
冨嶽三十六景
神奈川沖
浪裏

Welcome to

CS5242

Neural Networks and Deep Learning

Lee Hwee Kuan & Wang Wei

Teaching assistant

Connie Kou Khor Li, Ji Xin, Ouyang Kun

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

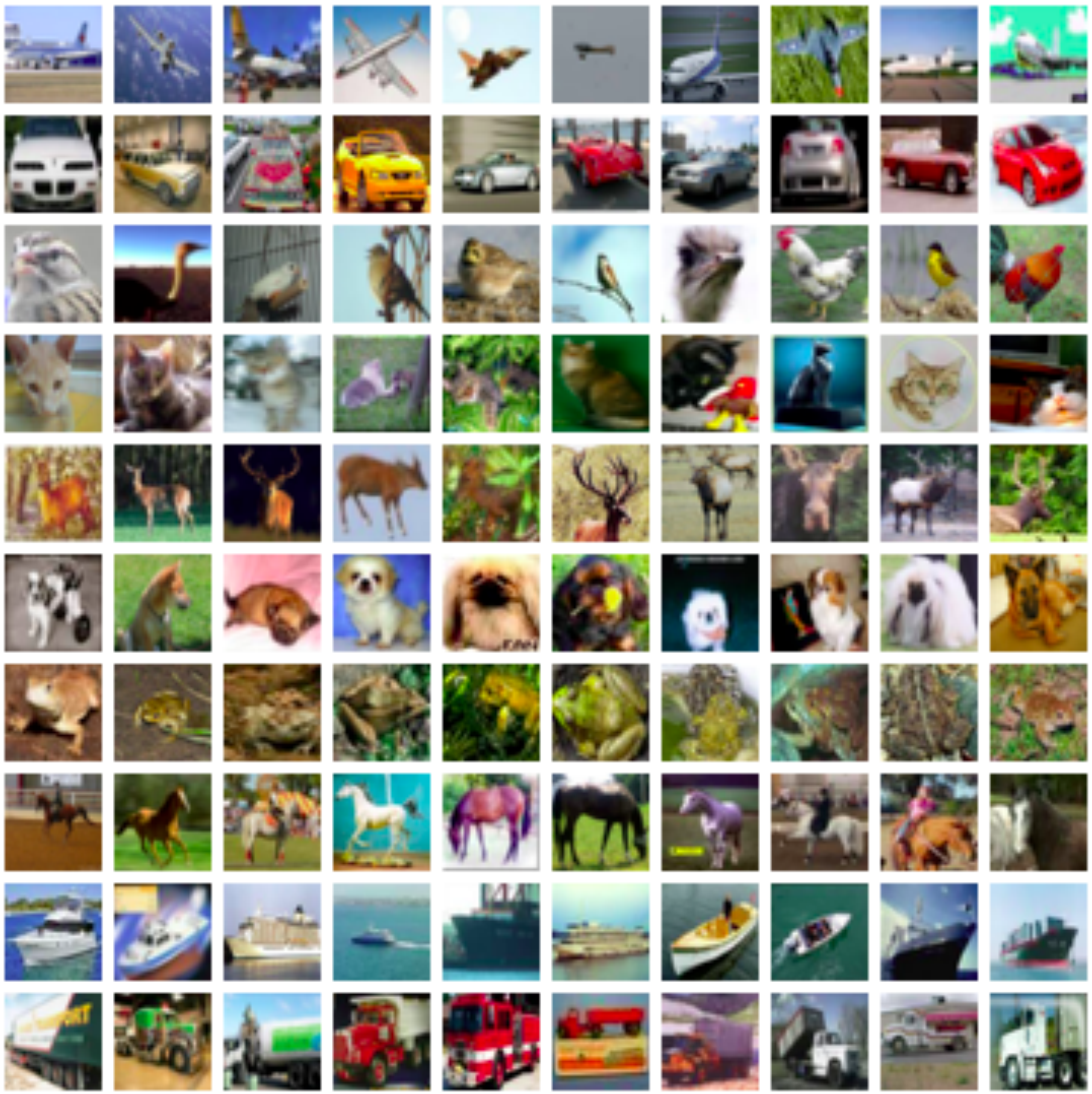
‘Contemporary’ history of neural nets

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

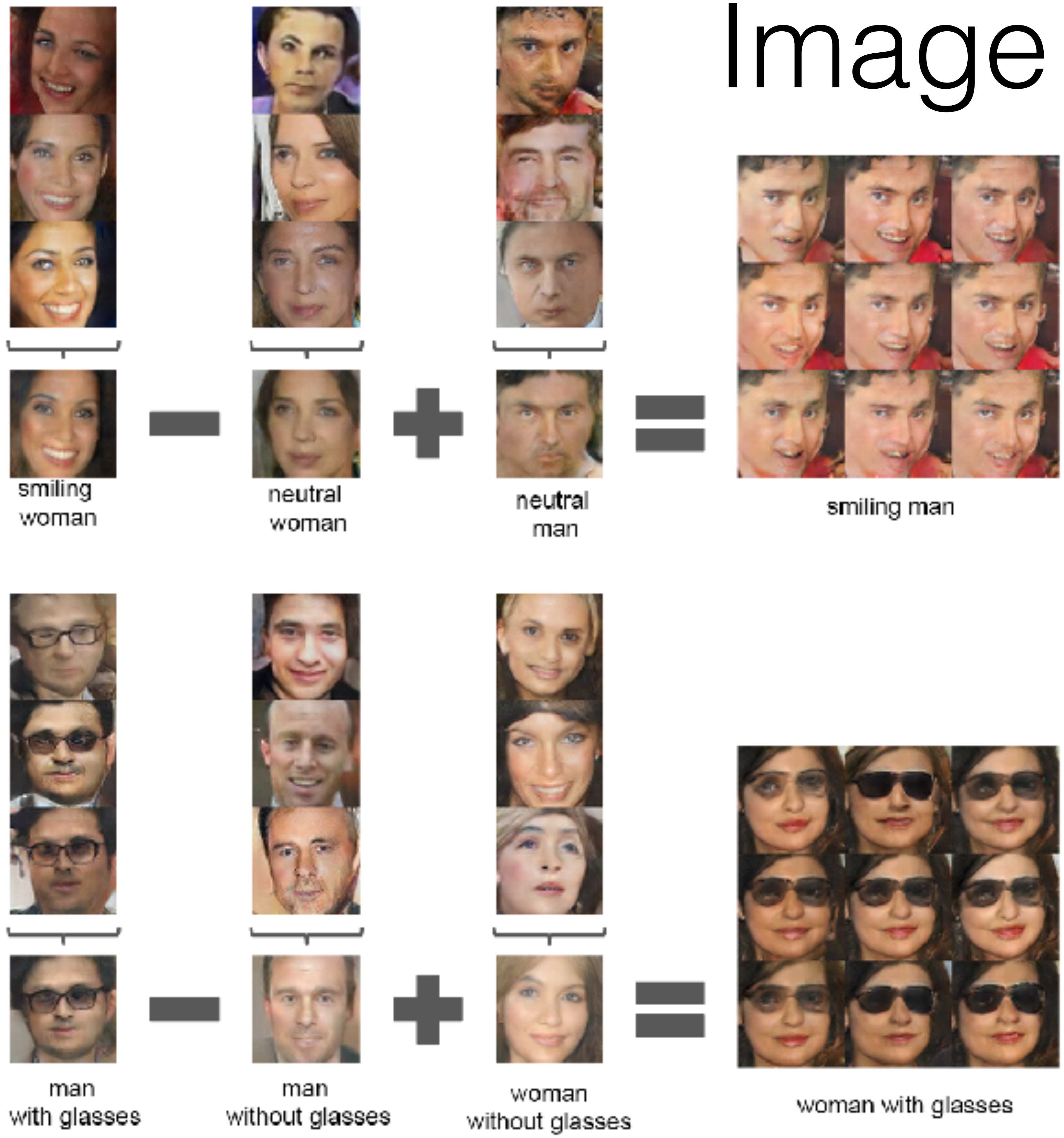


60,000 images, 50K train, 10K test, 10 classes

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

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

Image Generation

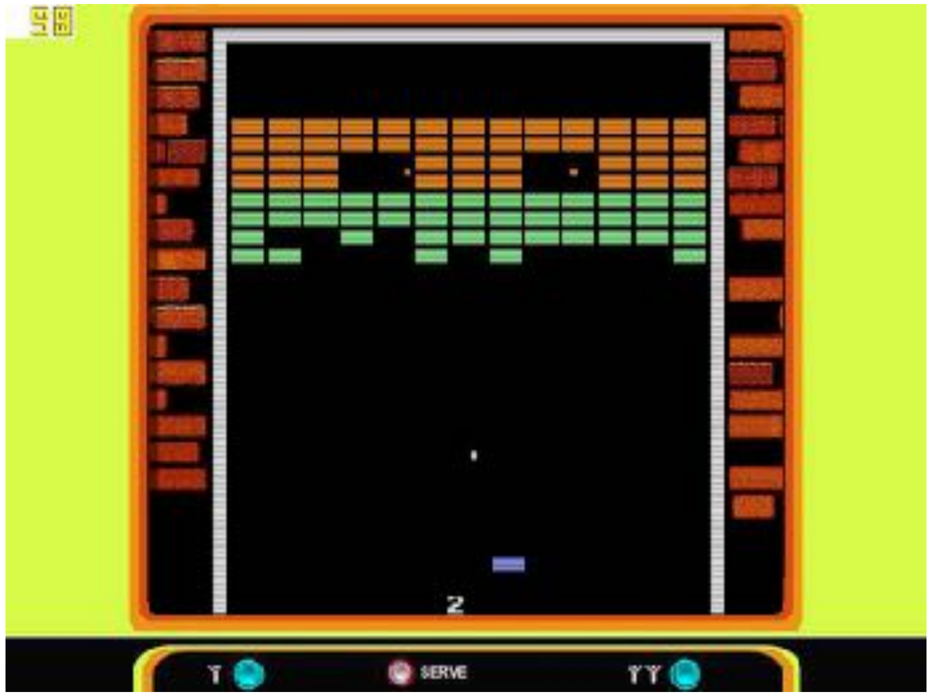


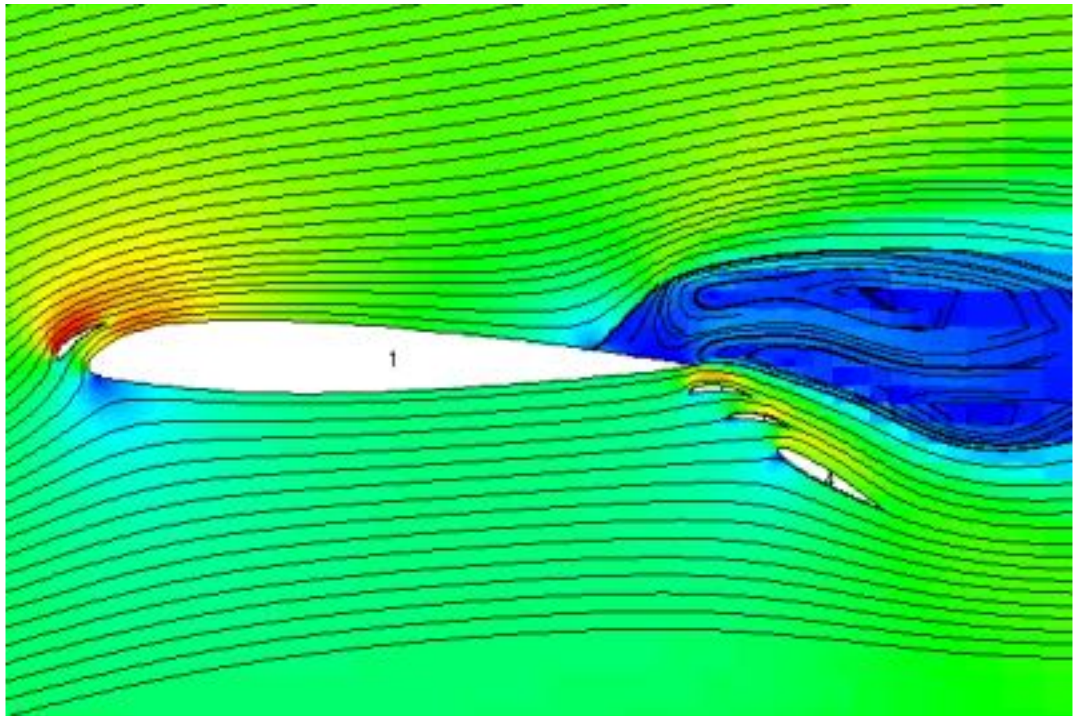
https://www.pinterest.com/pin/205124958008048718/

Alpha-Go

Our world versus computer world

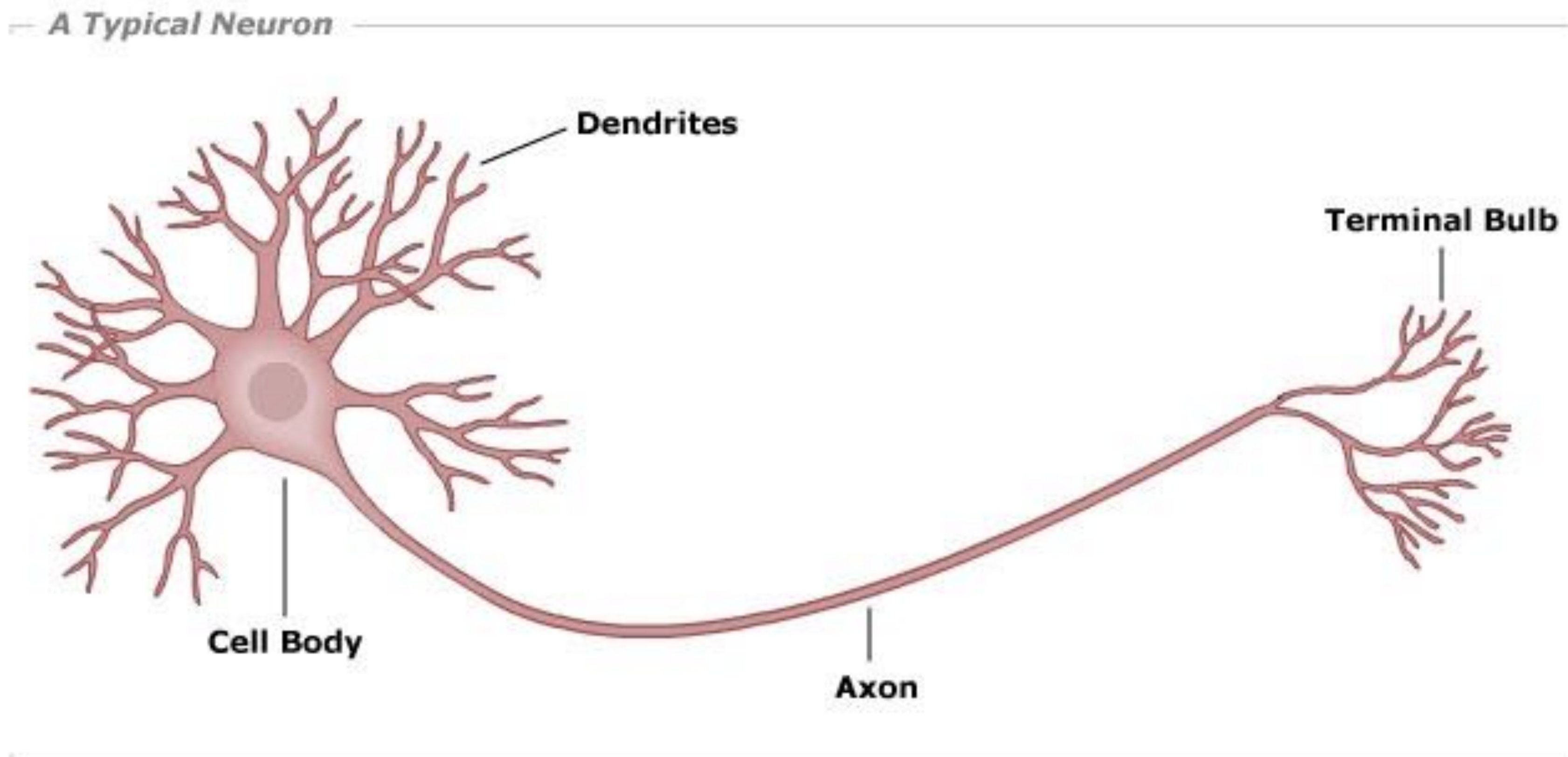
23684184 x 4729472 = 112013685070848



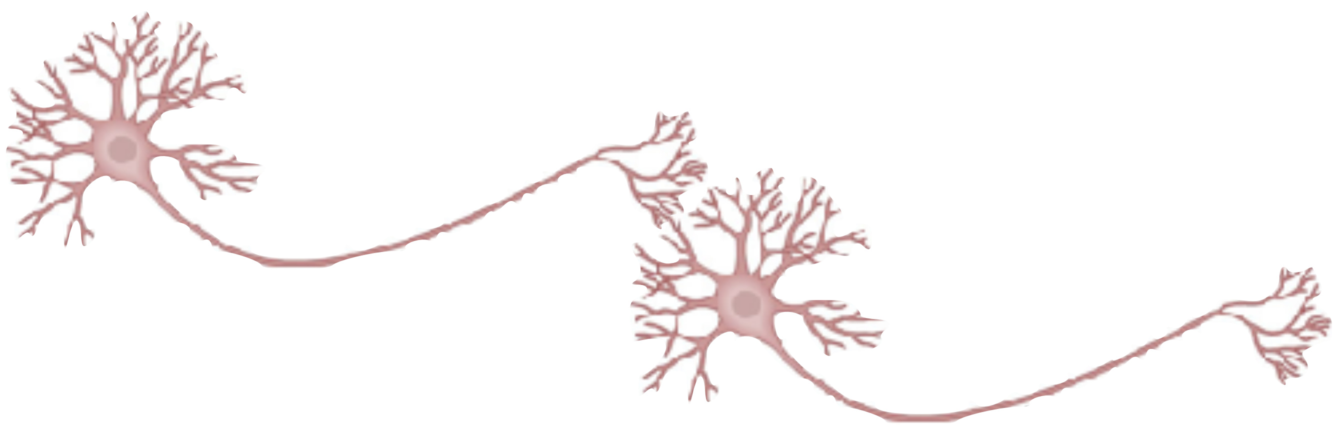


Our world versus computer world

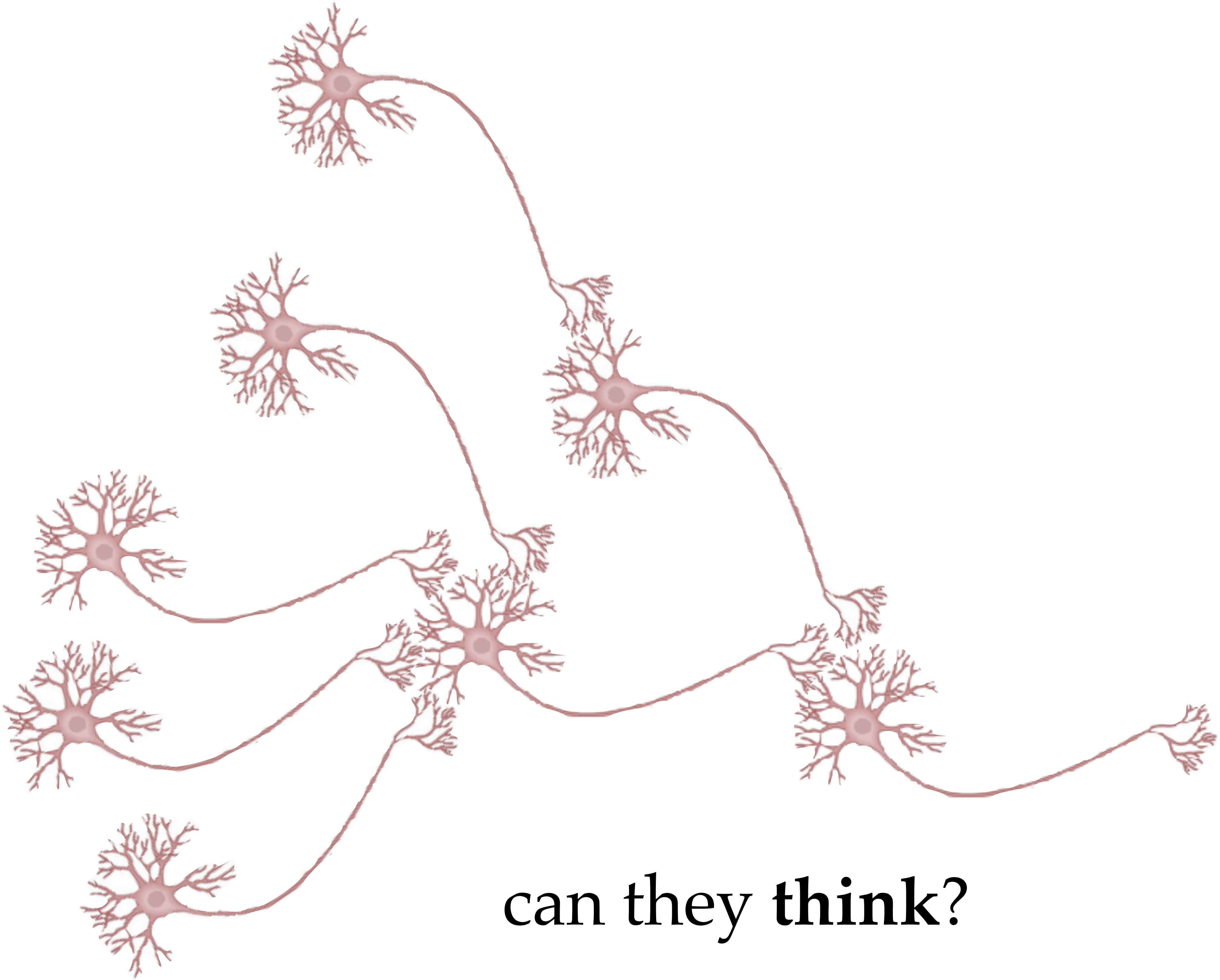
What makes Deep Leaning so good?



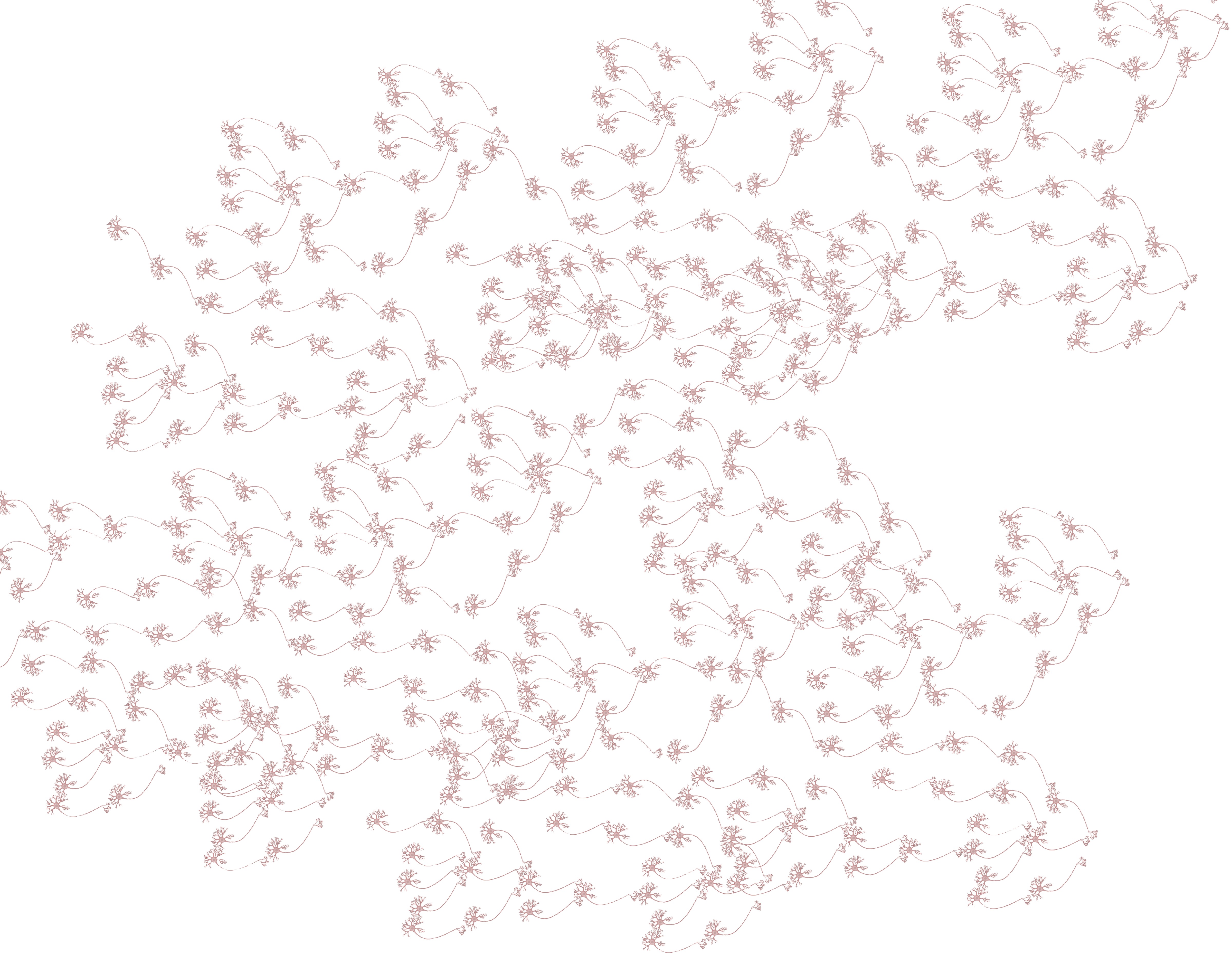
can this thing **think**?

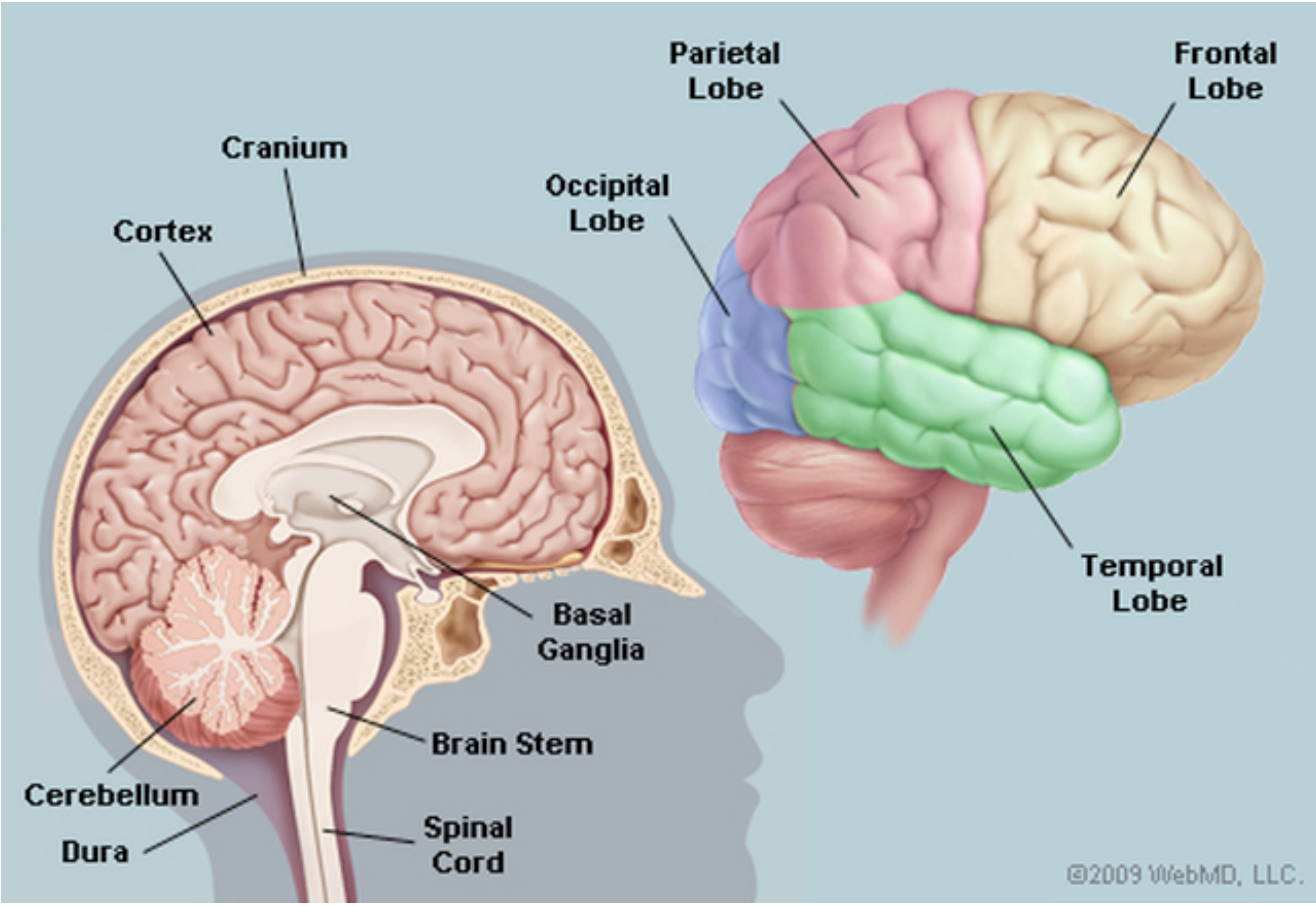


can they **think**?



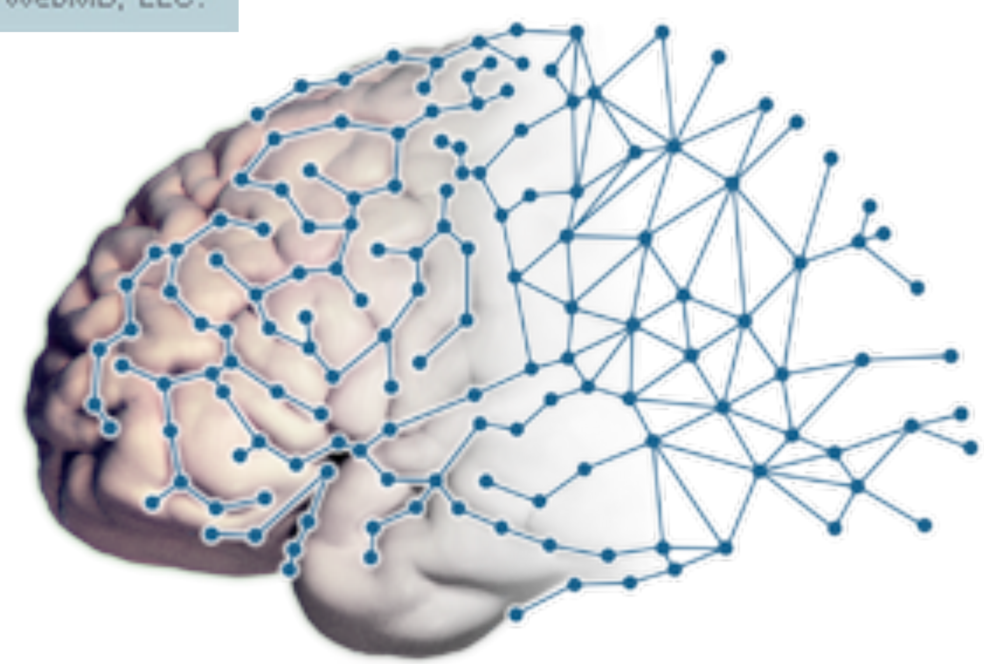
can they **think**?

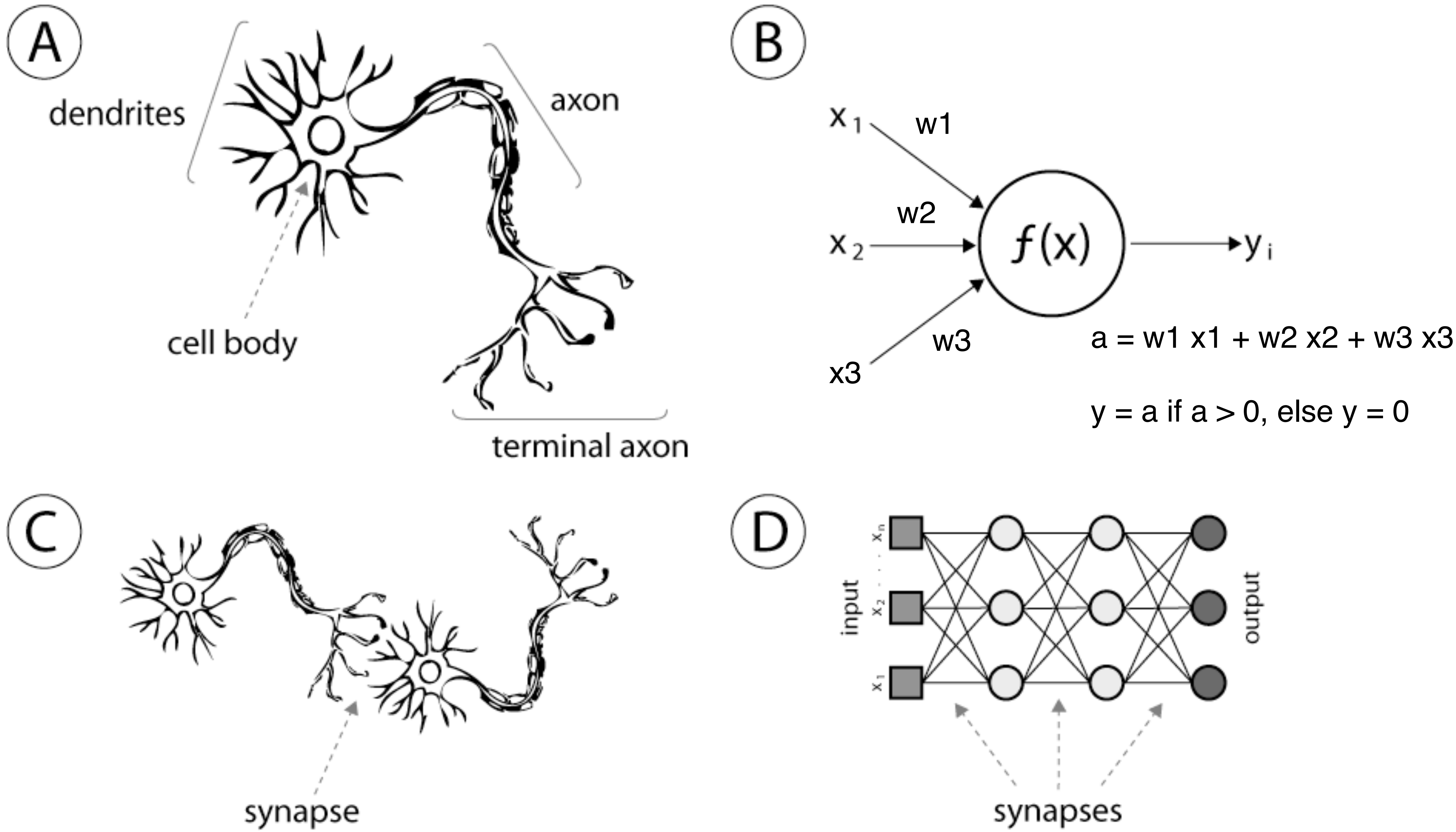




http://img.webmd.com/dtmcms/live/webmd/consumer_assets/site_images/articles/image_article_collections/anatomy_pages/brain2.jpg?resize=646px:*&output-quality=100

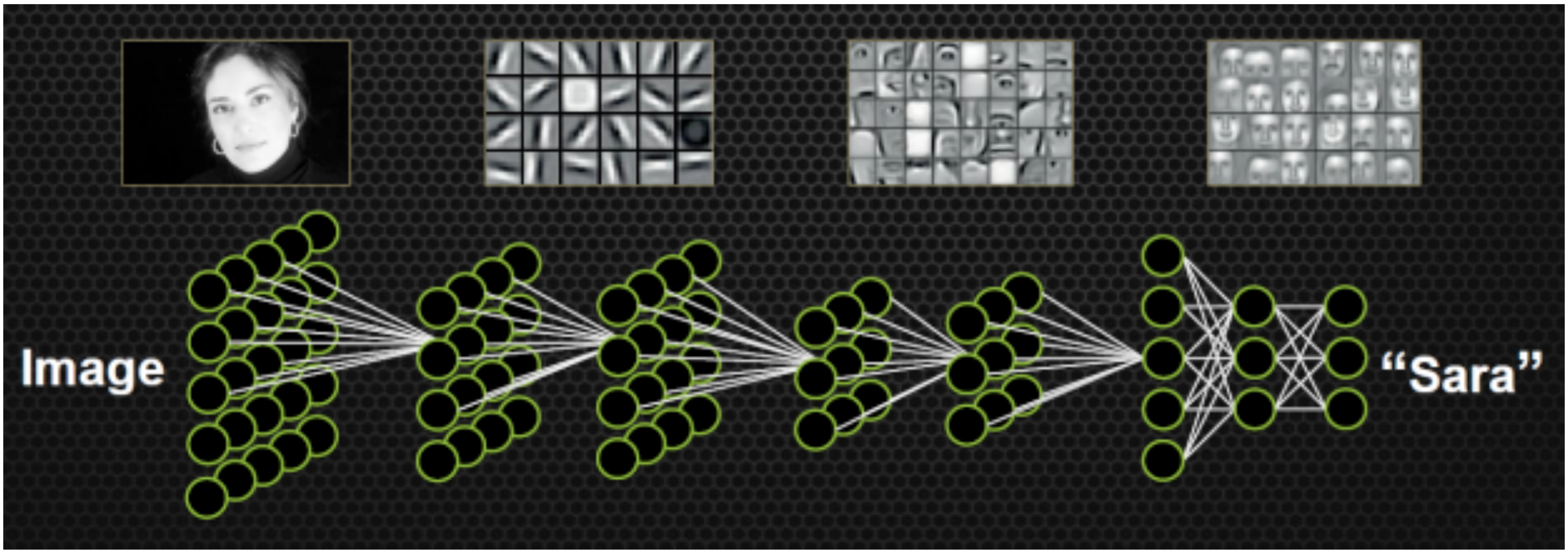
can it **think**?





can this thing **think**?

With millions of connections, it started to “think”



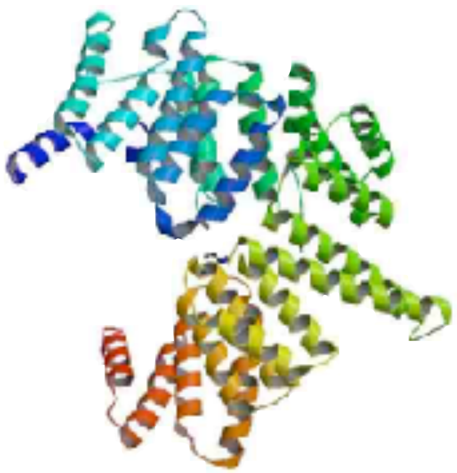
https://gogameguru.com/i/2016/03/AlphaGo-Lee-Sedol-game-2-Aja-Huang-Lee-Sedol.jpg

Simple emergent behaviour

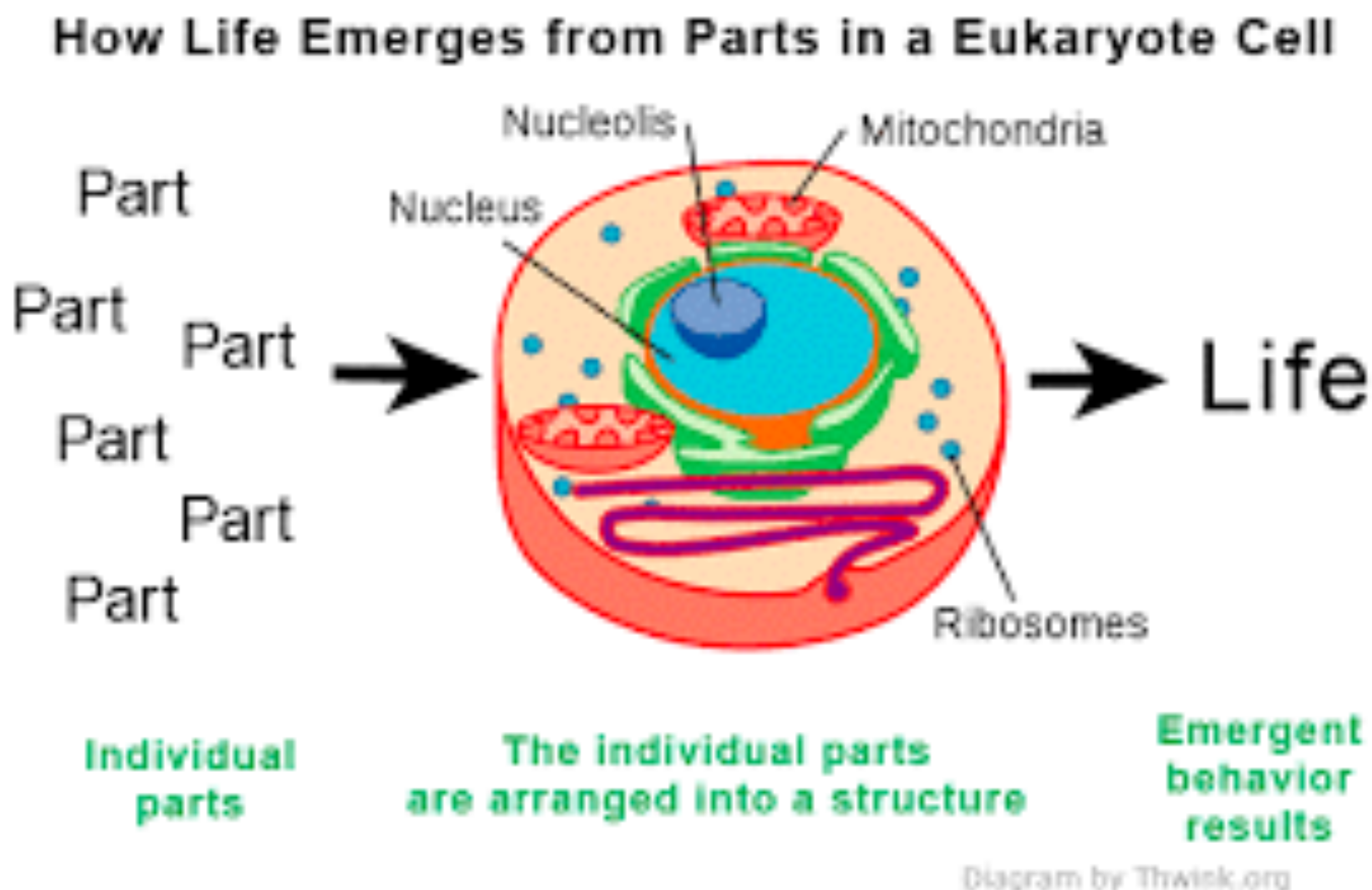
phase transitions - water, magnetism

show demo

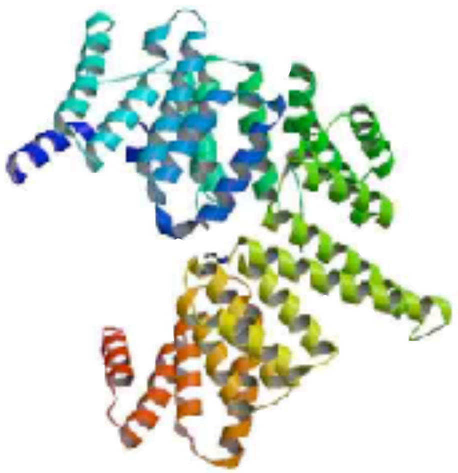
How do we get from molecules (proteins) to cell and then to life?



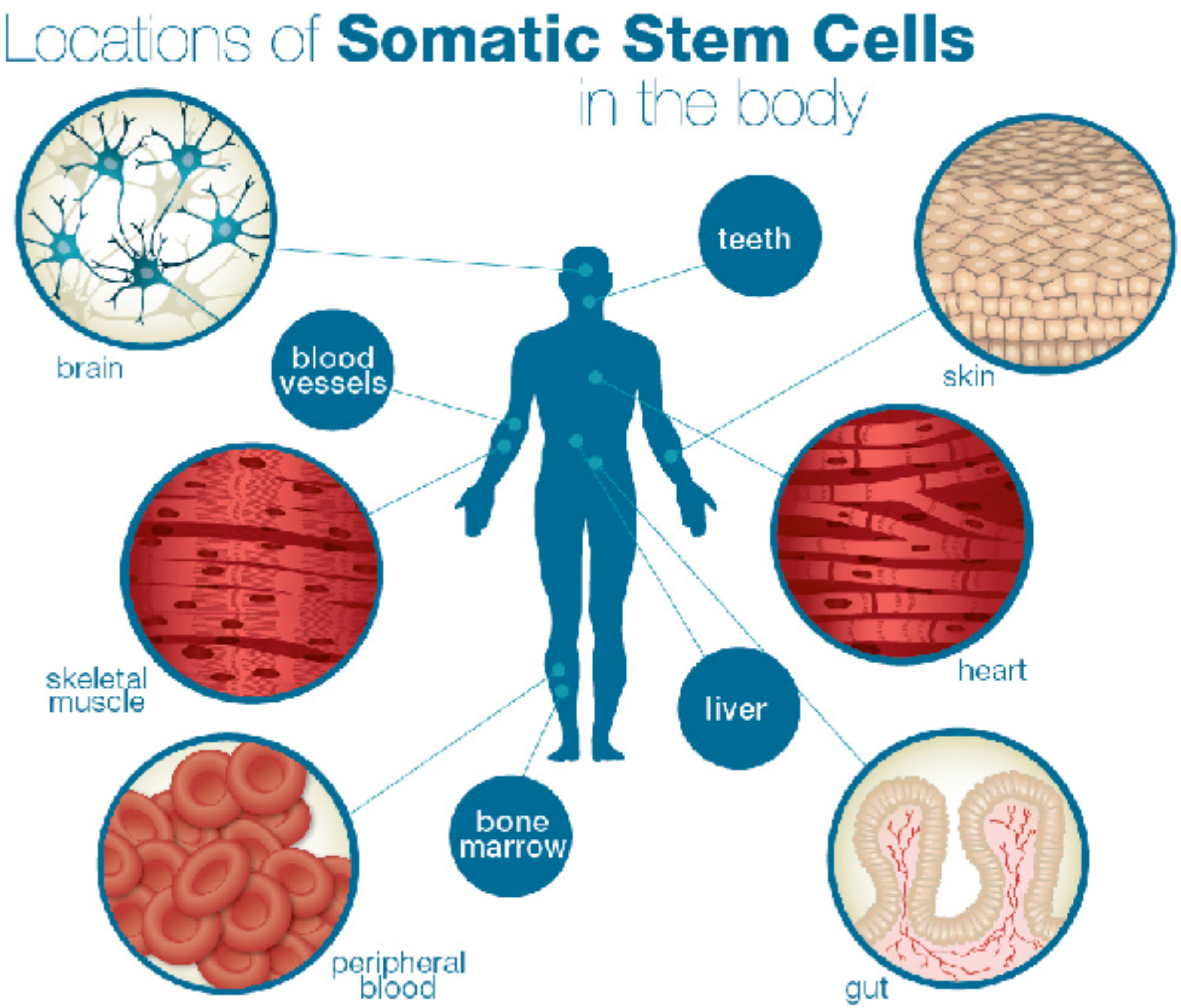
proteins



How do we get from molecules (proteins) to cell and then to life?

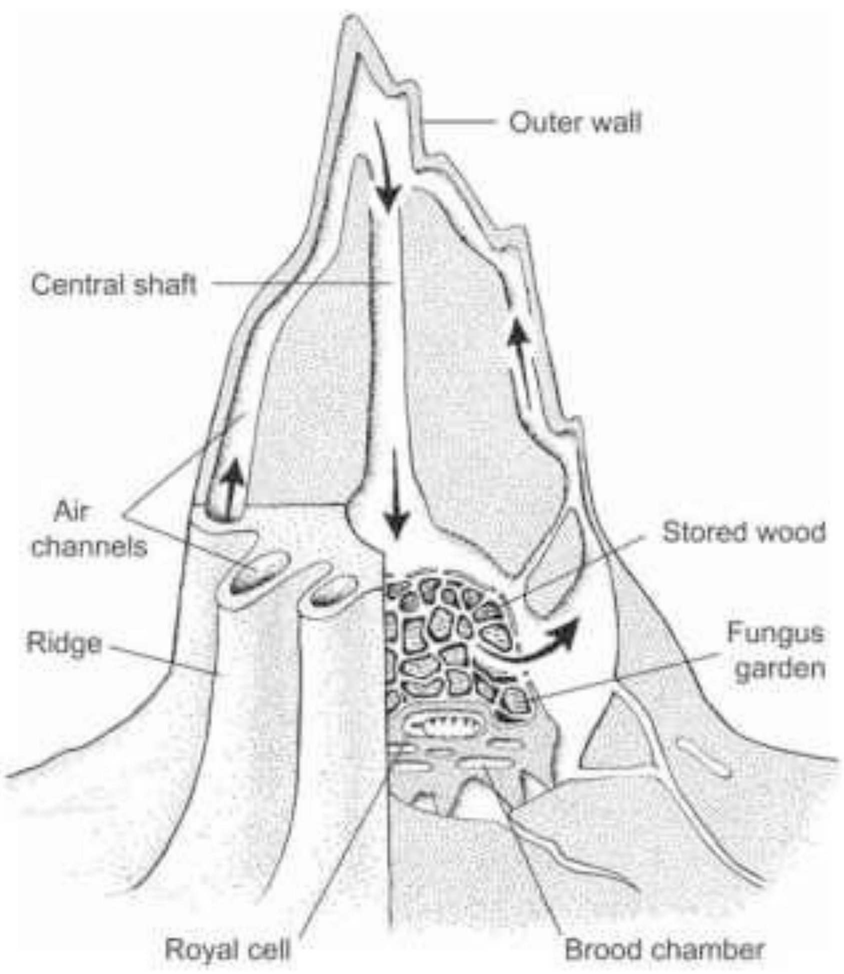


proteins



http://learn.genetics.utah.edu/content/stemcells/quickref/somaticstemcells.jpg

Life exhibits complexity



https://www.pinterest.com/sattele/termite-mound/

http://img.ev.mu/images/reportages/188/520x342/09.jpg

How extreme can life become?

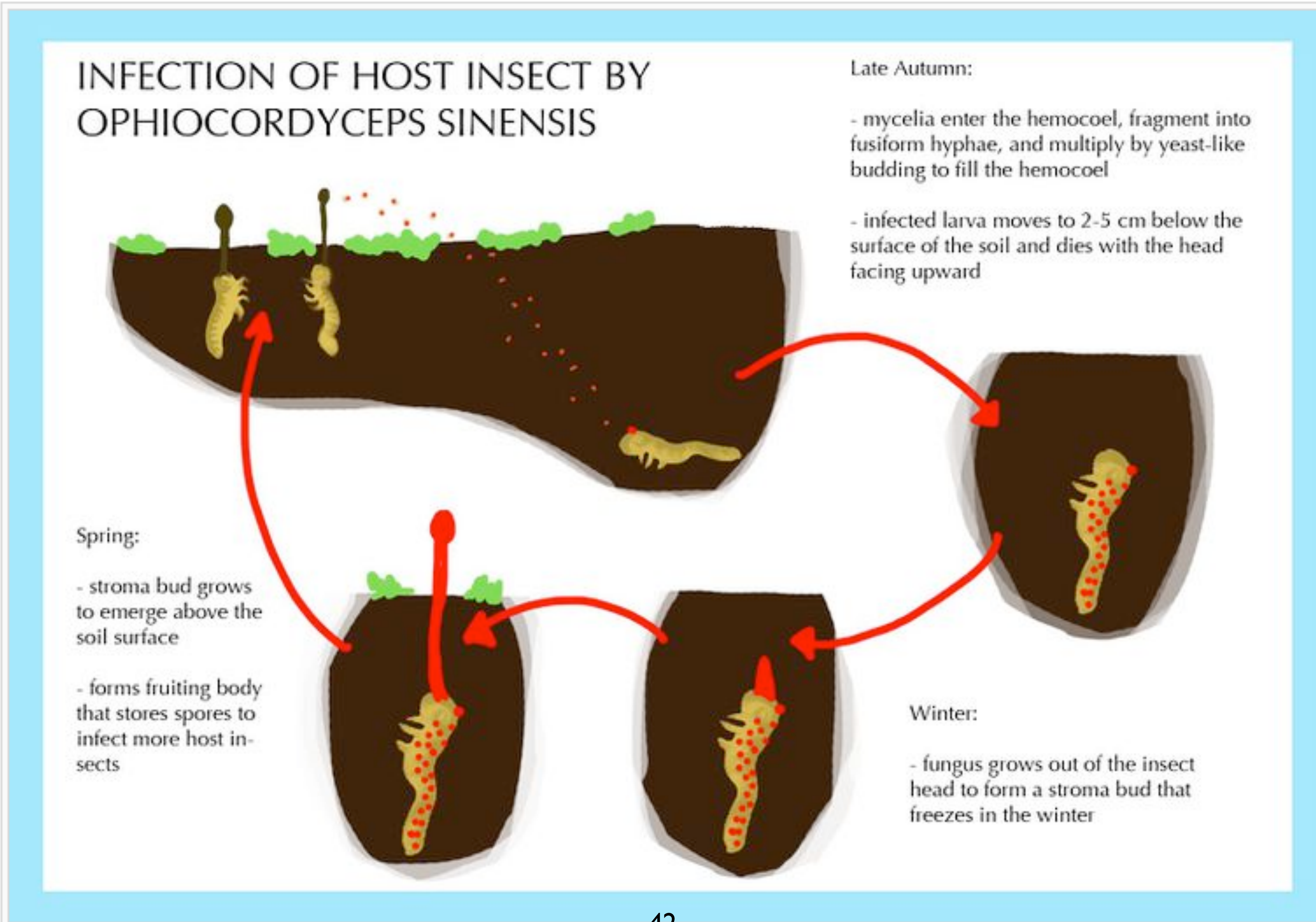
How extreme can emergent behaviour become?

very extreme indeed

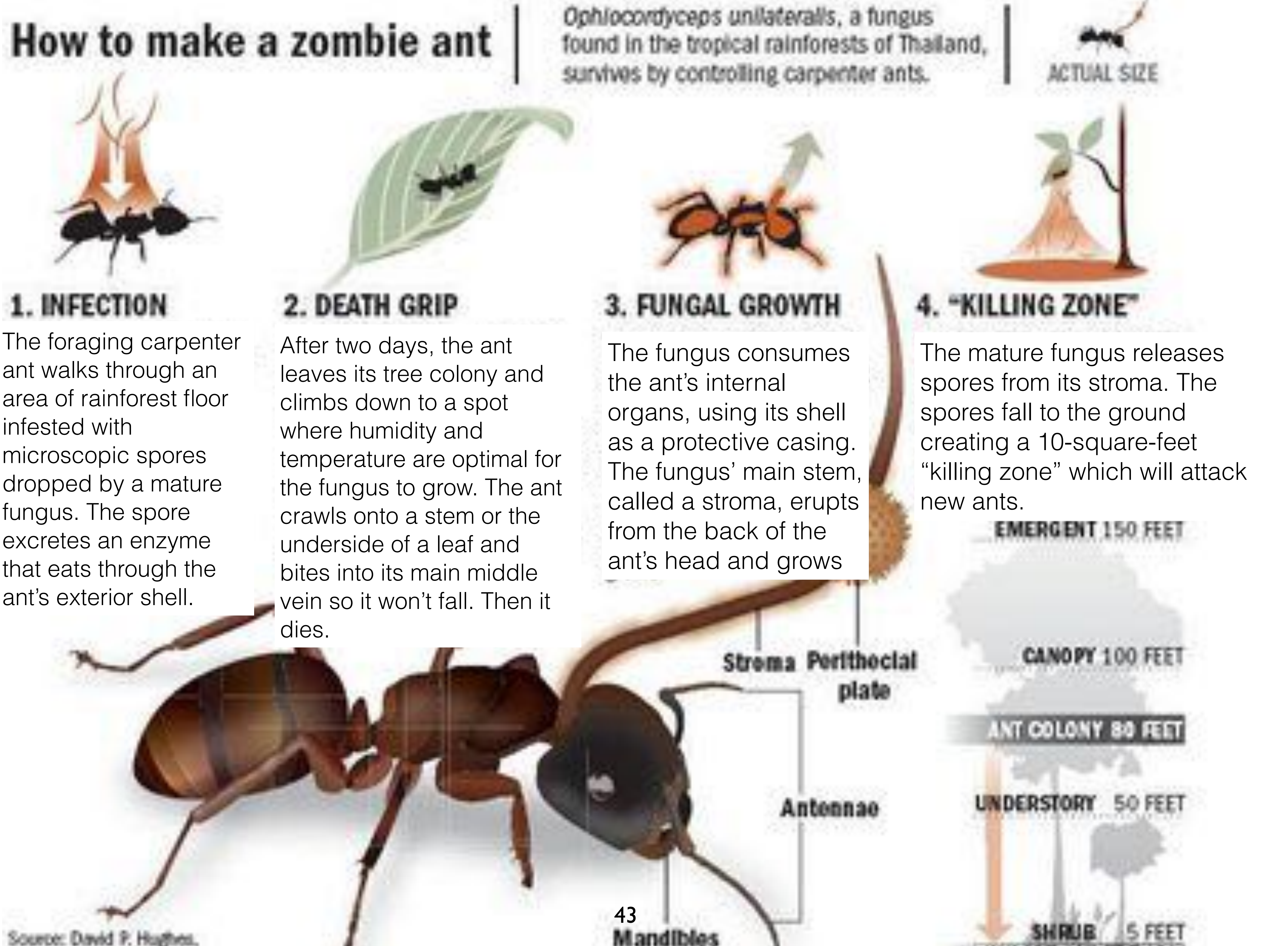
some life forms on earth evolved the ability to control the behaviours of another life form!

Ophiocordyceps sinensis
冬虫草

http://microbewiki.kenyon.edu/index.php/Ophiocordyceps_sinensis



https://microbewiki.kenyon.edu/index.php/Ophiocordyceps_unilateralis



1. INFECTION

The foraging carpenter ant walks through an area of rainforest floor infested with microscopic spores dropped by a mature fungus. The spore excretes an enzyme that eats through the ant's exterior shell.

2. DEATH GRIP

After two days, the ant leaves its tree colony and climbs down to a spot where humidity and temperature are optimal for the fungus to grow. The ant crawls onto a stem or the underside of a leaf and bites into its main middle vein so it won't fall. Then it dies.

3. FUNGAL GROWTH

The fungus consumes the ant's internal organs, using its shell as a protective casing. The fungus' main stem, called a stroma, erupts from the back of the ant's head and grows

4. "KILLING ZONE"

The mature fungus releases spores from its stroma. The spores fall to the ground creating a 10-square-feet "killing zone" which will attack new ants.

How extreme can machines become?

are we able to make a machine that is as complex as the most primitive life known to us?

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

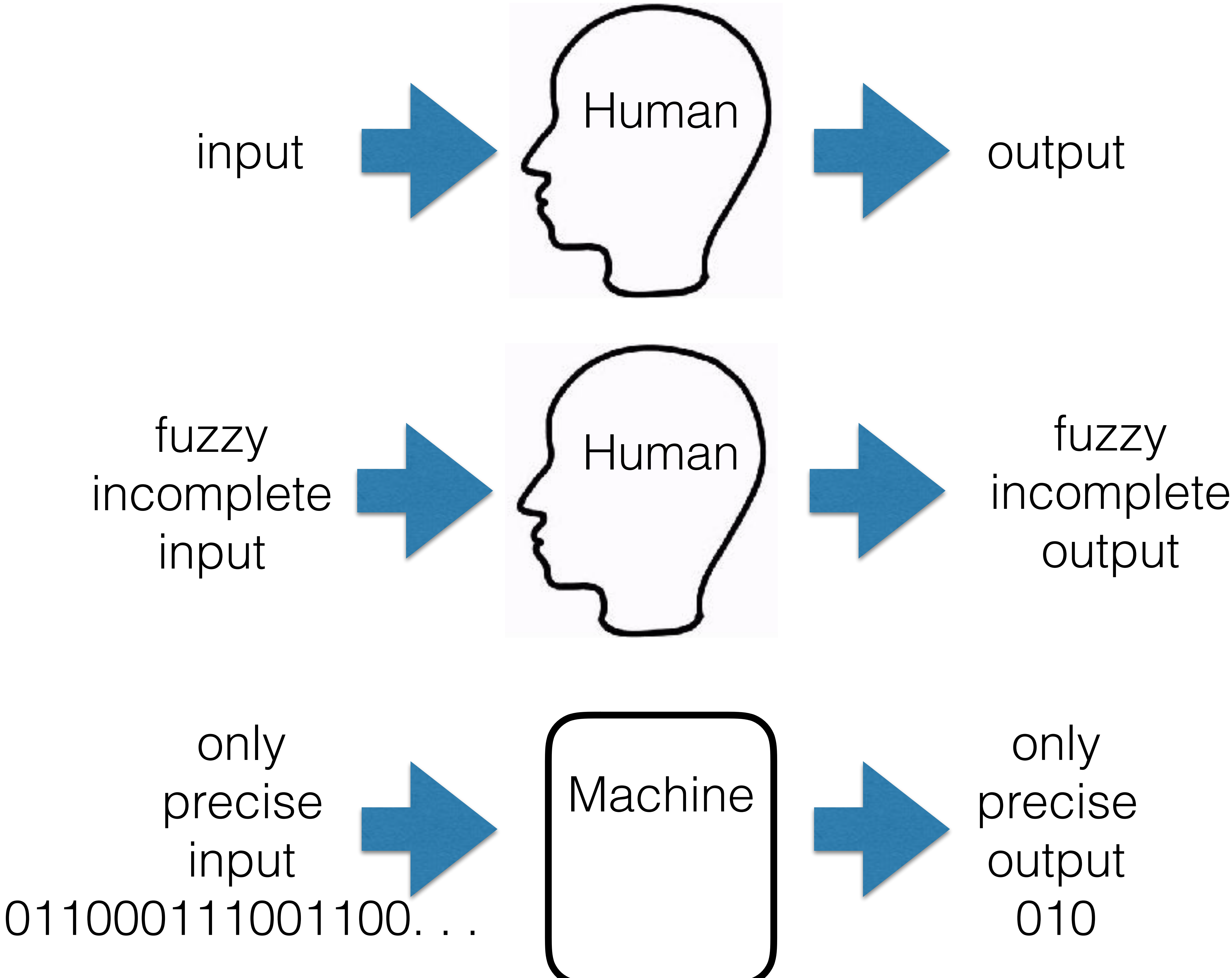
Need to know whats going on inside deep learning

a metaphor

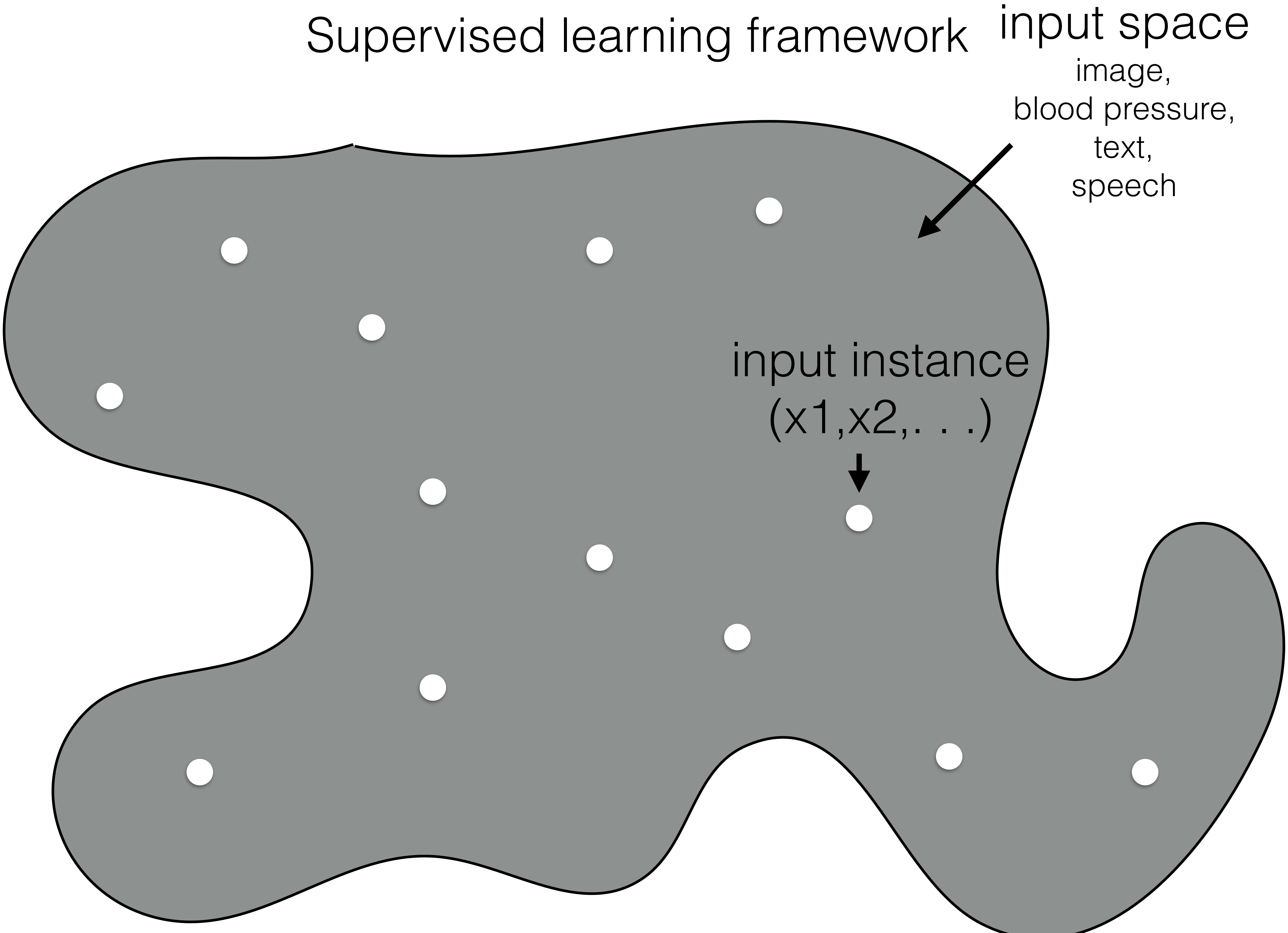
step on this pedal the car will go
step on that pedal the car will stop
why??

Very basic

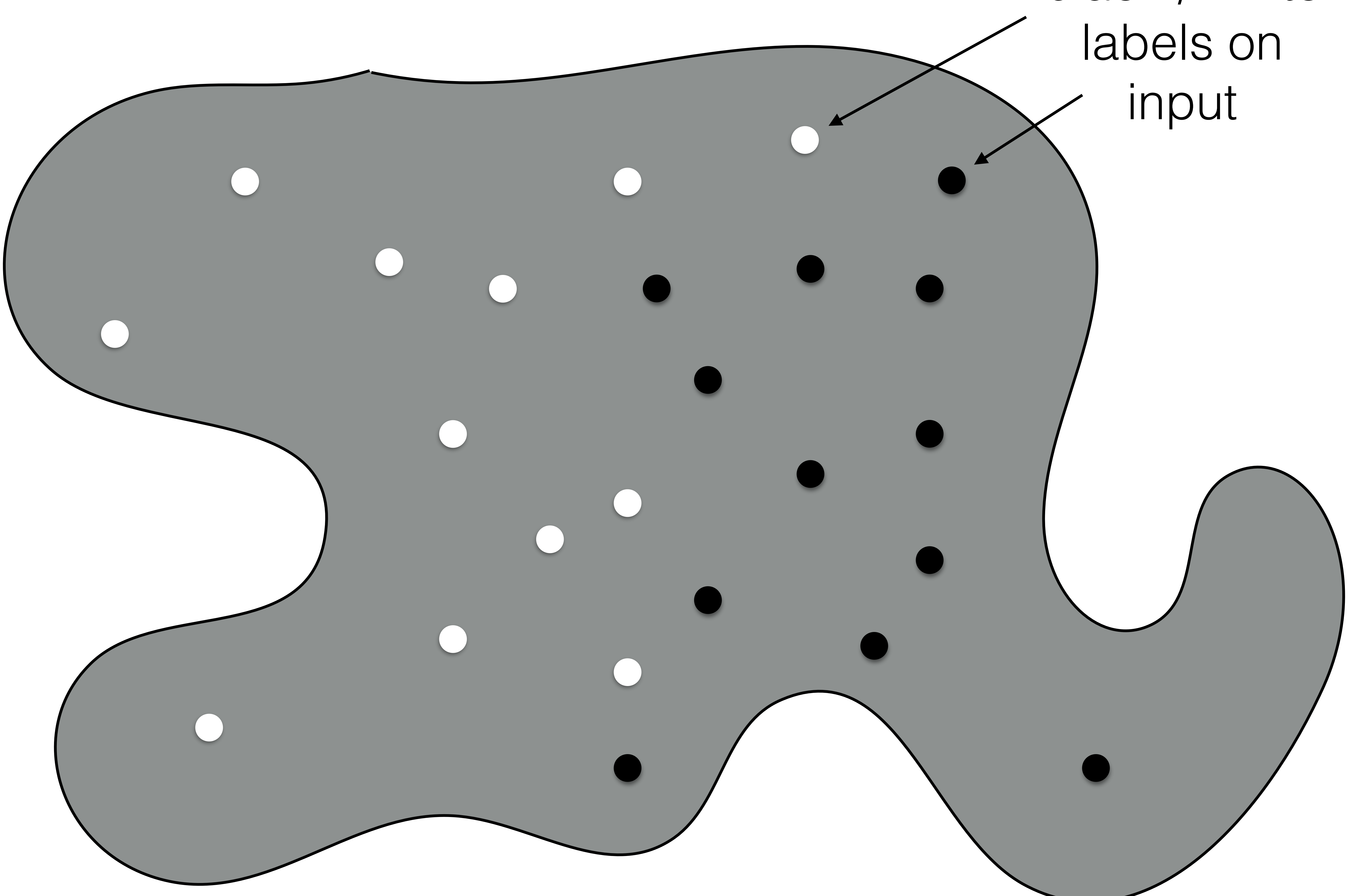
lets get everyone on the same level
sorry if this seems too simple to some of you


input
Human
output
fuzzy
incomplete
input
Human
fuzzy
incomplete
output
only
precise
input
01100011100 1100. . .
Machine
only
precise
output
010

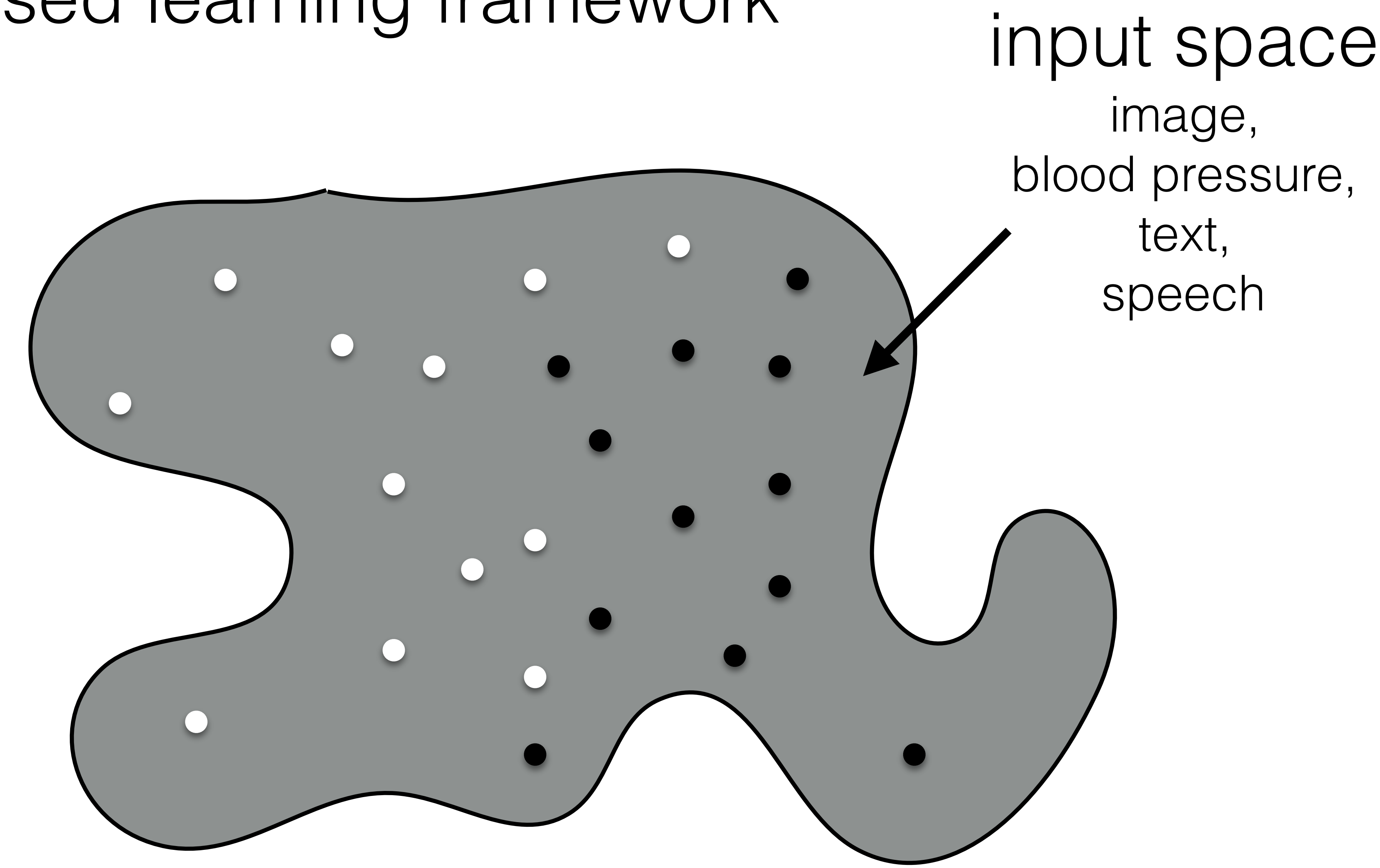
Supervised learning framework

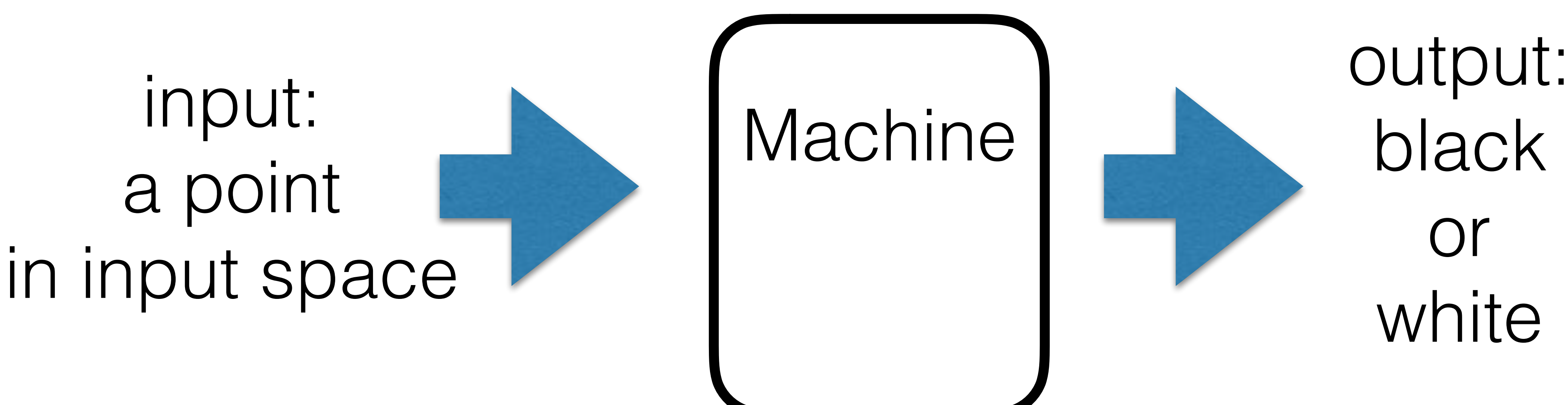


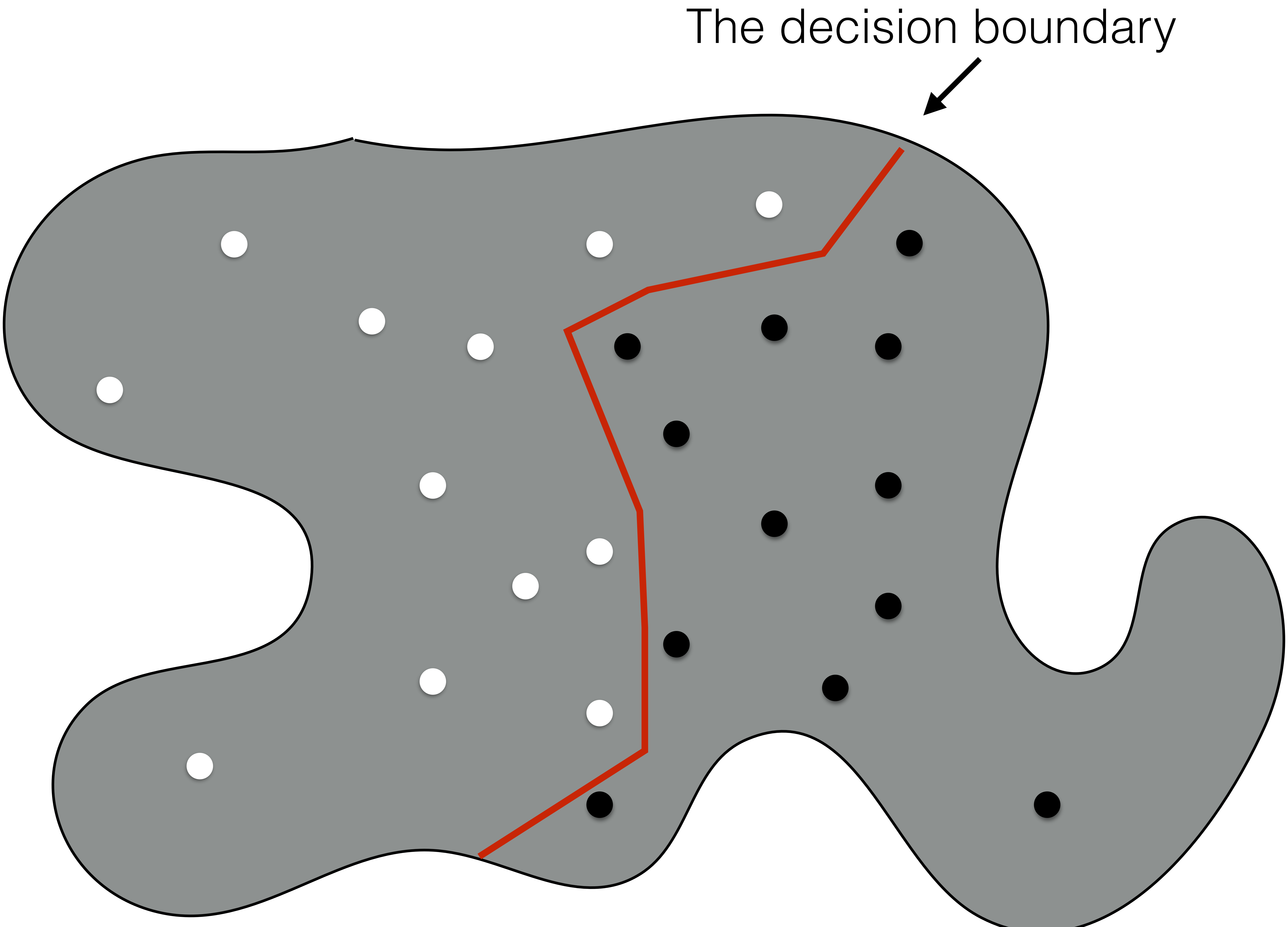
Supervised learning framework

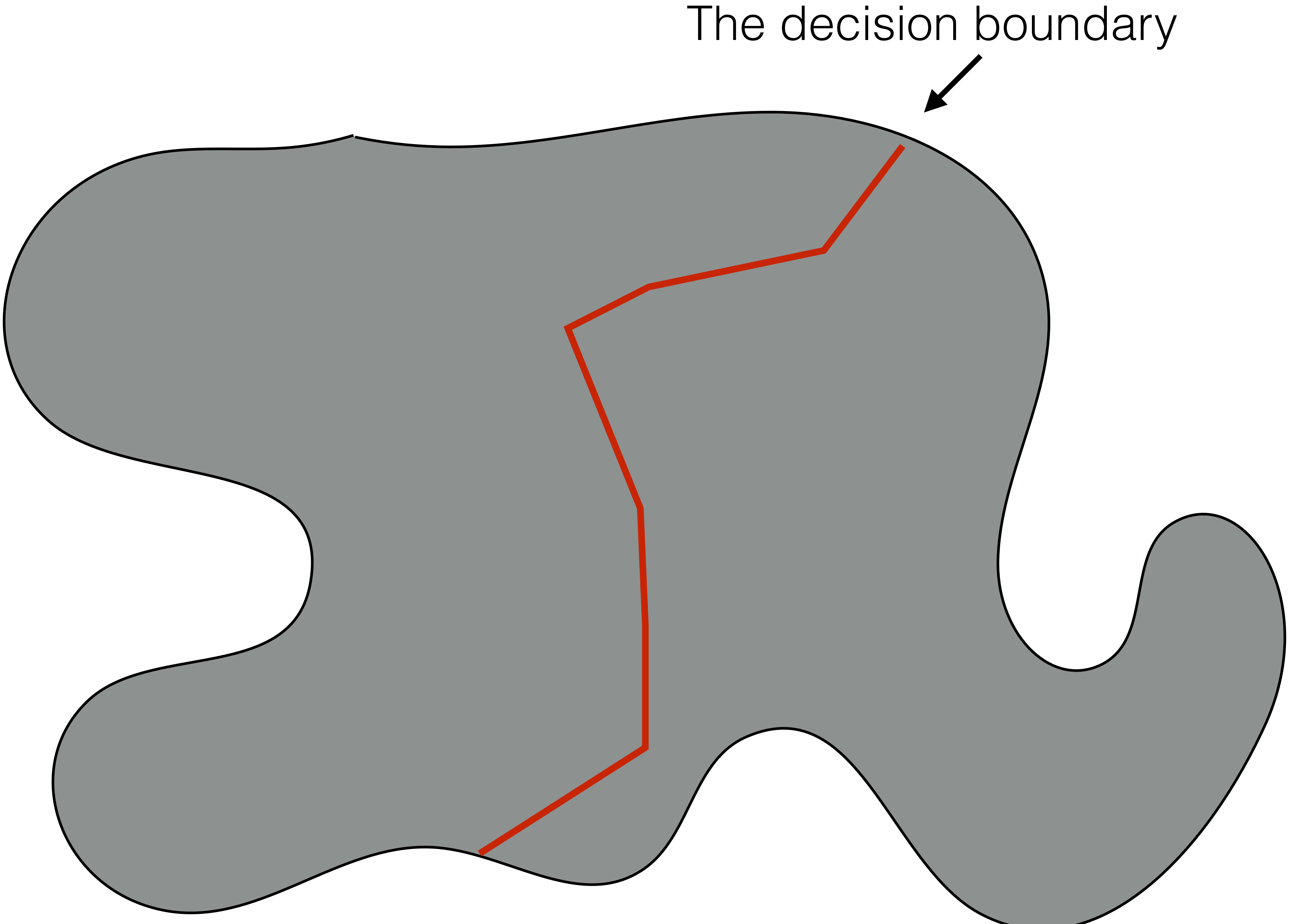


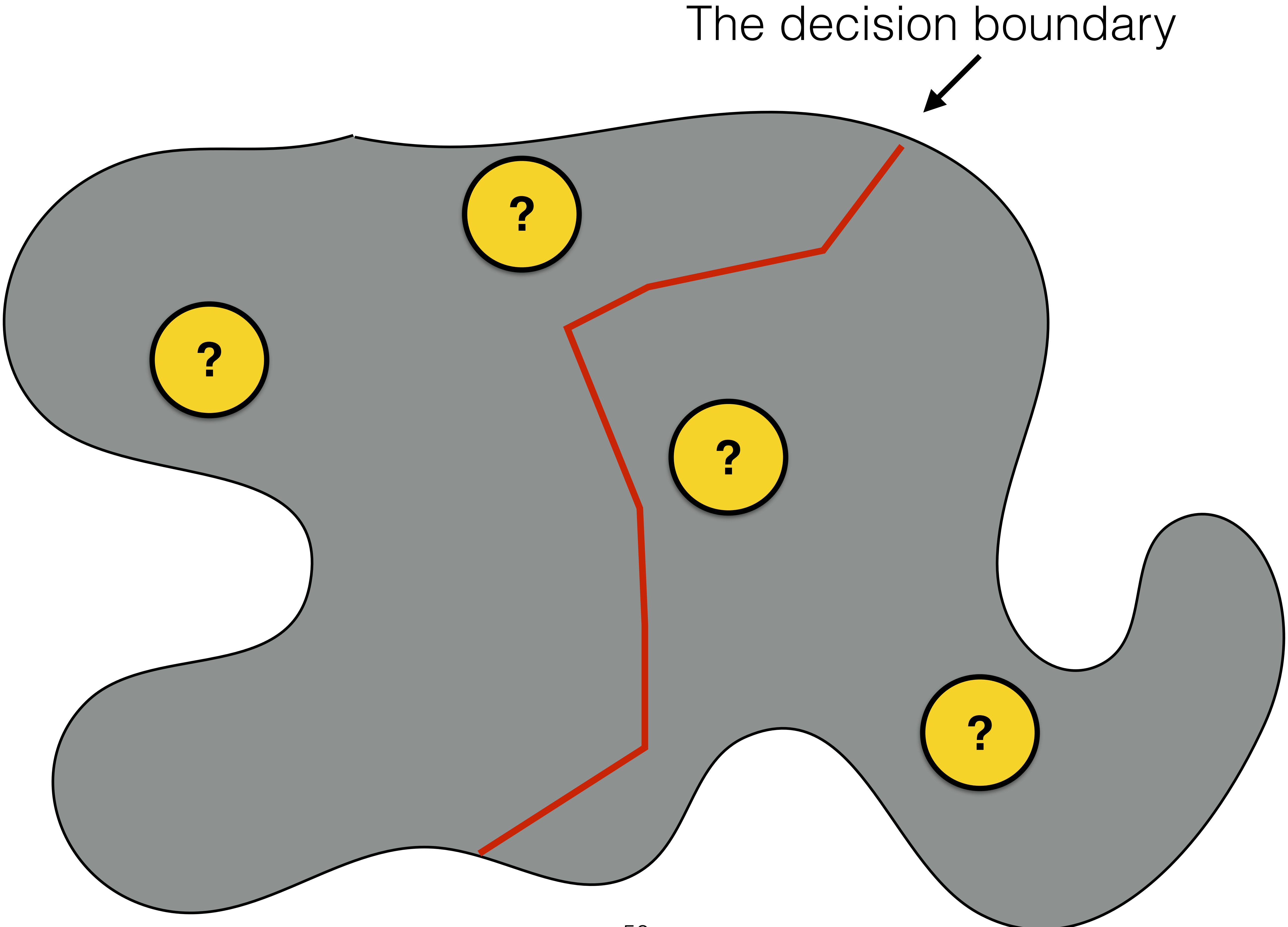
Supervised learning framework

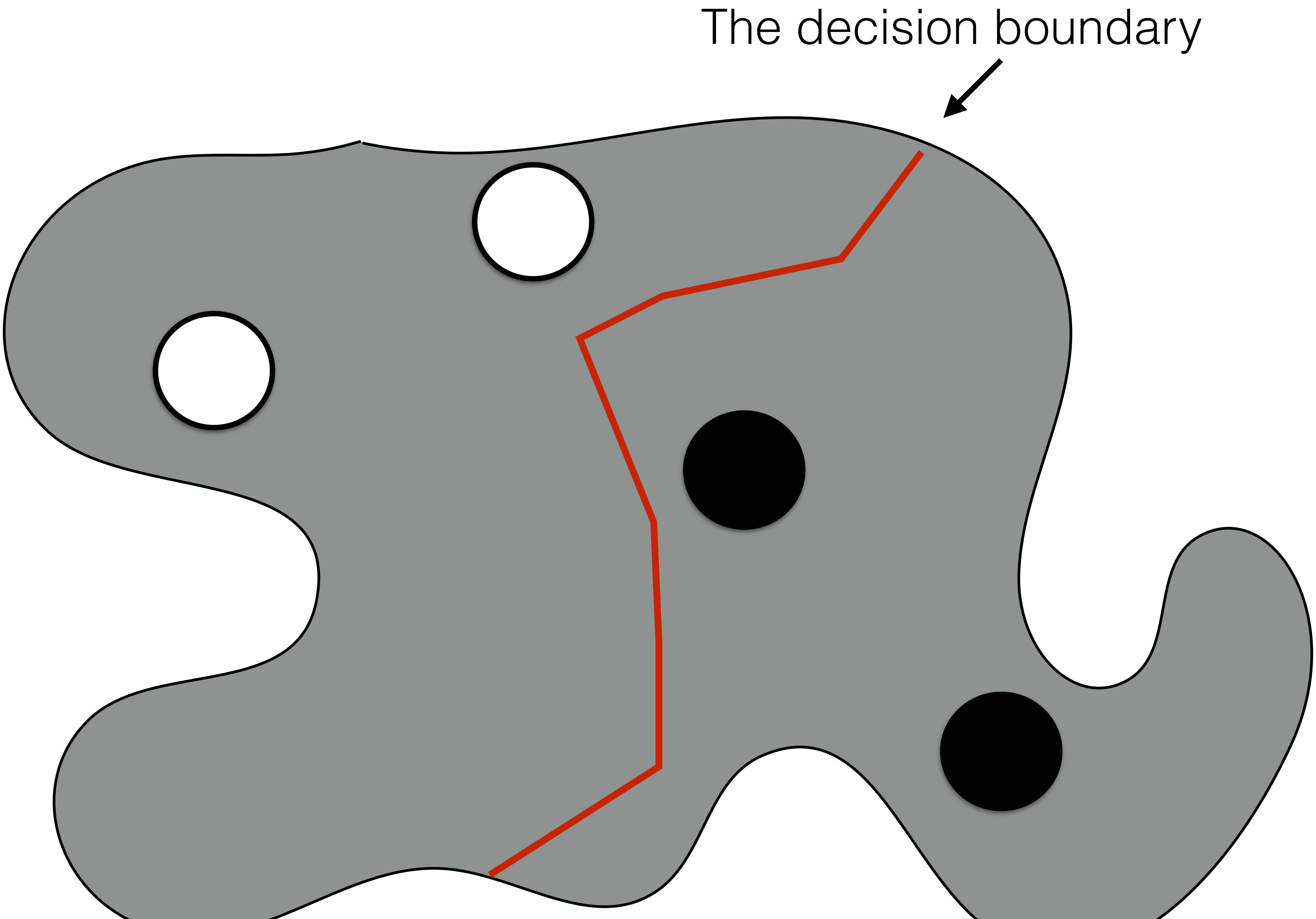





The decision boundary


The decision boundary

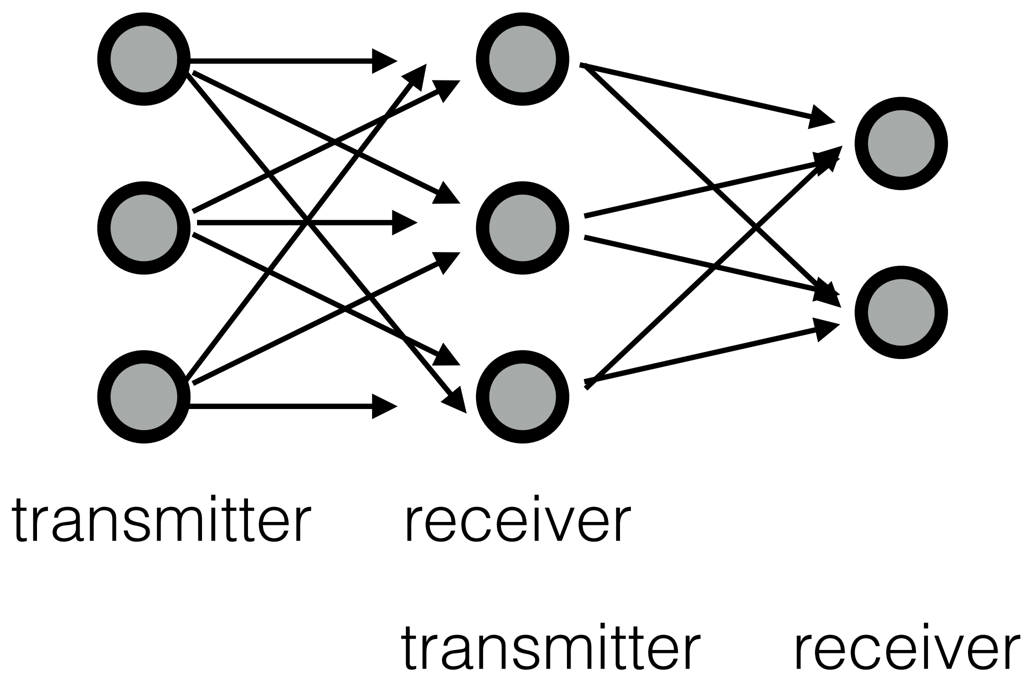

The decision boundary
?
?
?
?


The decision boundary

Forward Propagation

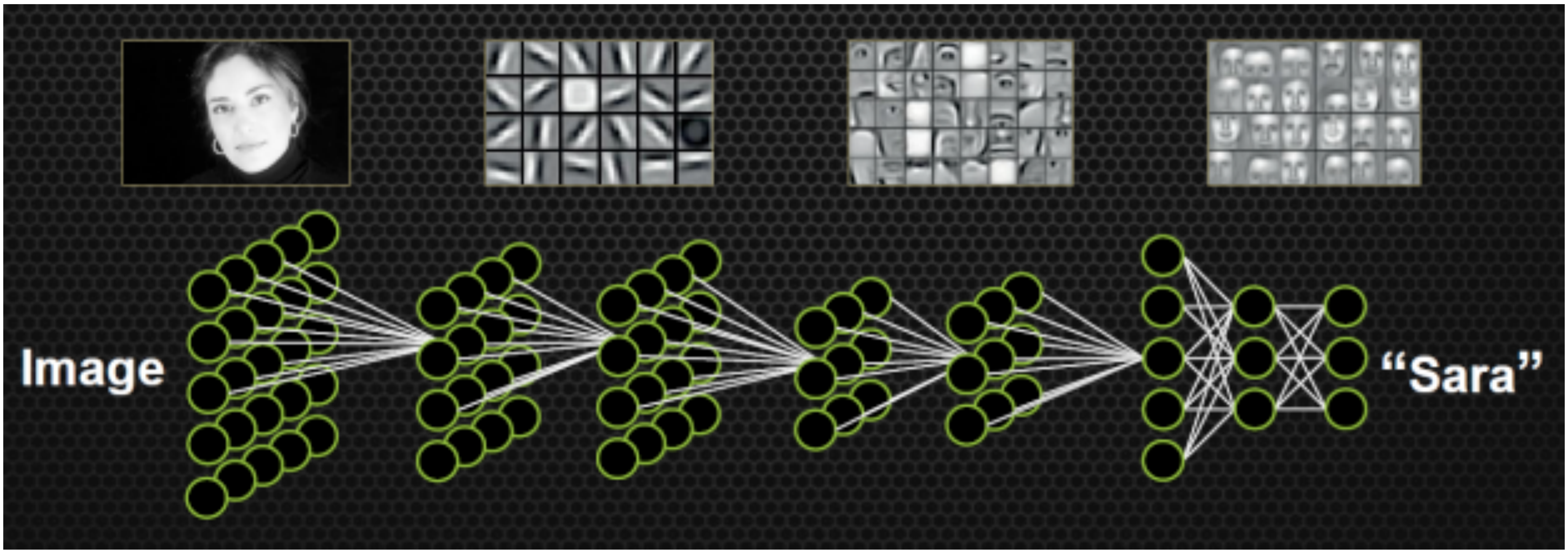
The network and information flow

feed in data into first layer



final layer presents the output of network

There can be millions of connections



Notation

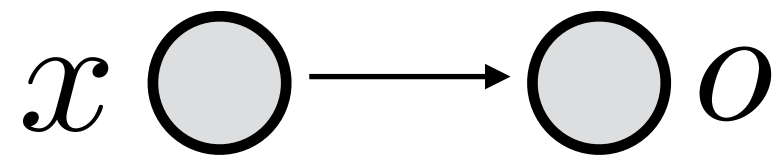
Let $x \in \mathbb{R}^d$ be the input space

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

$x \in \mathbb{R}$

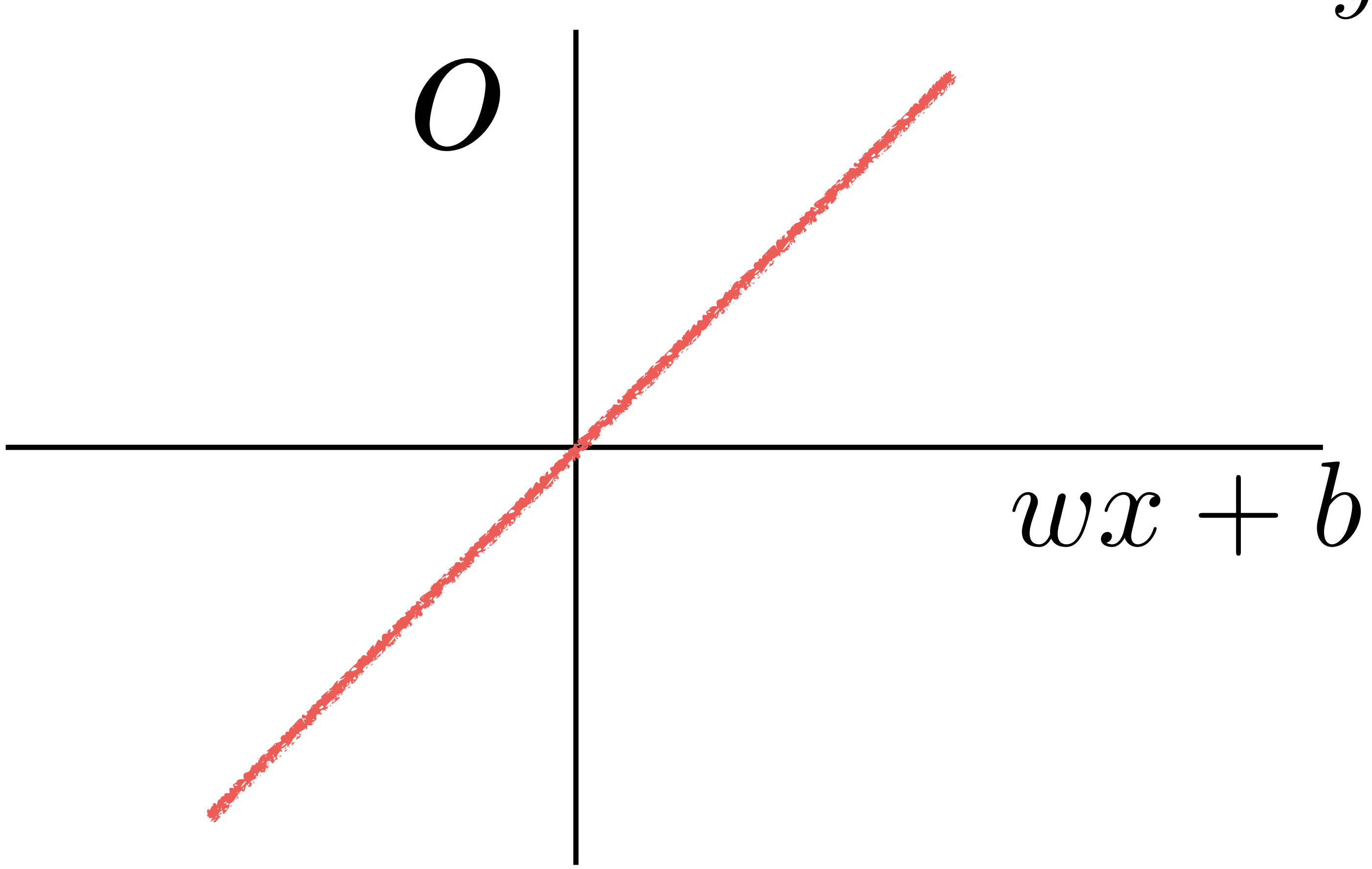


$$y = \begin{cases} 1 & \text{if } x > \tau \\ 0 & \text{otherwise} \end{cases}$$

$$o = wx + b$$

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

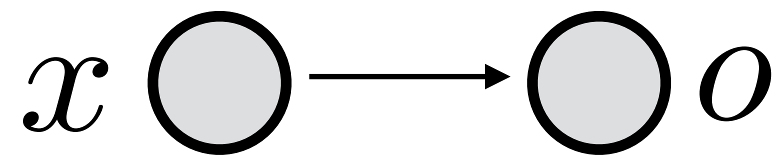
$$w\tau + b = 0$$



Simplest perceptron - rectilinear activation function

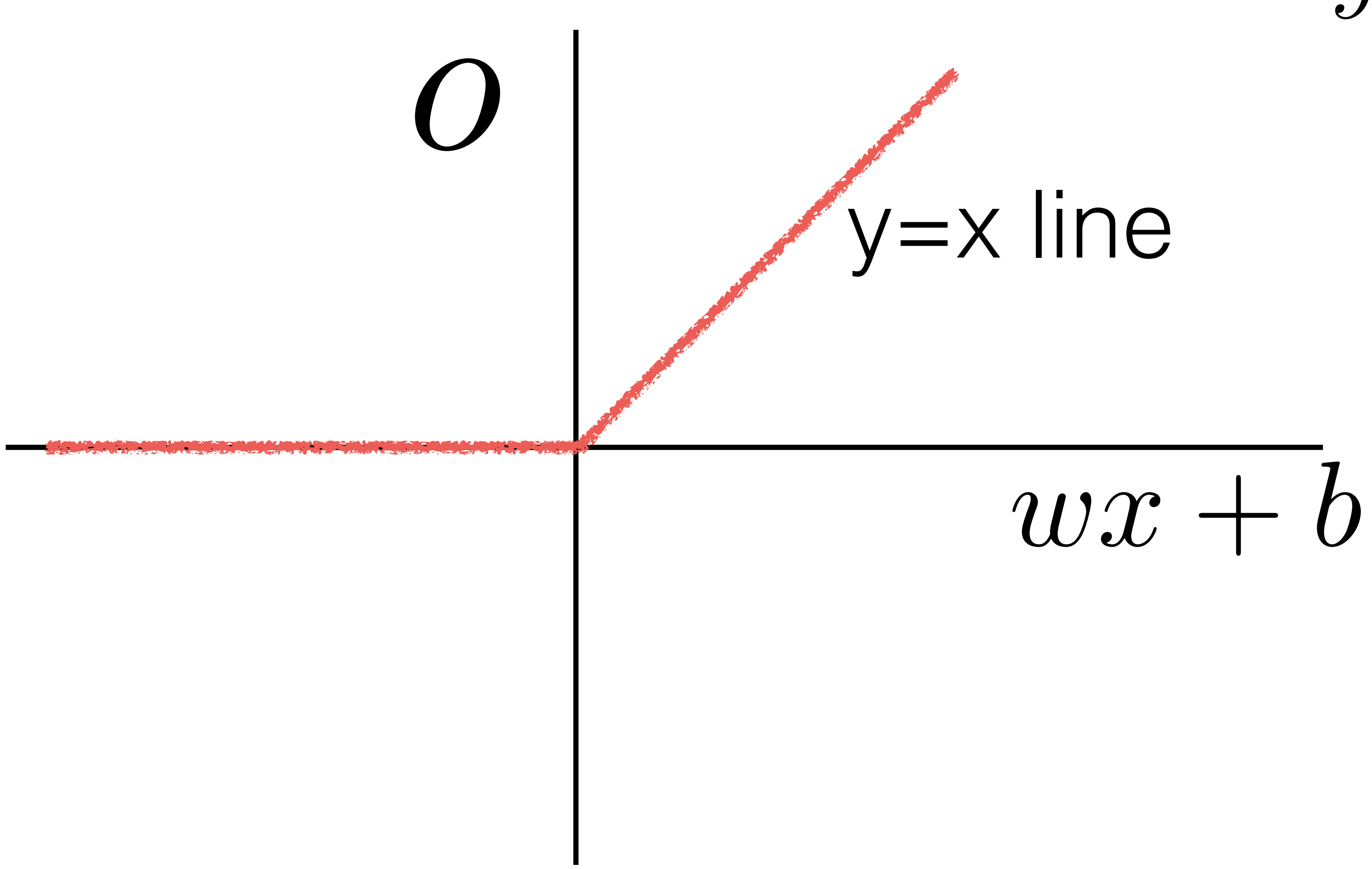
$x \in \mathbb{R}$

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



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

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



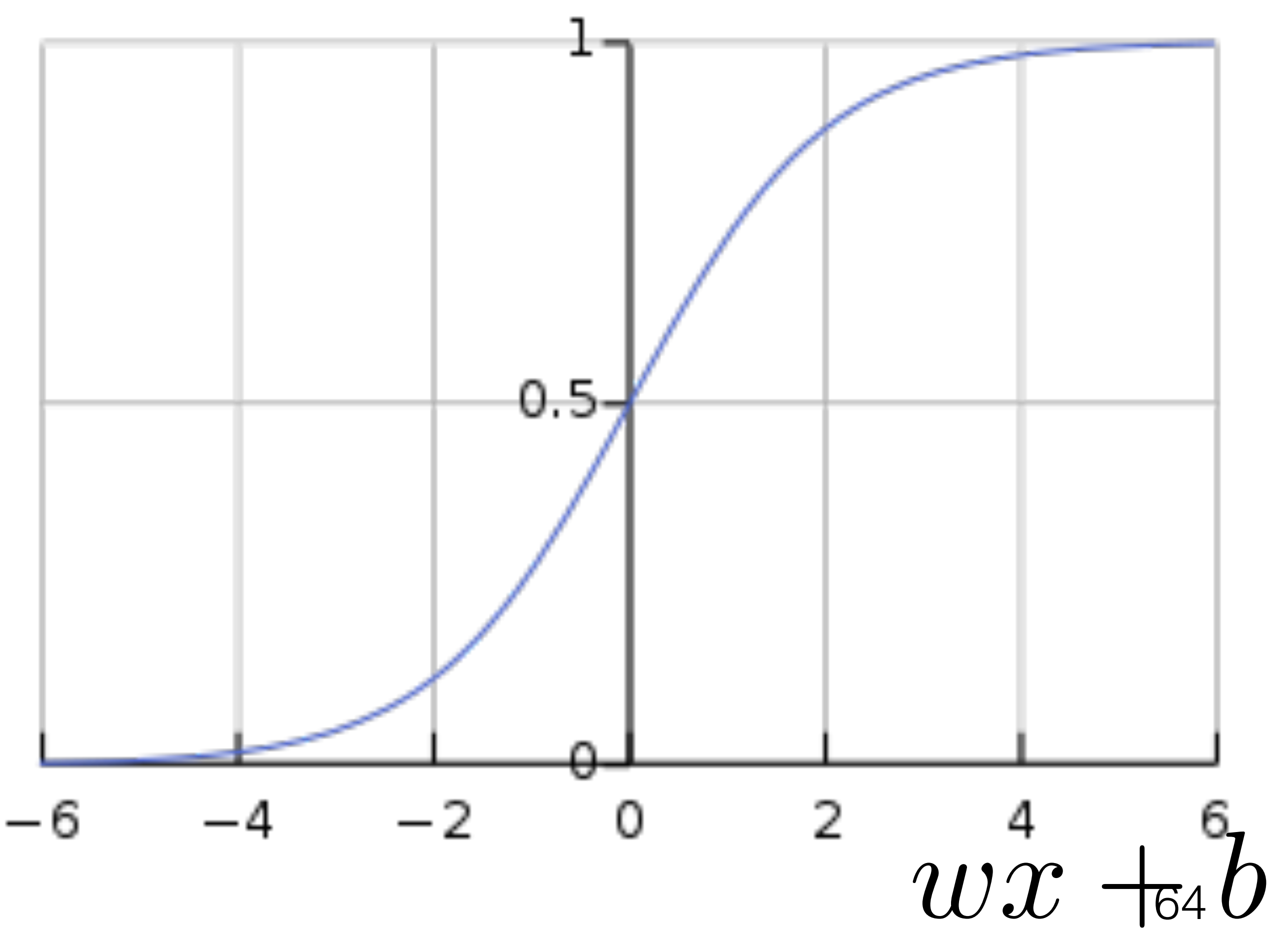
$$w\tau + b = 0$$

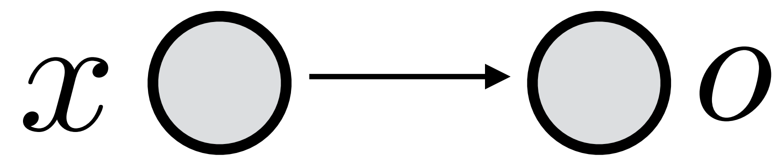
Simplest perceptron - sigmoid activation function

$x \in \mathbb{R}$

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

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

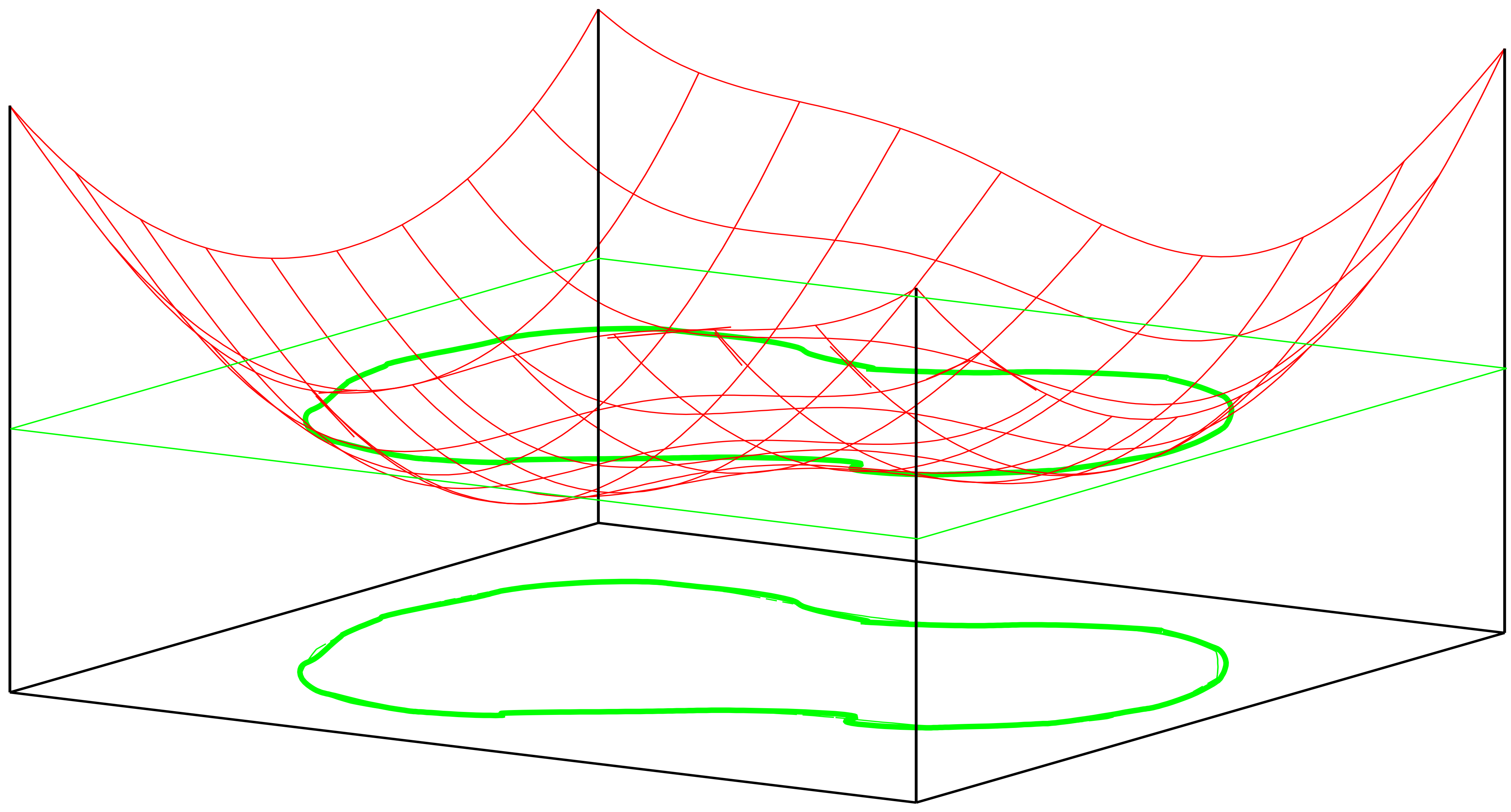




$$y = \begin{cases} 1 & \text{if } wx + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$w\tau + b = 0$$

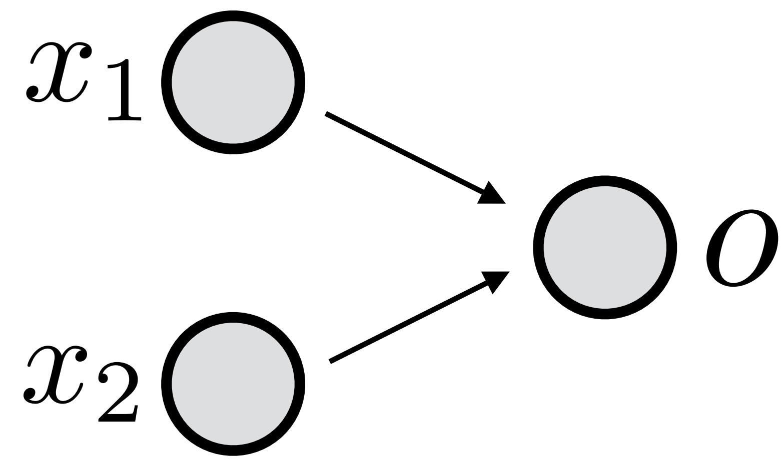
Concept of level sets

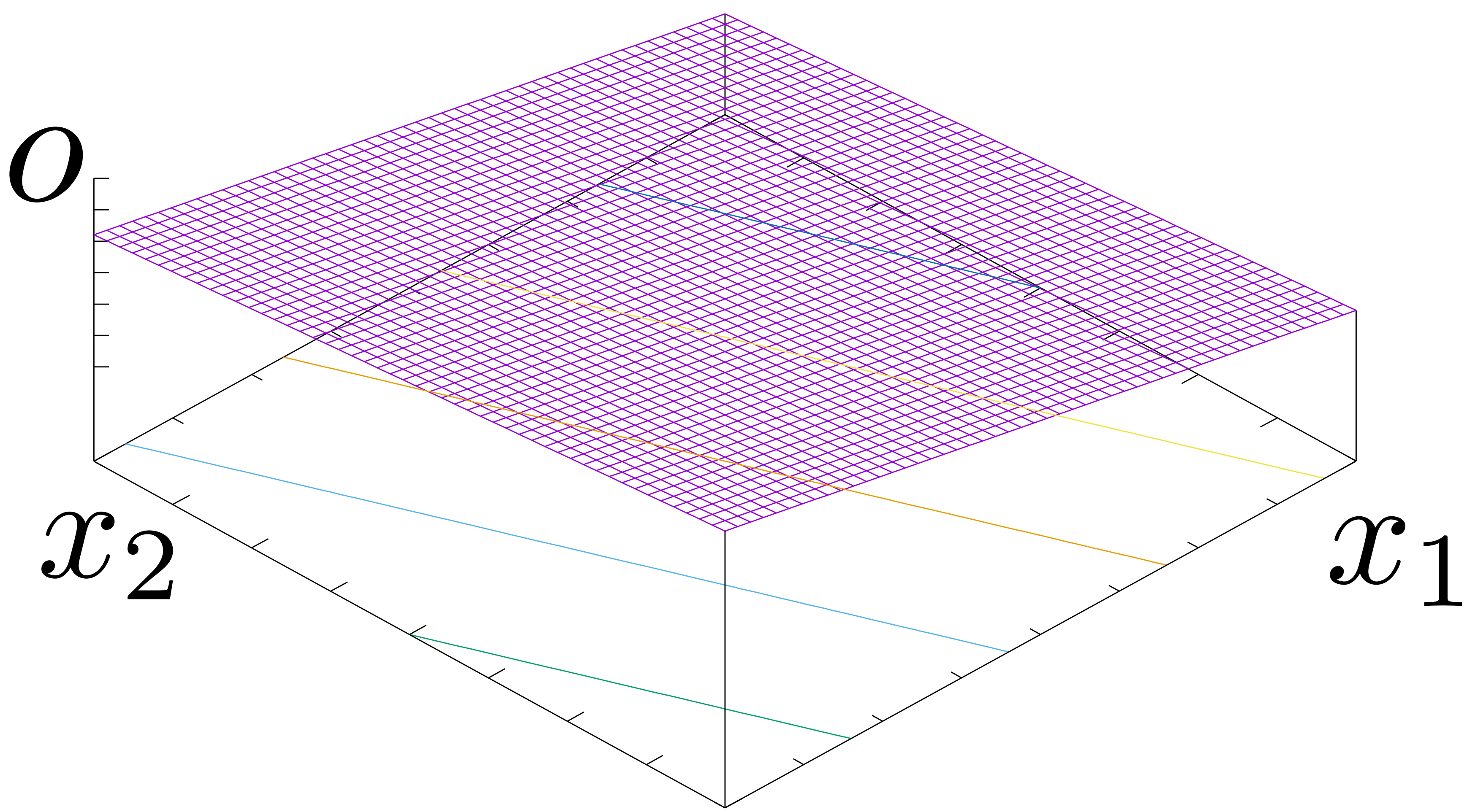


Next to simplest

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

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

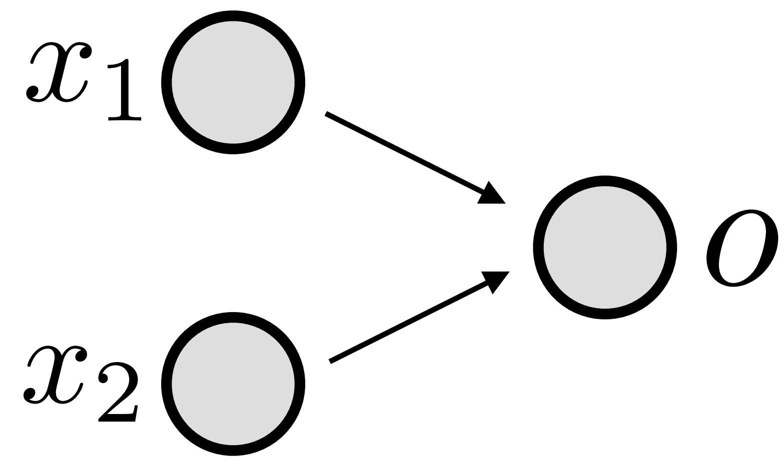


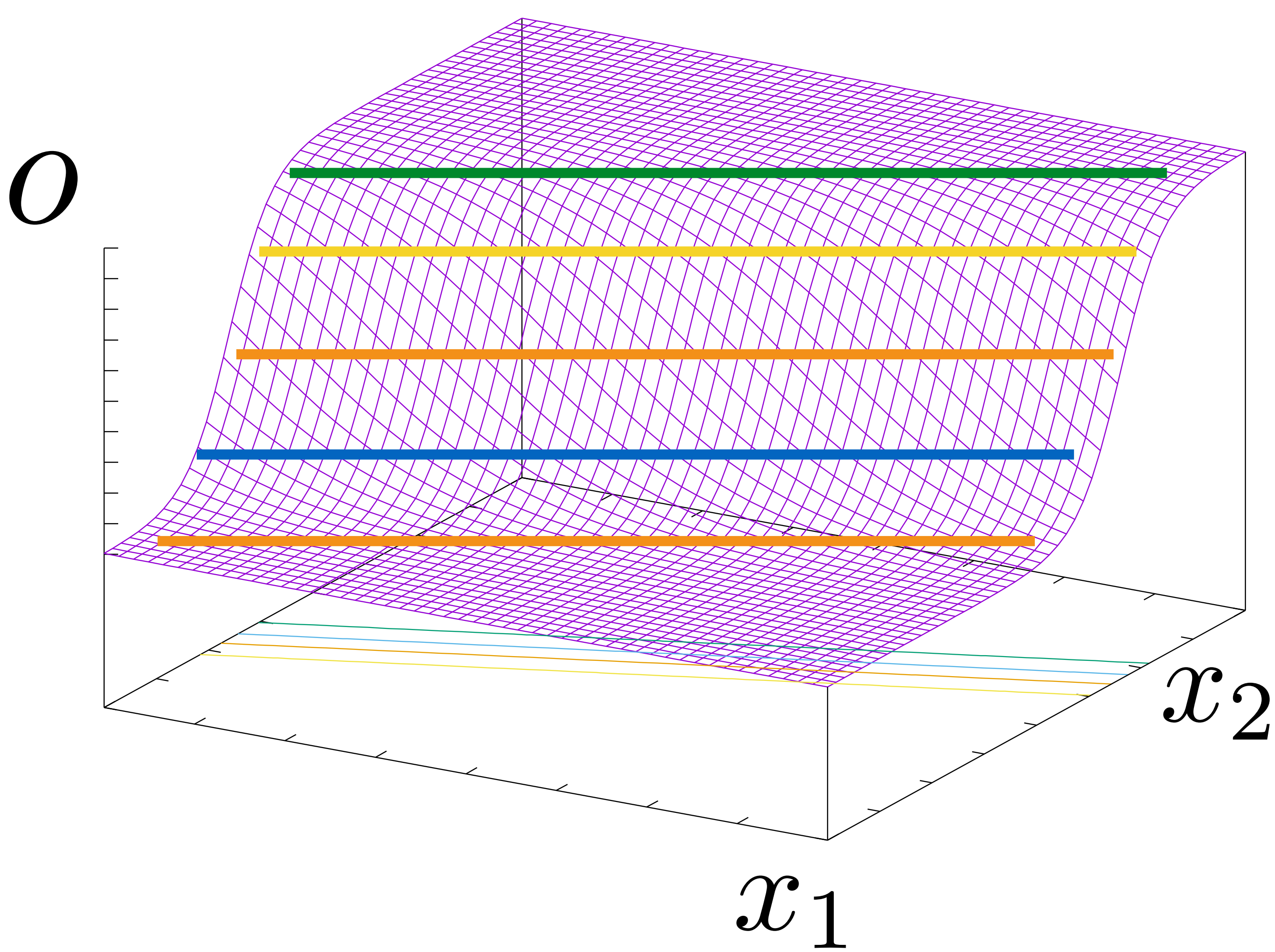


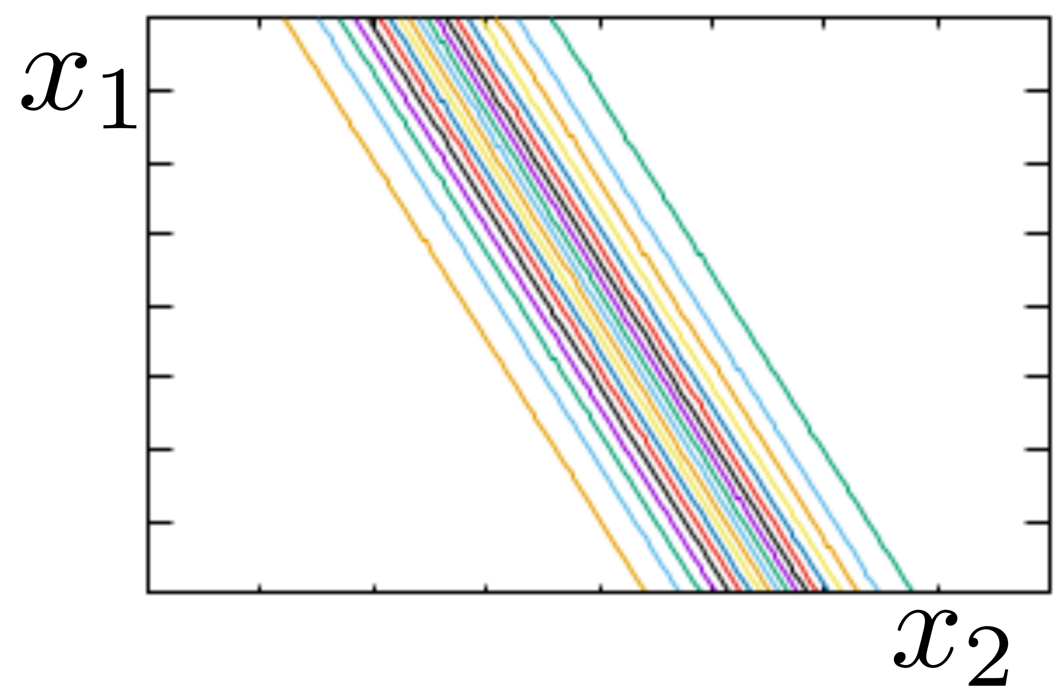
Next to simplest

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

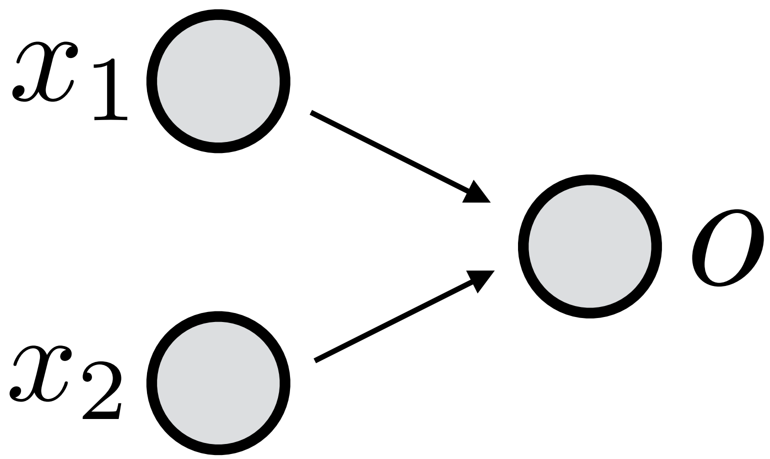
$$o = \sigma(w_1 x_1 + w_2 x_2 + b)$$





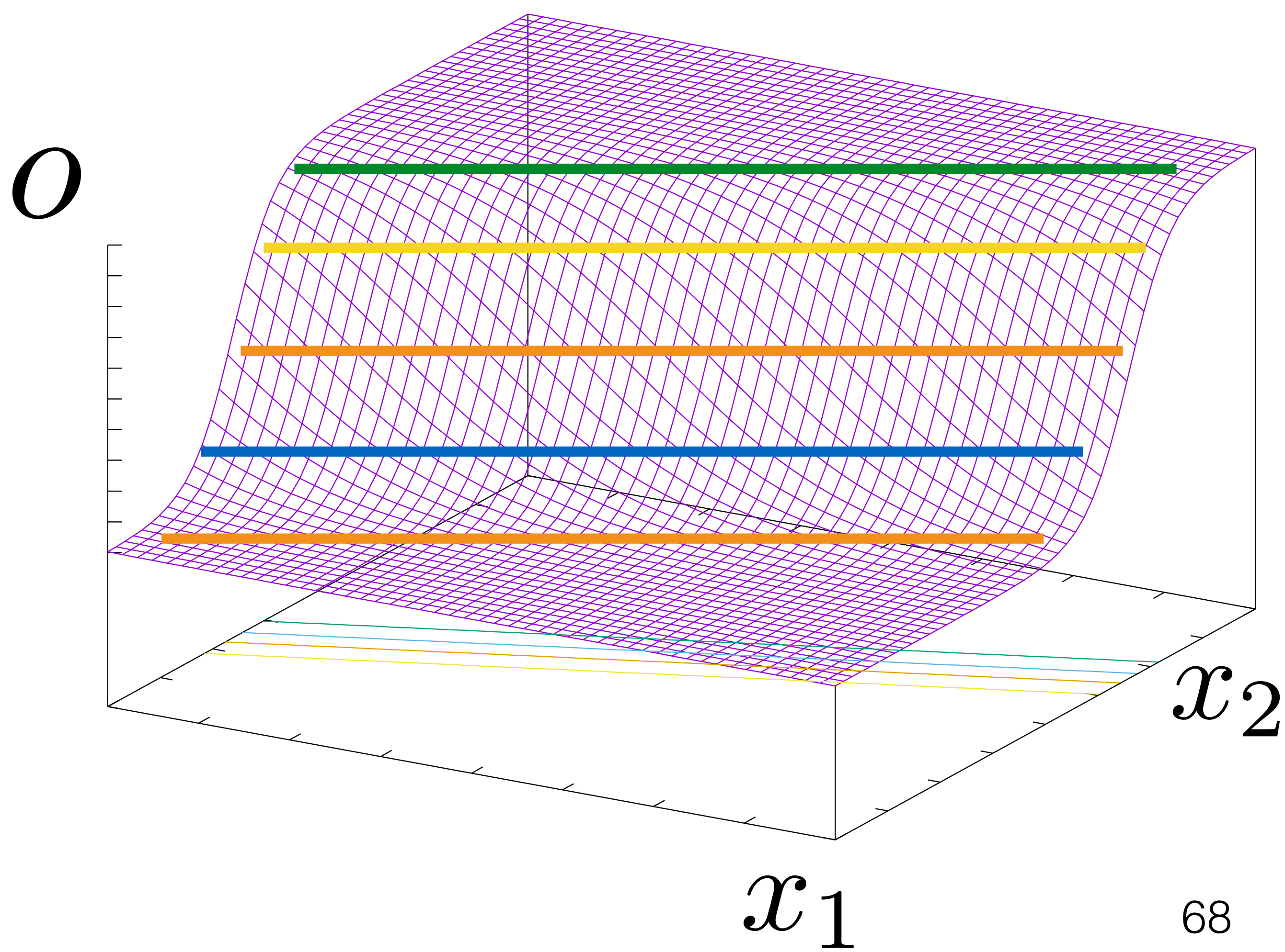


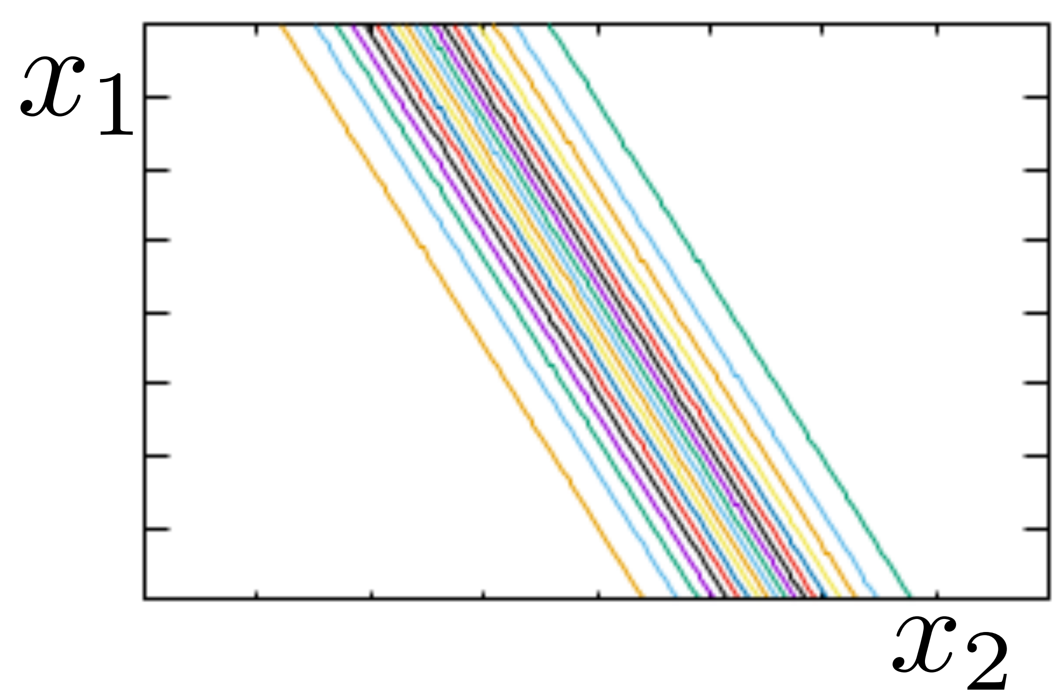
Network view and logistic regression view



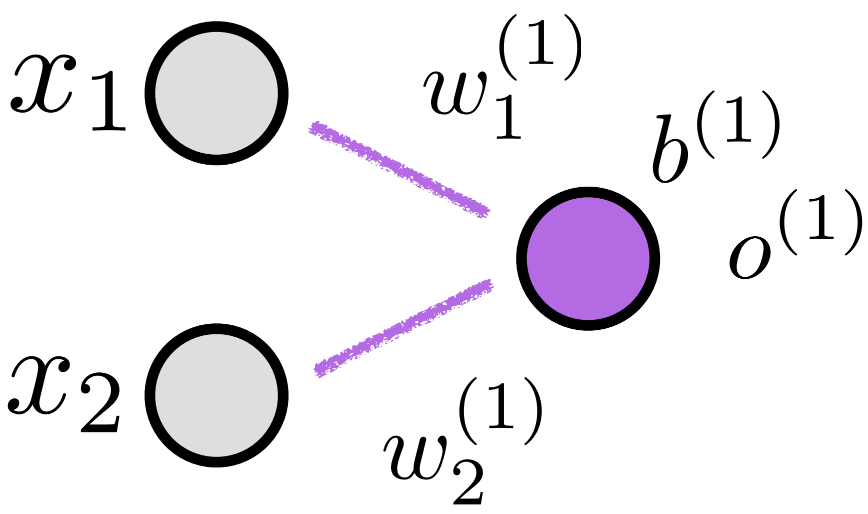
$$o = \frac{1}{1 + \exp(-w_1 x_1 - w_2 x_2 - b)}$$

Logistic regression

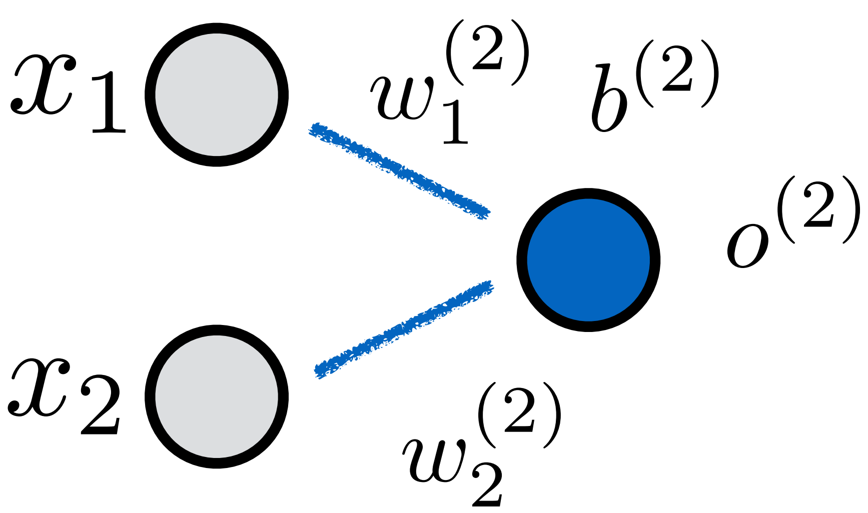




Stacked up

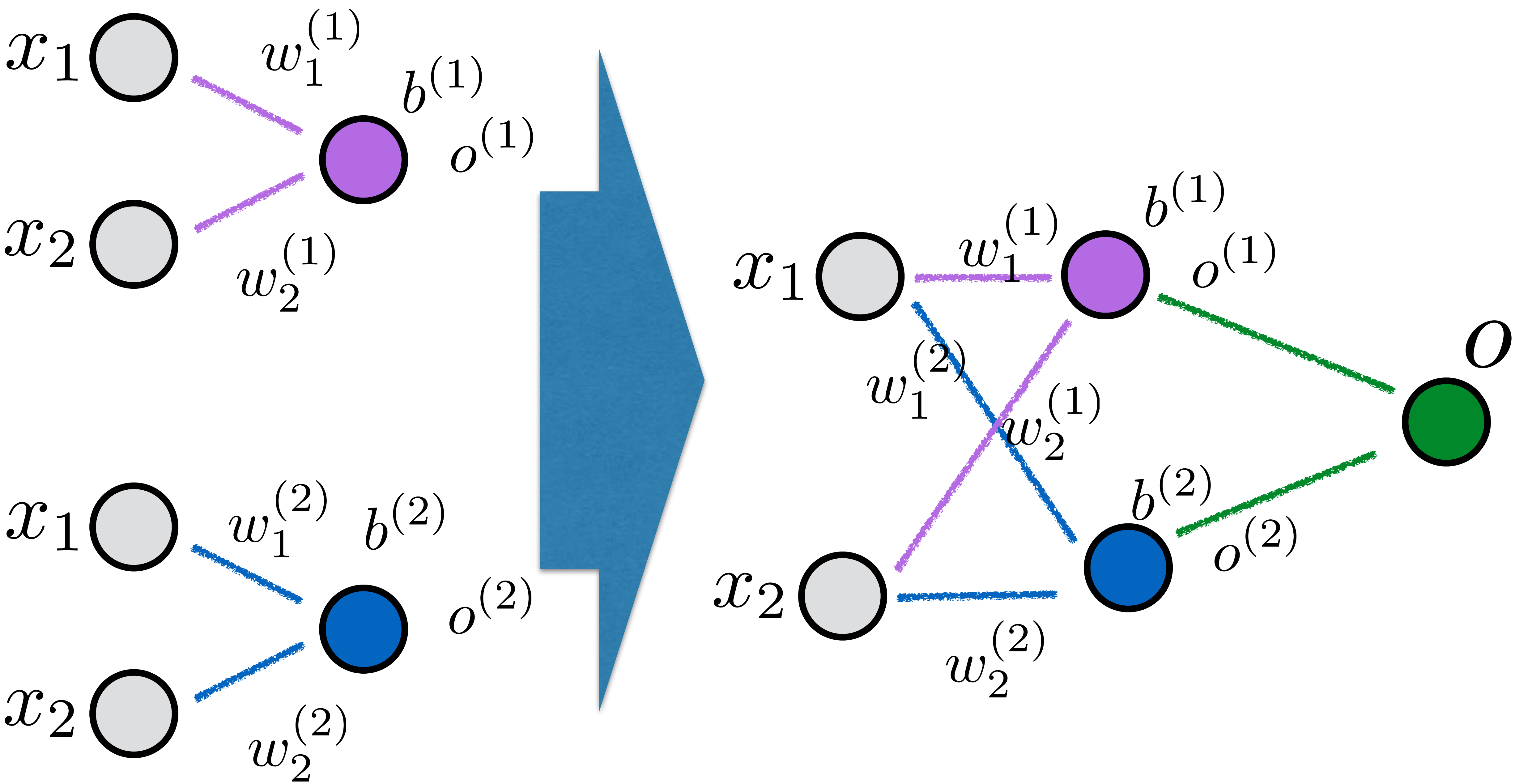


$$o^{(1)} = w_1^{(1)} x_1 + w_2^{(1)} x_2 + b^{(1)}$$



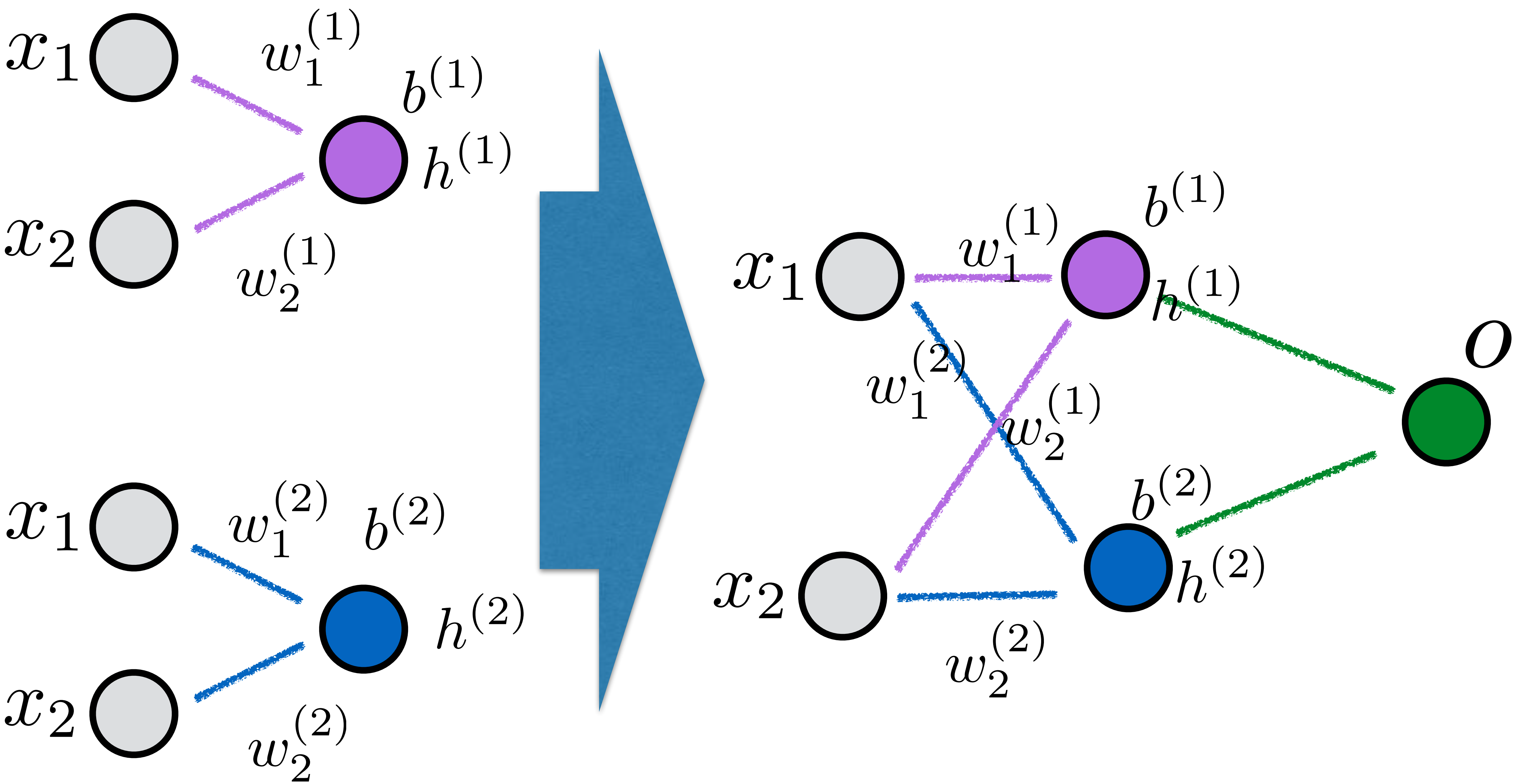
$$o^{(2)} = w_1^{(2)} x_1 + w_2^{(2)} x_2 + b^{(2)}$$

Stacked up



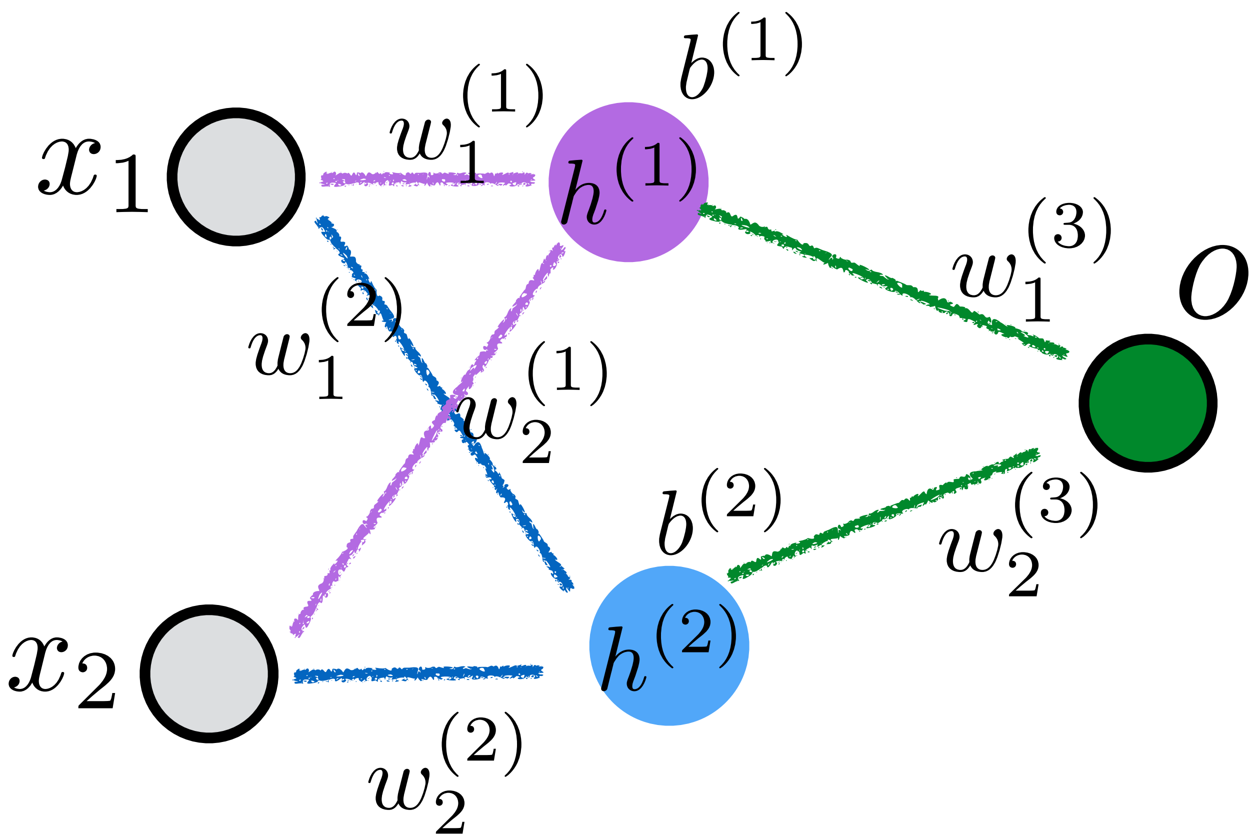
x_1
$w_1^{(1)}$
$b^{(1)}$
$o^{(1)}$
x_2
$w_2^{(1)}$
x_1
$w_1^{(2)}$
$b^{(2)}$
$o^{(2)}$
x_2
$w_2^{(2)}$
x_1
$w_1^{(1)}$
$b^{(1)}$
$o^{(1)}$
o
$w_1^{(2)}$
$w_2^{(1)}$
$b^{(2)}$
$o^{(2)}$
x_2
$w_2^{(2)}$

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

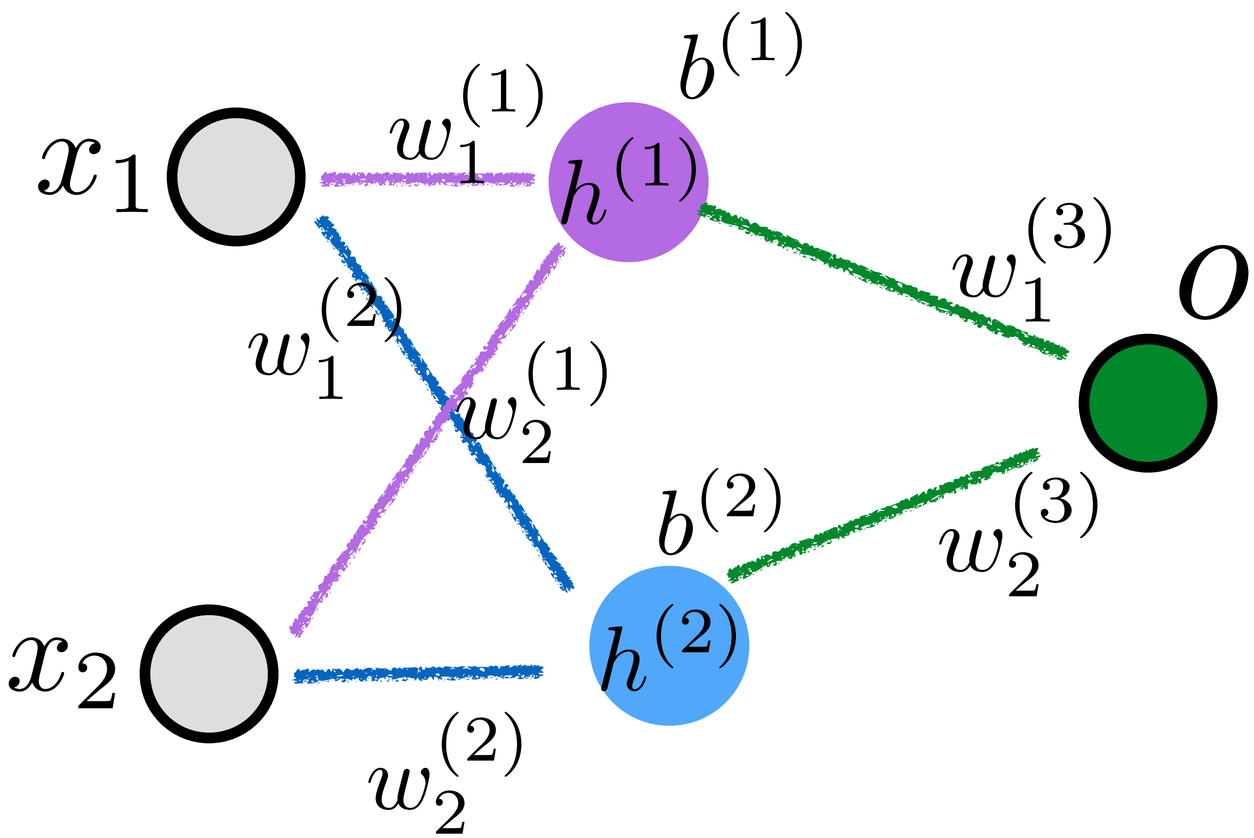


$$h^{(1)} = w_1^{(1)} x_1 + w_2^{(1)} x_2 + b^{(1)}$$

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

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

Stacked up with general activation function



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

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

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

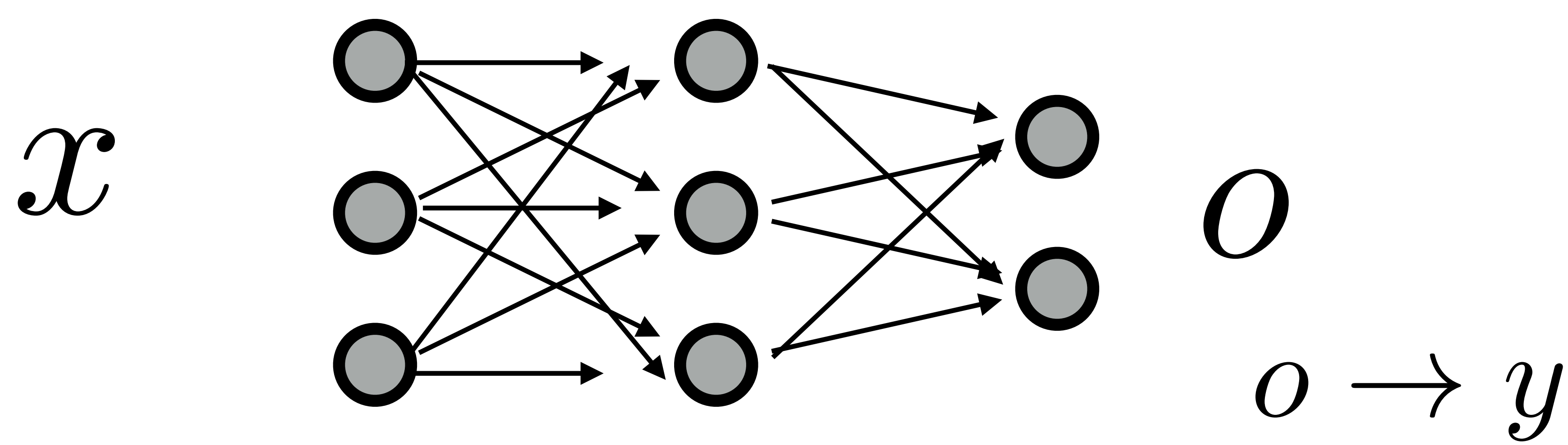
A general objective

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

$$o \approx y$$

How to tweak? Adjust the weights

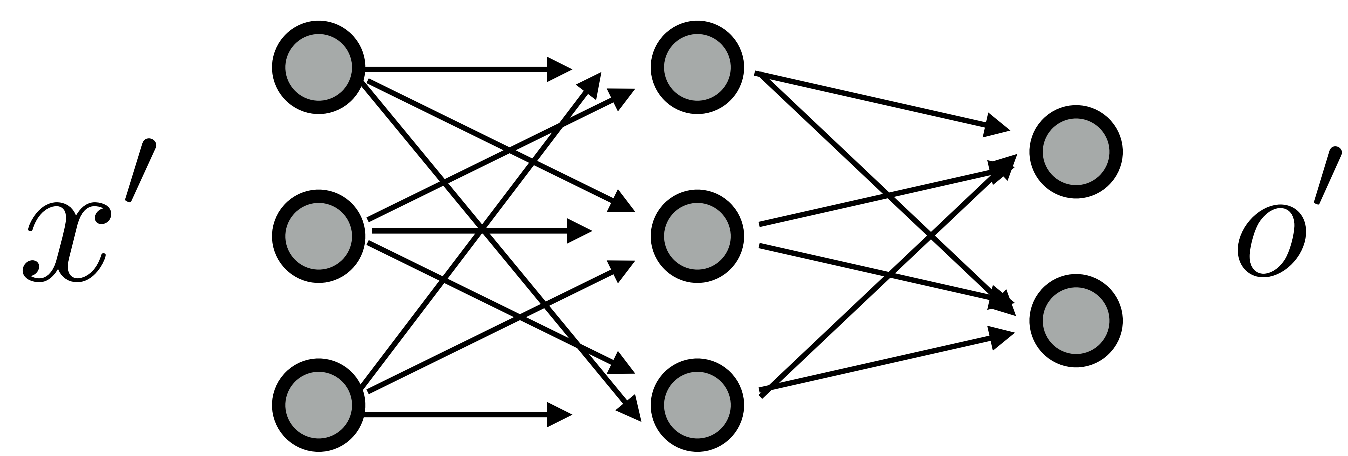
Given an example (x, y)



For this example, the neural network ‘behaves well’

After the first example, give another example

$$(x', ?)$$

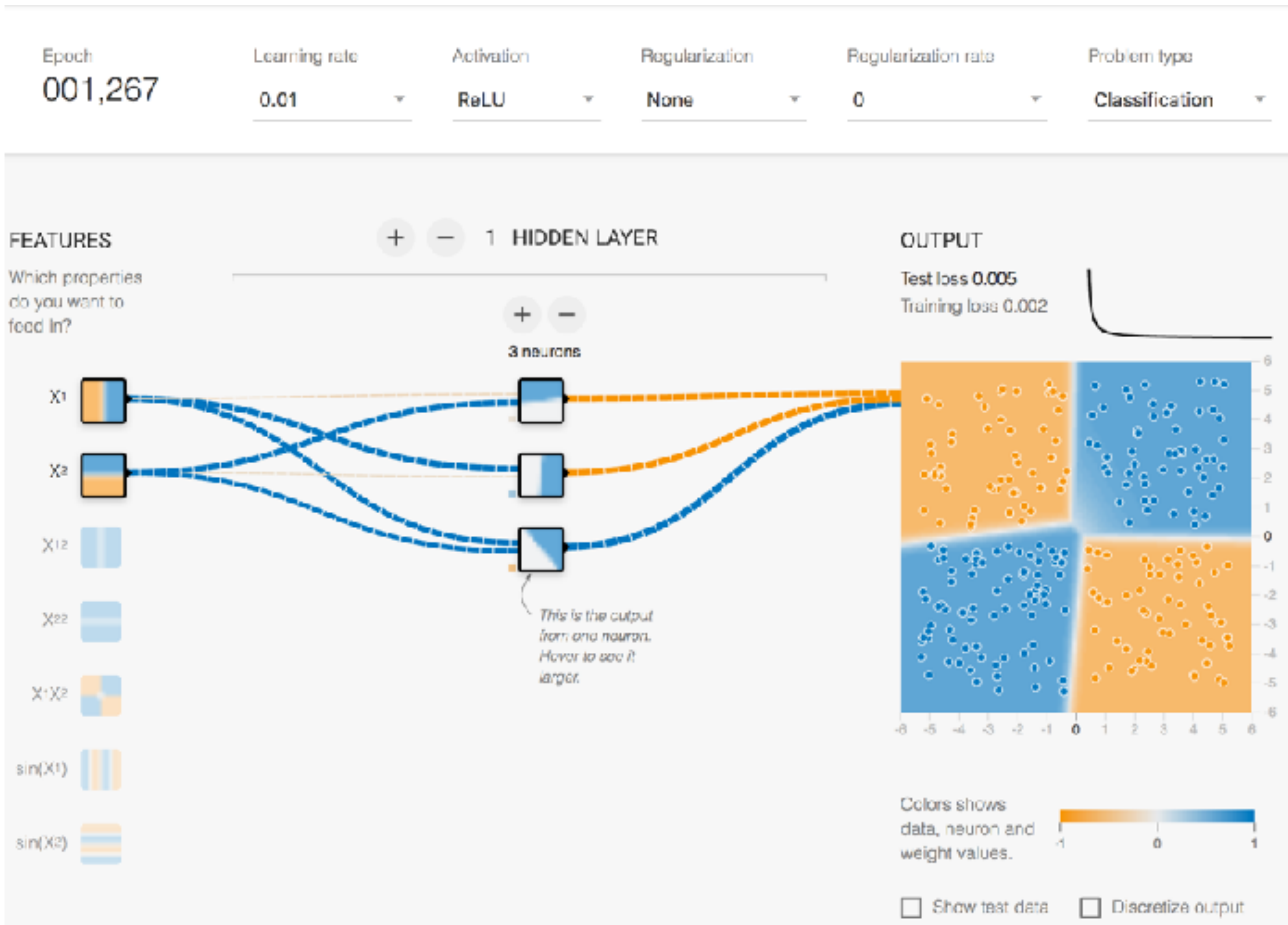


Now we say for a well behaved neural network

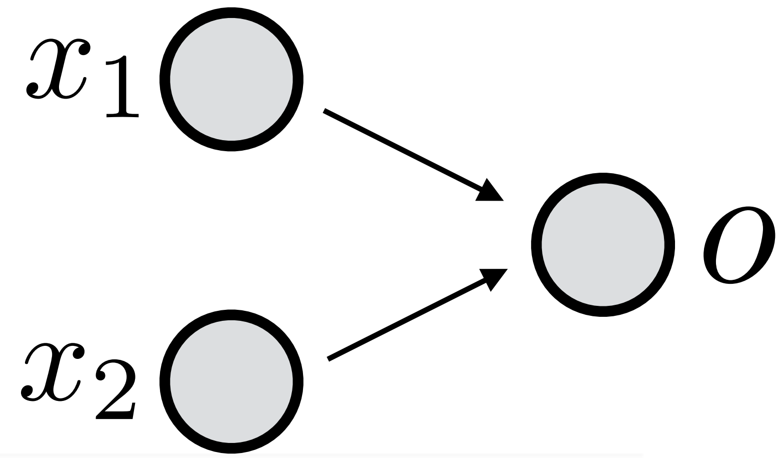
$$o' = y'$$

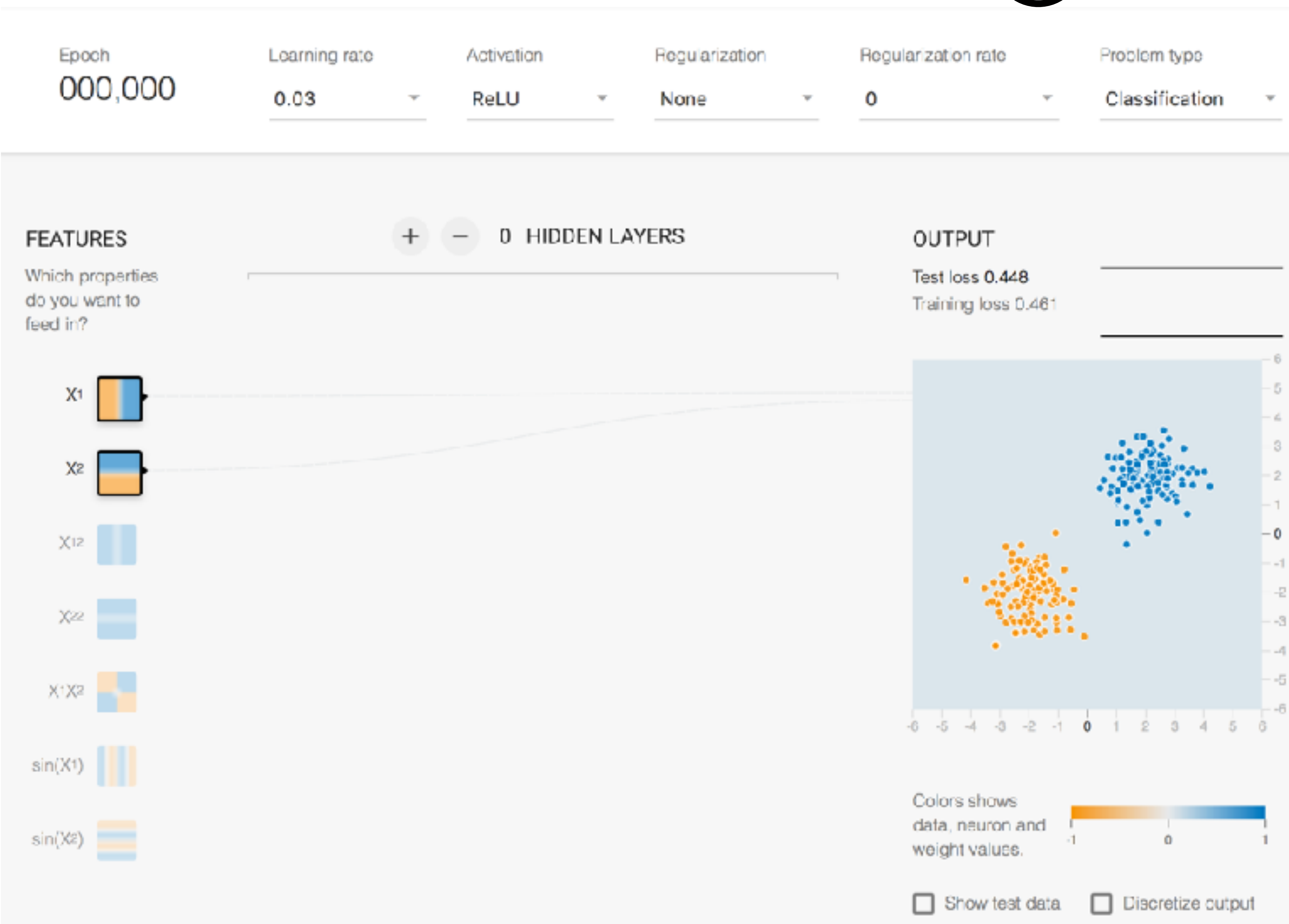
Animation @ playground search for "playground tensorflow"

introduction

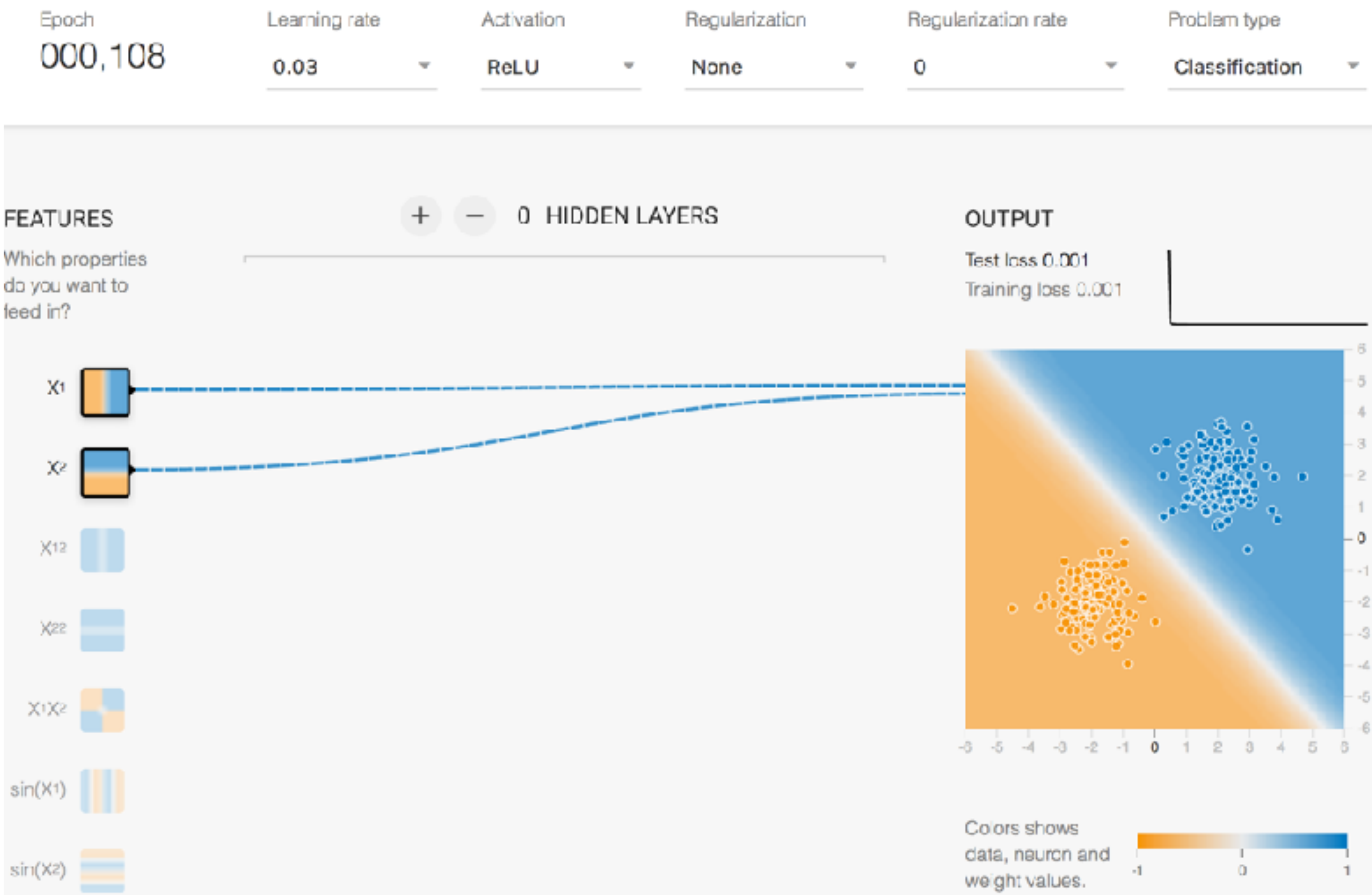


question


x_1
x_2
o

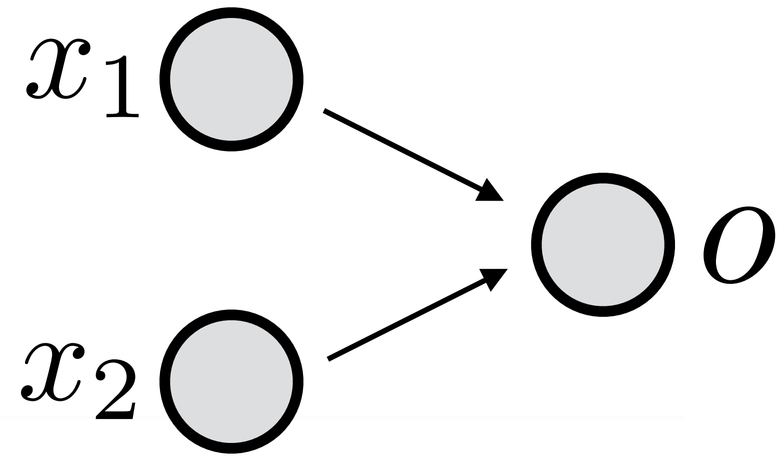

Epoch
000,000
Learning rate
0.03
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
0 HIDDEN LAYERS
OUTPUT
Test loss 0.448
Training loss 0.461
Colors shows data, neuron and weight values.
Show test data
Discretize output

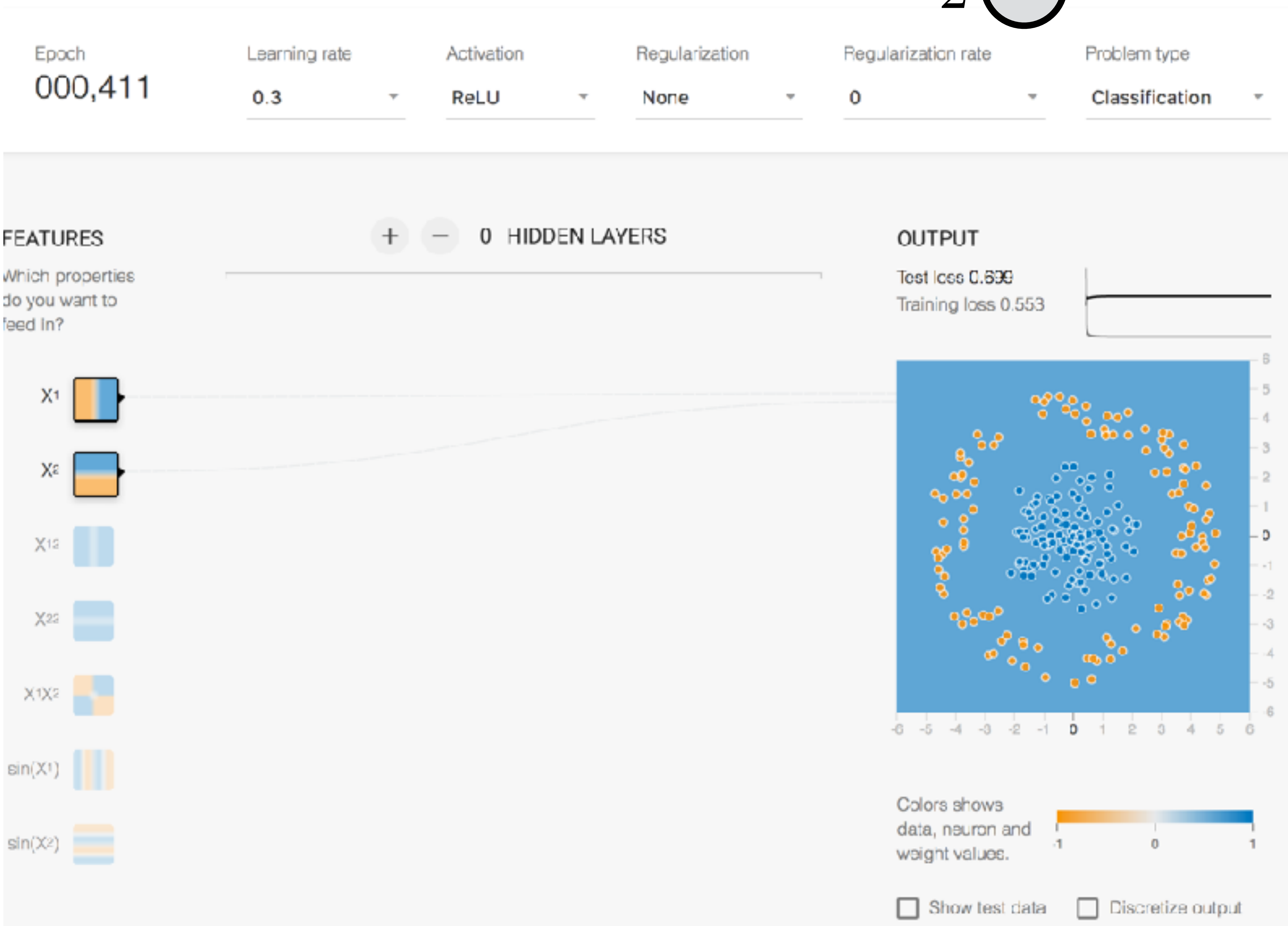
answer



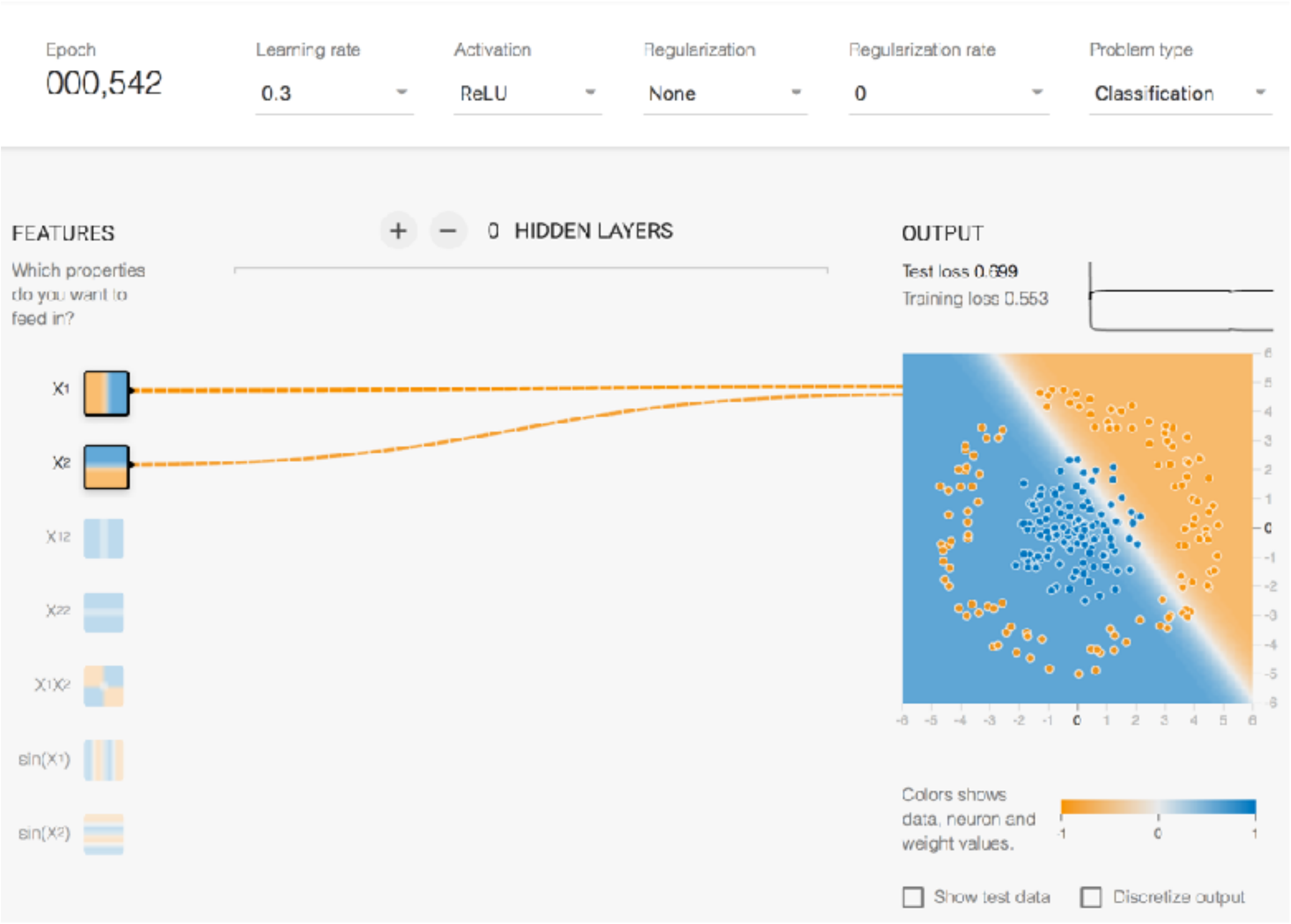
Epoch
000,108
Learning rate
0.03
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
0 HIDDEN LAYERS
OUTPUT
Test loss 0.001
Training loss 0.001
X1
X2
sin(X1)
sin(X2)
Colors shows data, neuron and weight values.

question


x_1
x_2
o

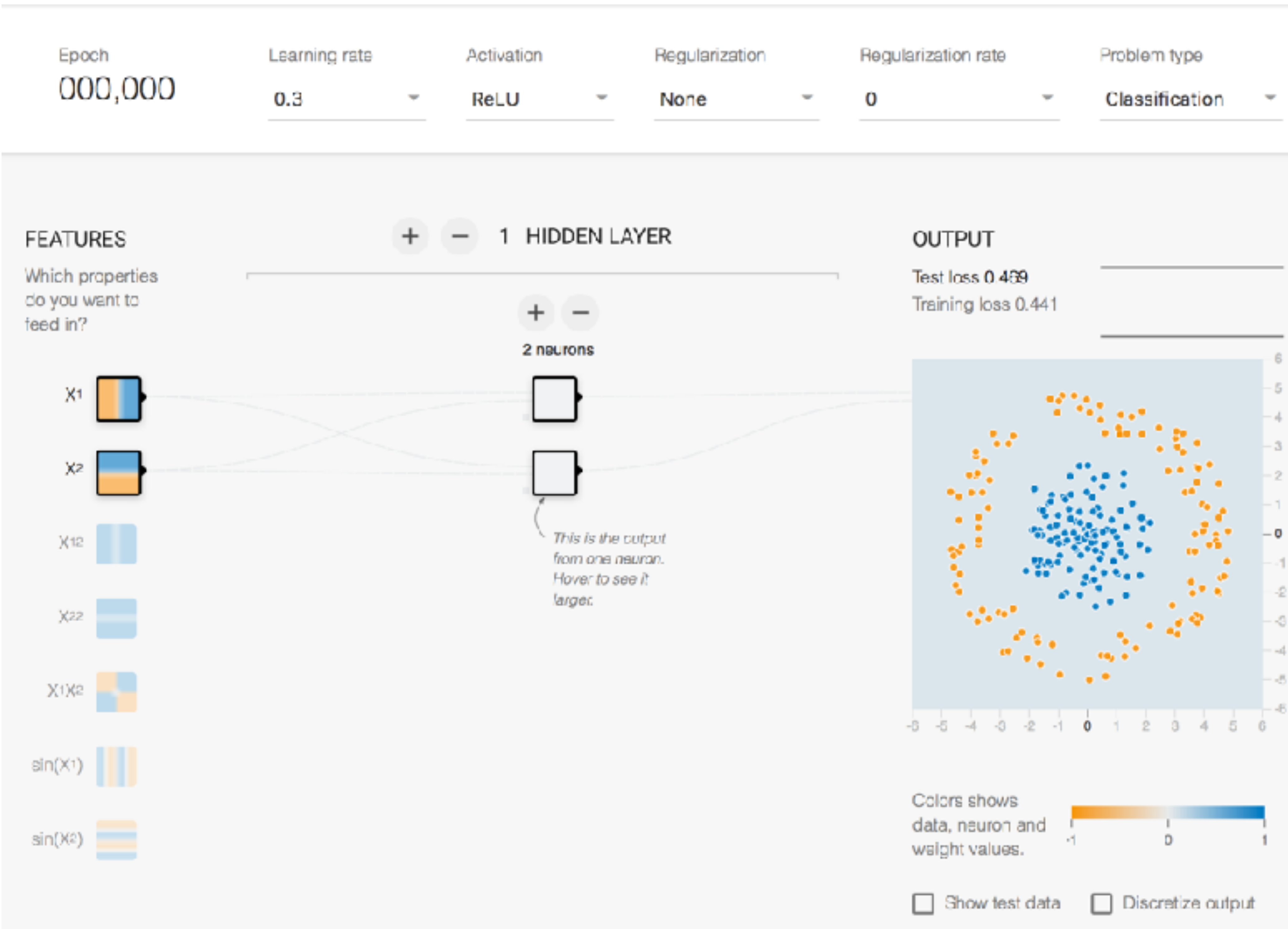

Epoch
000,411
Learning rate
0.3
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
0 HIDDEN LAYERS
OUTPUT
Training loss 0.553
Colors shows data, neuron and weight values.
Show test data
Discretize output

answer

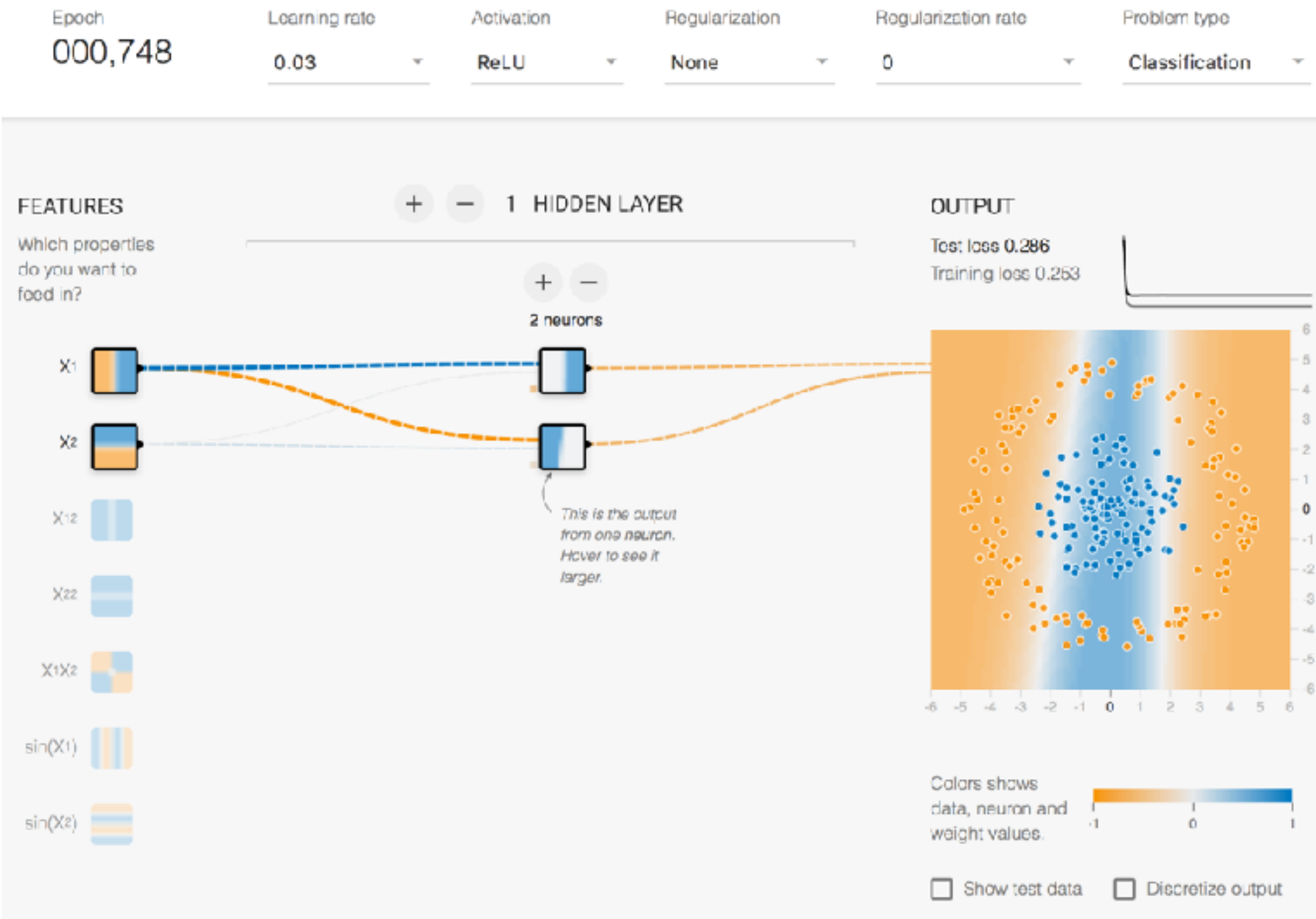


Epoch
000,542
Learning rate
0.3
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
0 HIDDEN LAYERS
OUTPUT
Test loss 0.699
Training loss 0.553
Colors shows data, neuron and weight values.
Show test data
Discretize output

question

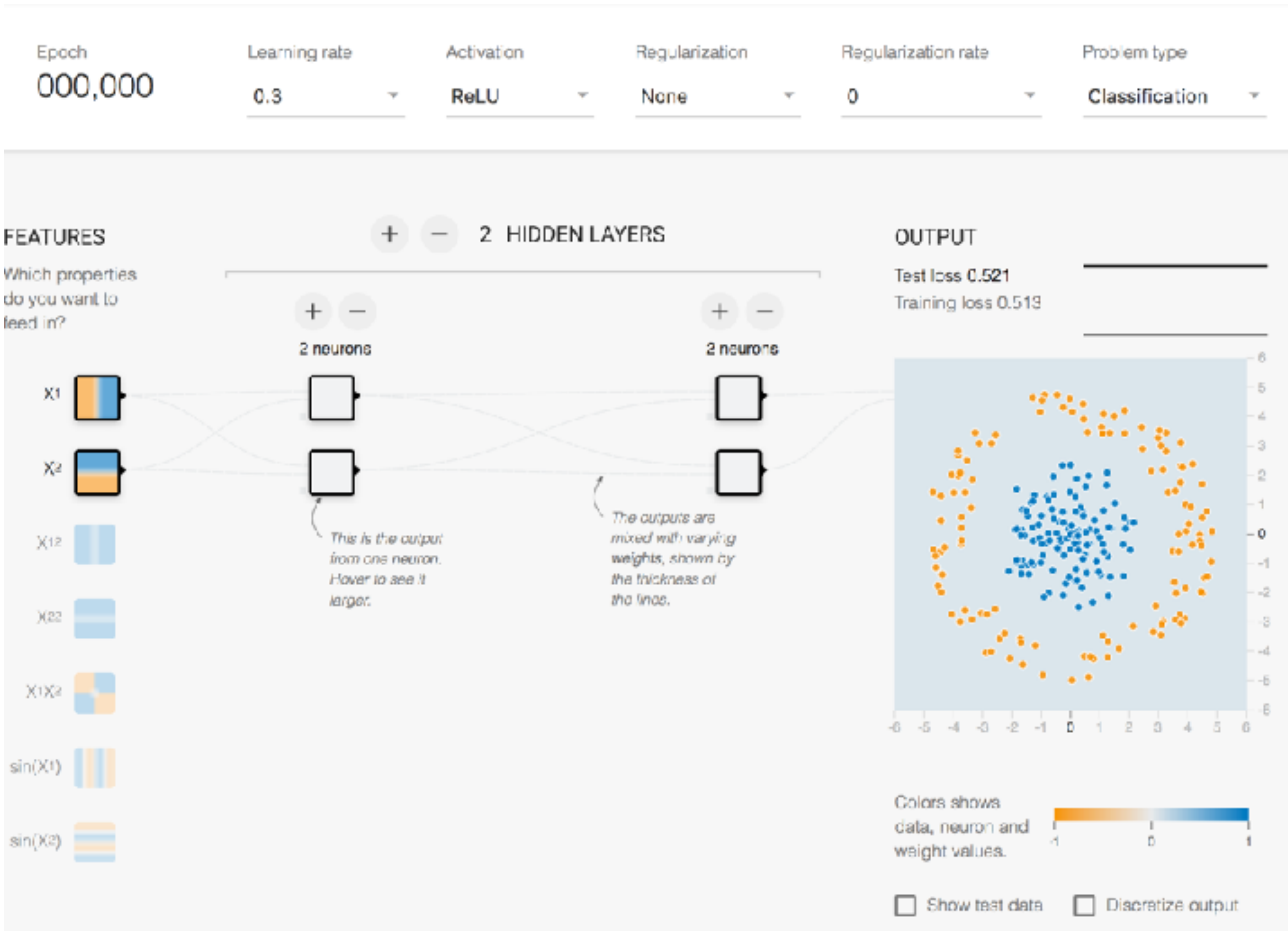

Epoch
000,000
Learning rate
0.3
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
2 neurons
This is the output from one neuron. Hover to see it larger.
OUTPUT
Test loss 0.459
Training loss 0.441
Colors shows data, neuron and weight values.
Show test data
Discretize output

answer

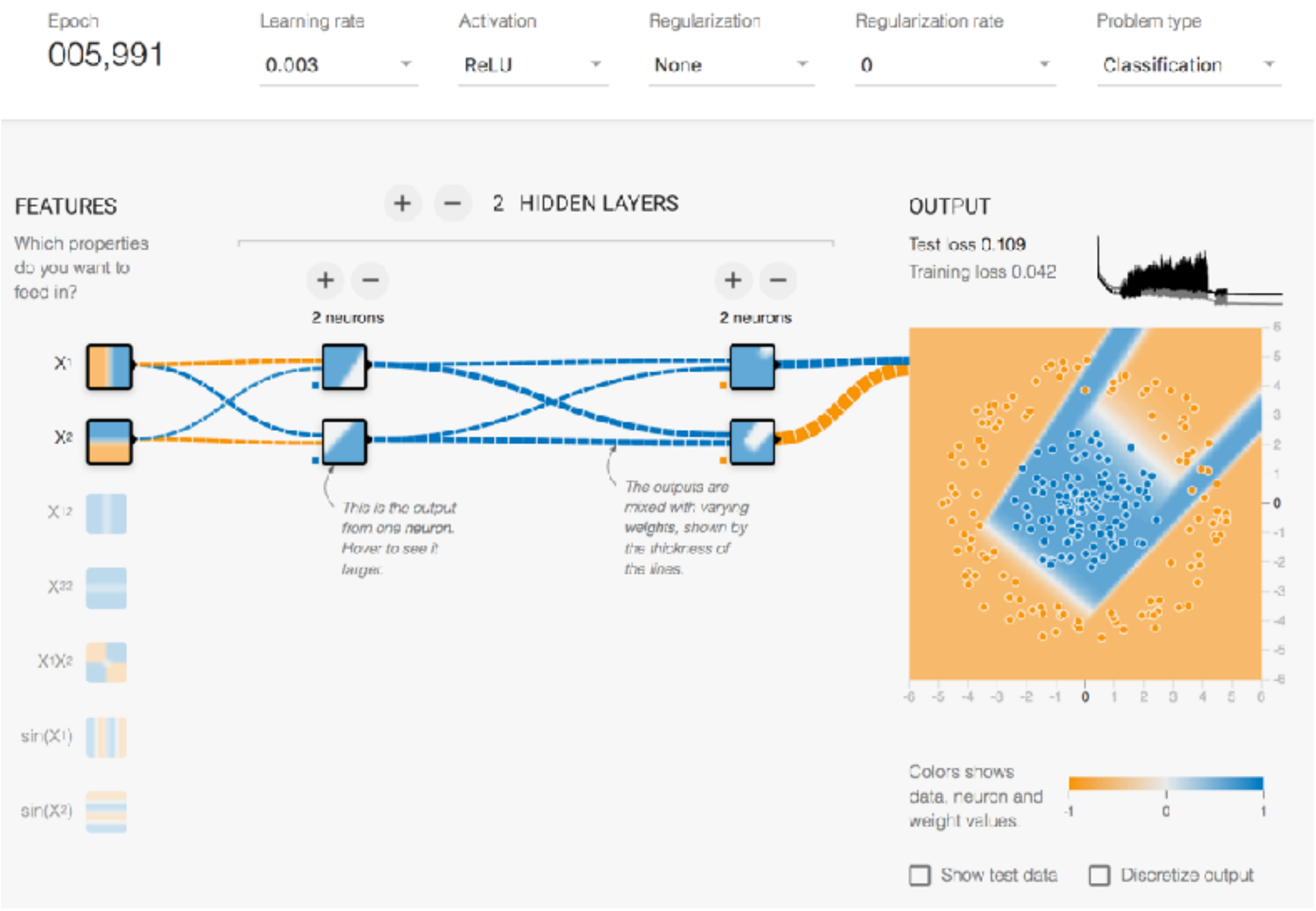


Epoch
000,748
Learning rate
0.03
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
2 neurons
This is the output from one neuron. Hover to see it larger.
OUTPUT
Test loss 0.286
Training loss 0.253
Colors shows data, neuron and weight values.
Show test data
Discretize output

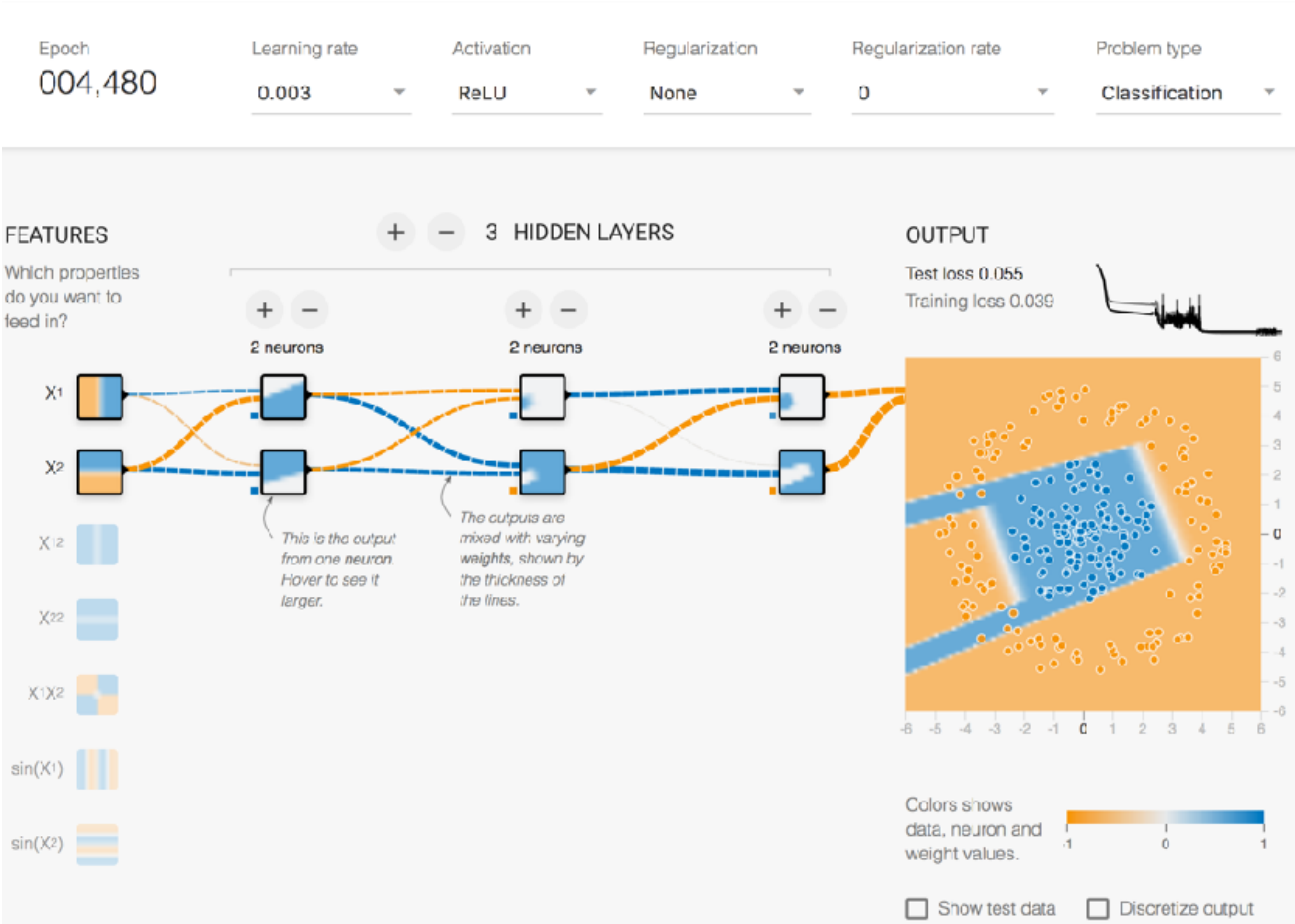
question


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

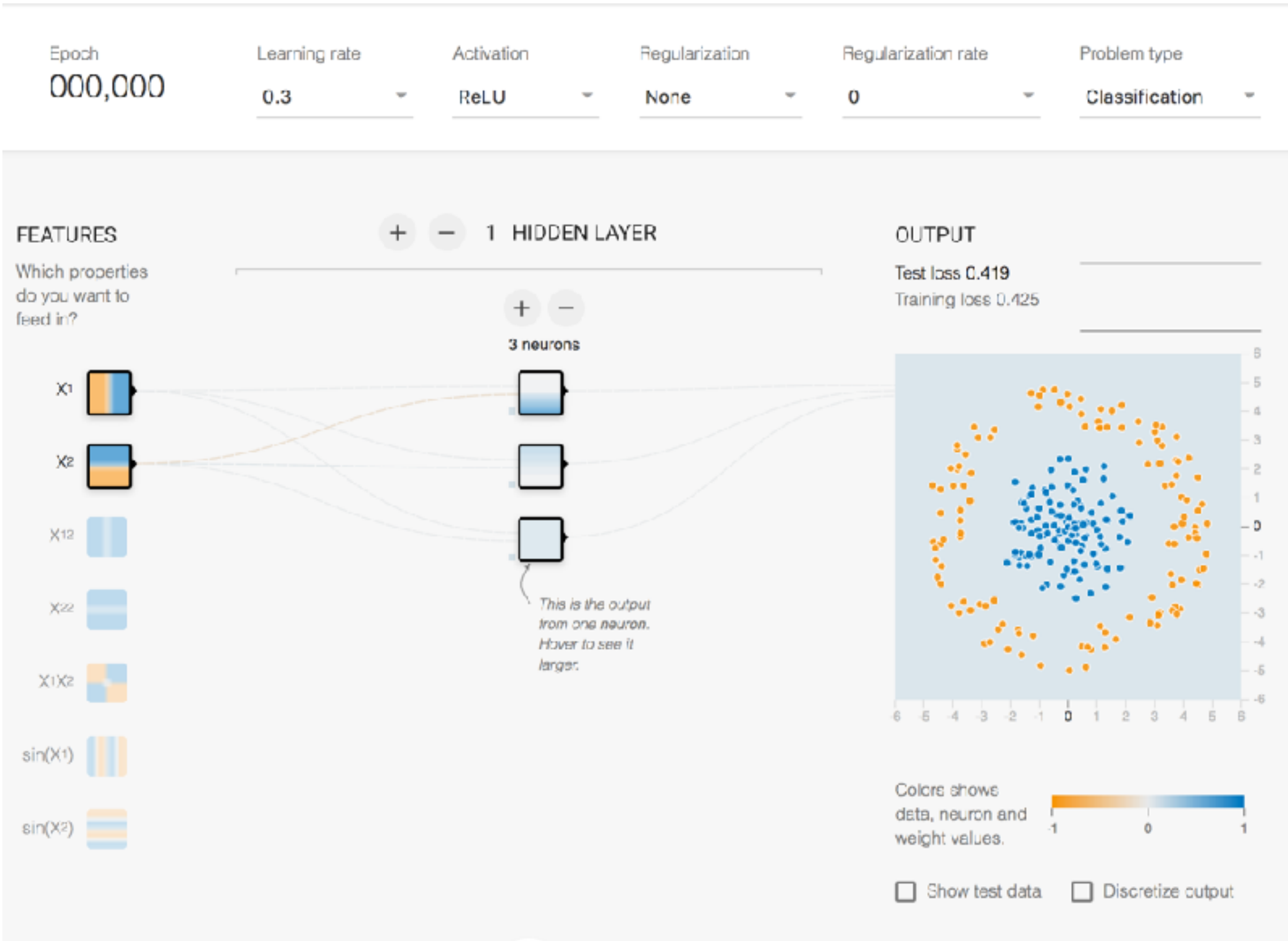

Epoch
005,991
Learning rate
0.003
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
2 HIDDEN LAYERS
2 neurons
2 neurons
This is the output from one neuron. Hover to see it larger.
The outputs are mixed with varying 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

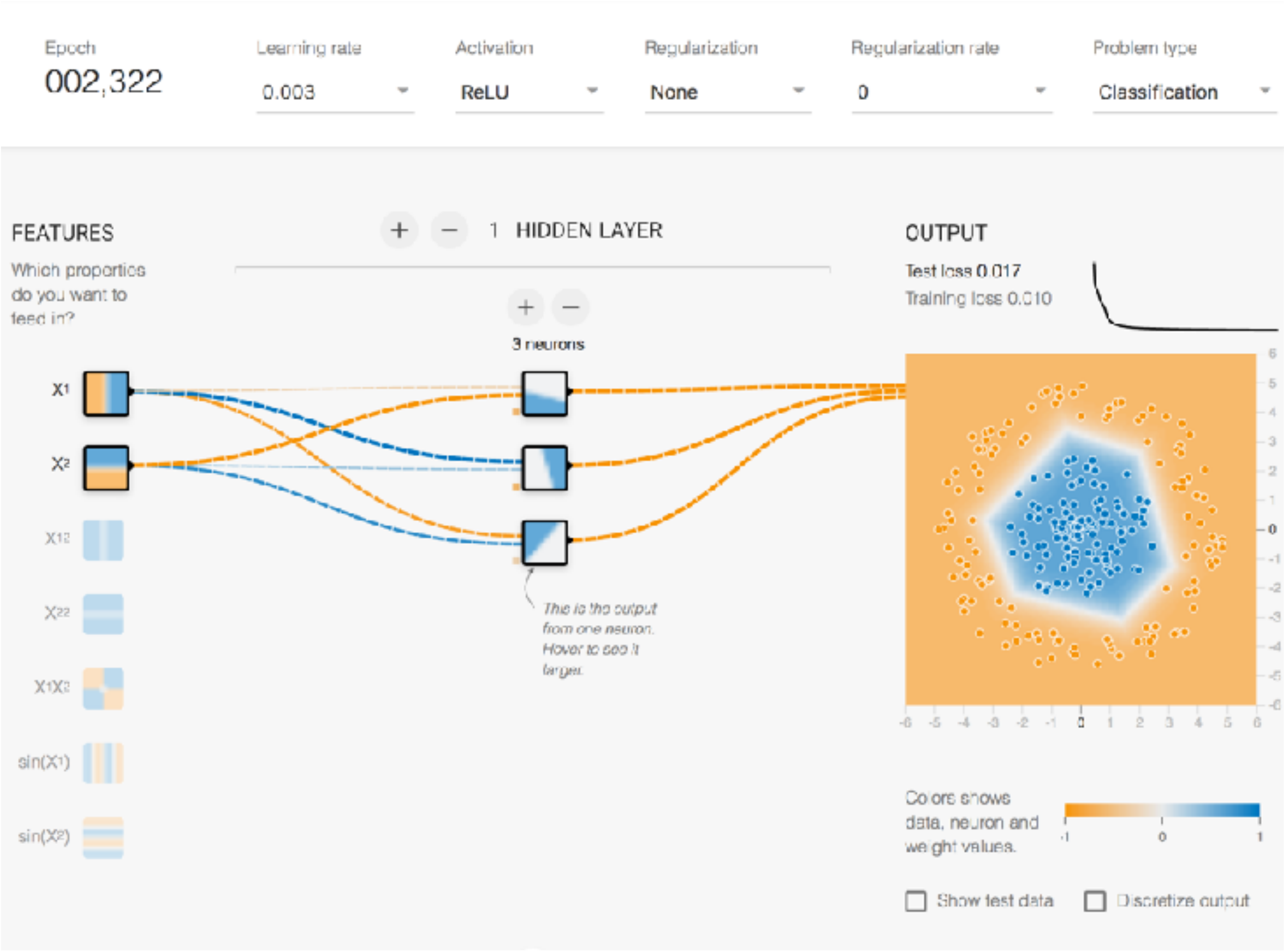


Epoch
004,480
Learning rate
0.003
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
3 HIDDEN LAYERS
2 neurons
2 neurons
2 neurons
This is the output from one neuron. Hover to see it larger.
The outputs are mixed with 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

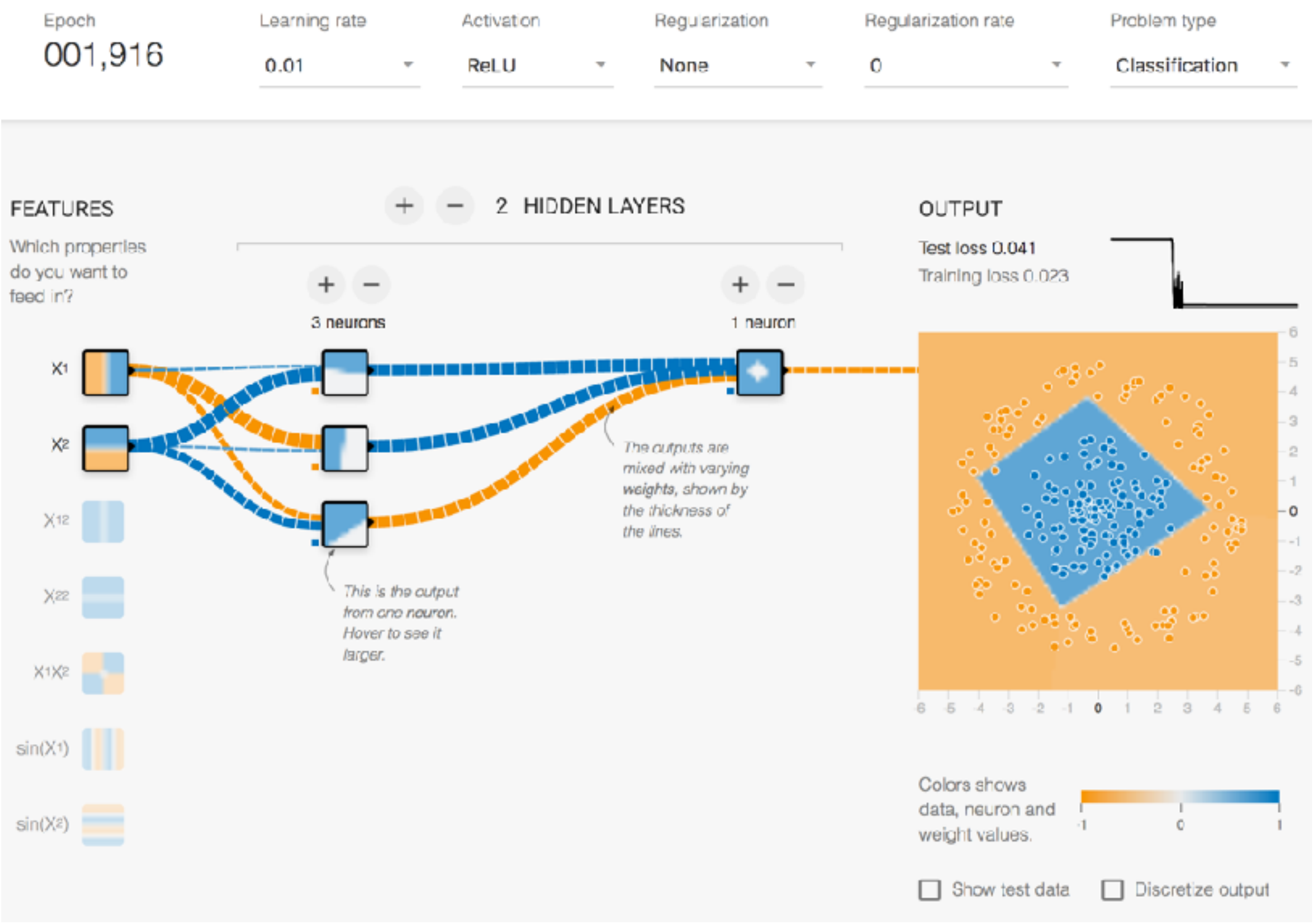

Epoch
000,000
Learning rate
0.3
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
3 neurons
This is the output from one neuron. Hover to see it larger.
OUTPUT
Test loss 0.419
Training loss 0.425
Colors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

answer



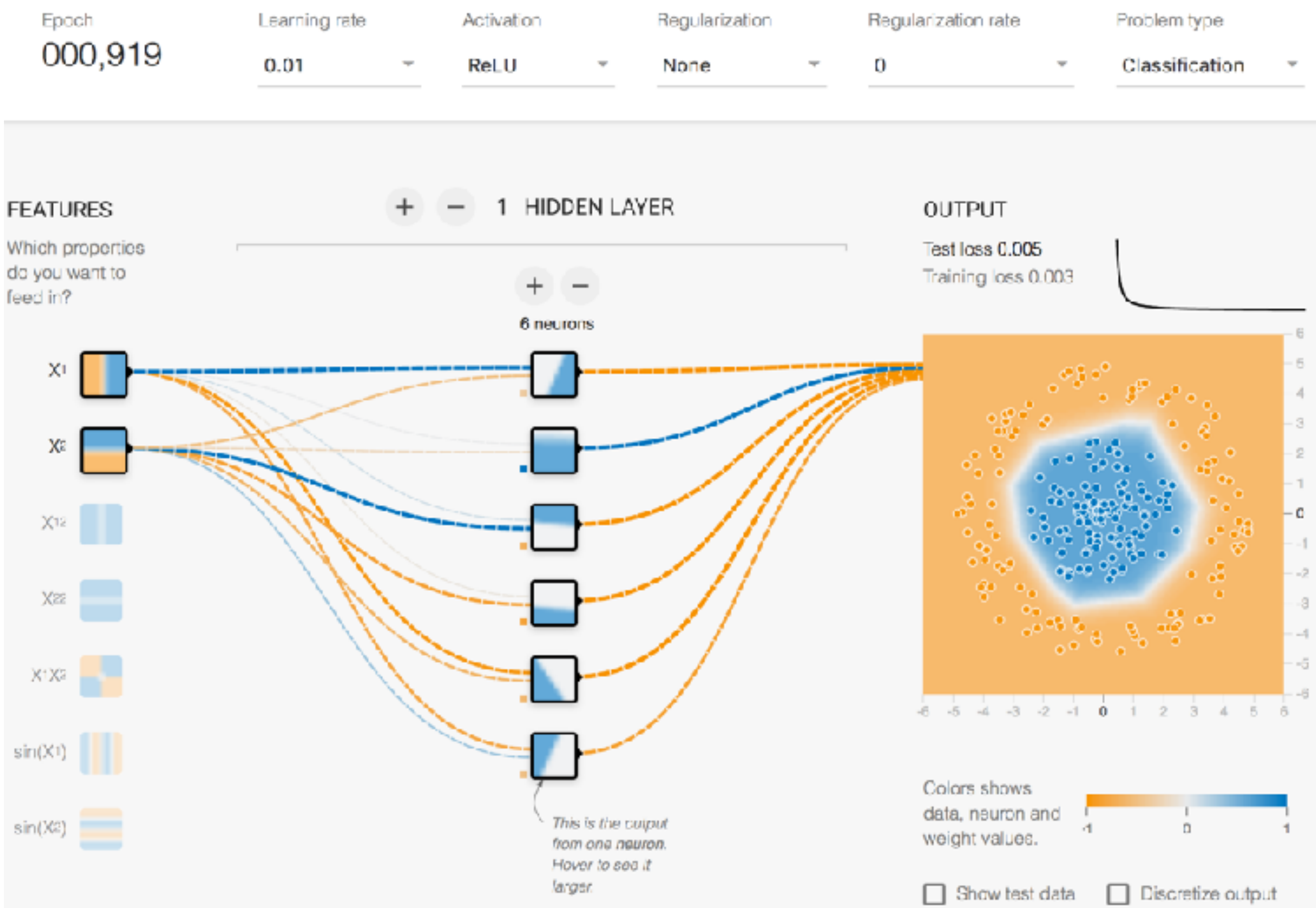
Epoch
002,322
Learning rate
0.003
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
3 neurons
This is the output from one neuron. Hover to see it larger.
OUTPUT
Test loss 0.017
Training loss 0.010
Colors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

answer



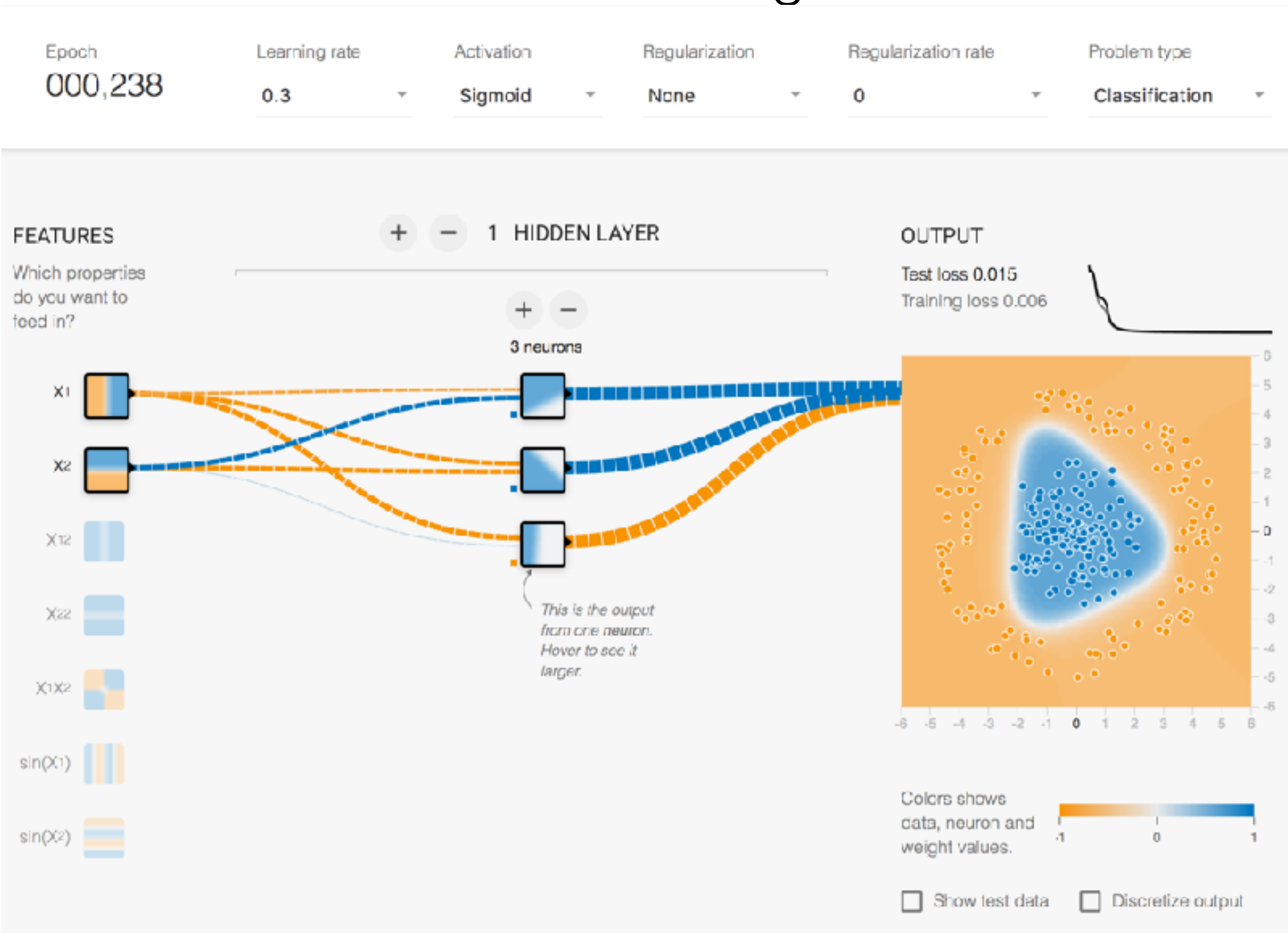
Epoch
001,916
Learning rate
0.01
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
2 HIDDEN LAYERS
3 neurons
1 neuron
OUTPUT
Test loss 0.041
Training loss 0.023
The outputs are mixed with varying weights, shown by the thickness of the lines.
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

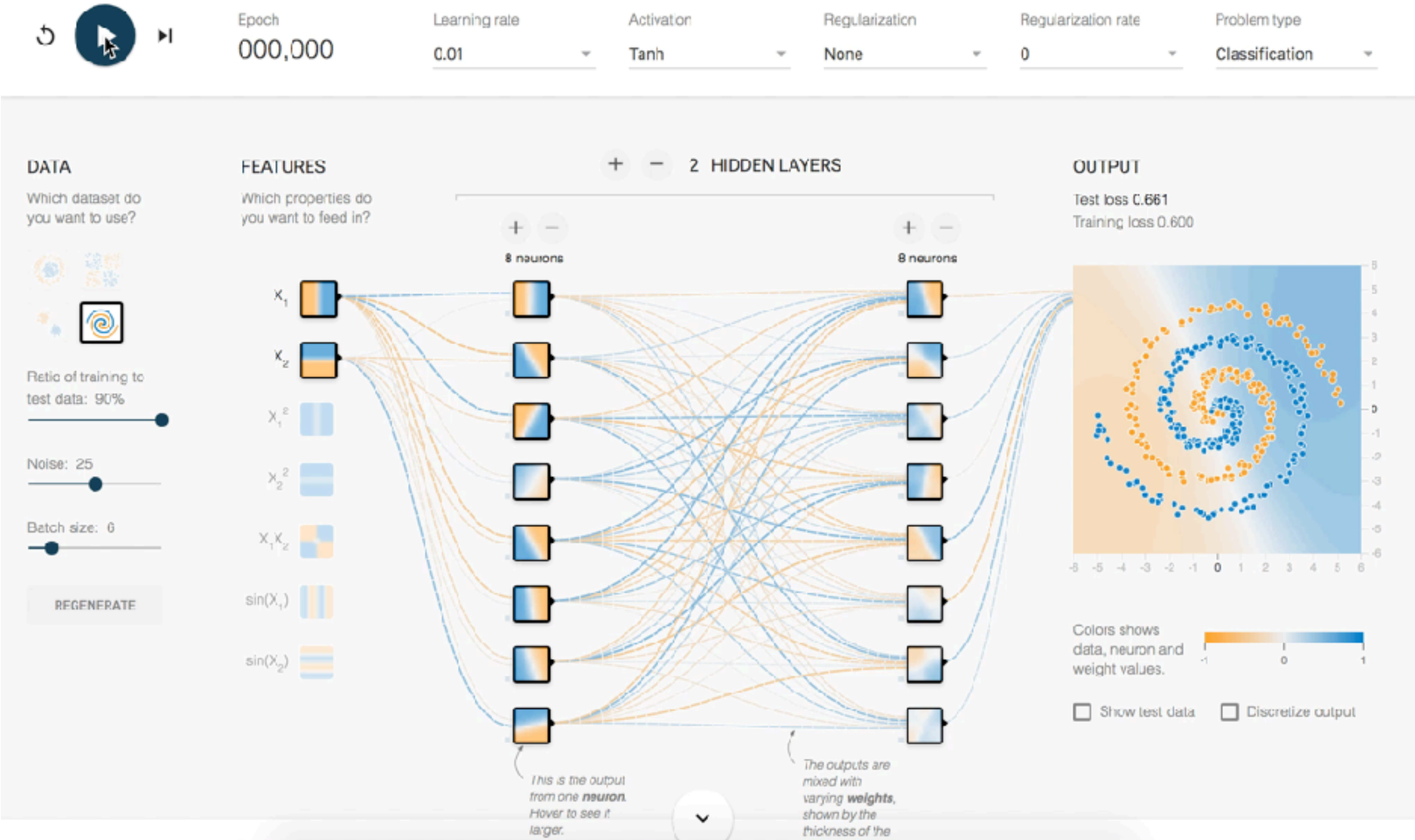


Epoch
000,919
Learning rate
0.01
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
6 neurons
OUTPUT
Test loss 0.005
Training loss 0.003
This is the output from one neuron. Hover to see it larger.
Colors 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
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
OUTPUT
Test loss 0.661
Training loss 0.600
Colors shows data, neuron and weight values.
Show test data
Discretize output

Administrative Issues

Communications matters

Use **IVLE forum** for communications. This is the fastest way to get response from us.

Only for private matters that are not convenient to announce in the IVLE forum, send us email directly

cs5242@comp.nus.edu.sg

Recipients of this email are:

Lee Hwee Kuan
Wang Wei
Connie Kou Khor Li
Ji Xin
Ouyang Kun

General grading and expectations

F/E There are those who do not know what they are doing. Their results are unreasonable

D/C There are those who know how to get some good results but cannot explain them.

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

A There are those who know enough to combine different methods and be creative in using Deep Learning

There are those who break the frontiers of Deep Learning research.

Assignments (10% * 4 = 40%)

- For each assignment
  - It has multiple sub-tasks, including coding tasks
  - You need to submit the answers, code and execution results

Quiz (20%)

- Closed-book
- 1 hour (or half-hour, TBD)
- To be held at week 9

Final Projects (30%)

- Projects will be held on Kaggle-in-class
  - Register using your nus email (https://inclass.kaggle.com/)
  - The project webpages will be announced once groups are finalized
  - No data disclosure
  - No additional data
  - Each group <=2 students
  - Submission deadline: Week 11. 04-Nov-2017 06:00 PM, Singapore Time
  - Report deadline: 11-Nov-2017 06:00 PM, Singapore Time
- Each group will be randomly assigned to
  - A computer vision project
  - A natural language processing project
- Assessment is based on the report (10%) and ranking (20%)
  - Top ranked groups of each project will be invited to do presentation at week 13.
- First task
  - Find a partner before **27 Aug 18:00** and register your group information on IVLE(Class & Groups -> Class Groups).

Final projects timeline summary

Date	Event
27 August 18:00	Register your project group @ IVLE 2 or less students per group
04 November 18:00	Results submission deadline
05 November 18:00	Selection and notification of selected projects for presentation
11 November 18:00	Report deadline
12 November 18:00	Selected project group send slides to cs5242@comp.nus.edu.sg
12 November 18:00	Make appointment to meet Hwee Kuan to do presentation rehearsal
16 November 18:30	Project presentations by students

Final Project (40%)

- Report template
  - Problem definition
  - Highlights (new algorithms, insights from the experiments)
  - Dataset pre-processing description
  - Training and testing procedure
  - Experimental study
  - (clarity, model understanding, and highlights are important for the assessment)
- Ranking -> score
  - The ranges are to be determined based on total number of groups
  - Rank top 10%, score 25 (max score)
  - Rank top 20%, score 21
  - Rank top 40%, score 17
  - Rank top 70%, score 14
  - Rank others, score 13
  - Prediction scores less than 10% better than random guess, score 5

Project details will be announced next week.

Start your project early

Do your project early

Due date: week 11, 04 Nov 6pm

Submit first draft on week 9

Enhance your submission with better results on week 10

Finally submit best results on week 11

Do not miss the deadline

email us early if you face any emergency situations

cs5242@comp.nus.edu.sg

GPU Machine/Cluster

- NSCC (National Super-Computing Center)
  - Free for NUS students
  - 128 GPU nodes, each with a NVIDIA Tesla K40 (12 GB)
  - Follow the manual and test program on IVLE workbin/misc/gpu to get yourself familiar with it.
- Google Cloud Platform
  - 300 USD credit for new register
  - NVIDIA Tesla K80 (2x12GB) is available for Region Taiwan
- Amazon EC2 (g2.8xlarge)
  - 100 USD credit for students
  - NVIDIA Grid GPU, 1536 CUDA cores, 4GB memory
  - Available for Singapore region
- Tips:
  - STOP/TERMINATE the instance immediately after your program terminates
  - Check the usage status frequently to avoid high outstanding bills
  - Use Amazon or Google cloud platform for debugging and use NSCC for actual training and tuning

Lecture 1: Introduction
Lecture 2: Neural Network Basics
Lecture 3: Training the Neural Network
Lecture 4: Convolutional Neural Network
Lecture 5: Convolutional Neural Network
Lecture 6: Convolutional Neural Network
Lecture 7: Recurrent Neural Network
Lecture 8: Recurrent Neural Network
Lecture 9: Recurrent Neural Network
Lecture 10: Advanced Topic: Generative Adversarial Network
Lecture 11: Advanced Topic: One-shot learning
Lecture 12: Advanced Topic: Distributed training
Lecture 13: Project demonstrations