Learn with
Google AI | Explore ML

Neural Networks

Explore ML

Date : 4th September,2019 | Event : Google AI, Explore ML | Speaker : Ayon Roy

Learn with Google AI | Explore ML

Hello **BPITians!**

I am **Ayon Roy**

B.Tech CSE (2017-2021)

Ex-Summer Intern at MateLabs, Bengaluru (**World's First Horizontal AI Startup**)

Email : ayon.roy2000@gmail.com

Telegram / Github / LinkedIn Username : ayonroy2000

Website : https://AYONROY.ML/

If you haven't heard about me yet, you might have been living under the rocks. Wake up !!

Agenda (4-09-2019)

- Introduction to Structure and operation of neural network
- Introduction to Tensorflow Playground
- Activity: TensorFlow Playground Neural Network Exercises
- Activity: Teachable Machine and the Limits of Neural Networks

Learn with Google AI | Explore ML

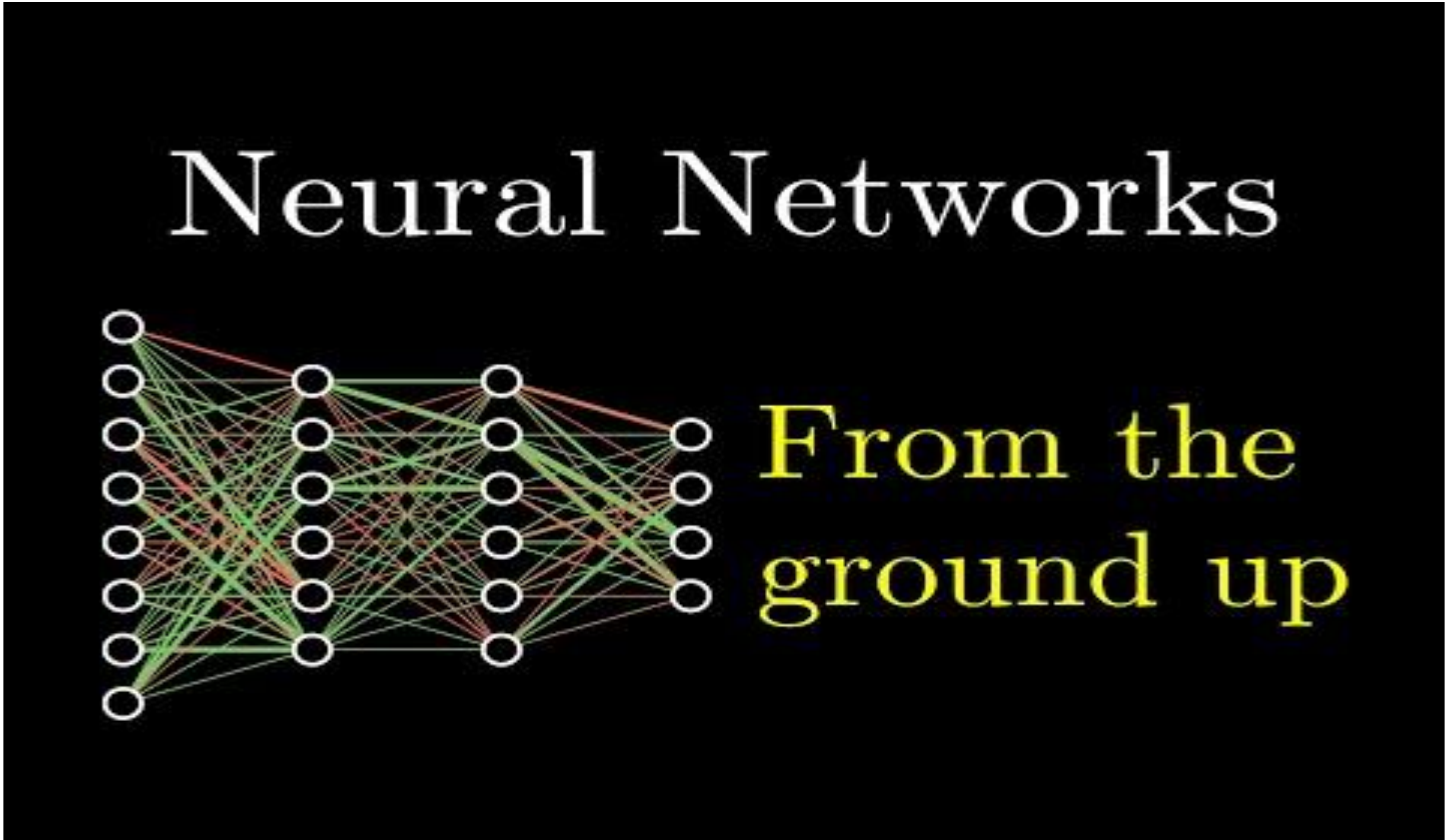
Neural Networks

Let's Play

g.co/teachablemachine

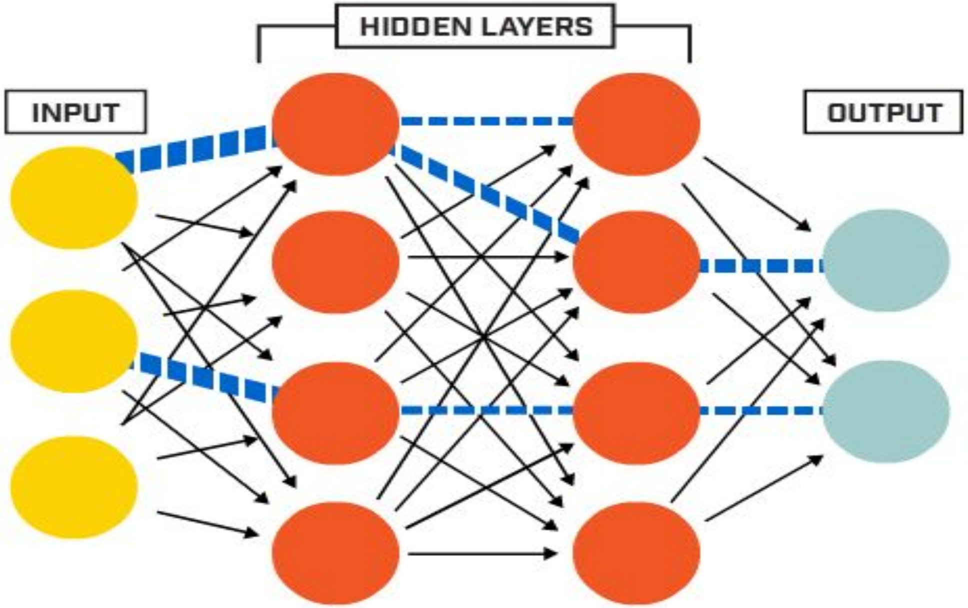
If privacy is not an issue,
please use your laptop's webcam for this activity.

- See a Neural Network Demo at http://bit.do/BPIT1


Neural Networks
From the ground up

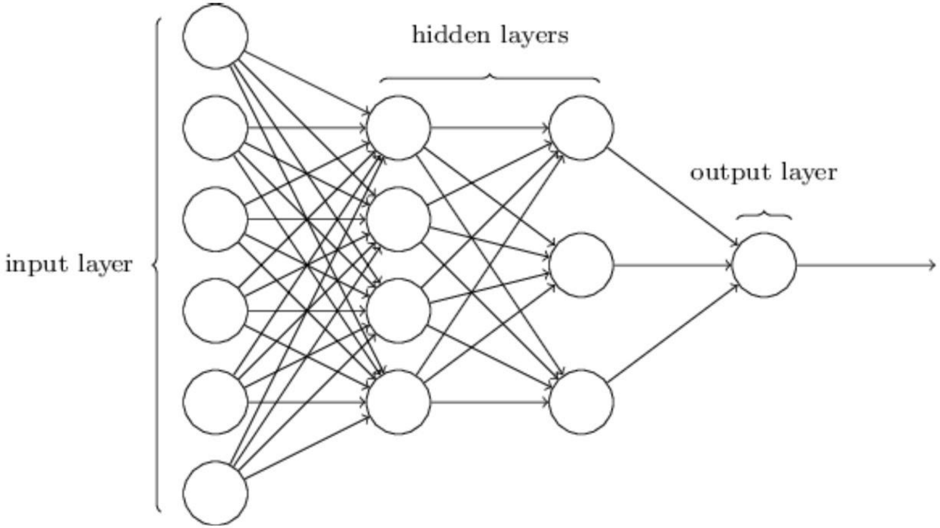
Learn with Google AI | Explore ML

What are Neural Networks ?



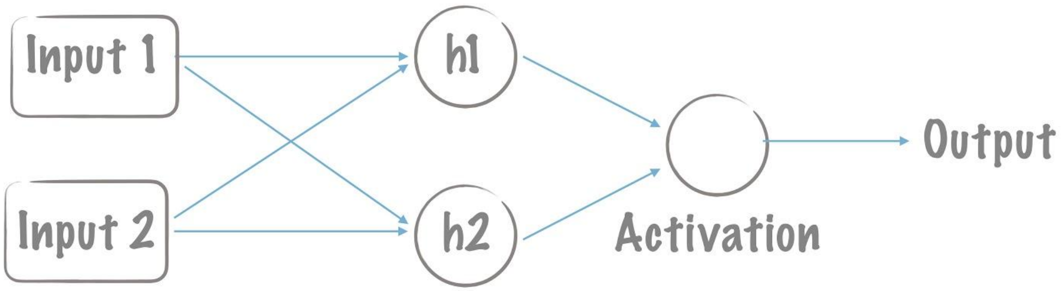
Learn with Google AI | Explore ML

Neural Networks



Its like Perceptrons are together in a defined fashion

Two Layered Neural Network

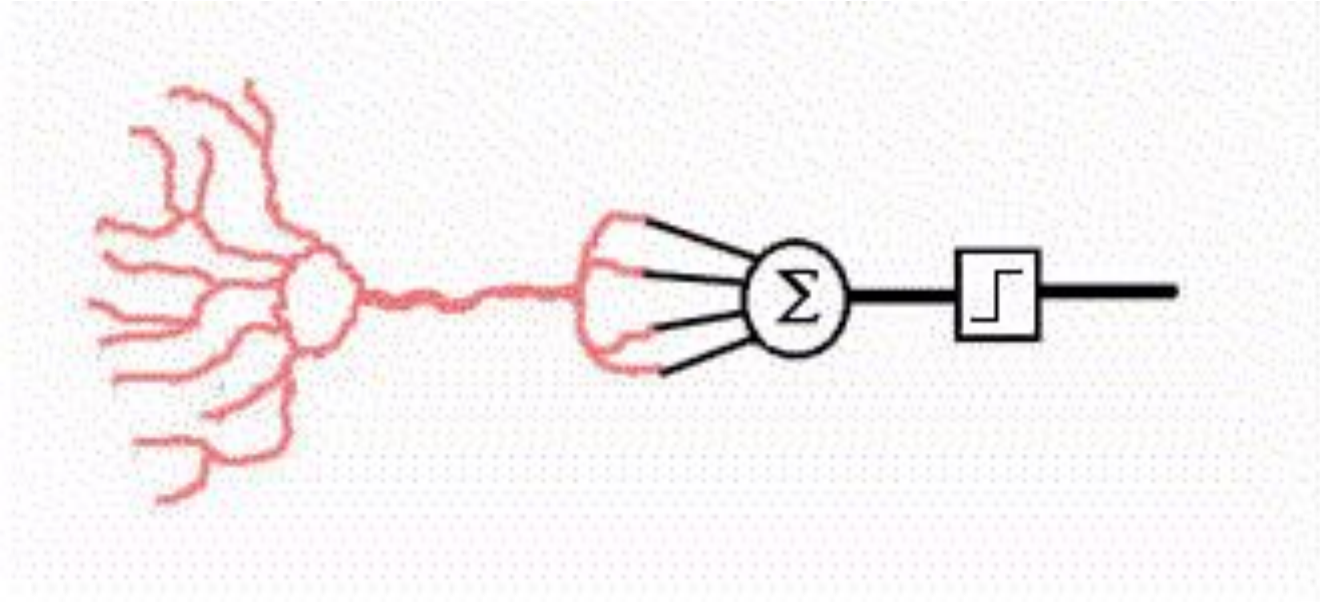


These array of perceptrons form the Neural Nets

Learn with Google AI | Explore ML

Computer System inspired by biological networks of neurons that learn progressively i.e which improves performance to do tasks; by considering examples generally without task specific programming.

Why Neural Networks ?



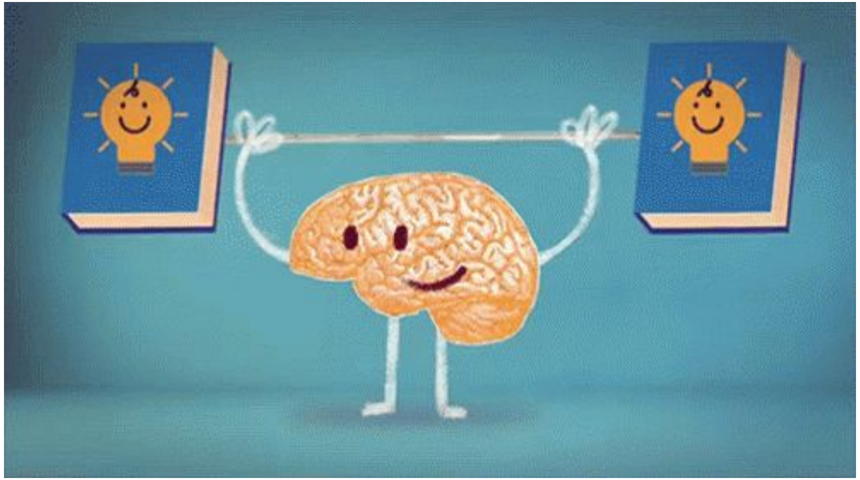
Problem before Neural Networks were introduced

Computers used to follow a set of instructions to solve a problem

But what now ?

Neural Networks learn by example. So now computers can do things what we don’t exactly know how to do.

Motivation behind Neural Networks ?



Application of Neural Networks

Is it only here ?

It's here too

Speech Recognition

- Multilayer networks with recurrent connections
- Kohonen self-organizing feature map

Character Recognition

- Backpropagation neural networks.
- Neocognitron

Signature Verification, Face Recognition etc.

But how it works ?

Single Layer & Multilayer Perceptron

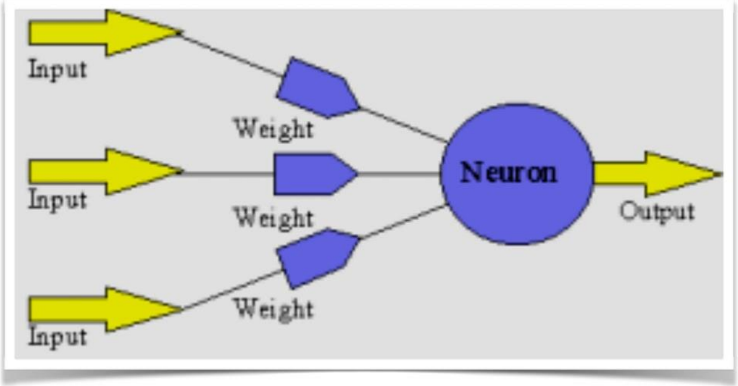
Perceptron

The elementary entity of a neural network.

The most basic form of neural network which can also learn

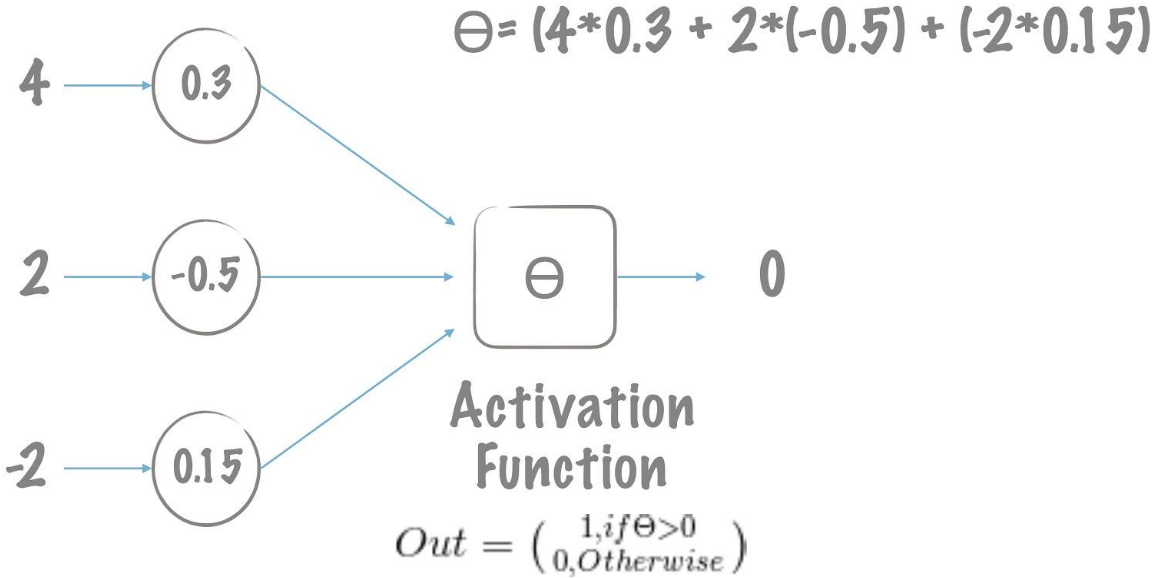
Learn with Google AI | Explore ML

Perceptron

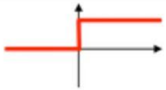
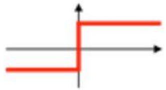
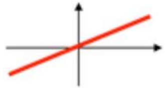
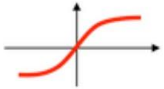


A Basic Linear discriminant Classifier

How Perceptron Work



Activation Functions

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer NN	

Training a Perceptron

Let's train a Perceptron to mimic this pattern

Training Set

x	y	z
0	0	1
1	1	1
1	0	1
0	1	1

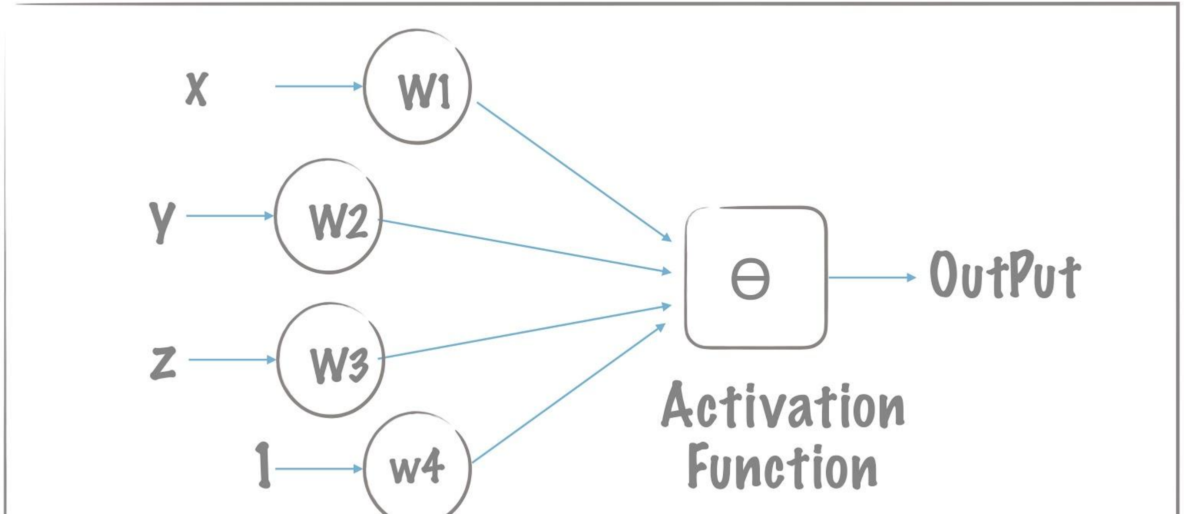
T(out)

out
0
1
1
0

Learn with Google AI | Explore ML

Training a Perceptron

Perceptron Model



$$Out = \begin{pmatrix} 1, if\ \Theta > 0 \\ 0, Otherwise \end{pmatrix}$$

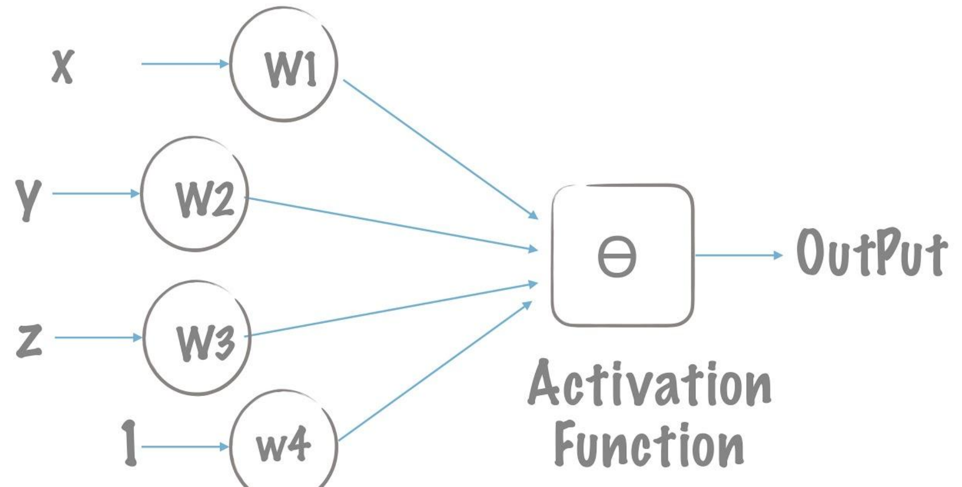
Learn with Google AI | Explore ML

Training a Perceptron

Training Rules : $Wi = Wi + \Delta Wi$

$\Delta W = -\eta * (\text{target output} - \text{perceptron output}) * X$

η is learning rate for perceptron



$$Out = \left({1, if\,\Theta > 0 \atop 0, Otherwise} \right)$$

Learn with Google AI | Explore ML

Training a Perceptron

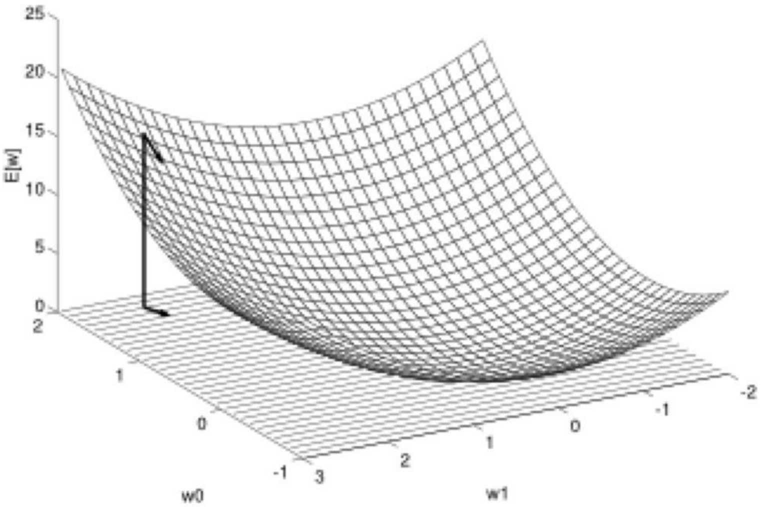
Let's learn w_i such that it minimizes the squared error

$$E[\vec{w}] \equiv \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

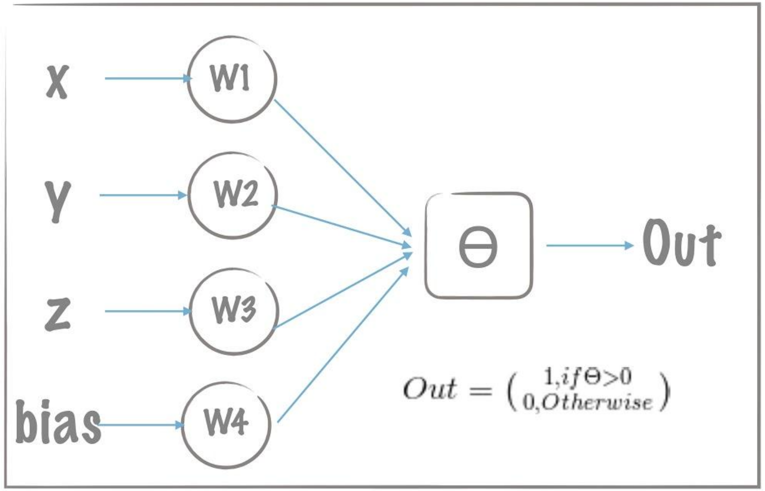
$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

On simplifying

$$\Delta w_i = \eta \sum_{d \in D} (t_d - o_d) x_{id}$$



Learn with Google AI | Explore ML



$W_i = W_i + \Delta W_i$

$\Delta W_i = -\eta * [P(out) - T(Out)] * x_i$

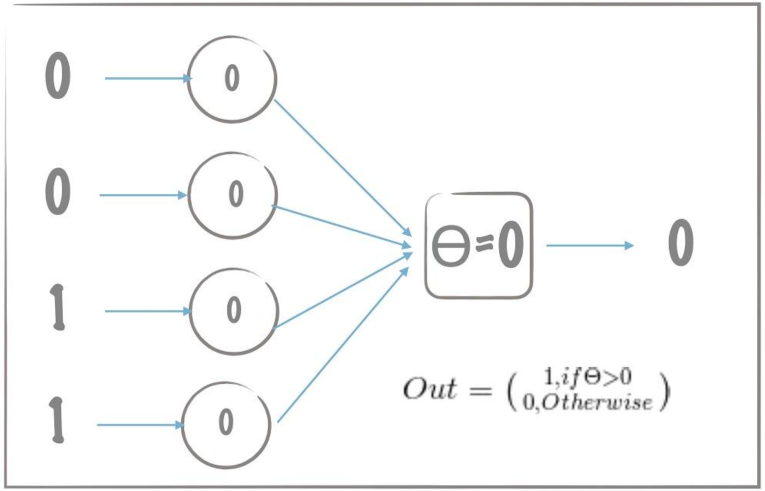
η is learning rate for perceptron

Training Set T(out)

	x	y	z	bias	out
epoch 1	0	0	1	1	0
	1	1	1	1	1
	1	0	1	1	1
	0	1	1	1	0
epoch 2	0	0	1	1	0
	1	1	1	1	1
	1	0	1	1	1
	0	1	1	1	0

Assign random values to Weight Vector([W1,W2,W3,W4])

Learn with Google AI | Explore ML



$W_i = W_i + \Delta W_i$

$\Delta W_i = -\eta * [P(out) - T(Out)] * x_i$

η is learning rate for perceptron

Training Set

	x	y	z	bias
	0	0	1	1
epoch 1	1	1	1	1
epoch 1	1	0	1	1
epoch 1	0	1	1	1
epoch 2	0	0	1	1
epoch 2	1	1	1	1
epoch 2	1	0	1	1
epoch 2	0	1	1	1

T(out)

out
0
1
1
0
0
1
1
0

Weights

w1	w2	w3	w4
0	0	0	0

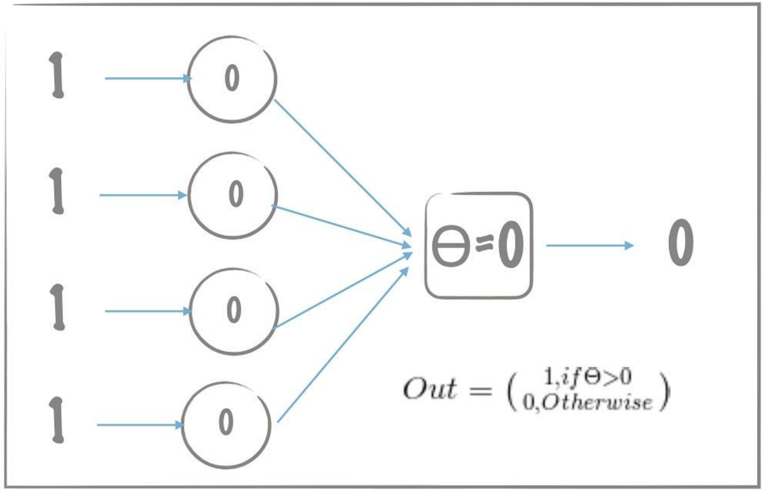
P(out)

out
0

ΔWeights

Δw1	Δw2	Δw3	Δw4
0	0	0	0

Learn with Google AI | Explore ML



$W_i = W_i + \Delta W_i$

$\Delta W_i = -\eta * [P(out) - T(Out)] * x_i$

η is learning rate for perceptron

Training Set

	x	y	z	bias
epoch 1	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1
epoch 2	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1

T(out)

out
0
1
1
0
0
1
1
0

Weights

w1	w2	w3	w4
0	0	0	0
0	0	0	0

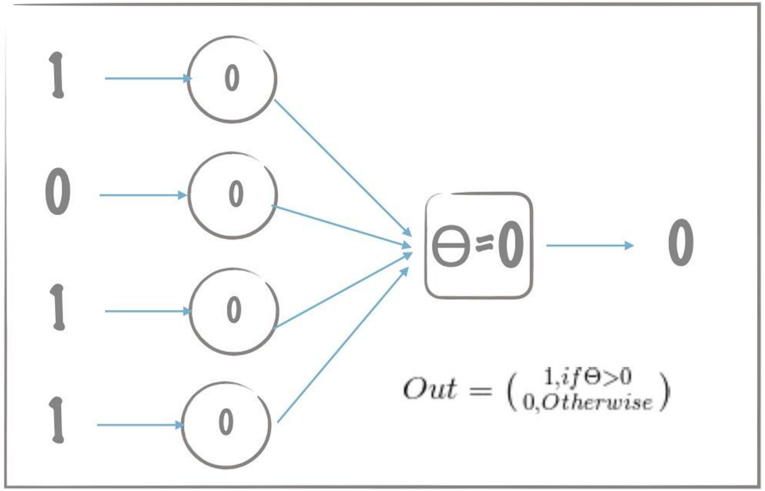
P(out)

out
0
0

ΔWeights

Δw1	Δw2	Δw3	Δw4
0	0	0	0
1	1	1	1

Learn with Google AI | Explore ML



$W_i = W_i + \Delta W_i$

$\Delta W_i = -\eta * [P(out) - T(Out)] * x_i$

η is learning rate for perceptron

Training Set

	x	y	z	bias
epoch 1	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1
epoch 2	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1

T(out)

out
0
1
1
0
0
1
1
0

Weights

w1	w2	w3	w4
0	0	0	0
0	0	0	0
1	1	1	1

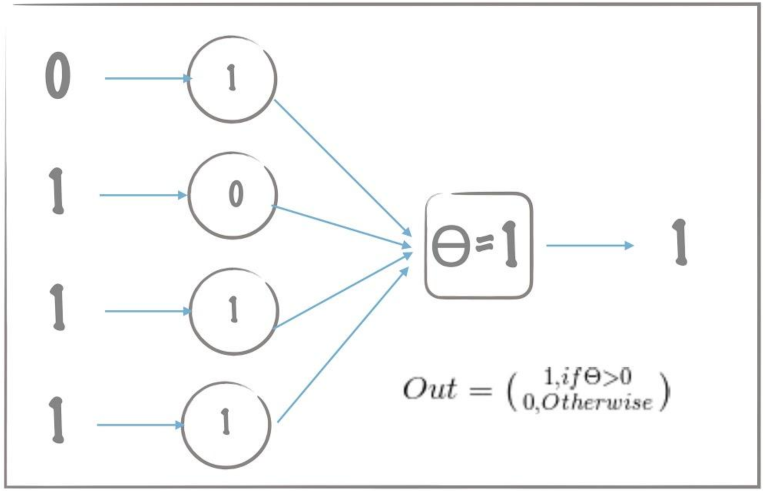
P(out)

out
0
0
1

ΔWeights

Δw1	Δw2	Δw3	Δw4
0	0	0	0
1	1	1	1
0	0	0	0

Learn with Google AI | Explore ML



Wi = Wi + ΔWi

ΔWi = -η * [P(out) -T(Out)] * xi

η is learning rate for perceptron

Training Set

	x	y	z	bias
epoch 1	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1
epoch 2	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1

T(out)

out
0
1
1
0
0
1
1
0

Weights

w1	w2	w3	w4
0	0	0	0
0	0	0	0
1	1	1	1
1	1	1	1

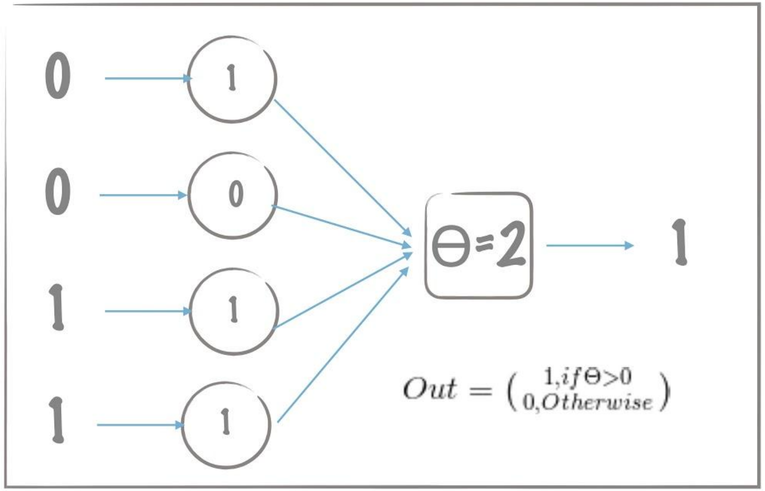
P(out)

out
0
0
1
1

ΔWeights

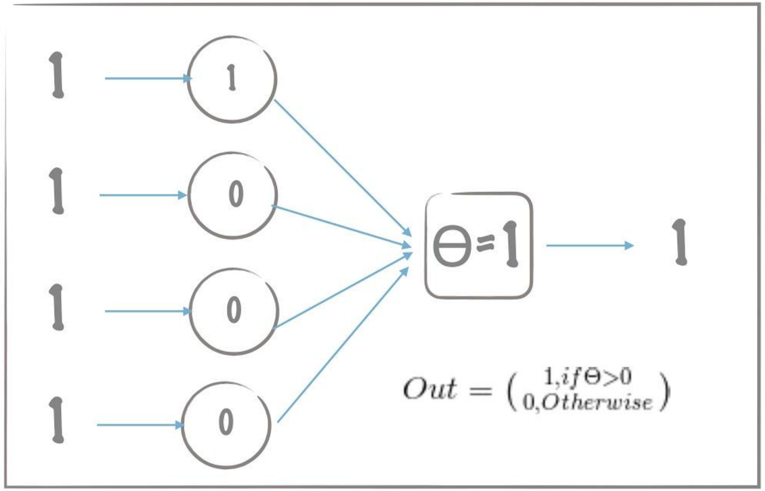
Δw1	Δw2	Δw3	Δw4
0	0	0	0
1	1	1	1
0	0	0	0
0	-1	-1	-1

Learn with Google AI | Explore ML



$Wi = Wi + \Delta Wi$
$\Delta Wi = -\eta * [P(out) - T(Out)] * xi$
η is learning rate for perceptron

	Training Set				T(out)	Weights				P(out)	ΔWeights			
	x	y	z	bias	out	w1	w2	w3	w4	out	Δw1	Δw2	Δw3	Δw4
epoch 1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	0	0	0	0	0	1	1	1	1
	1	0	1	1	1	1	1	1	1	1	0	0	0	0
	0	1	1	1	0	1	1	1	1	1	0	-1	-1	-1
epoch 2	0	0	1	1	0	1	0	0	0	0	0	0	0	0
	1	1	1	1	1									
	1	0	1	1	1									
	0	1	1	1	0									



Wi = Wi + ΔWi

ΔWi = -η * [P(out) -T(Out)] * xi

η is learning rate for perceptron

Training Set

	x	y	z	bias
epoch 1	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1
epoch 2	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1

T(out)

out
0
1
1
0
0
1
1
0

Weights

w1	w2	w3	w4
0	0	0	0
0	0	0	0
1	1	1	1
1	1	1	1
1	0	0	0
1	0	0	0

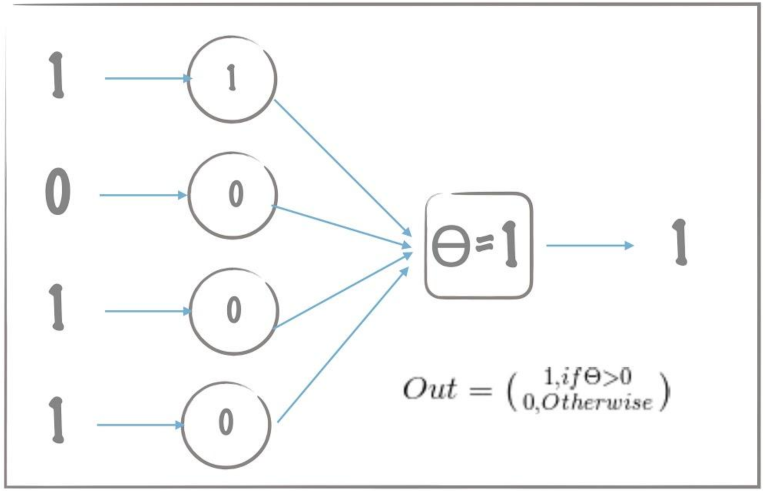
P(out)

out
0
0
1
1
1
1

ΔWeights

Δw1	Δw2	Δw3	Δw4
0	0	0	0
1	1	1	1
0	0	0	0
0	-1	-1	-1
0	0	0	0
0	0	0	0

Learn with Google AI | Explore ML



$W_i = W_i + \Delta W_i$

$\Delta W_i = -\eta * [P(out) - T(Out)] * x_i$

η is learning rate for perceptron

Training Set

	x	y	z	bias
epoch 1	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1
epoch 2	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1

T(out)

out
0
1
1
0
0
1
1
0

Weights

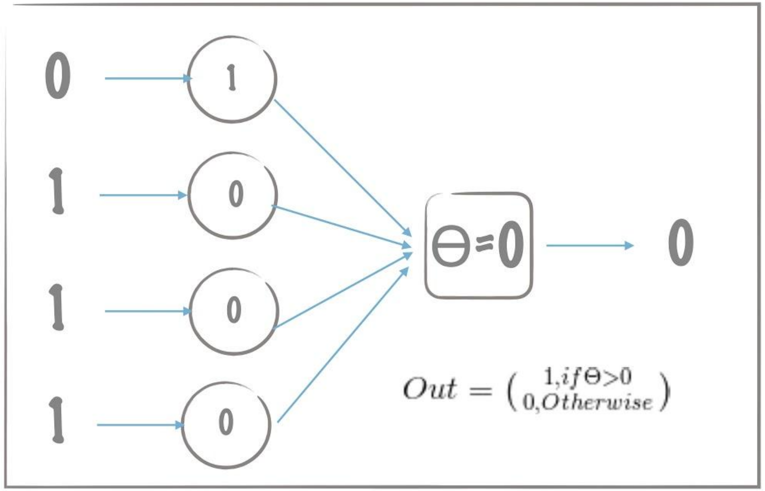
w1	w2	w3	w4
0	0	0	0
0	0	0	0
1	1	1	1
1	1	1	1
1	0	0	0
1	0	0	0
1	0	0	0

P(out)

out
0
0
1
1
1
1
1

ΔWeights

Δw1	Δw2	Δw3	Δw4
0	0	0	0
1	1	1	1
0	0	0	0
0	-1	-1	-1
0	0	0	0
0	0	0	0
0	0	0	0



Wi = Wi + ΔWi
ΔWi = -η * [P(out) -T(Out)] * xi
η is learning rate for perceptron

Training Set

	x	y	z	bias
epoch 1	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1
epoch 2	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1

T(out)

out
0
1
1
0
0
1
1
0

Weights

w1	w2	w3	w4
0	0	0	0
0	0	0	0
1	1	1	1
1	1	1	1
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0

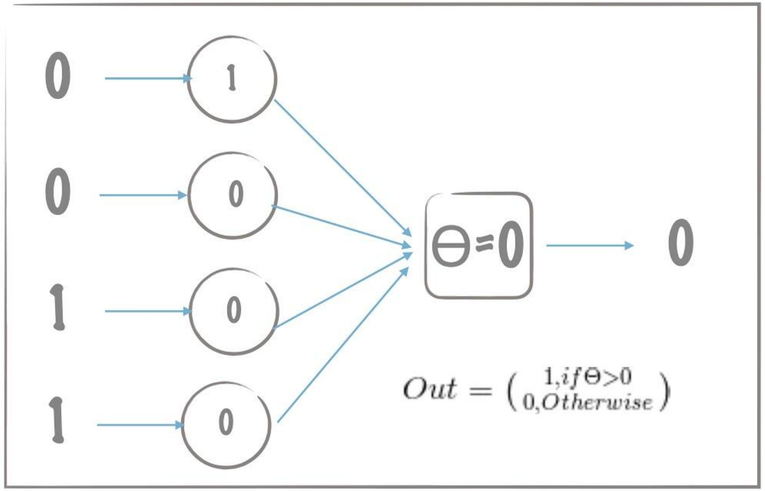
P(out)

out
0
0
1
1
1
1
1
0

ΔWeights

Δw1	Δw2	Δw3	Δw4
0	0	0	0
1	1	1	1
0	0	0	0
0	-1	-1	-1
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Learn with Google AI | Explore ML



Wi = Wi + ΔWi

ΔWi = -η * [P(out) -T(Out)] * xi

η is learning rate for perceptron

Training Set

	x	y	z	bias
	0	0	1	1
	1	1	1	1
epoch3	1	0	1	1
	0	1	1	1
	0	0	1	1
epoch4	1	1	1	1
	1	0	1	1
	0	1	1	1

T(out)

out
0
1
1
0
0
1
1
0

Weights

w1	w2	w3	w4
1	0	0	0

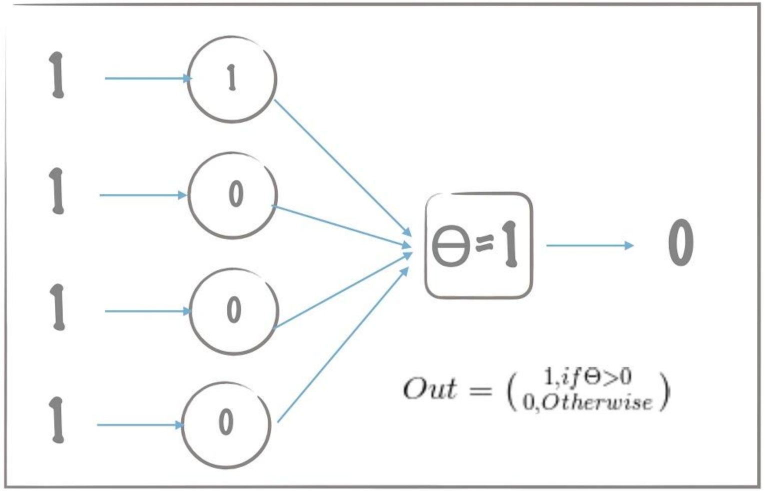
P(out)

out
0

ΔWeights

Δw1	Δw2	Δw3	Δw4
0	0	0	0

Learn with Google AI | Explore ML



Wi = Wi + ΔWi
ΔWi = -η * [P(out) -T(Out)] * xi
η is learning rate for perceptron

Training Set

	x	y	z	bias
epoch3	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1
epoch4	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1

T(out)

out
0
1
1
0
0
1
1
0

Weights

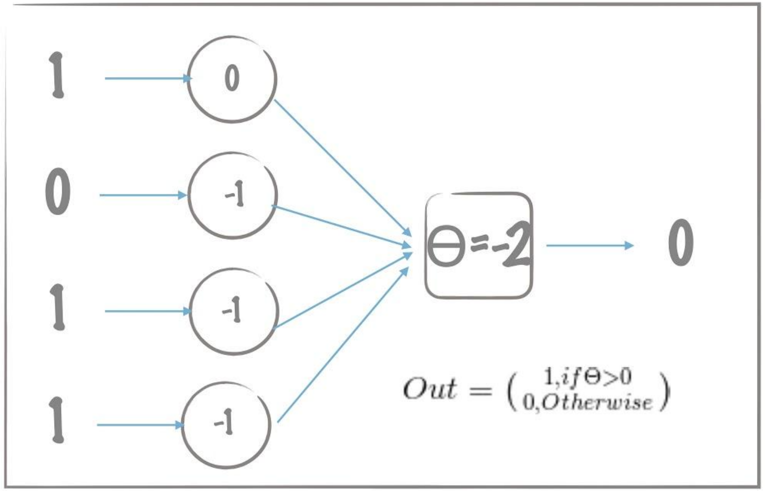
w1	w2	w3	w4
1	0	0	0
1	0	0	0

P(out)

out
0
1

ΔWeights

Δw1	Δw2	Δw3	Δw4
0	0	0	0
0	0	0	0



Wi = Wi + ΔWi
ΔWi = -η * [P(out) -T(Out)] * xi
η is learning rate for perceptron

Training Set

	x	y	z	bias
epoch3	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1
epoch4	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1

T(out)

out
0
1
1
0
0
1
1
0

Weights

w1	w2	w3	w4
1	0	0	0
1	0	0	0
1	0	0	0

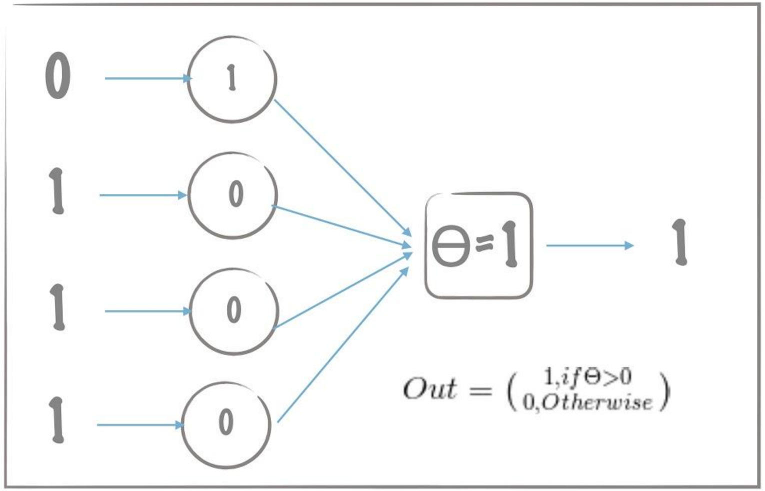
P(out)

out
0
1
1

ΔWeights

Δw1	Δw2	Δw3	Δw4
0	0	0	0
0	0	0	0
0	0	0	0

Learn with Google AI | Explore ML

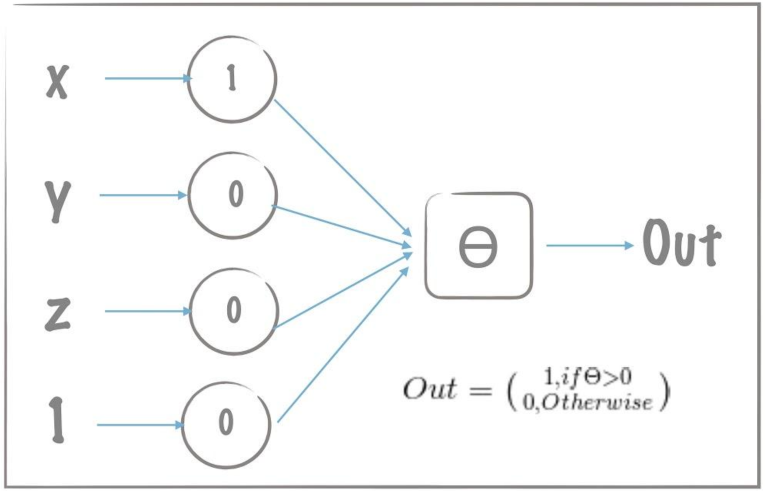


$W_i = W_i + \Delta W_i$

$\Delta W_i = -\eta * [P(out) - T(Out)] * x_i$

η is learning rate for perceptron

	Training Set				T(out)	Weights				P(out)	ΔWeights			
	x	y	z	bias	out	w1	w2	w3	w4	out	Δw1	Δw2	Δw3	Δw4
epoch3	0	0	1	1	0	1	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	0	0	0	1	0	0	0	0
	1	0	1	1	1	1	0	0	0	1	0	0	0	0
	0	1	1	1	0	1	0	0	0	0	0	0	0	0
epoch4	0	0	1	1	0									
	1	1	1	1	1									
	1	0	1	1	1									
	0	1	1	1	0									



Wi = Wi + ΔWi

ΔWi = -η * [P(out) -T(Out)] * xi

η is learning rate for perceptron

Training Set

	x	y	z	bias
epoch3	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1
epoch4	0	0	1	1
	1	1	1	1
	1	0	1	1
	0	1	1	1

T(out)

out
0
1
1
0
0
1
1
0

Weights

w1	w2	w3	w4
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0

P(out)

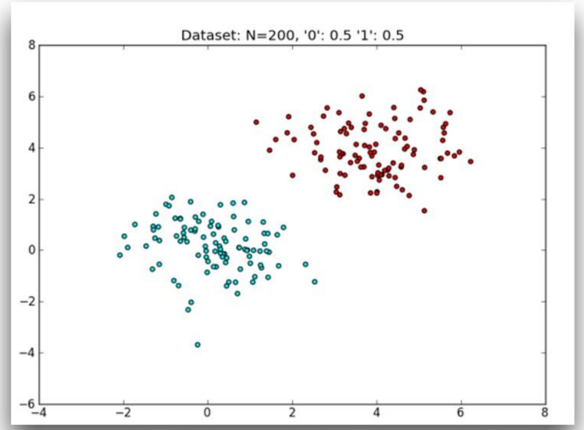
out
0
1
1
0

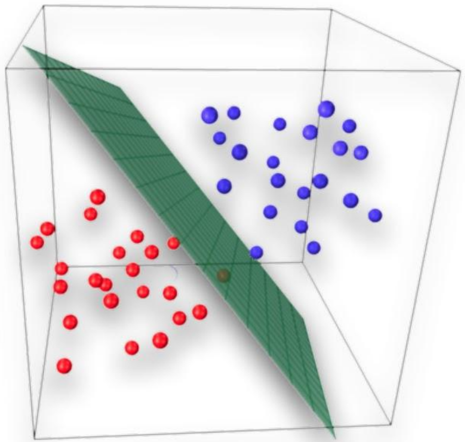
ΔWeights

Δw1	Δw2	Δw3	Δw4
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

In epoch3, 'loss' is 0. So, we can assume that Perceptron has learnt the pattern

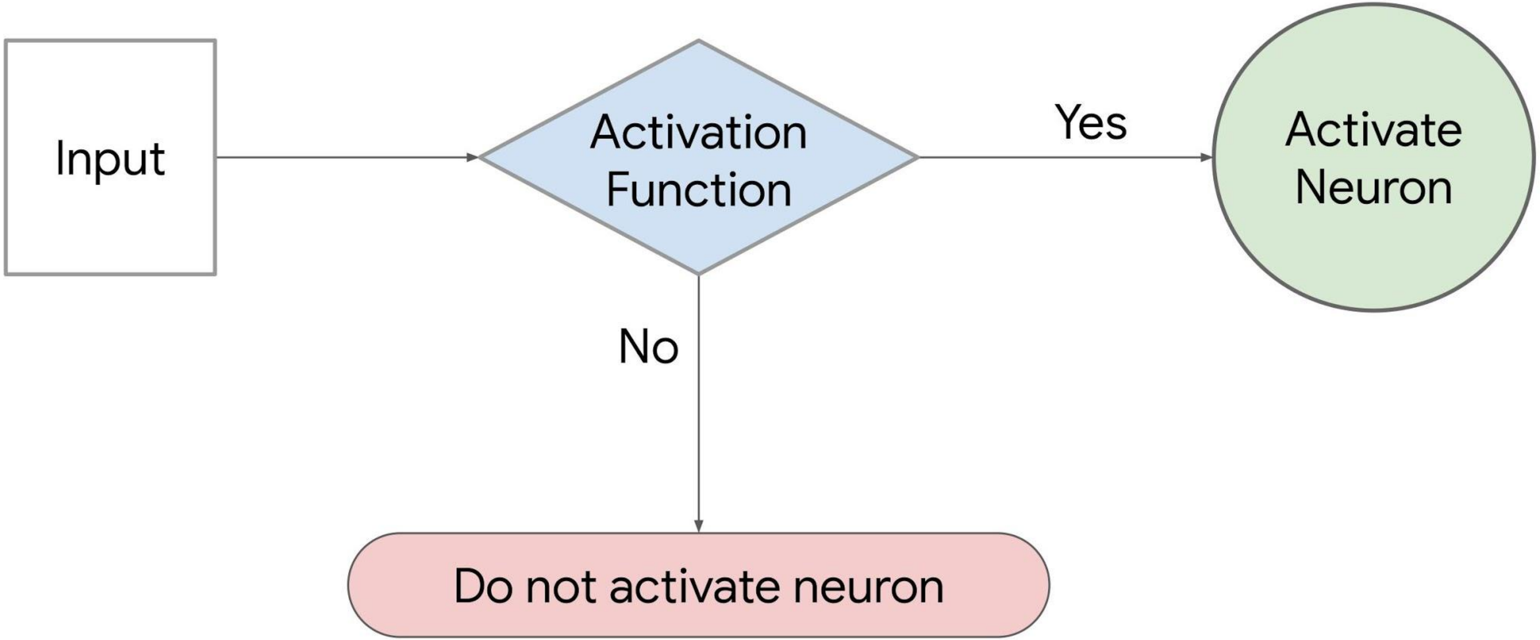
We trained our Perceptron Model



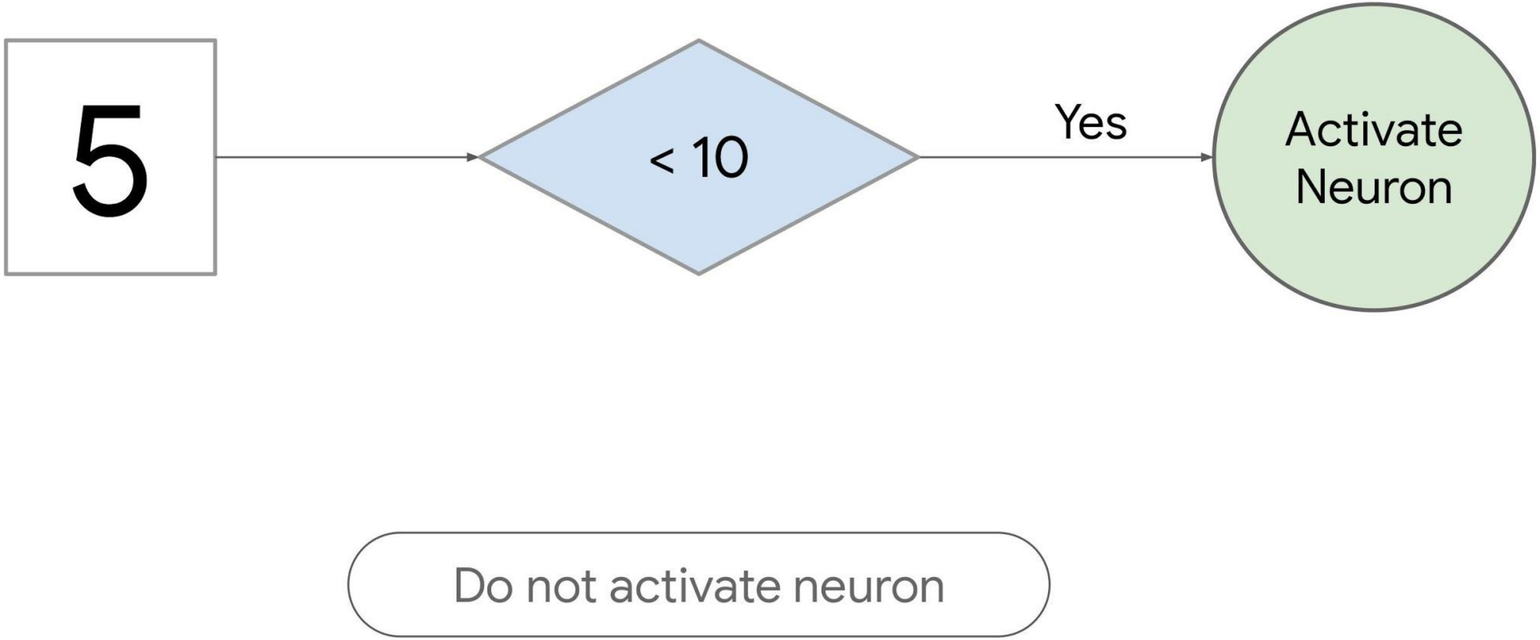


On a Linear Separable Distribution

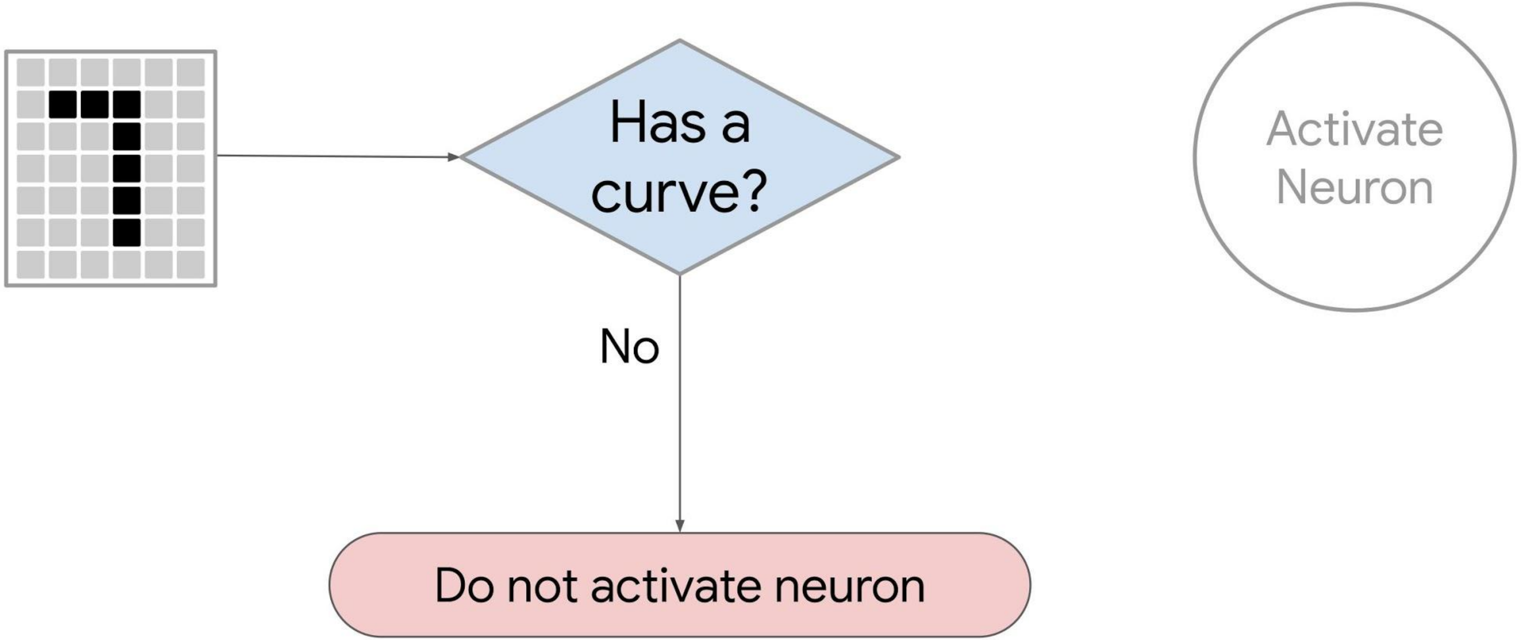
Perception of Artificial Neuron



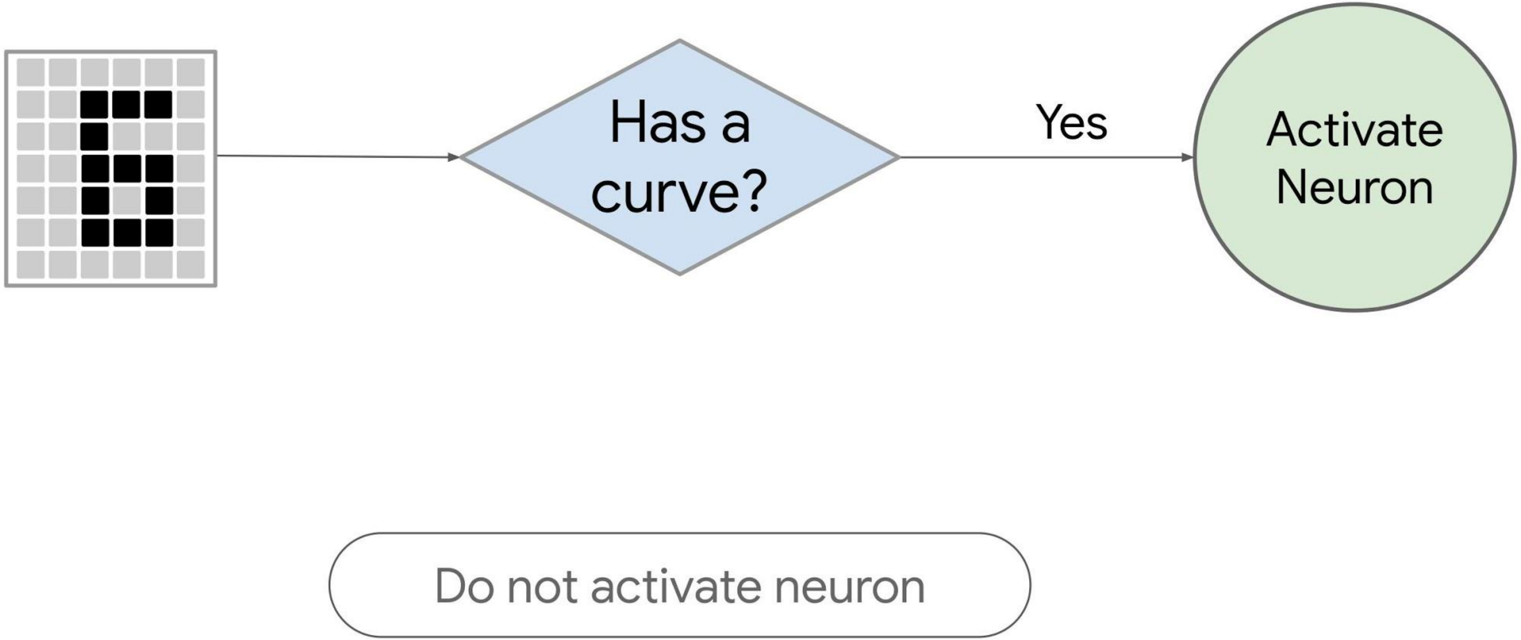
Perception of Artificial Neuron



Perception of Artificial Neuron



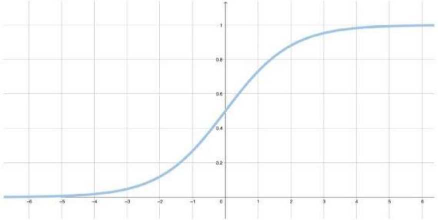
Perception of Artificial Neuron



Common Activation Functions

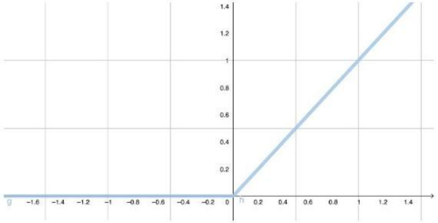
Sigmoid activation function converts the weighted sum to a value between **0** and **1**.

$$F(x) = \frac{1}{1 + e^{-x}}$$



ReLU (Rectified Linear Unit) activation function often works a little better than a smooth function like the sigmoid, while also being significantly easier to compute.

$$F(x) = max(0, x)$$



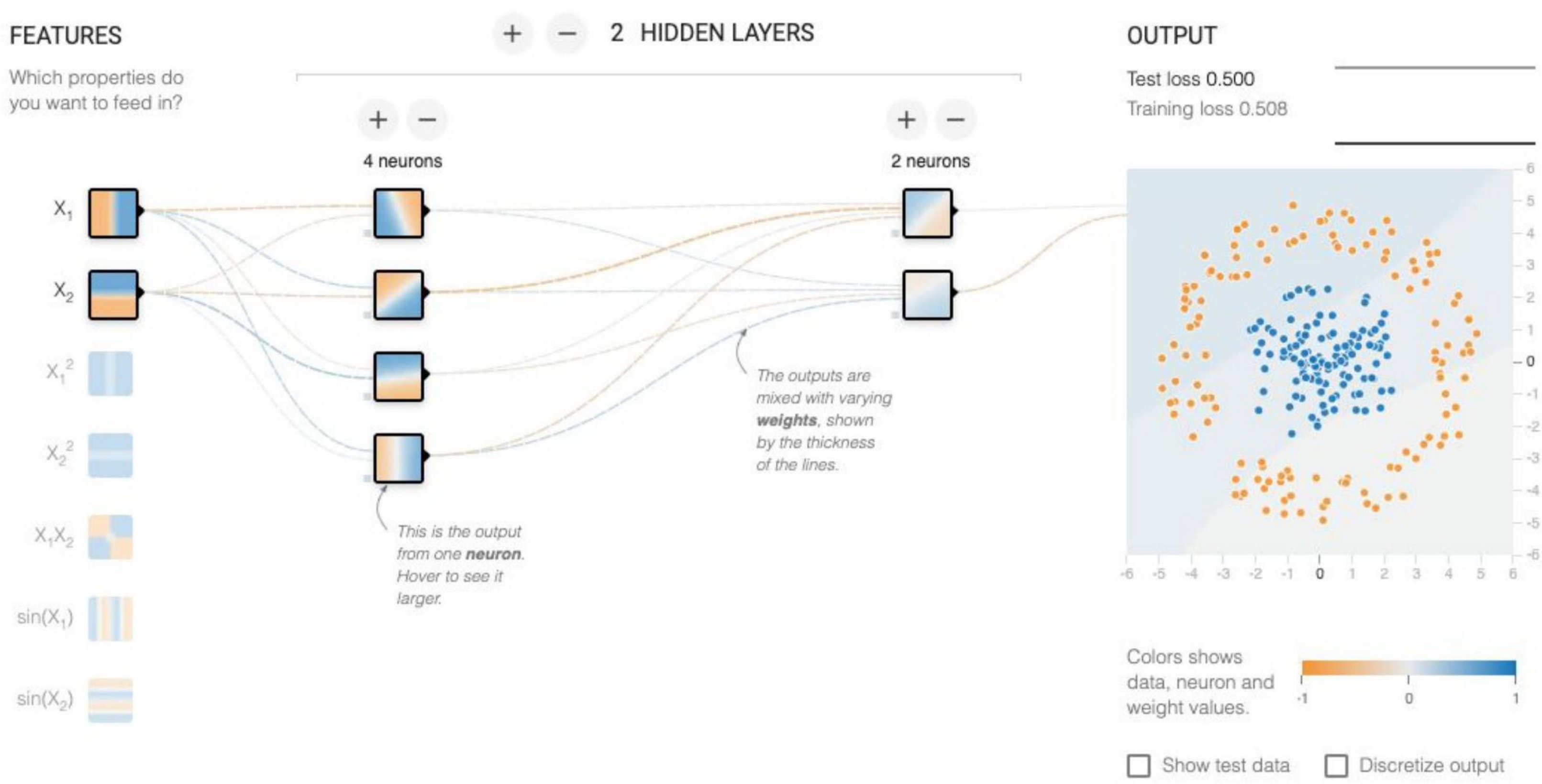
Getting Bored ?

Let's Play with Neural Networks

But where to play in Edusat Lab ?

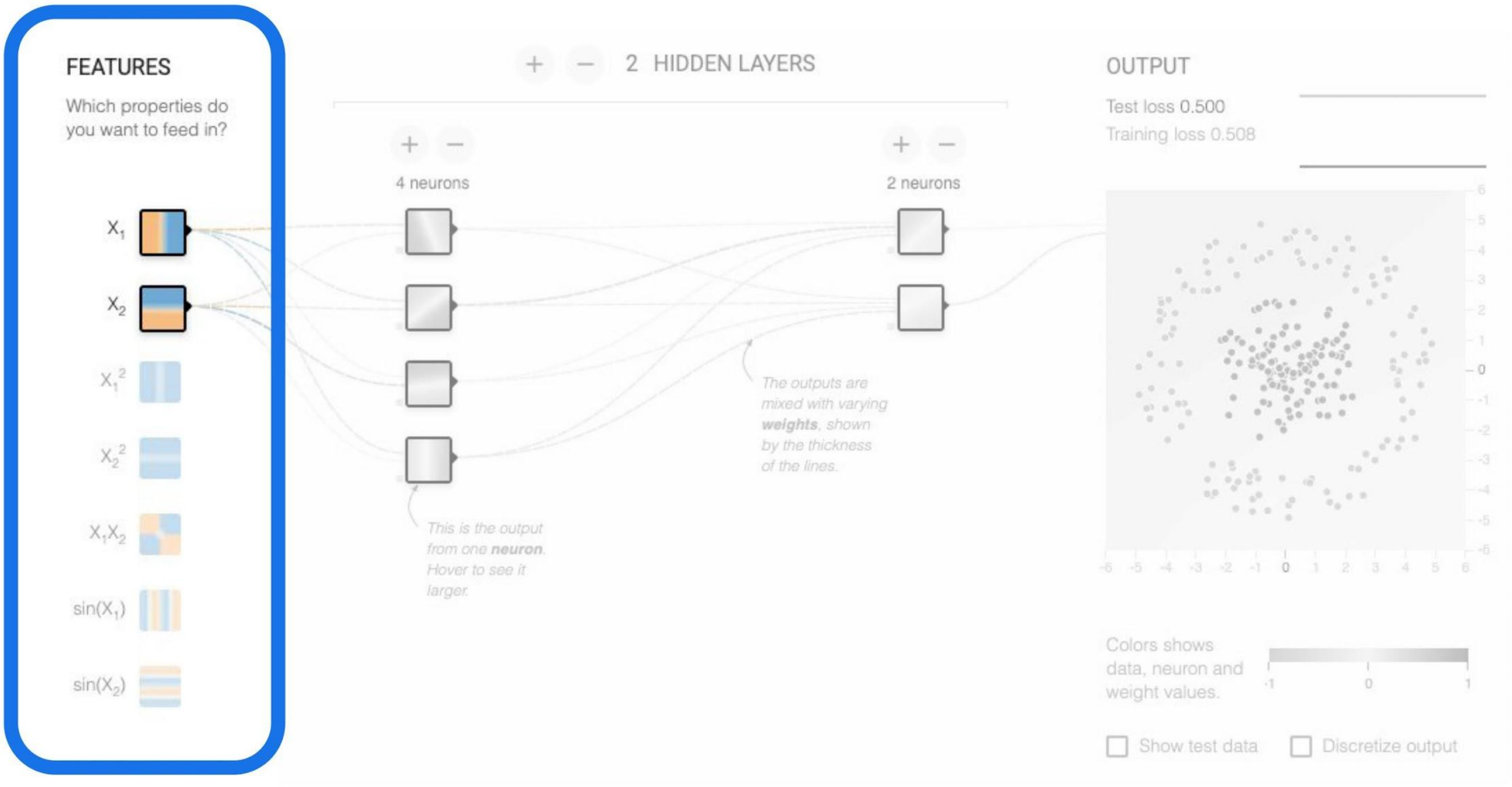
Let's play in the Tensorflow Playground

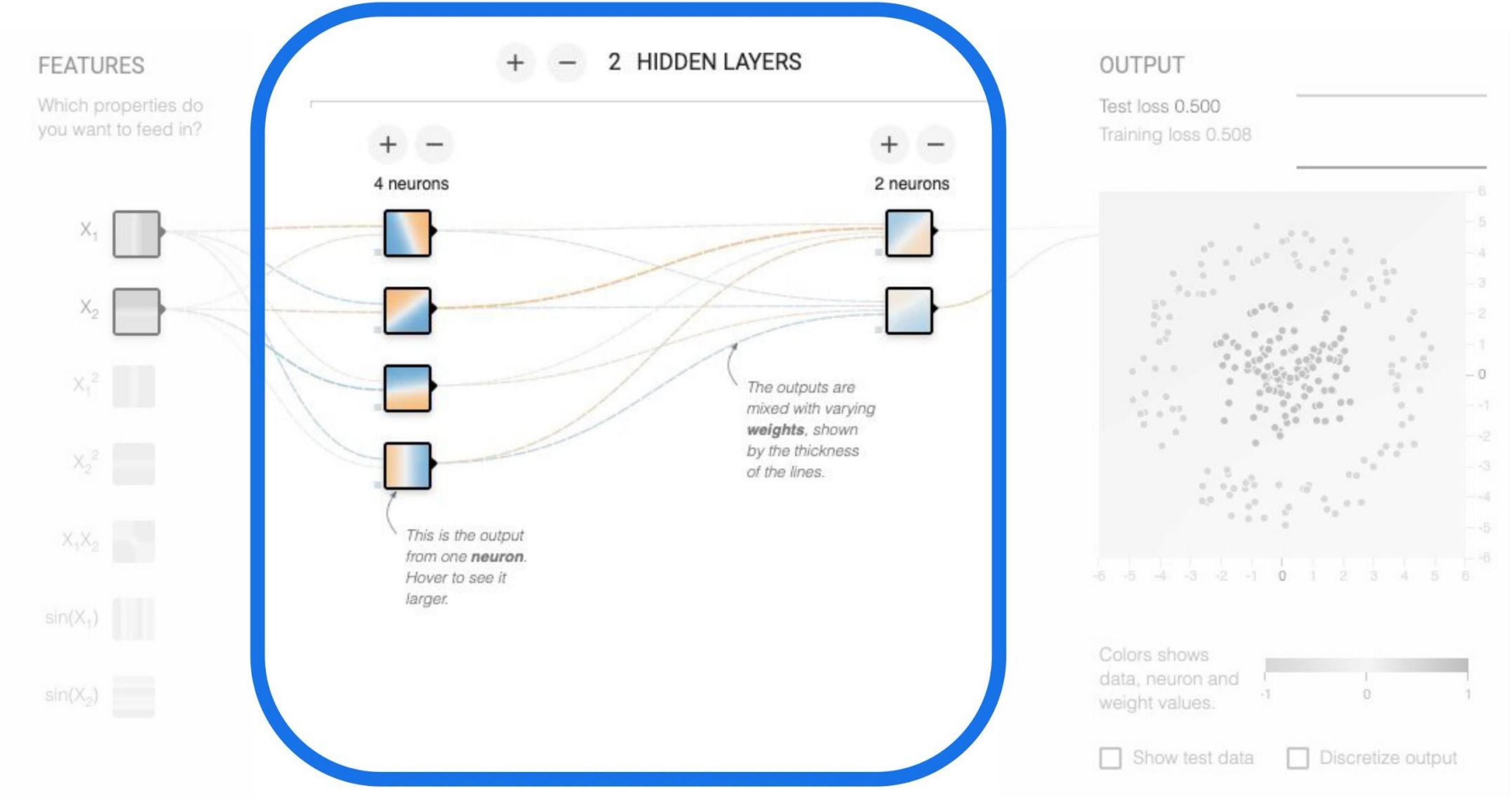
http://bit.do/BPITPlay



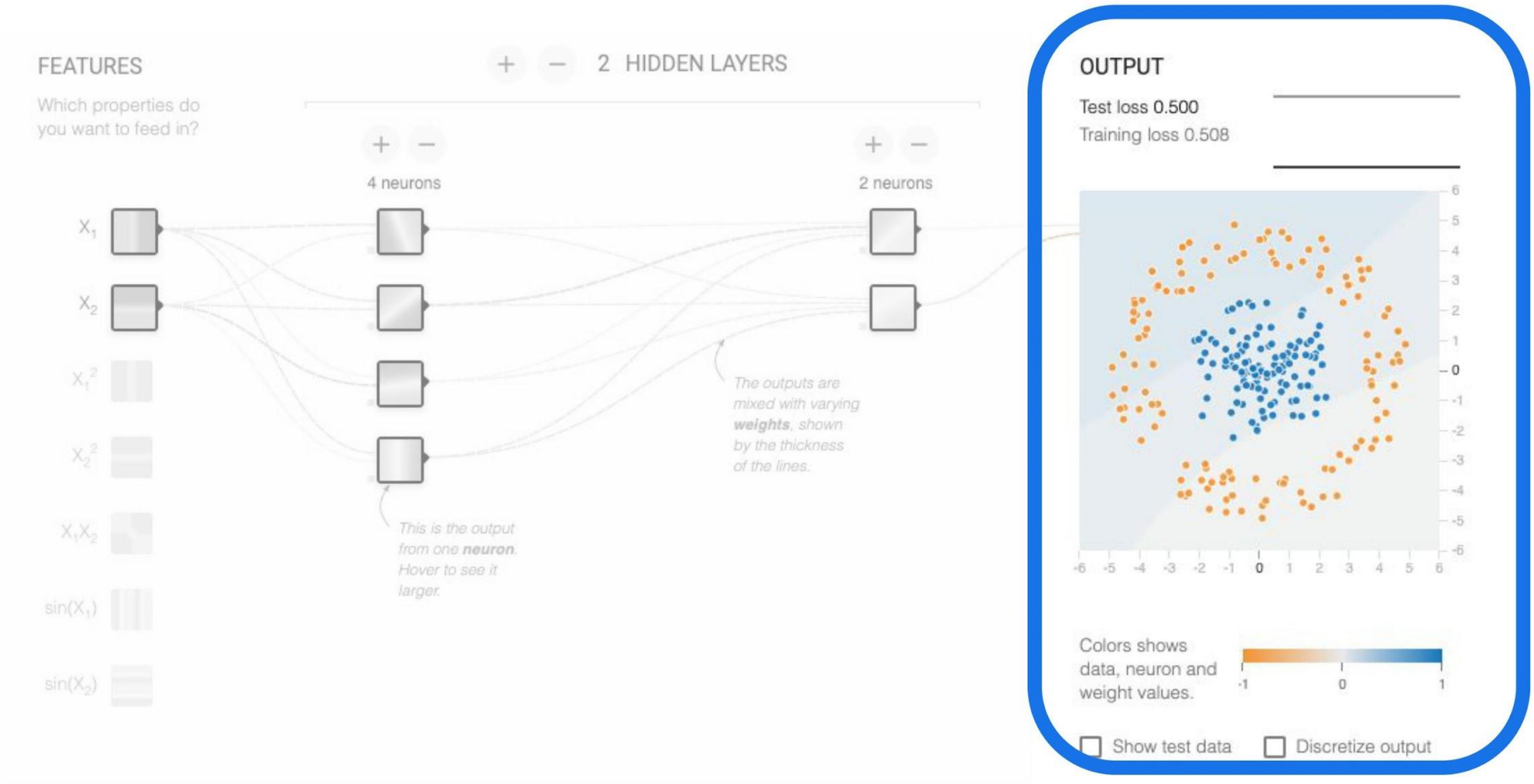
FEATURES
Which properties do you want to feed in?
X1
X2
X1^2
X2^2
X1X2
sin(X1)
sin(X2)
2 HIDDEN LAYERS
4 neurons
2 neurons
This is the output from one **neuron**. Hover to see it larger.
The outputs are mixed with varying **weights**, shown by the thickness of the lines.
OUTPUT
Test loss 0.500
Training loss 0.508
Colors shows data, neuron and weight values.
-1
0
1
Show test data
Discretize output

Learn with Google AI | Explore ML




FEATURES
Which properties do you want to feed in?
X1
X2
X1²
X2²
X1X2
sin(X1)
sin(X2)
+ − 2 HIDDEN LAYERS
+ −
4 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 0.500
Training loss 0.508
Colors shows data, neuron and weight values.
-1
0
1
Show test data
Discretize output

Learn with Google AI | Explore ML



Learn with Google AI | Explore ML

Still feeling Bored ?

Let’s play an awesome game now.

http://bit.do/BPITPlay1

- Wanna play more ? Visit g.co/teachablemachine
- See a Neural Network Demo at http://bit.do/BPIT1

Let me answer your Questions now.

Finally, it’s your time to speak.

Danke Scheon

Learn with Google AI | Explore ML

Questions ? Any Feedbacks ? Did you like the talk?
Tell me about it.

If you think I can help you,
connect with me via

Email : ayonroy2000@gmail.com

LinkedIn / Github / Telegram Username : ayonroy2000

Website : https://AYONROY.ML/