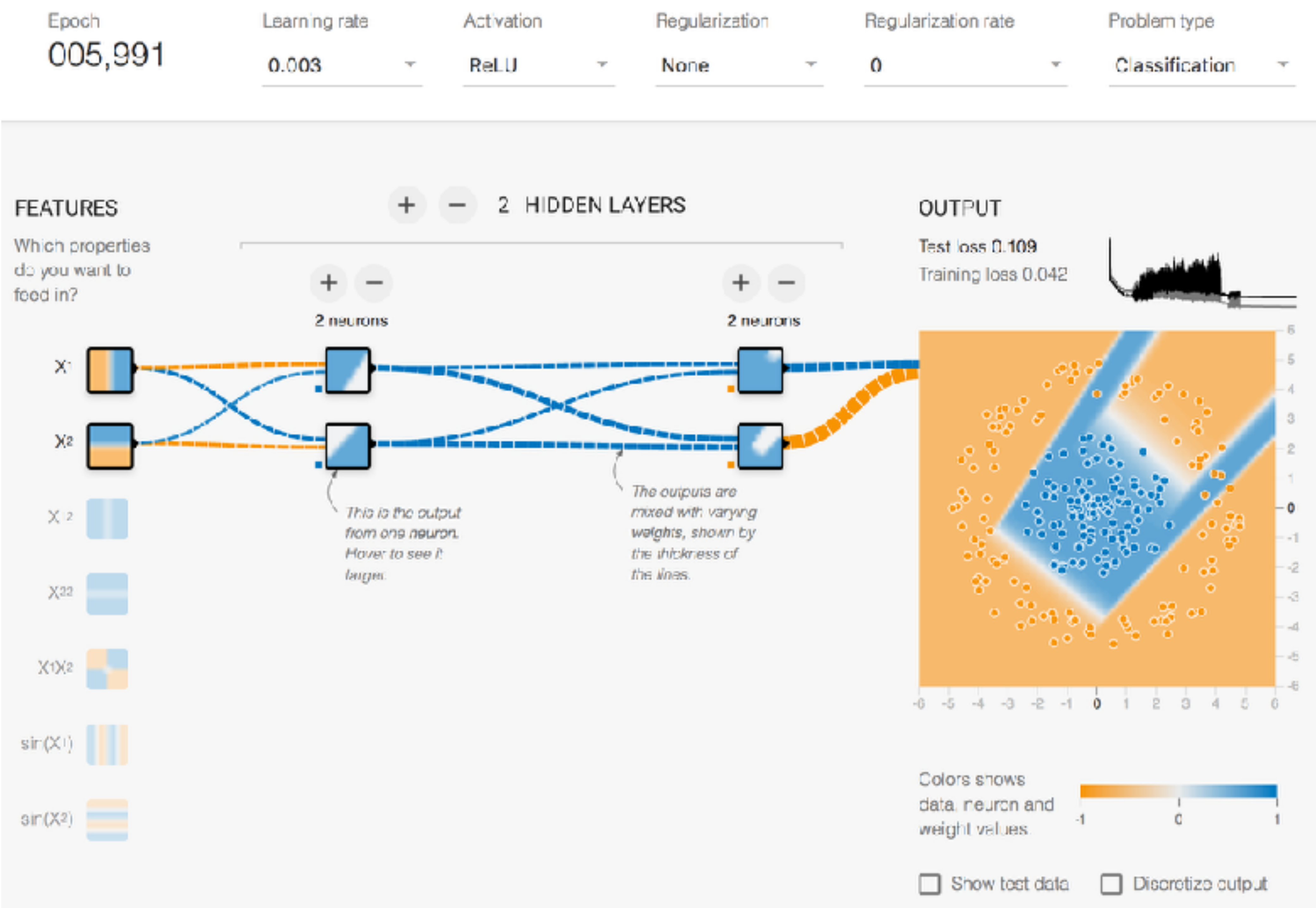

Epoch
000,748
Learning rate
0.03
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
2 neurons
OUTPUT
Test loss 0.286
Training loss 0.263
This is the output from one neuron. Hover to see it larger.
Colors shows data, neuron and weight values.
Show test data
Discretize output

question

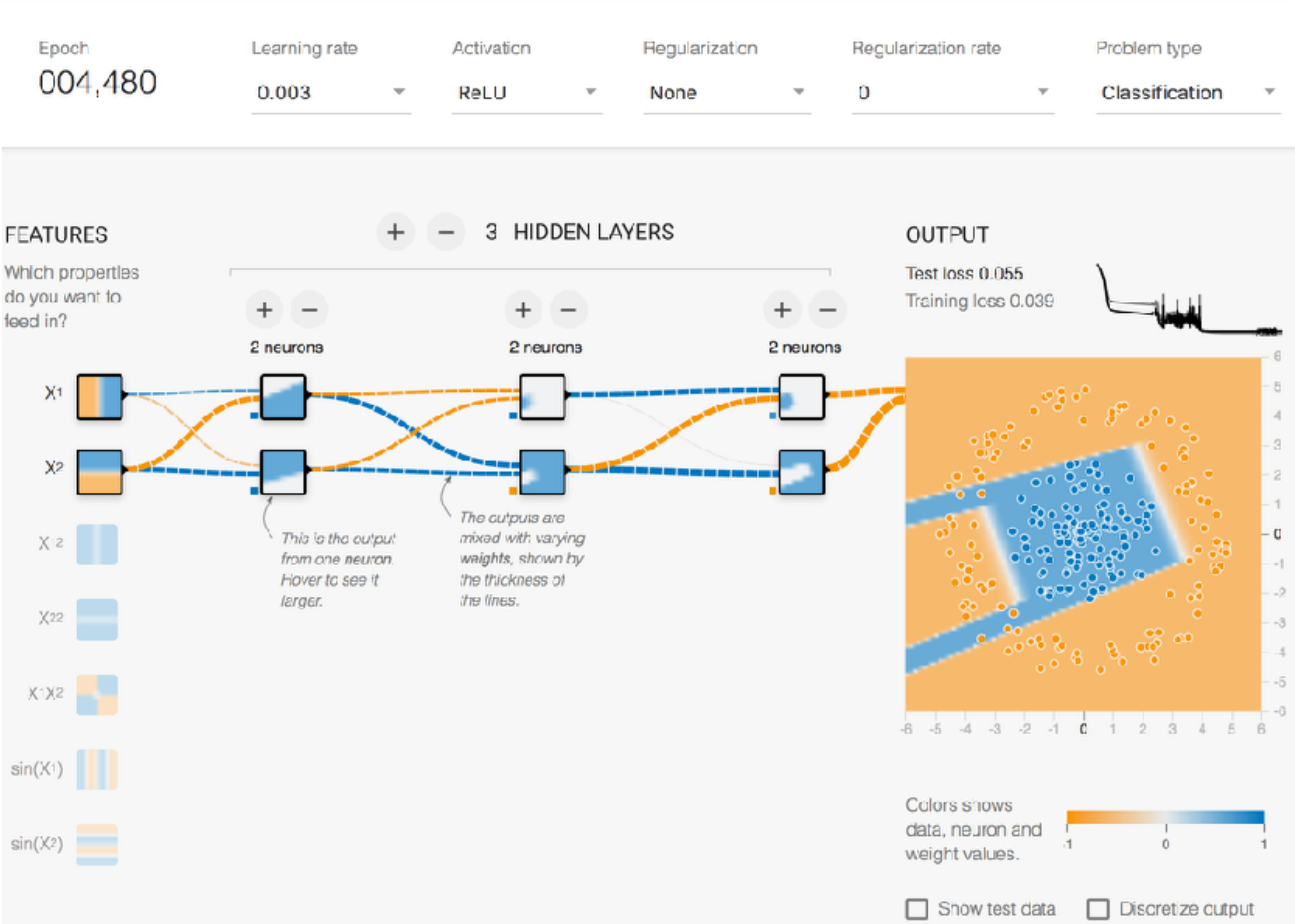


Epoch
000,000
Learning rate
0.3
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
2 HIDDEN LAYERS
2 neurons
2 neurons
This is the output from one neuron. Hover to see it larger.
The outputs are mixed with varying weights, shown by the thickness of the lines.
OUTPUT
Test loss 0.521
Training loss 0.513
Colors shows data, neuron and weight values.
Show test data
Discretize output

answer

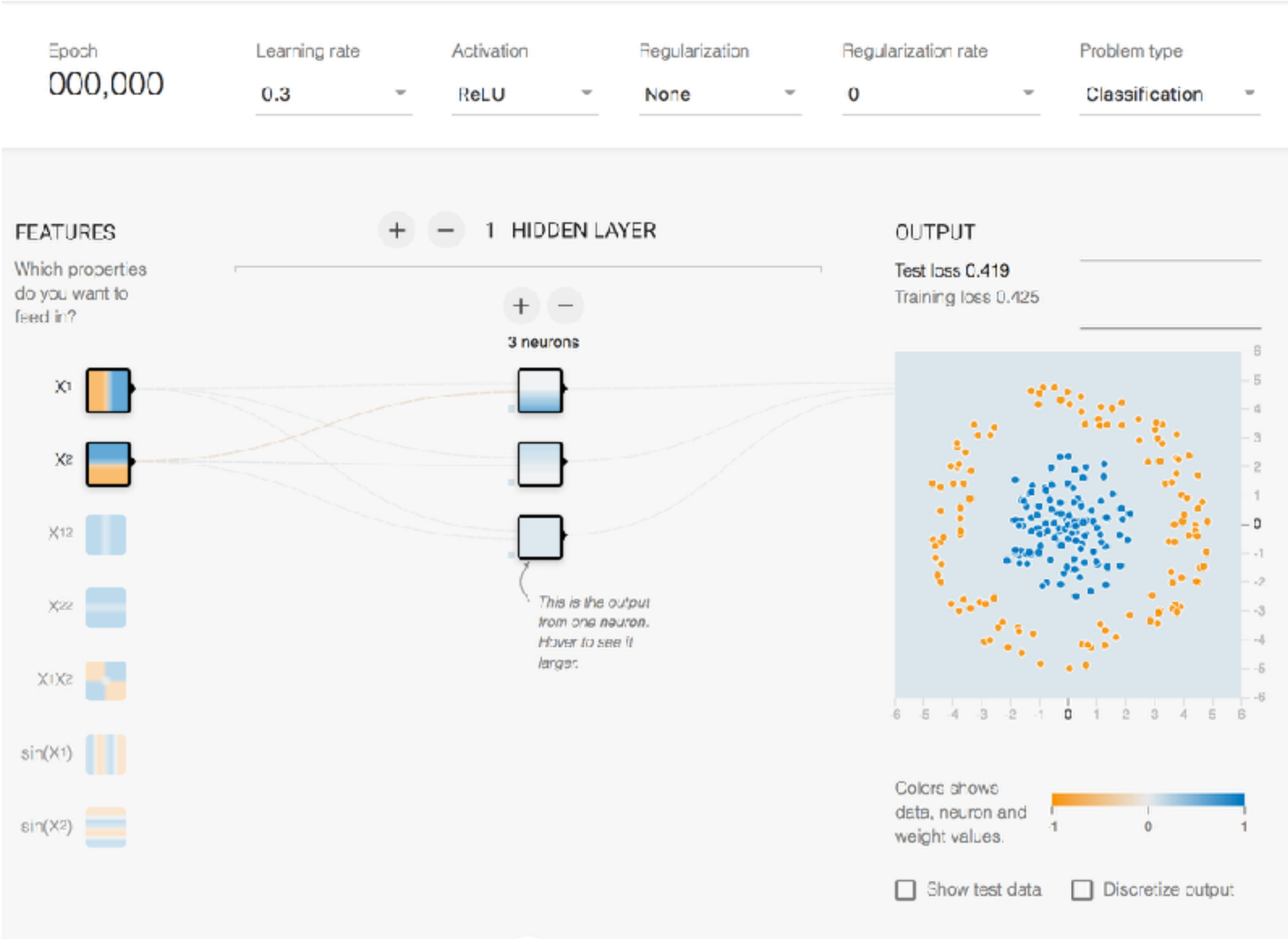

Epoch
005,991
Learning rate
0.003
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
2 HIDDEN LAYERS
2 neurons
2 neurons
This is the output from one neuron. Hover to see it larger.
The outputs are mixed with varying weights, shown by the thickness of the lines.
OUTPUT
Test loss 0.109
Training loss 0.042
Colors shows data, neuron and weight values.
Show test data
Discretize output

answer



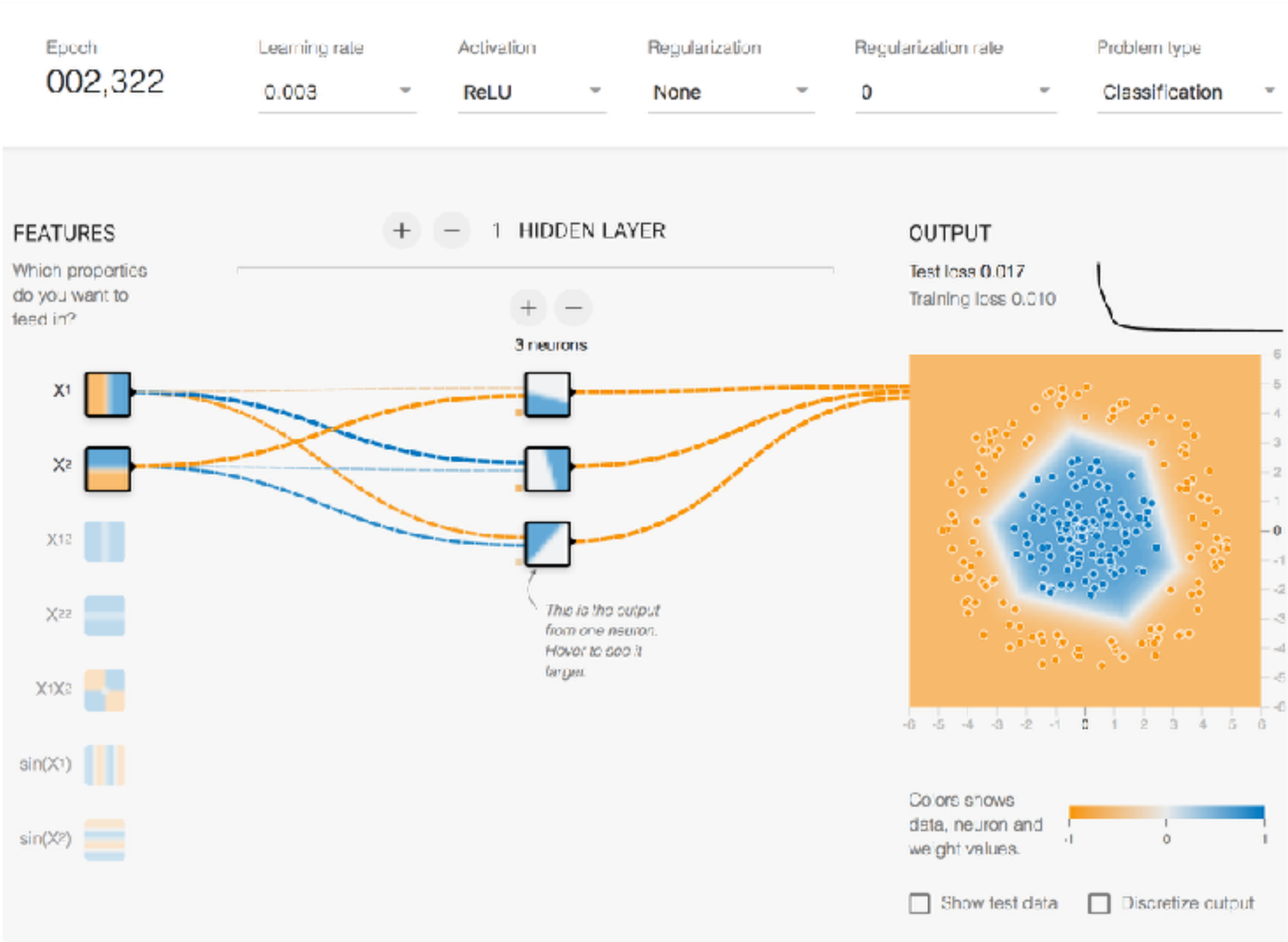
Epoch
004,480
Learning rate
0.003
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
3 HIDDEN LAYERS
2 neurons
2 neurons
2 neurons
This is the output from one neuron. Hover to see it larger.
The outputs are mixed with varying weights, shown by the thickness of the lines.
OUTPUT
Test loss 0.055
Training loss 0.039
Colors shows data, neuron and weight values.
Show test data
Discretize output

question

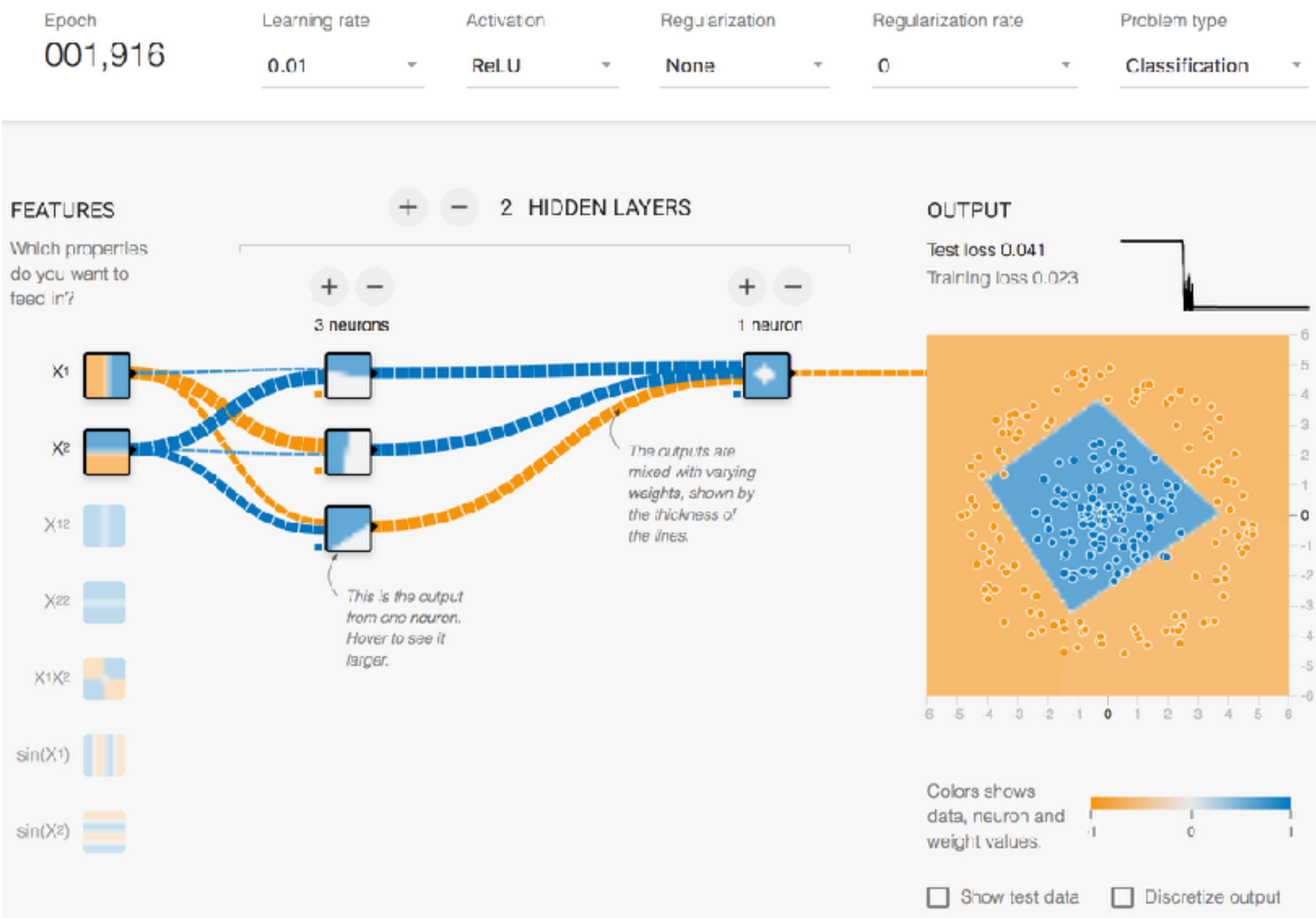


Epoch
000,000
Learning rate
0.3
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
3 neurons
This is the output from one neuron. Hover to see it larger.
OUTPUT
Test loss 0.419
Training loss 0.425
Colors shows data, neuron and weight values.
Show test data
Discretize output

answer

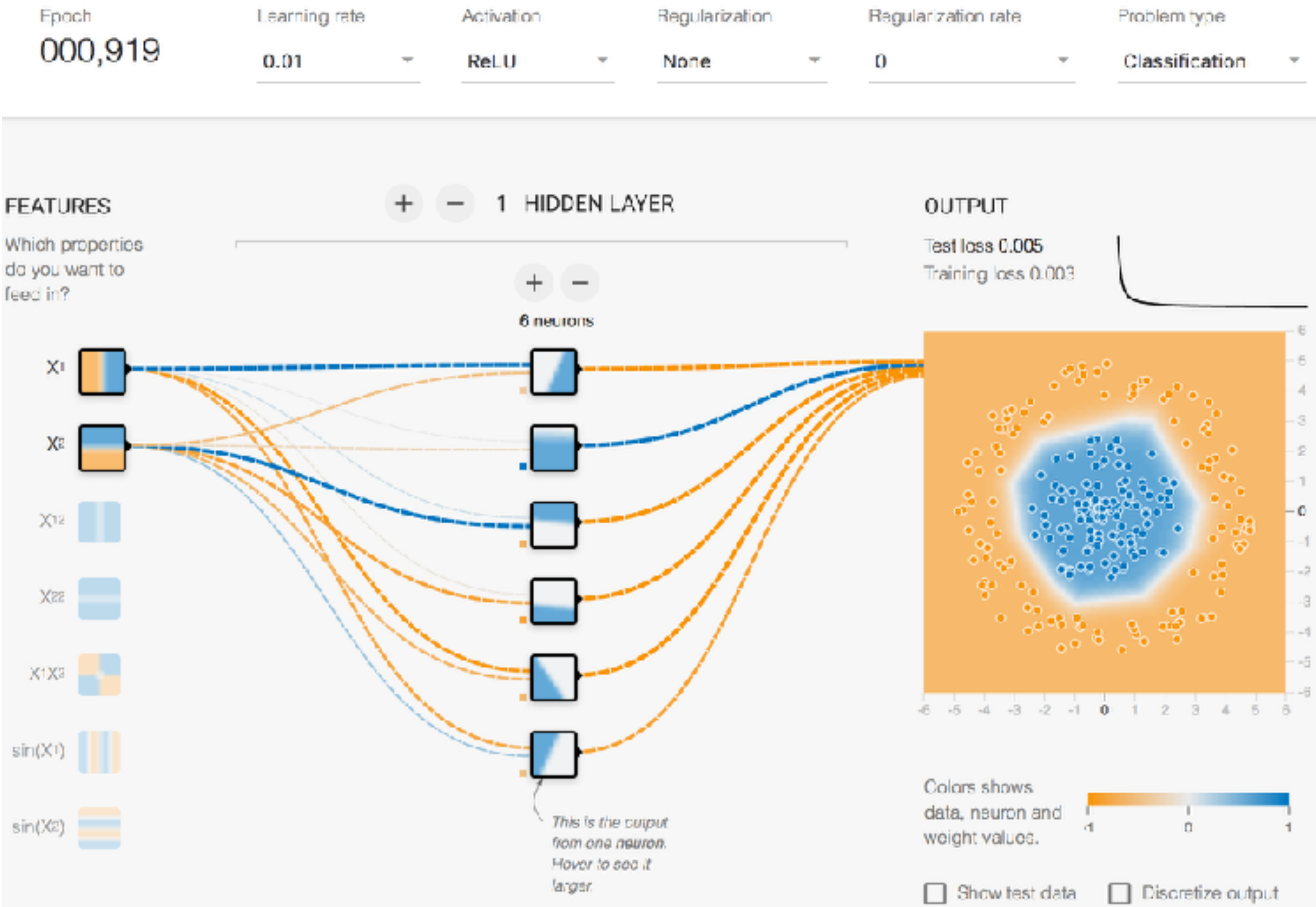

Epoch
002,322
Learning rate
0.003
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
3 neurons
OUTPUT
Test loss 0.017
Training loss 0.010
This is the output from one neuron. Hover to see it larger.
Colors shows data, neuron and weight values.
Show test data
Discretize output

answer

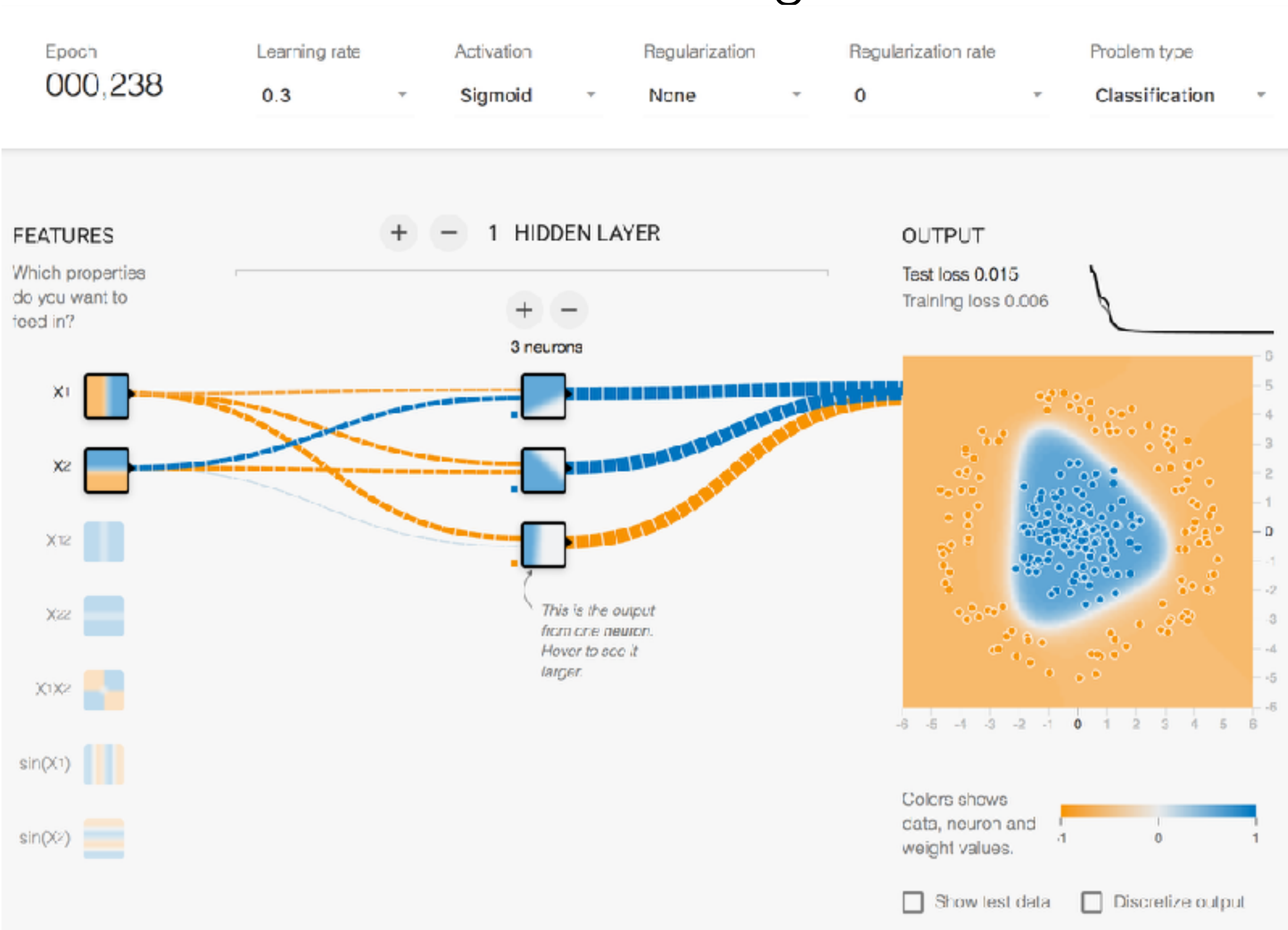


Epoch
001,916
Learning rate
0.01
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
2 HIDDEN LAYERS
3 neurons
1 neuron
OUTPUT
Test loss 0.041
Training loss 0.023
X1
X2
sin(X1)
sin(X2)
This is the output from one neuron. Hover to see it larger.
The outputs are mixed with varying weights, shown by the thickness of the lines.
Colors shows data, neuron and weight values.
Show test data
Discretize output

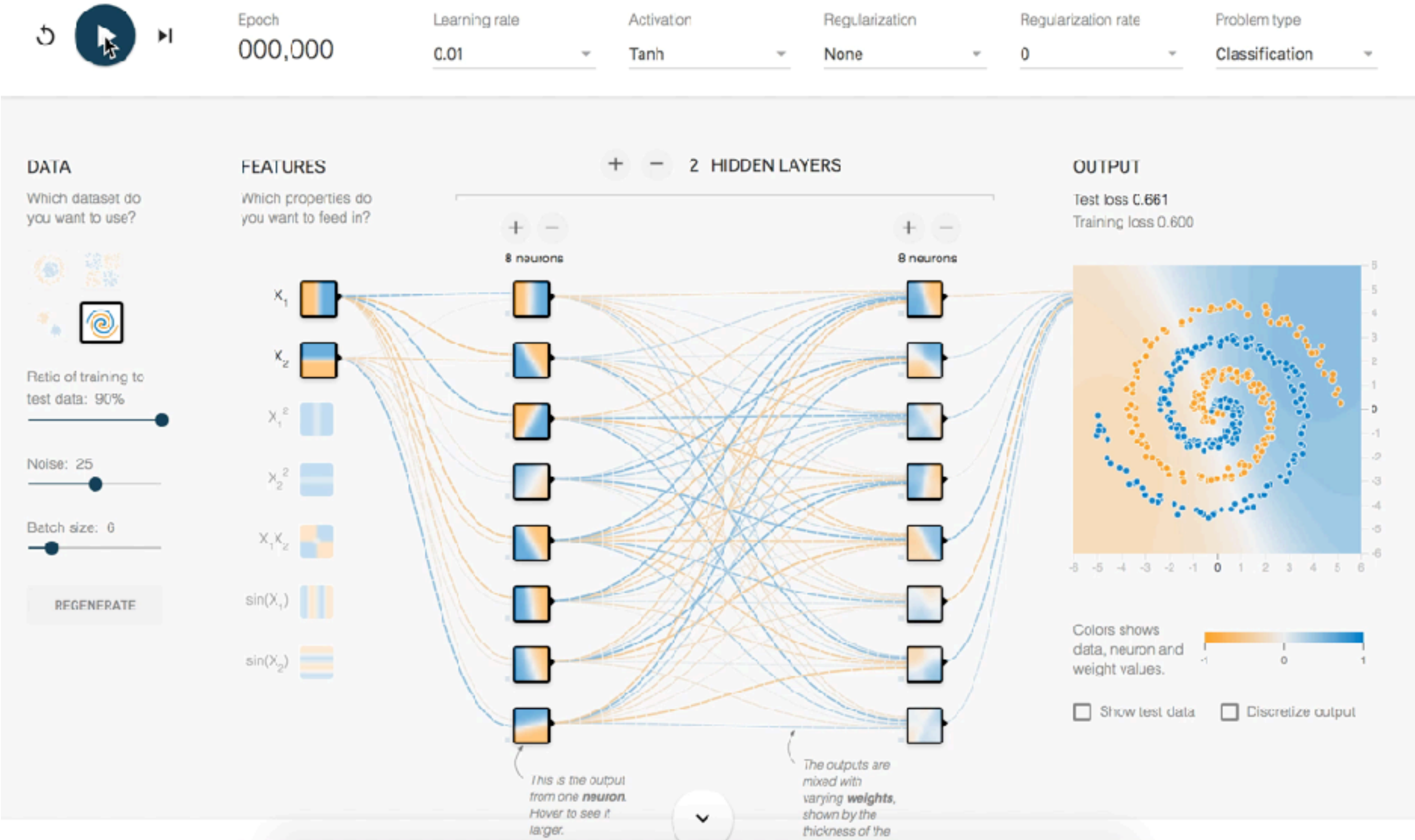
answer


Epoch
000,919
Learning rate
0.01
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
6 neurons
OUTPUT
Test loss 0.005
Training loss 0.003
This is the output from one neuron. Hover to see it larger.
Colors shows data, neuron and weight values.
Show test data
Discretize output

answer - with sigmoid



question



Epoch
000,000
Learning rate
0.01
Activation
Tanh
Regularization
None
Regularization rate
0
Problem type
Classification
DATA
Which dataset do you want to use?
Ratio of training to test data: 90%
Noise: 25
Batch size: 6
REGENERATE
FEATURES
Which properties do you want to feed in?
2 HIDDEN LAYERS
8 neurons
8 neurons
OUTPUT
Test loss 0.661
Training loss 0.600
Colors shows data, neuron and weight values.
Show test data
Discretize output
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