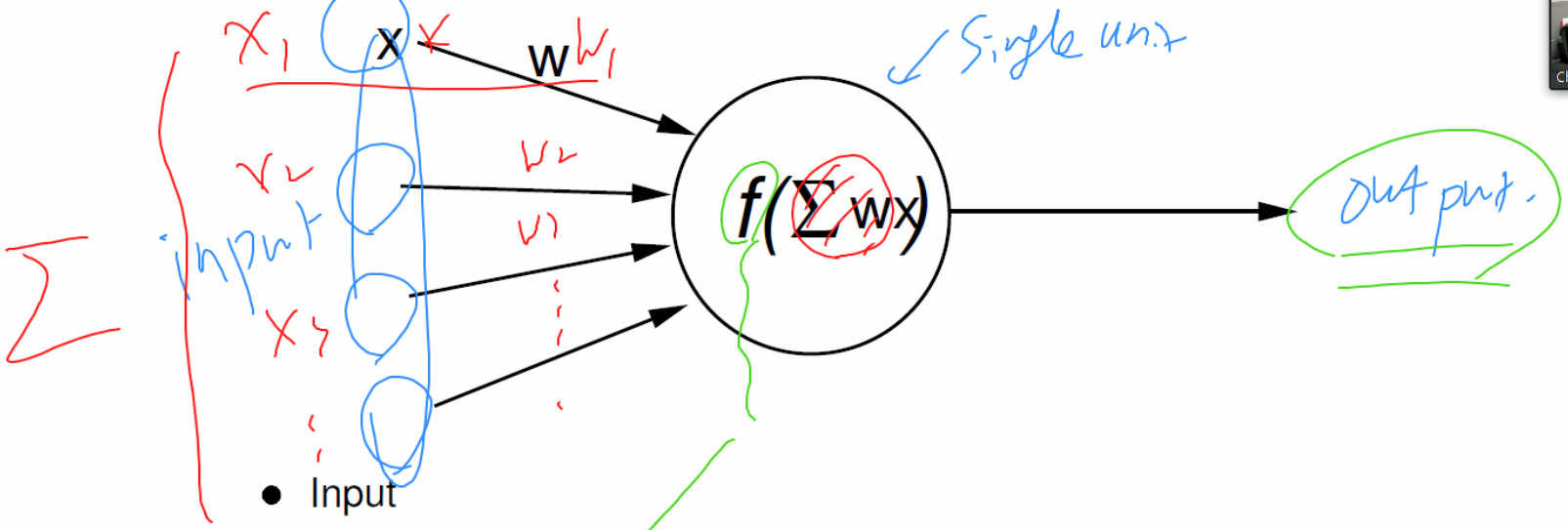
Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Abstraction of the Neuron in Neural Networks



- Input
- Connection weight
- Transfer function: $f(\cdot)$

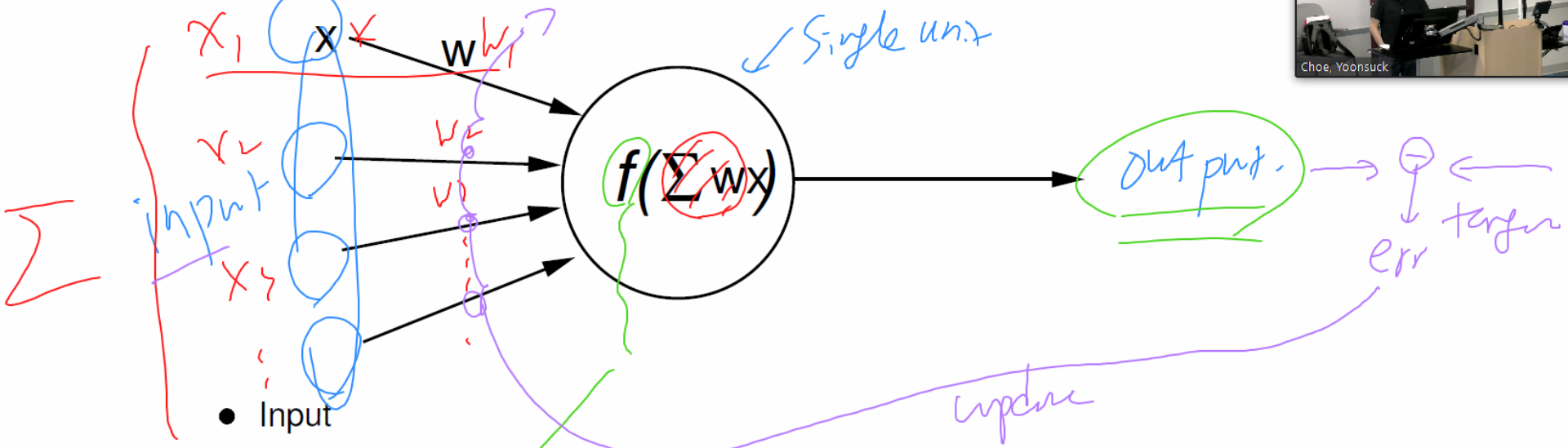
Typical transfer functions: step-function or sigmoid.

Then you get the output, so that's very simple and there could be many different forms of transfer functions will look

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Abstraction of the Neuron in Neural Networks

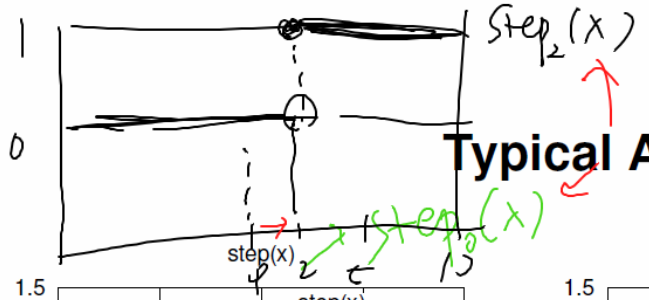


- Input
- Connection weight
- Transfer function: $f(\cdot)$

Typical transfer functions: step-function or sigmoid.

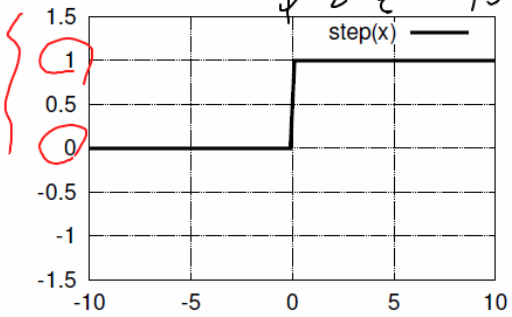
this arrow to update these question links so that the error is reduced when you present the same input with the

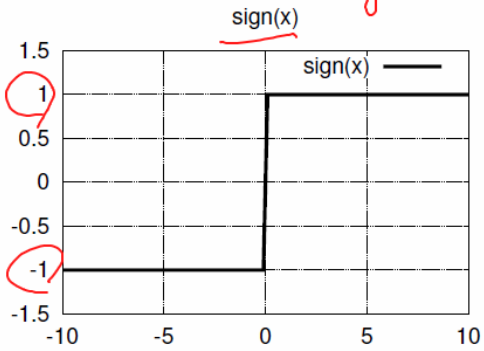
You are screen sharing Stop Share

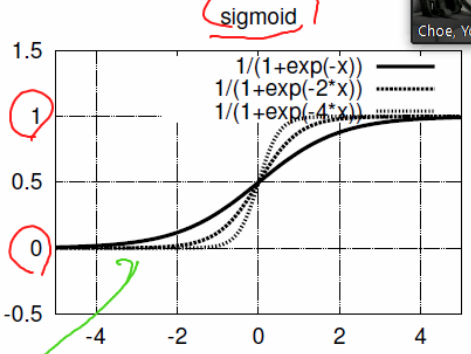


Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

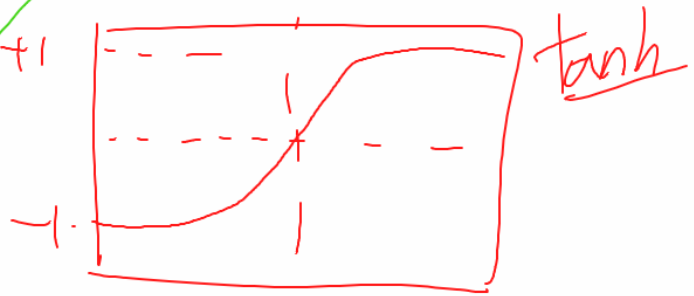
Typical Activation Functions







- $Step_t(x) = 1$ if $x \geq t$, 0 if $x < t$
- $Sign(x) = +1$ if $x \geq 0, -1$ if $x < 0$
- $Sigmoid(x) = \frac{1}{1+e^{-x}}$



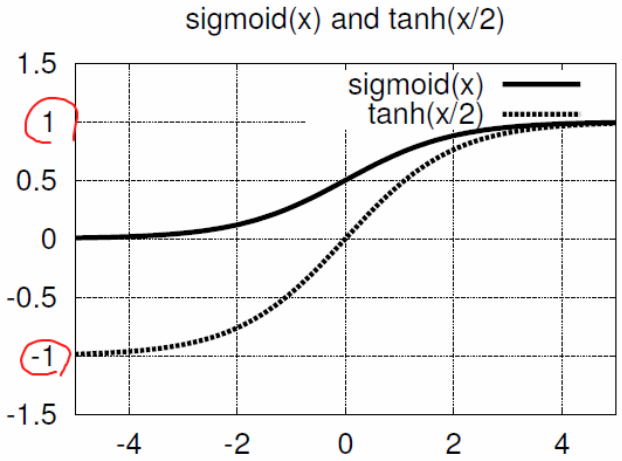
Note that $Step_t(x) = Step_0(x - t)$, which we will simply call $Step(x - t)$.

Then these 2 are the same

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

More Activation Functions: $tanh(\frac{x}{2})$



- $Sigmoid(x) = \frac{1}{1+e^{-x}}$
- $tanh(\frac{x}{2}) = \frac{1-e^{-x}}{1+e^{-x}}$

And the the functional form is this one over one minus the exponential of minus 6, divided by one plus exponential of minus X

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Classification of Neural Networks

Teacher exists?

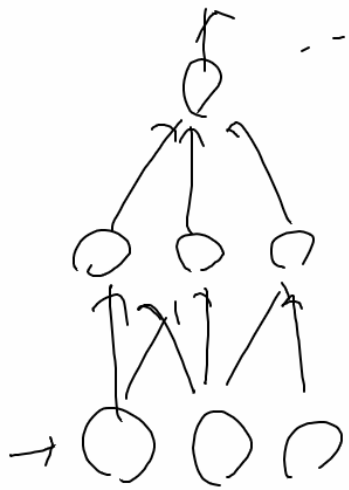
- Supervised (with teacher): perceptrons, backpropagation network, etc.
- Unsupervised (no teacher): self-organizing maps, etc.

multilayer perceptrons, neural networks

Recurrent connections?

- Feed-forward: perceptrons, backpropagation network, etc.
- Recurrent: Hopfield network, Boltzmann machines, SRN (simple recurrent network), LSTM, GRU, etc.

So that's one way of the candidates
You are screen sharing Stop Share



Feed forward.

i^p

Classification of Neural Networks

Teacher exists?

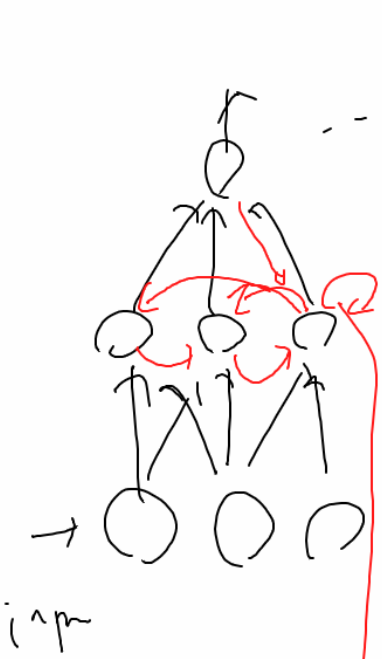
- Supervised (with teacher): perceptrons, backpropagation network, etc.
- Unsupervised (no teacher): self-organizing maps, etc.

multilayer perceptrons, neural networks

Recurrent connections?

- Feed-forward: perceptrons, backpropagation network, etc.
- Recurrent: Hopfield network, Boltzmann machines, SRN (simple recurrent network), LSTM, GRU, etc.

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On
Choe, Yoonsuck
So if there's no loop this is called the feed forward you know
You are screen sharing Stop Share



Classification of Neural Networks

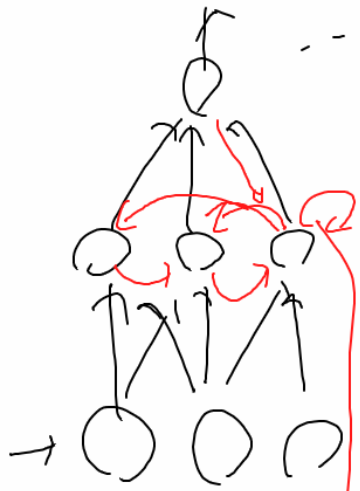
Teacher exists?

- Supervised (with teacher): perceptrons, backpropagation network, etc.
- Unsupervised (no teacher): self-organizing maps, etc.

Recurrent connections?

- Feed-forward: perceptrons, backpropagation network, etc.
- Recurrent: Hopfield network, Boltzmann machines, SRN (simple recurrent network), LSTM, GRU, etc.

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On



Classification of Neural Networks

Teacher exists?

- Supervised (with teacher): perceptrons, backpropagation network, etc.
- Unsupervised (no teacher): self-organizing maps, etc.

Recurrent connections?

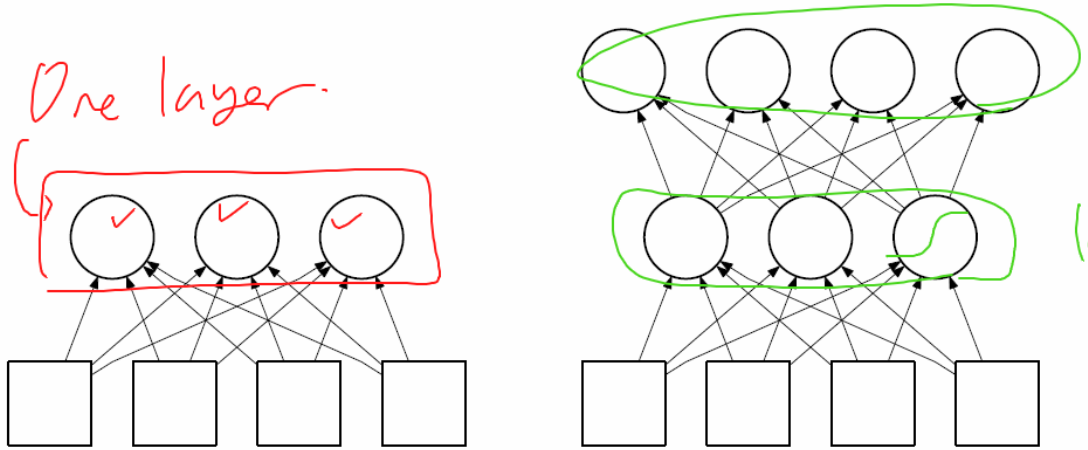
- Feed-forward: perceptrons, backpropagation network, etc.
- Recurrent: Hopfield network, Boltzmann machines, SRN (simple recurrent network), LSTM, GRU, etc.

Okay, and there are several different instances of this and we'll probably go through goes through some of these when we look into deep learning

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Feedforward Networks

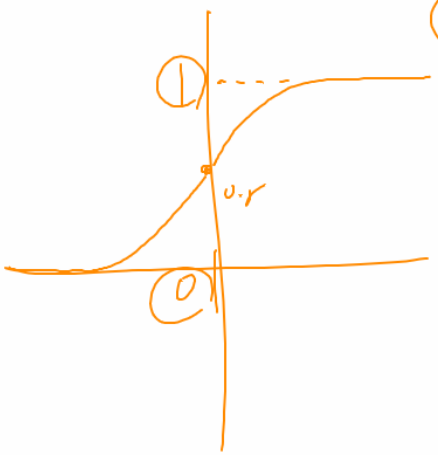


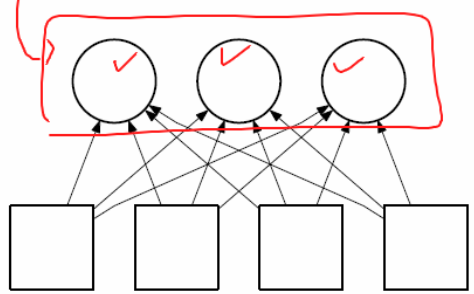
- Perceptrons: single layer, threshold-gated
- Backpropagation networks: multiple layers, sigmoid (or tanh) activation function.

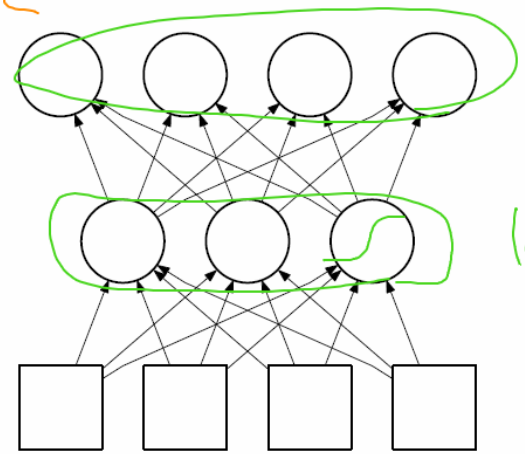
And again, these appear in deep learning, so we will only consider Sigmoid or hyperbolic tangent

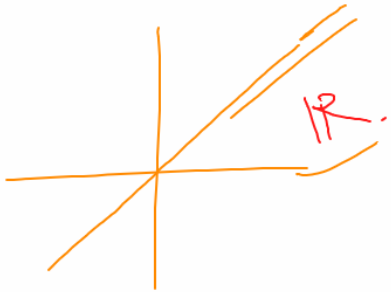
You are screen sharing Stop Share

Feedforward Networks



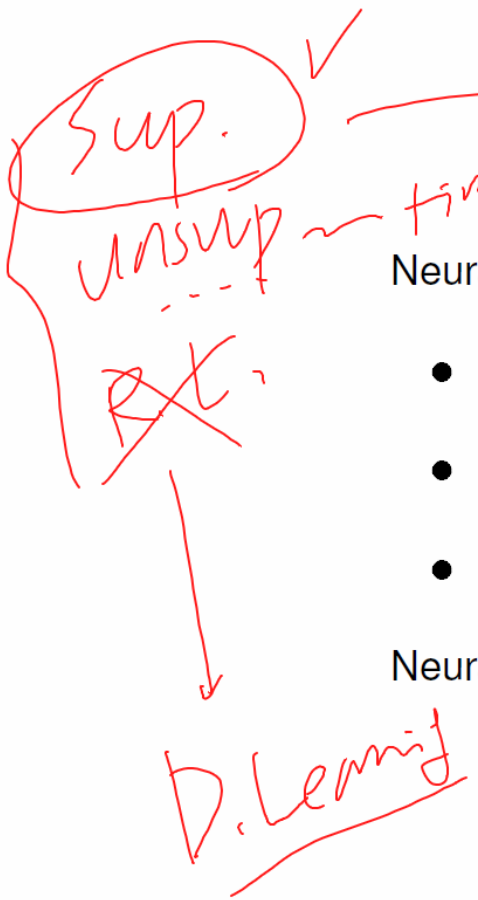






- Perceptrons: single layer, threshold-gated
- Backpropagation networks: multiple layers, sigmoid (or tanh) activation function.

But this is not



Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Neural Networks

Neural networks is one particular form of learning from data.

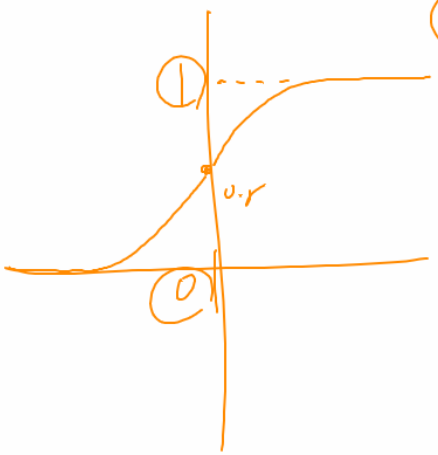
- simple processing elements: called “units”, or “neurons”
- connective structure and associated connection weights
- learning: adaptation of connection weights

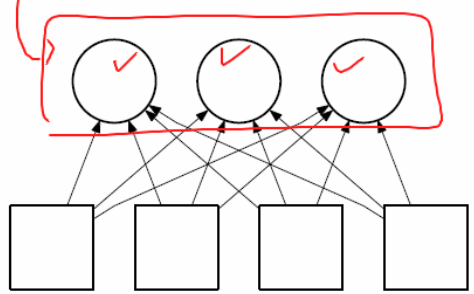
Neural networks mimic the human (or animal) nervous system.

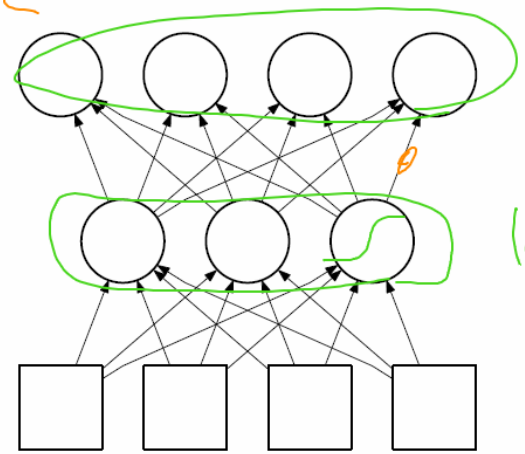
After this machine learning slides, well look at one specific reinforcement learning algorithm using neural networks

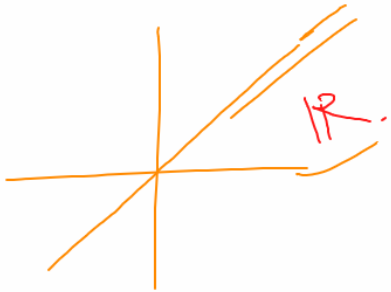
You are screen sharing Stop Share

Feedforward Networks





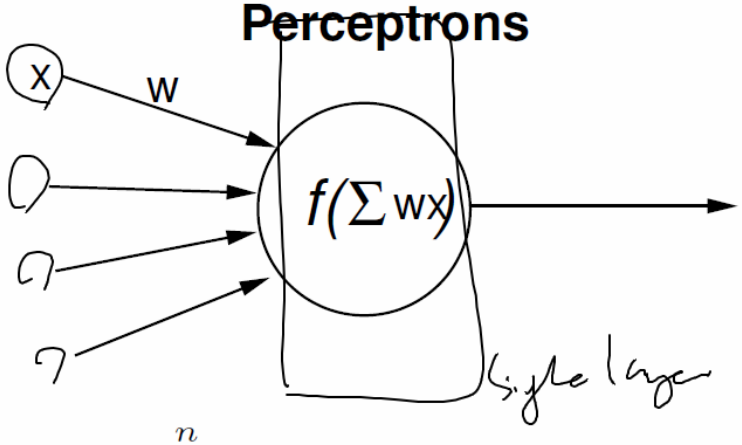




- Perceptrons: single layer, threshold-gated
- Backpropagation networks: multiple layers, sigmoid (or tanh) activation function.

one layer

Perceptrons



$$a_i = step_t(\sum_{j=1}^{n} W_{ij} a_j)$$

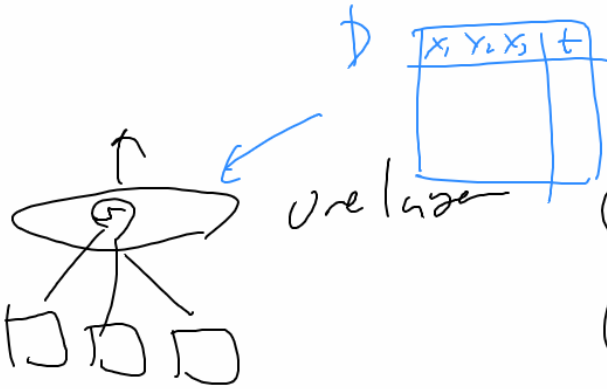
$$= step_0(\sum_{j=1}^{n} W_{ij} a_j - t)$$

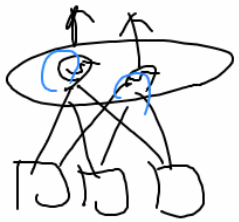
$$= step_0(t \times (-1) + \sum_{j=1}^{n} W_{ij} a_j)$$

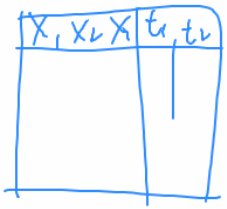
$$= step_0(\sum_{j=0}^{n} W_{ij} a_j), \text{ where } W_{i0} = t \text{ and } a_0 = -1 \quad (1)$$

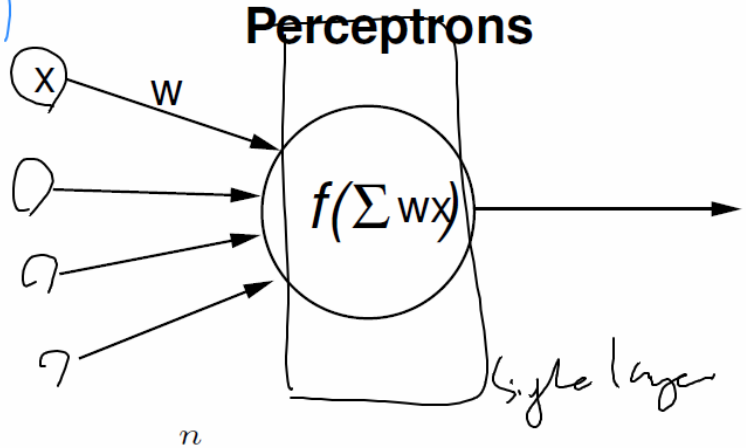
And then activation function is this step function

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On
Choe, Yoonsuck
You are screen sharing Stop Share









$$a_i = step_t(\sum_{j=1}^{n} W_{ij} a_j)$$

$$= step_0(\sum_{j=1}^{n} W_{ij} a_j - t)$$

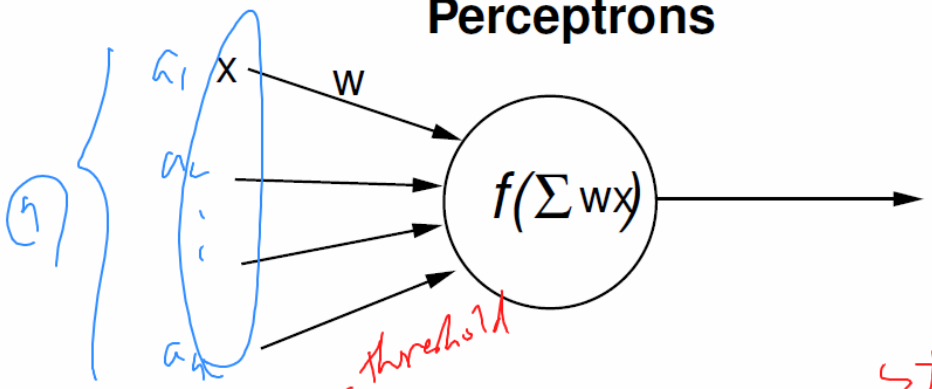
$$= step_0(t \times (-1) + \sum_{j=1}^{n} W_{ij} a_j)$$

$$= step_0(\sum_{j=1}^{n} W_{ij} a_j), \text{ where } W_{i0} = t \text{ and } a_0 = -1 \quad (1)$$

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On
Choe, Yoonsuck
Let's say t one t 2 right because you have 2 output values. Alright.
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Perceptrons

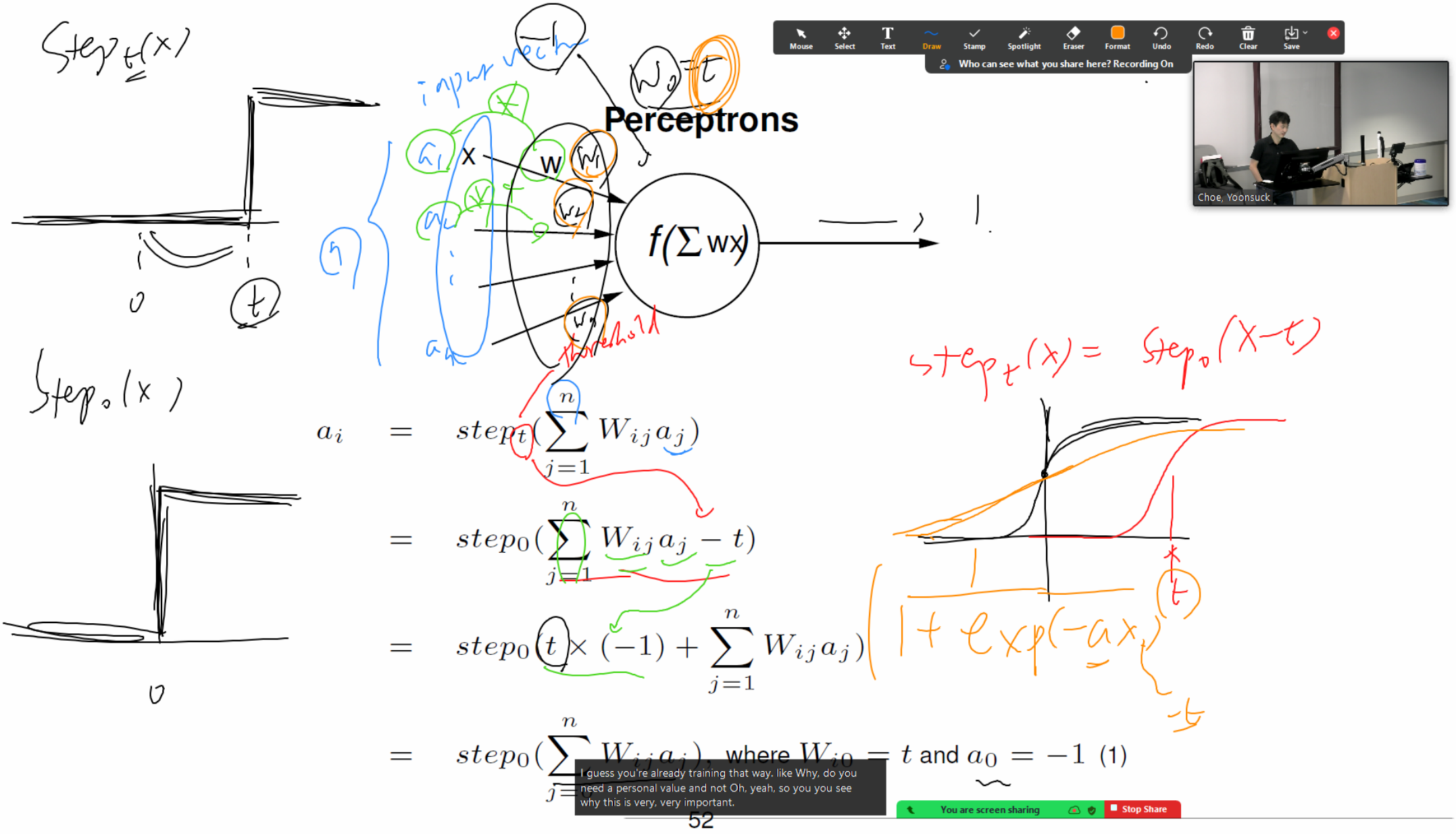


$$step_t(x) = step_0(x - t)$$

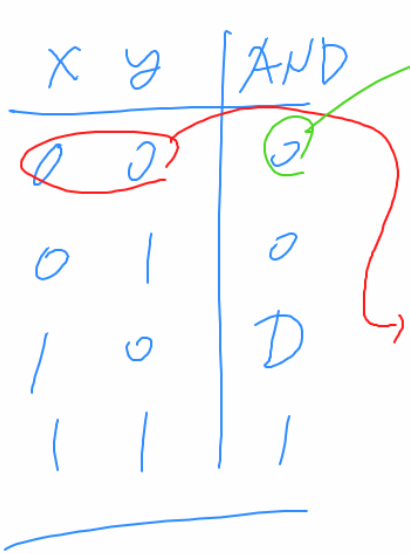
$$\begin{aligned} a_i &= step_t\left(\sum_{j=1}^{n} W_{ij} a_j\right) \\ &= step_0\left(\sum_{j=1}^{n} W_{ij} a_j - t\right) \\ &= step_0\left(t \times (-1) + \sum_{j=1}^{n} W_{ij} a_j\right) \\ &= step_0\left(\sum_{j=0}^{n} W_{ij} a_j\right), \text{ where } W_{i0} = t \text{ and } a_0 = -1 \quad (1) \end{aligned}$$

We saw that step T. of X equals step 0 x minus t so that's how this becomes that

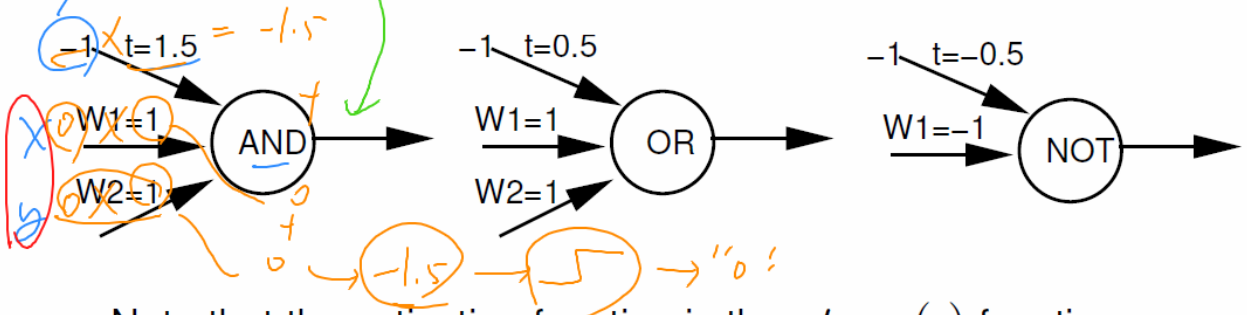
You are screen sharing Stop Share


Perceptrons
f(Σwx)
x
w
threshold
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
Who can see what you share here? Recording On
Choe, Yoonsuck
$a_i = step_t(\sum_{j=1}^{n} W_{ij} a_j)$
$= step_0(\sum_{j=1}^{n} W_{ij} a_j - t)$
$= step_0(t \times (-1) + \sum_{j=1}^{n} W_{ij} a_j)$
$= step_0(\sum_{j=1}^{n} W_{ij} a_j)$, where $W_{i0} = t$ and $a_0 = -1$ (1)
$\frac{1}{1+exp(-ax)}$
I guess you're already training that way, like Why, do you need a personal value and not Oh, yeah, so you you see why this is very, very important.
You are screen sharing
Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On



Boolean Logic Gates with Perceptron Units [a]



- Note that the activation function is the $step_0(\cdot)$ function.
- Perceptrons can represent basic boolean functions.
- Thus, a network of perceptron units can compute any Boolean function.

What about XOR or EQUIV?

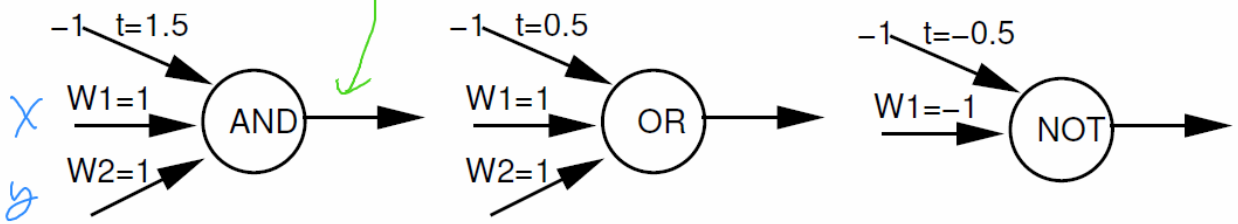
[a] Same as Russel & Norvig p.570, figure 19.6

Okay, So that's the first
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

x	y	AND
0	0	0
0	1	0
1	0	0
1	1	1

Boolean Logic Gates with Perceptron Units [a]



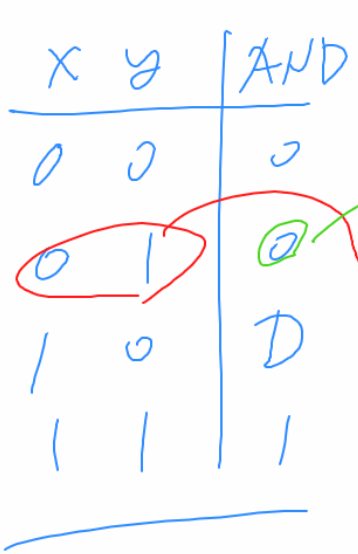
- Note that the activation function is the $step_0(\cdot)$ function.
- Perceptrons can represent basic boolean functions.
- Thus, a network of perceptron units can compute any Boolean function.

What about XOR or EQUIV?

[a] Same as Russel & Norvig p.570, figure 19.6

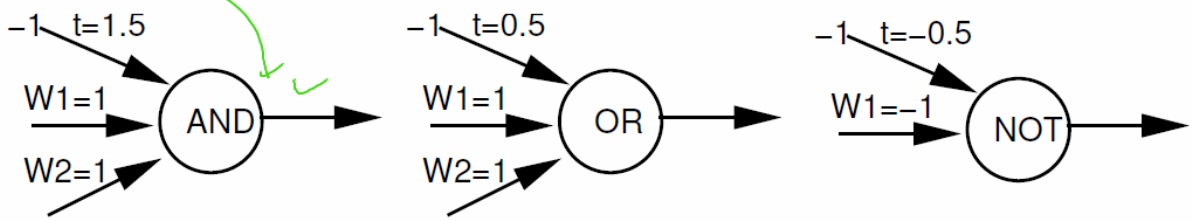
Let me just write it down here. This the second one is hold on.
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On



bias unit

Boolean Logic Gates with Perceptron Units [a]



- Note that the activation function is the $step_0(\cdot)$ function.
- Perceptrons can represent basic boolean functions.
- Thus, a network of perceptron units can compute any Boolean function.

What about XOR or EQUIV?

[a] Same as Russel & Norvig p.570, figure 19.6

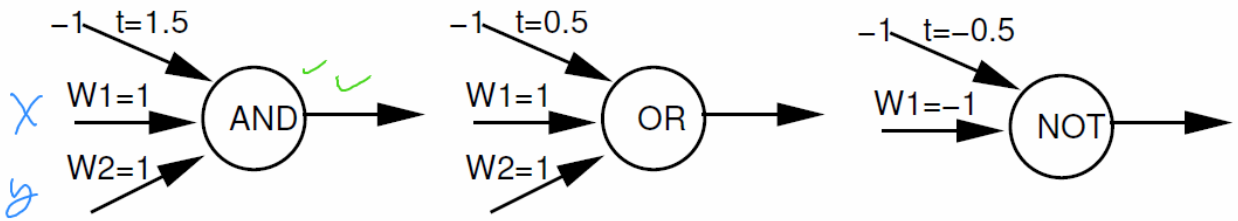
Now you want to check whether the output is 0
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

x	y	AND
0	0	0
0	1	0
1	0	0
1	1	1

bias unit

Boolean Logic Gates with Perceptron Units [a]



- Note that the activation function is the $step_0(\cdot)$ function.
- Perceptrons can represent basic boolean functions.
- Thus, a network of perceptron units can compute any Boolean function.

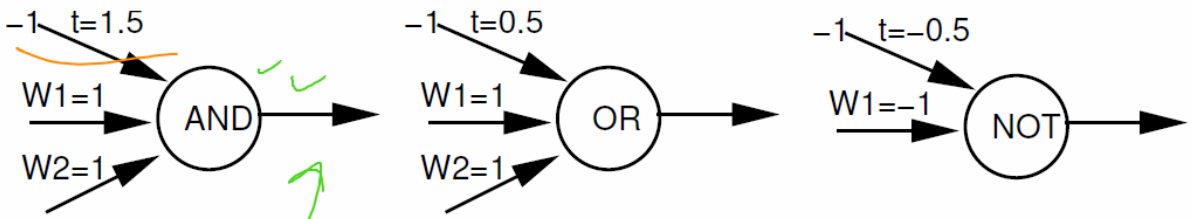
What about XOR or EQUIV?

[a] Same as Russel & Norvig p.570, figure 19.6

The outcome is the same, and Finally, you try 1 one let's see
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Boolean Logic Gates with Perceptron Units [a]



- Note that the activation function is the $step_0(\cdot)$ function.
- Perceptrons can represent basic boolean functions.
- Thus, a network of perceptron units can compute any Boolean function.

What about XOR or EQUIV?

[a] Same as Russel & Norvig p.570, figure 19.6

Okay, So this is greater than 0. So the output is one which is the correct output
You are screen sharing Stop Share

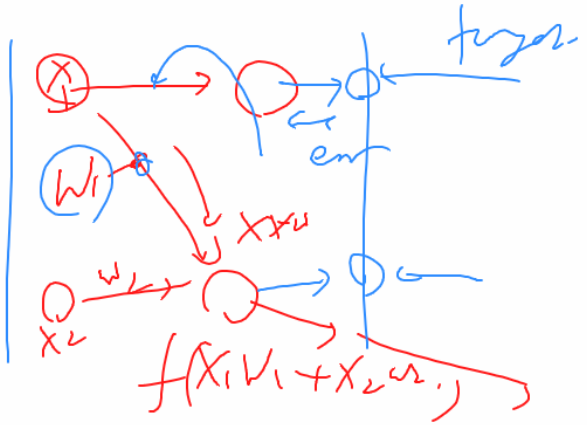
Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Neural Networks

Neural networks is one particular form of learning from data.

- simple processing elements: called "units", or "neurons"
- connective structure and associated connection weights
- learning: adaptation of connection weights

Neural networks mimic the human (or animal) nervous system.



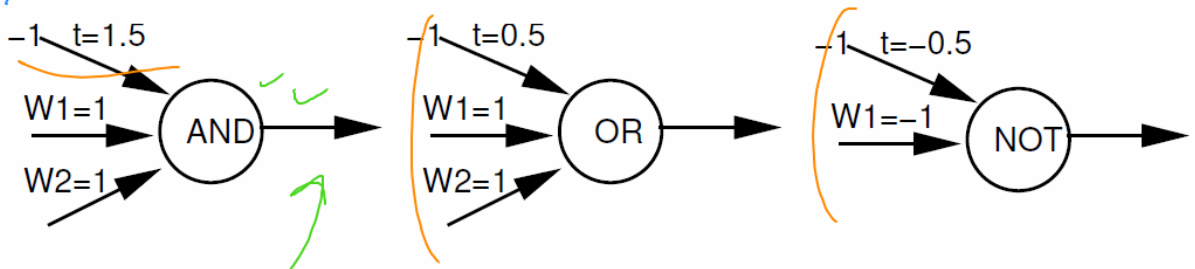
how learning
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

x	y	AND
0	0	0
0	1	0
1	0	0
1	1	1

bias unit

Boolean Logic Gates with Perceptron Units [a]



- Note that the activation function is the $step_0(\cdot)$ function.
- Perceptrons can represent basic boolean functions.
- Thus, a network of perceptron units can compute any Boolean function.

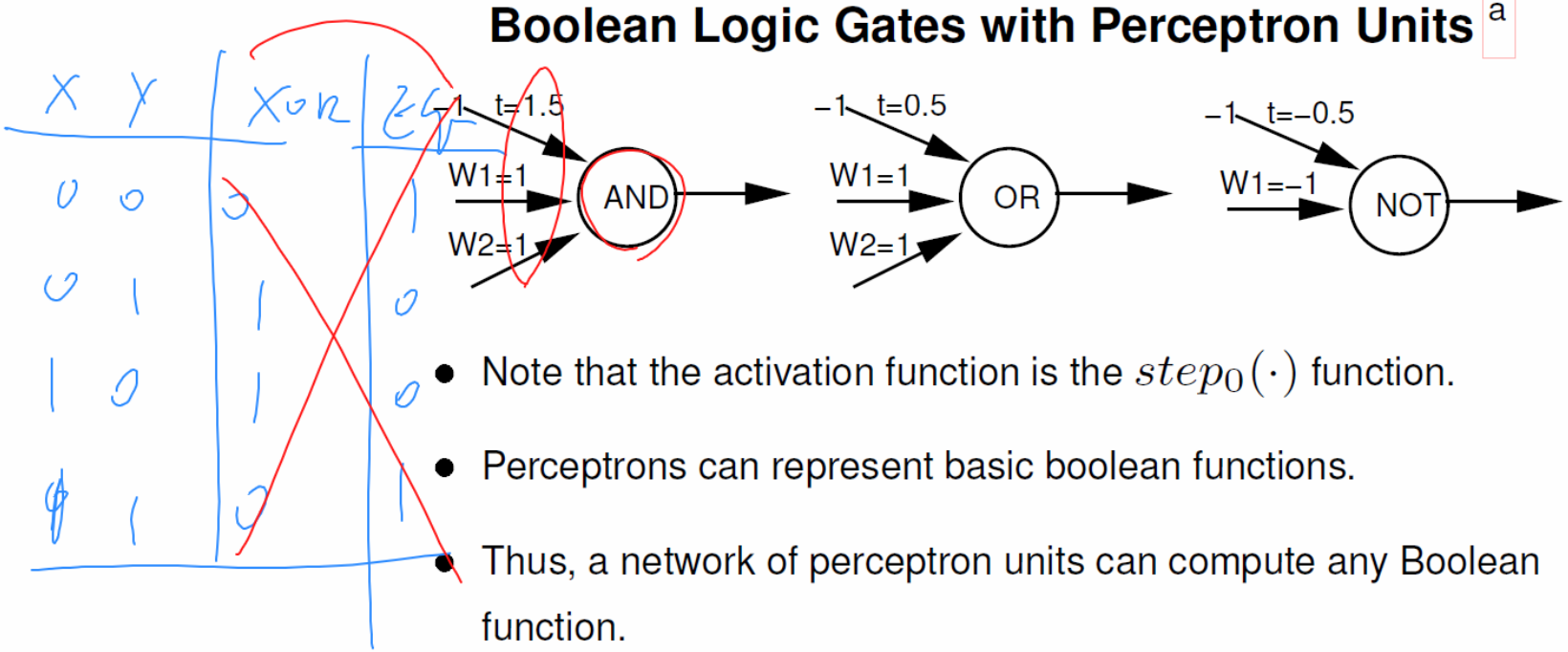
What about XOR or EQUIV?

[a] Same as Russel & Norvig p.570, figure 19.6

So so you can try the same with all of these other ones.
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Boolean Logic Gates with Perceptron Units [a]



- Note that the activation function is the $step_0(\cdot)$ function.
- Perceptrons can represent basic boolean functions.
- Thus, a network of perceptron units can compute any Boolean function.

What about XOR or EQUIV?

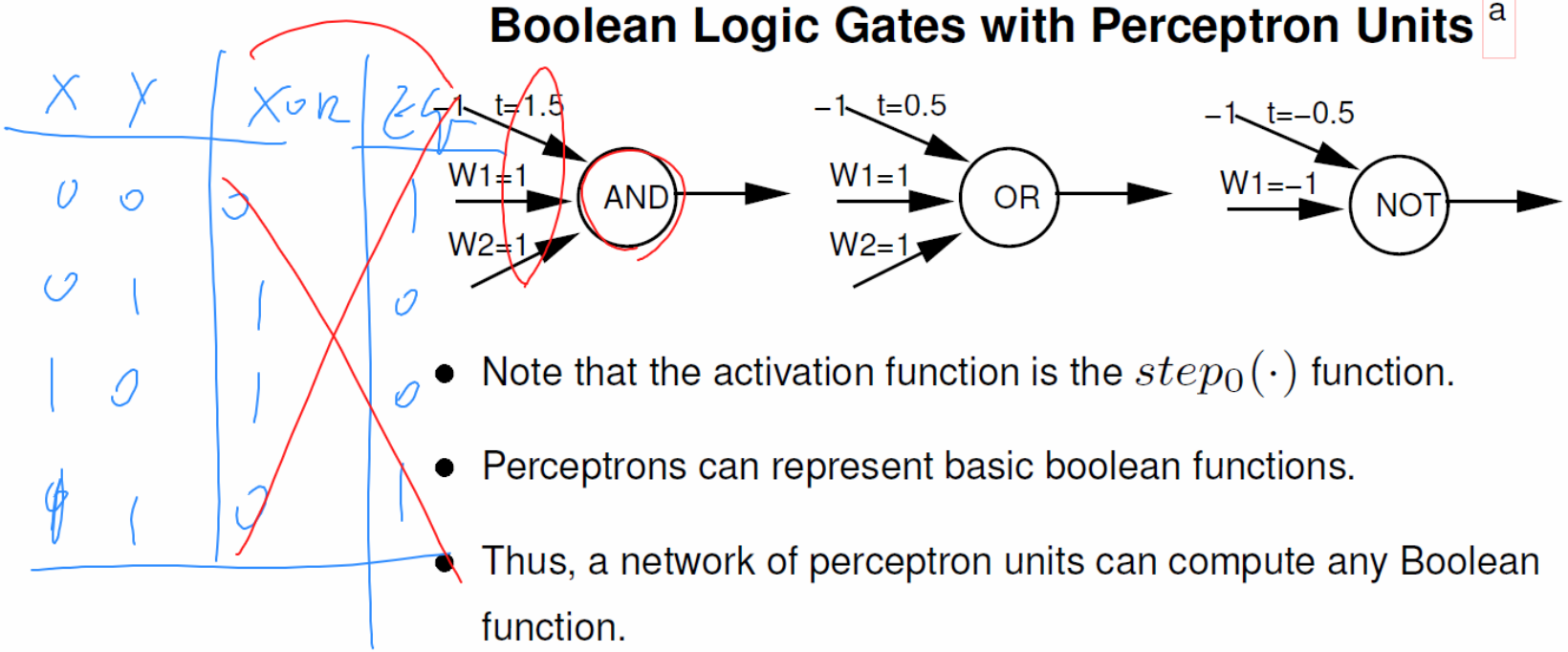
[a] Same as Russel & Norvig p.570, figure 19.6

realize only on is that this particular kind of sitting, regardless of what connection rate you said, You cannot learn these functions

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Boolean Logic Gates with Perceptron Units [a]



1985

- Note that the activation function is the $step_0(\cdot)$ function.
- Perceptrons can represent basic boolean functions.
- Thus, a network of perceptron units can compute any Boolean function.

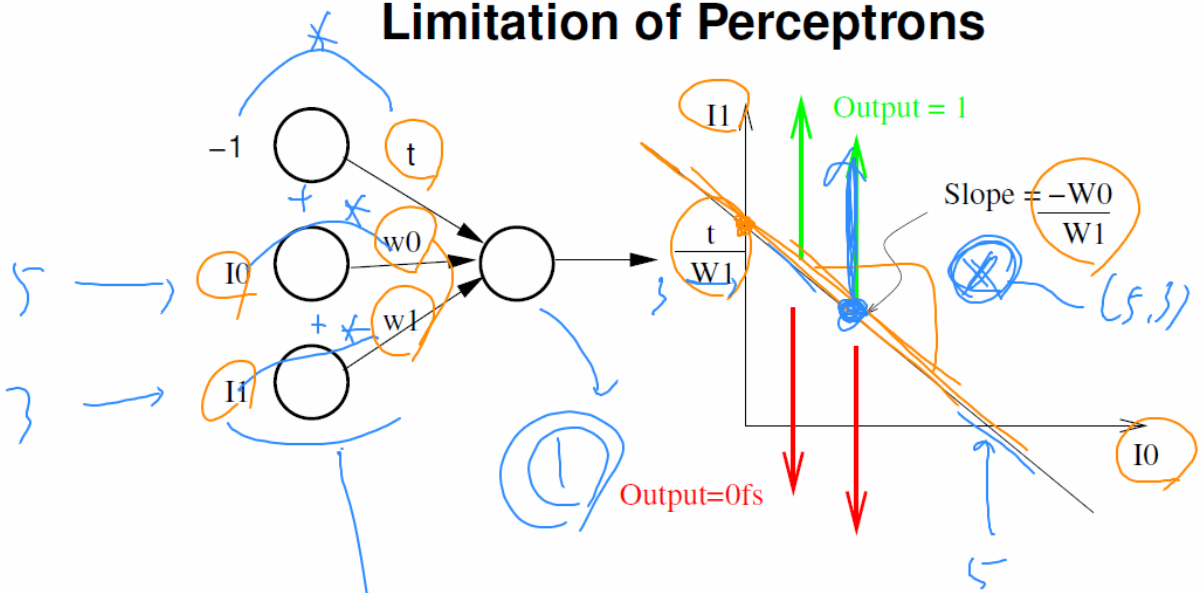
What about XOR or EQUIV?

[a] Same as Russel & Norvig p.570, figure 19.6

The mid 19 eighties to for people to figure out how how this issue
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Limitation of Perceptrons



Perceptrons can only represent **linearly-separable** functions.

- Output of the perceptron:

$$W_0 \times I_0 + W_1 \times I_1 - t \geq 0, \text{ then output is } 1$$

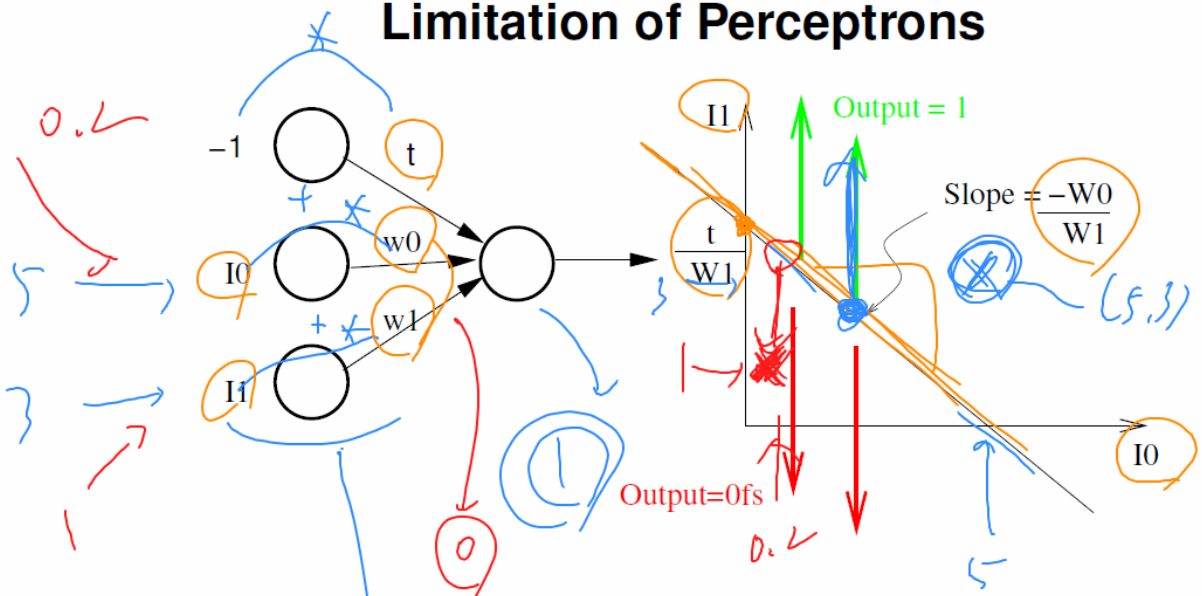
$$W_0 \times I_0 + W_1 \times I_1 - t < 0, \text{ then output is } 0$$

Input, this is input, 5, 3, And if it is all, or above this decision boundary, then this would give you the output of one

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Limitation of Perceptrons



Perceptrons can only represent **linearly-separable** functions.

- Output of the perceptron:

$$W_0 \times I_0 + W_1 \times I_1 - t \geq 0, \text{ then output is } 1$$

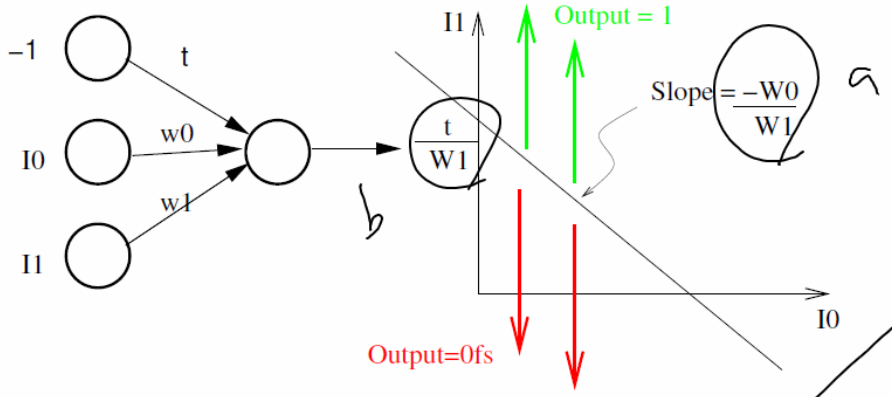
$$W_0 \times I_0 + W_1 \times I_1 - t < 0, \text{ then output is } 0$$

2, and then let's say the input was here and one so if you plug in point 2 and one then because it's below that line the output would be 0

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Limitation of Perceptrons (cont'd)



$W_0 I_0 + W_1 I_1 - t = 0$

$W_1 I_1 = -W_0 I_0 + t$

$I_1 = \frac{-W_0}{W_1} I_0 + \frac{t}{W_1}$

- A geometrical interpretation of this is:

$$I_1 = \frac{-W_0}{W_1} \times I_0 + \frac{t}{W_1},$$

where points on or above the line, the output is 1, and 0 for those below the line (when W_1 is positive). Compare with

$$y = \frac{-W_0}{W_1} \times x + \frac{t}{W_1}.$$

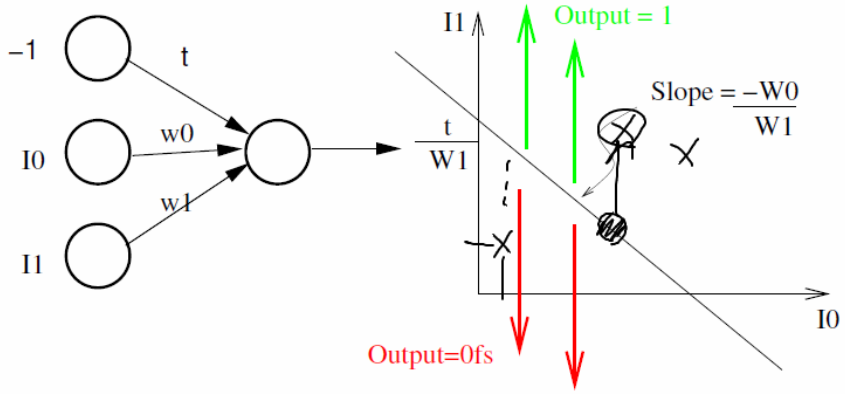
$y = ax + b$

Note: When dividing both sides with W_1, depending on the sign, the inequality can flip its direction (see previous page).

This is P.
You are screen sharing
Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Limitation of Perceptrons (cont'd)



- A geometrical interpretation of this is:

$$I_1 = \frac{-W_0}{W_1} \times I_0 + \frac{t}{W_1},$$

where points on or above the line, the output is 1, and 0 for those below the line (when W_1 is positive). Compare with

$$y = \frac{-W_0}{W_1} \times x + \frac{t}{W_1}.$$

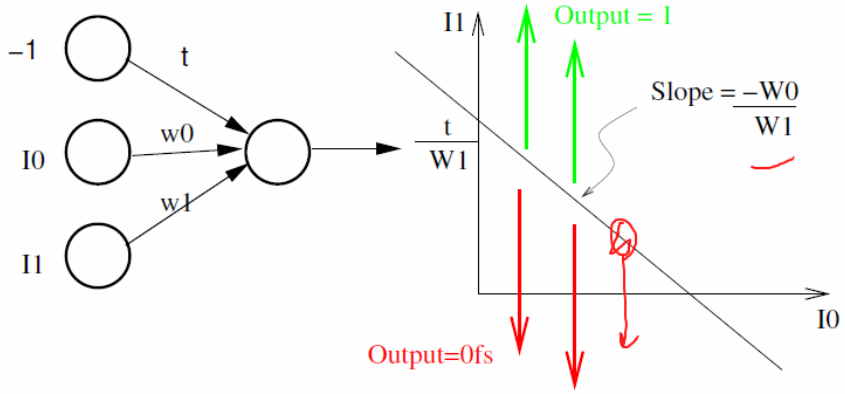
So now, if it is on or above, then it becomes I'll put this one.
If this particular look input is on our above that this
decision boundary is one otherwise, if it is below did the

Note: When dividing both sides with W_1, depending on the sign, the inequality can flip its direction (see previous page).

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Limitation of Perceptrons (cont'd)



- A geometrical interpretation of this is:

$$I_1 = \frac{-W_0}{W_1} \times I_0 + \frac{t}{W_1},$$

where points on or above the line, the output is 1, and 0 for those below the line (when W_1 is positive). Compare with

$$y = \frac{-W_0}{W_1} \times x + \frac{t}{W_1}.$$

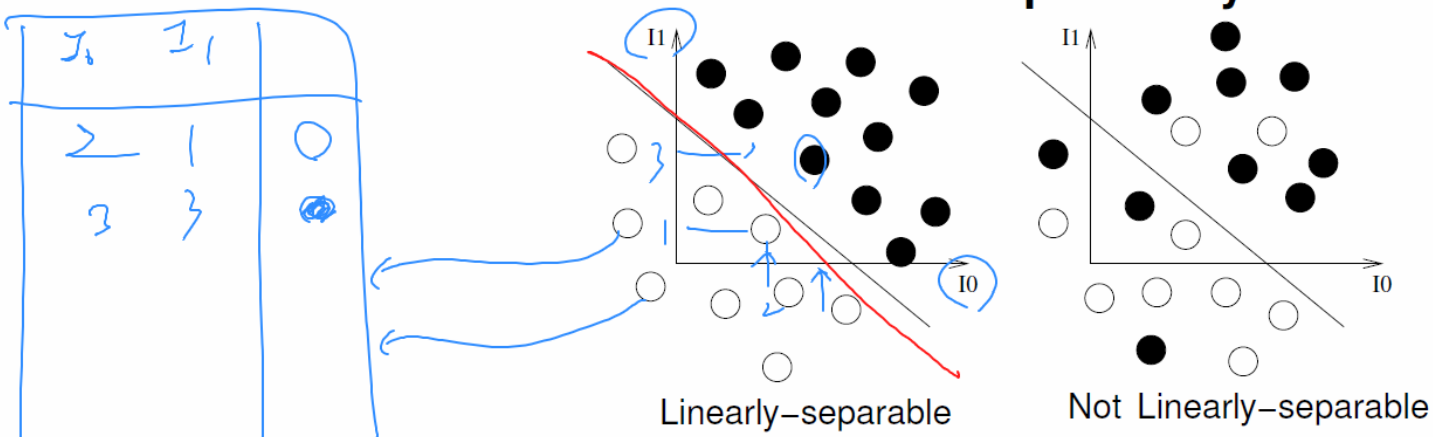
One is positive, because it divided it by W. one. If this was negative, then now everything below, on or below that line is positive. otherwise

Note: When dividing both sides by W_1, depending on the sign, the inequality can flip its direction (see previous page).

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Linear Separability

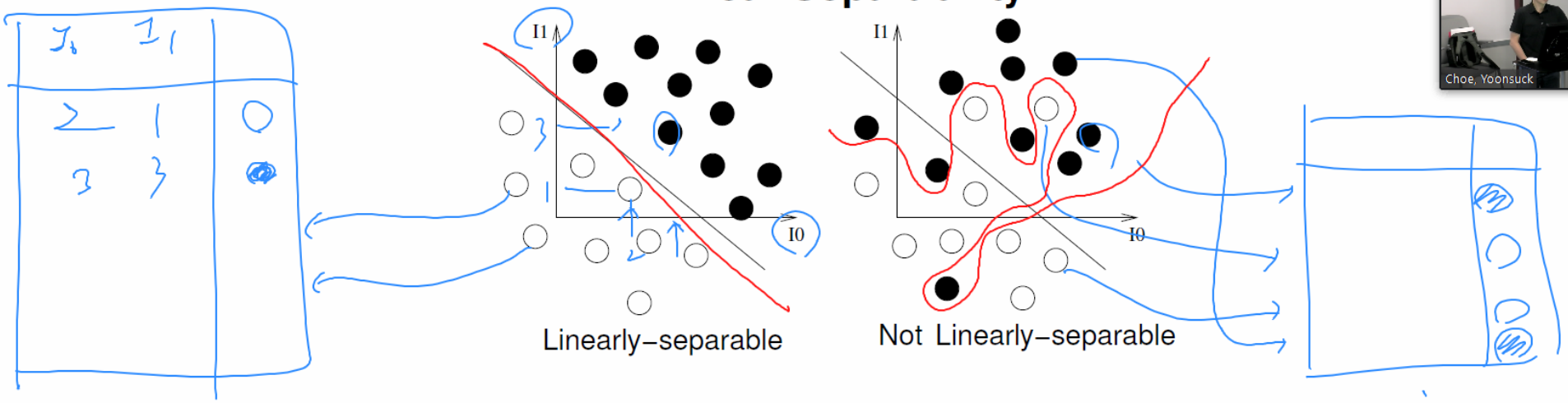


- For functions that take integer or real values as arguments and output either 0 or 1.
- Left: linearly-separable (i.e., can draw a straight line between the classes).
- Right: not linearly-separable (i.e., perceptrons cannot represent such a function)

Okay, then, this is called a linear it's a probable data sets. Well.
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Linear Separability

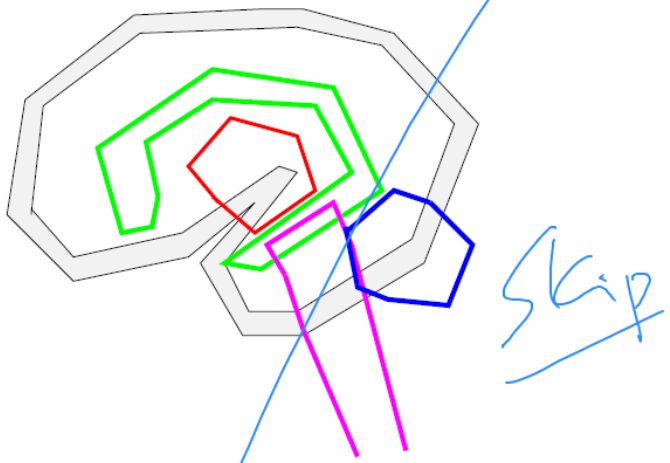


- For functions that take integer or real values as arguments and output either 0 or 1.
- Left: linearly-separable (i.e., can draw a straight line between the classes).
- Right: not linearly-separable (i.e., perceptrons cannot represent such a function)

To separate, clean separate between the the filled discs, and then, yeah, blank discs
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

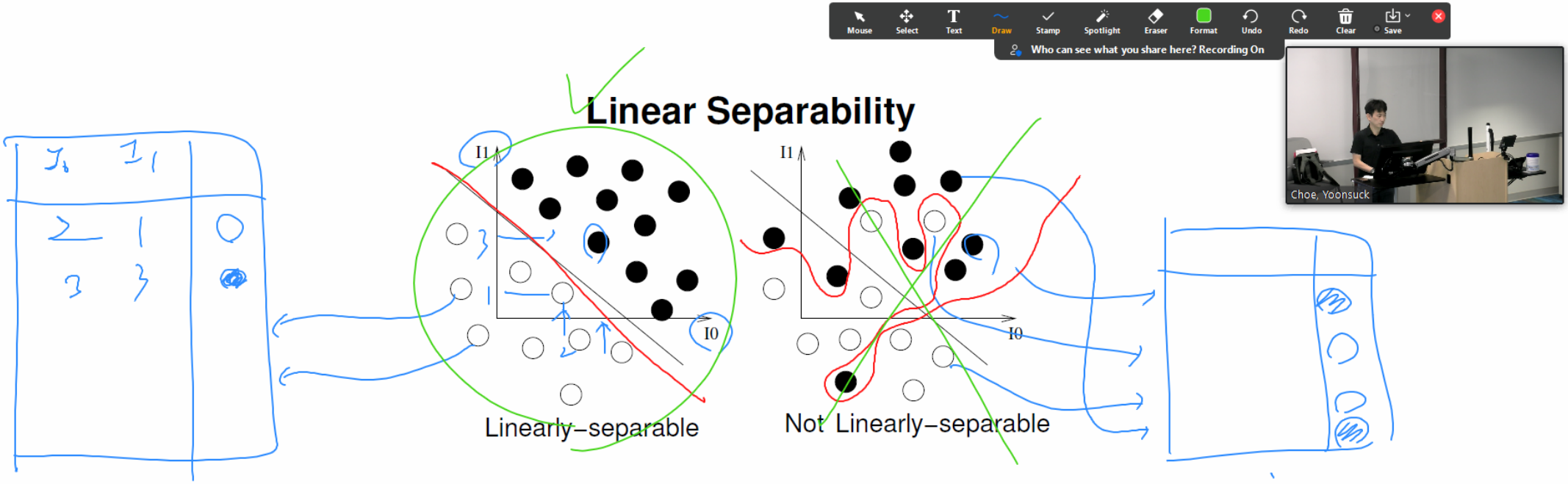
The Central Nervous System



- Cortex: thin outer sheet where most of the neurons are.
- Sub-cortical nuclei: thalamus, hippocampus, basal ganglia, etc.
- Midbrain, pons, and medulla, connects to the spinal cord.
- Cerebellum (hind brain, or small brain)

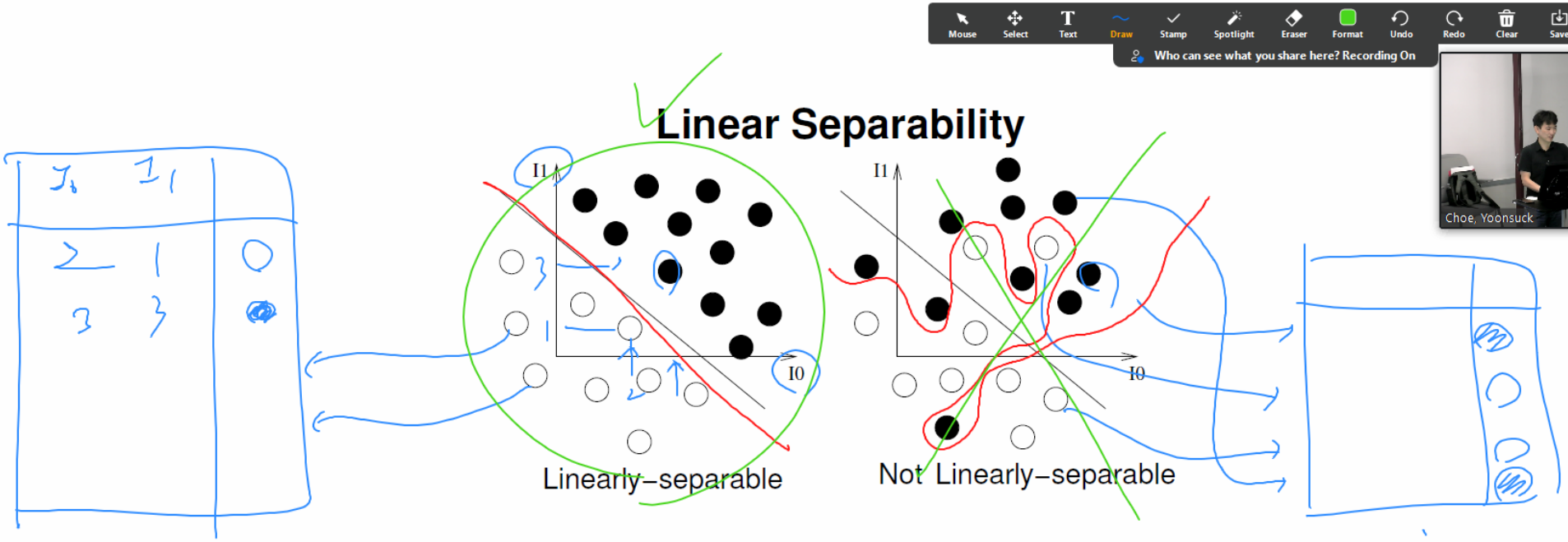
But i'm gonna gonna skip this because this is a little bit too detailed. I'm going to skip it. You can read it

You are screen sharing Stop Share



- For functions that take integer or real values as arguments and output either 0 or 1.
- Left: linearly-separable (i.e., can draw a straight line between the classes).
- Right: not linearly-separable (i.e., perceptrons cannot represent such a function)

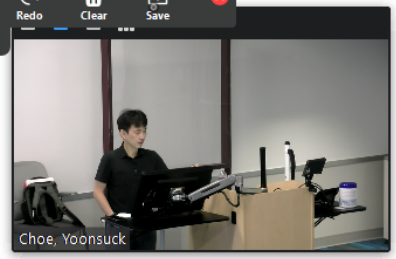
Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On
Choe, Yoonsuck
now know that it can only deal with the tests that have this kind of characteristic, so it cannot deal with any of these details that are not
You are screen sharing Stop Share



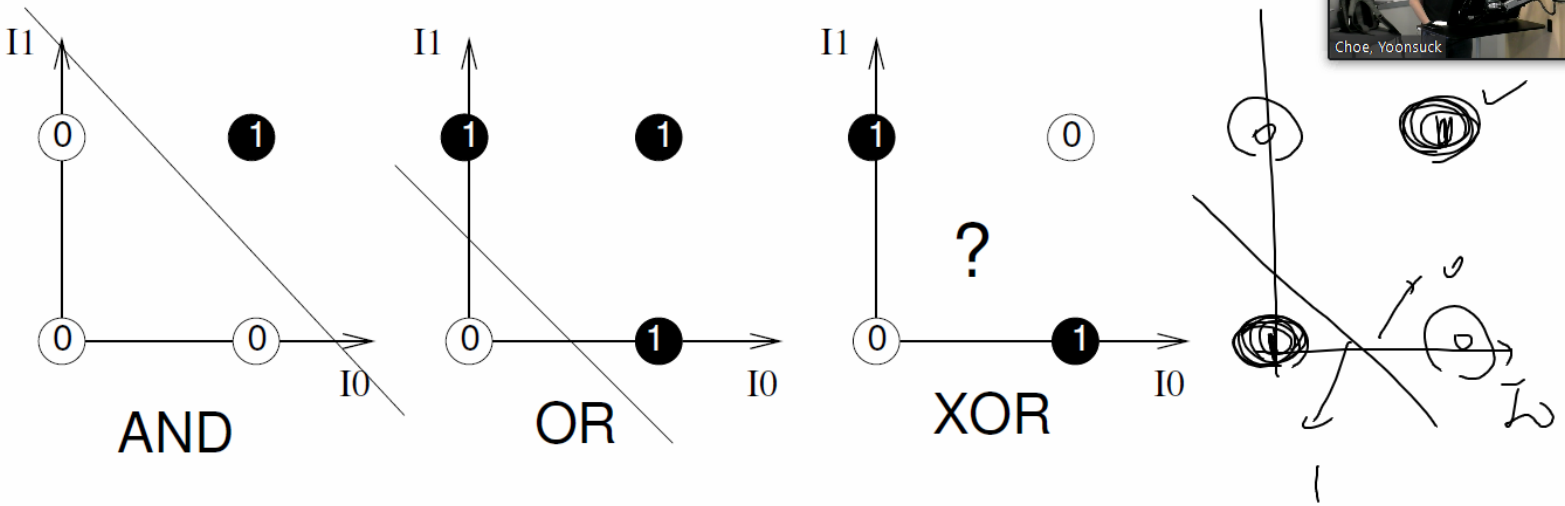
- For functions that take integer or real values as arguments and output either 0 or 1.
- Left: linearly-separable (i.e., can draw a straight line between the classes).
- Right: not linearly-separable (i.e., perceptrons cannot represent such a function)

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On
Choe, Yoonsuck
now know that it can only deal with the tests that have this kind of characteristic, So it cannot, deal with any of these details that are not linear is a probable
You are screen sharing
Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On



Linear Separability (cont'd)

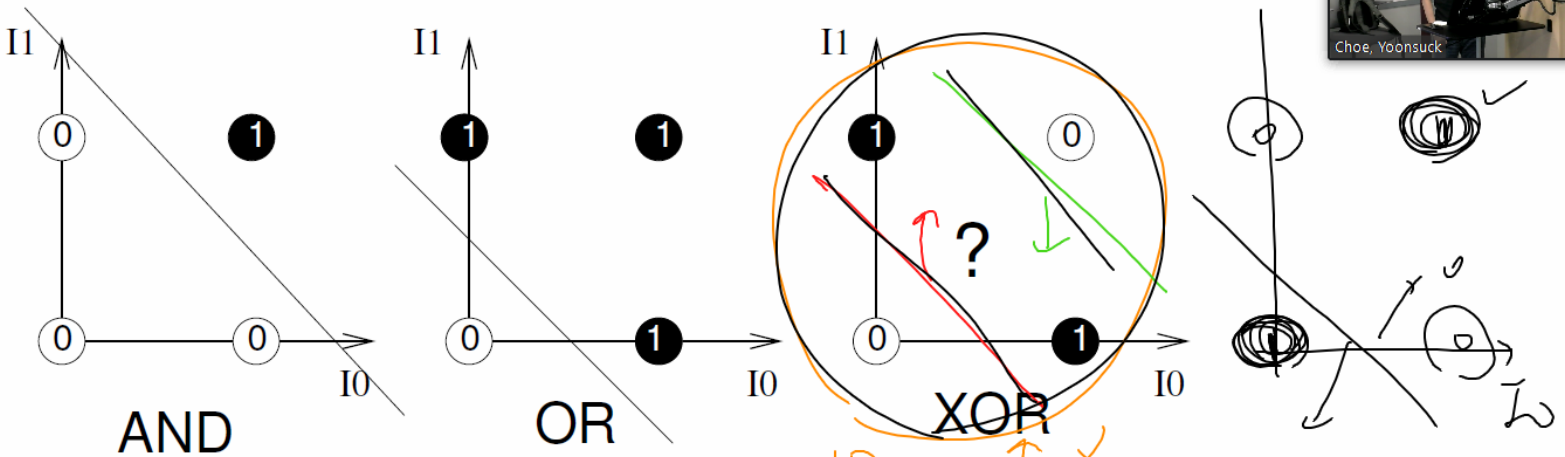


- Perceptrons cannot represent XOR!
- Minsky and Papert (1969)

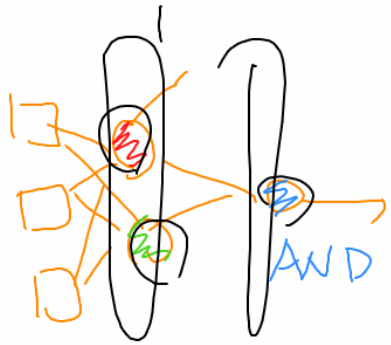
Yeah, and so on. so you cannot throw straight line
You are screen sharing
Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Linear Separability (cont'd)



- Perceptrons cannot represent XOR!
- Minsky and Papert (1969)

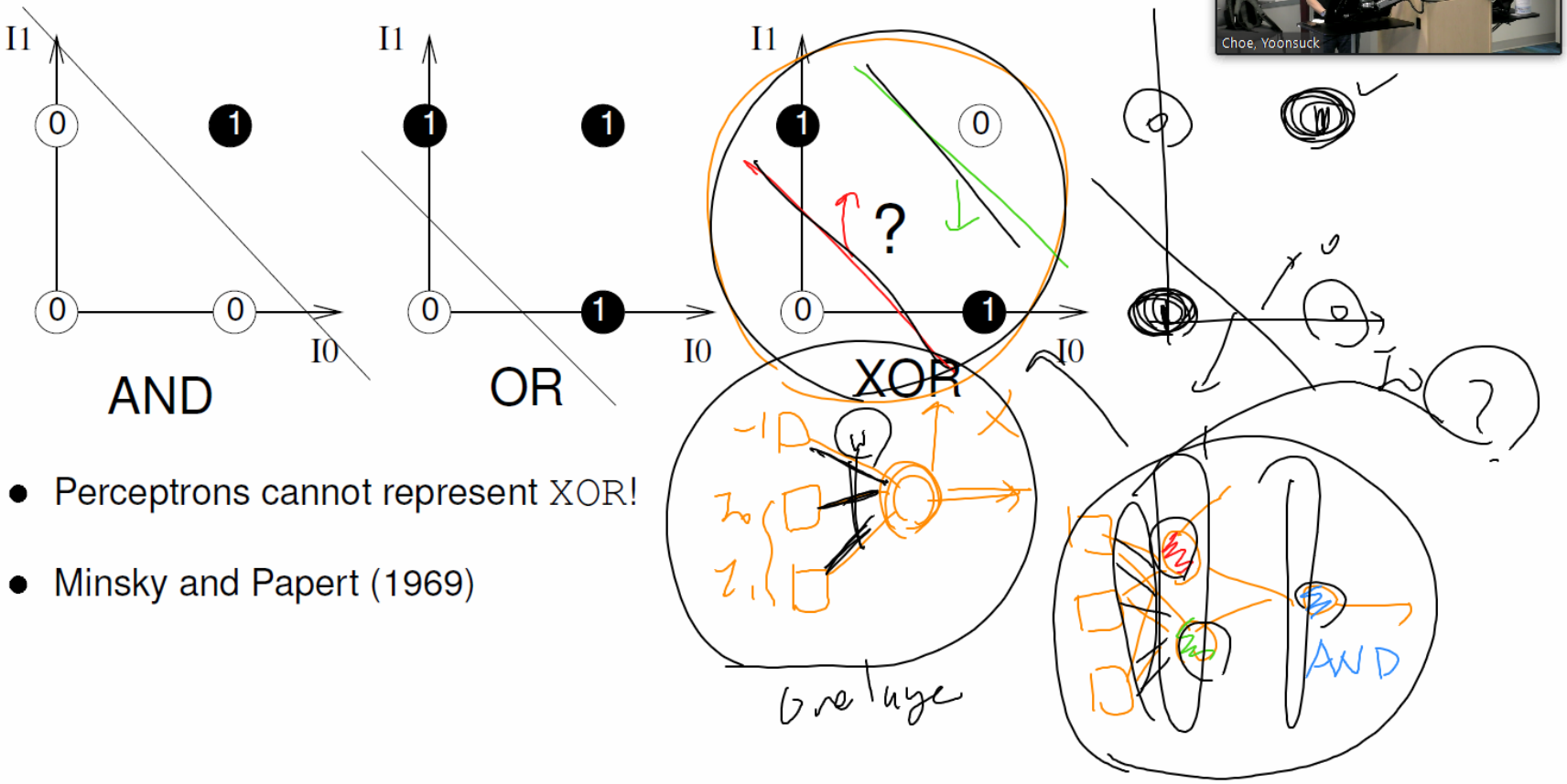


Because it has 2 boundaries. Yeah, that is somehow combined

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Linear Separability (cont'd)



- Perceptrons cannot represent XOR!
- Minsky and Papert (1969)

But they did not know how to update these connection weights, so they kind of abandoned the support

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Learning in Perceptrons

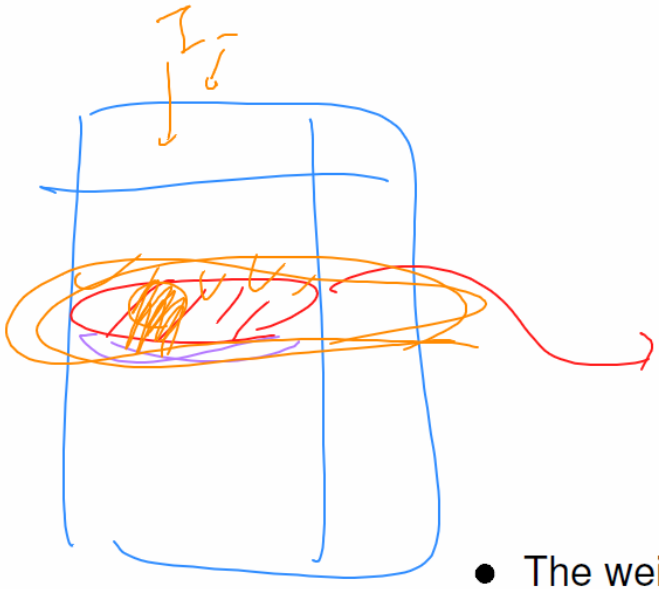
- The weights do not have to be calculated manually.
- We can train the network with (input,output) pair according to the following weight update rule:

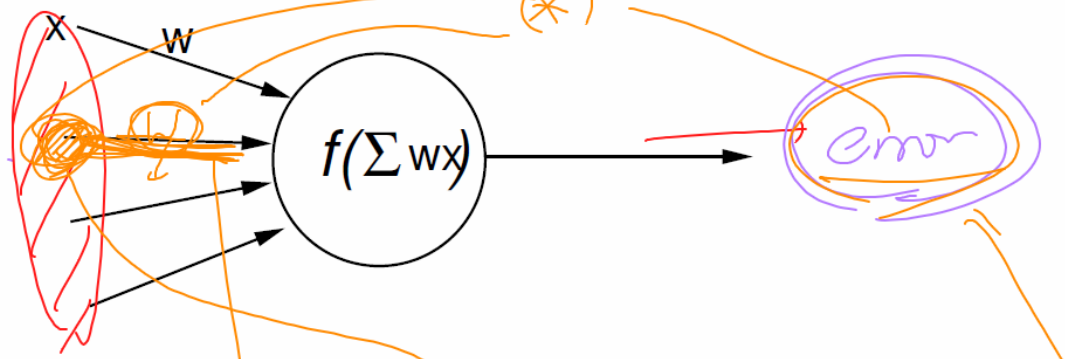
$$W_j \leftarrow W_j + \alpha \times I_j \times Err,$$

where α is the learning rate parameter, I_j is the input, and

$$Err = DesiredOutput - NetworkOutput.$$

Yeah, well, it's applied by us
You are screen sharing Stop Share



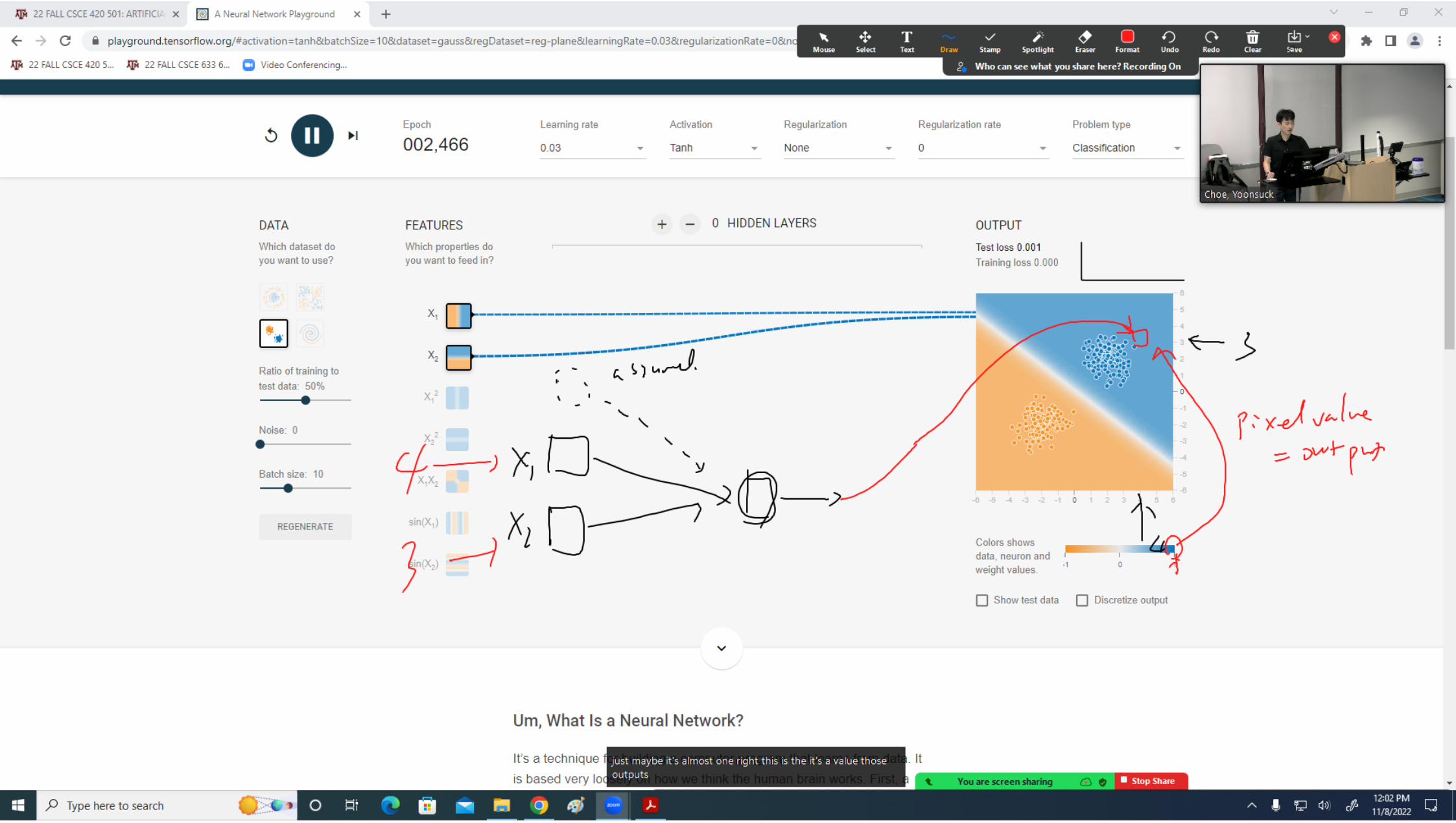


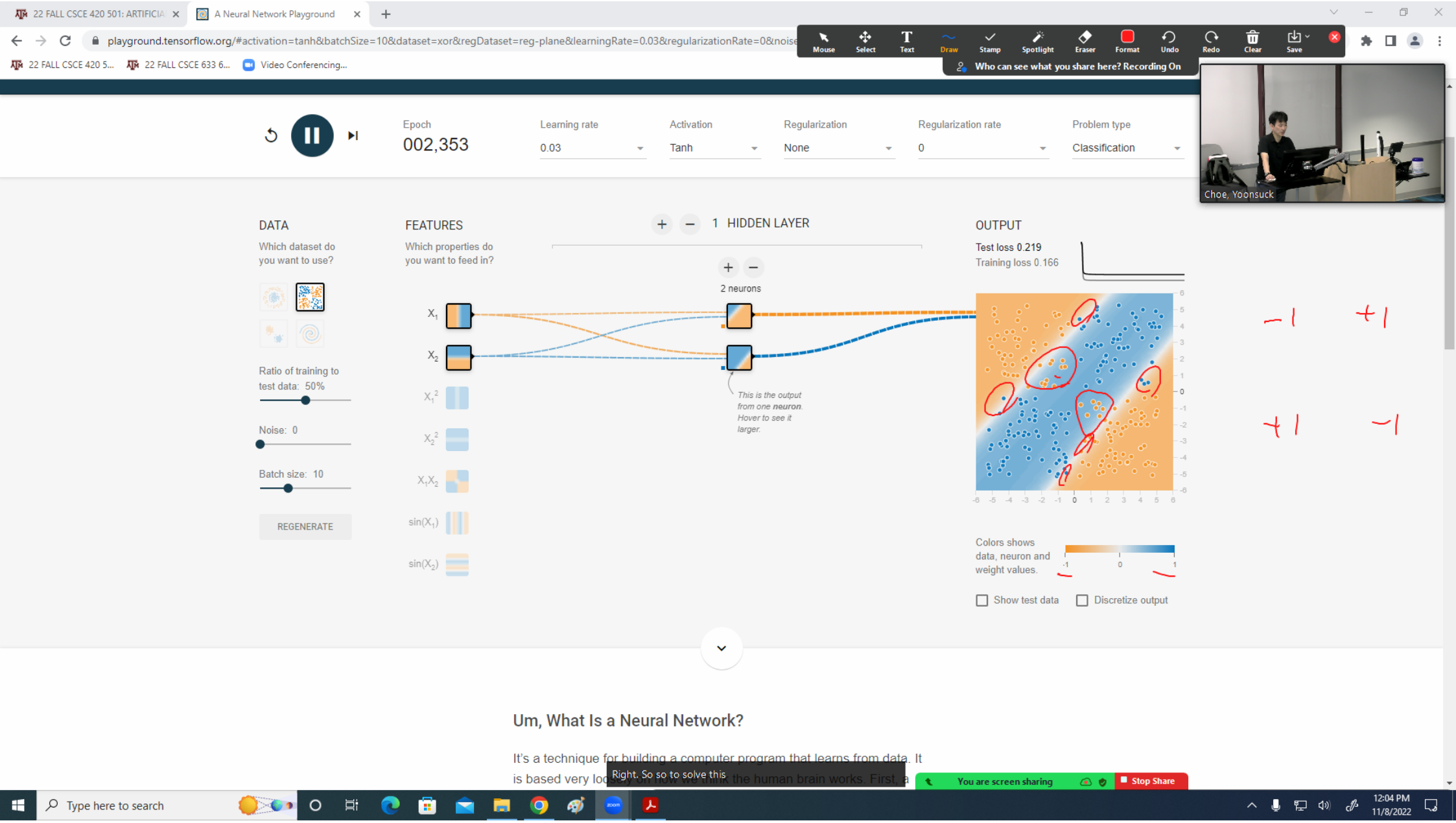
Learning in Perceptrons

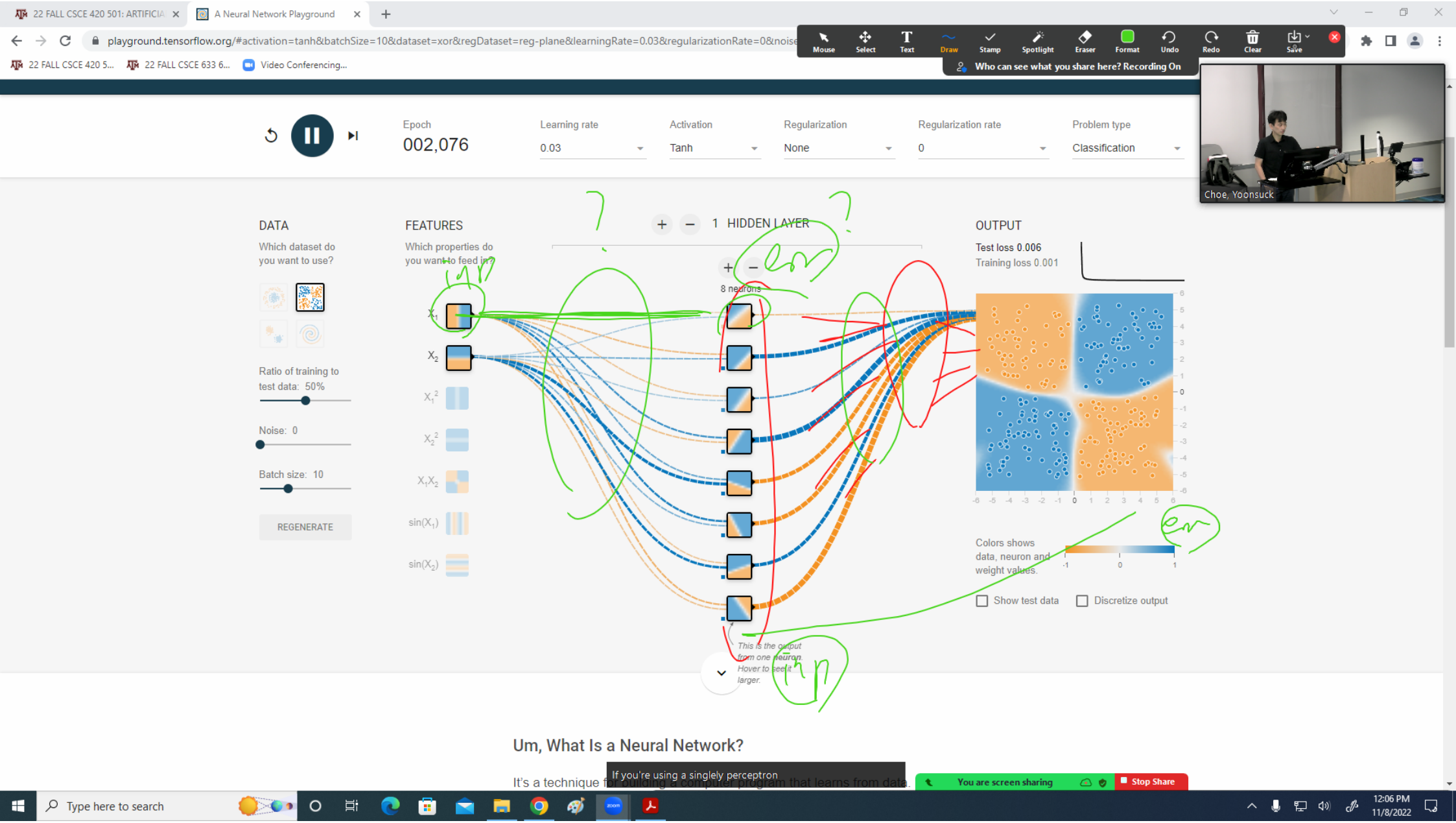
- The weights do not have to be calculated manually.
- We can train the network with (input,output) pair according to the following weight update rule:

$$W_j \leftarrow W_j + \alpha \times I_j \times Err,$$

where α is the learning rate parameter, I_j is the input, and $Err = DesiredOutput - NetworkOutput.$


22 FALL CSCE 420 501: ARTIFICIA
A Neural Network Playground
playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=gauss&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&nc
22 FALL CSCE 420 5...
22 FALL CSCE 633 6...
Video Conferencing...
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
Who can see what you share here? Recording On
Choe, Yoonsuck
Epoch
002,466
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?
0 HIDDEN LAYERS
OUTPUT
Test loss 0.001
Training loss 0.000
Colors shows data, neuron and weight values.
Show test data
Discretize output
pixel value = output
Um, What Is a Neural Network?
It's a technique f
is based very loosely on how we think the human brain works. First, a
just maybe it's almost one right this is the it's a value those outputs
You are screen sharing
Stop Share
Type here to search
12:02 PM
11/8/2022


22 FALL CSCE 420 501: ARTIFICIA
A Neural Network Playground
playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=xor&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
Who can see what you share here? Recording On
22 FALL CSCE 420 5...
22 FALL CSCE 633 6...
Video Conferencing...
Choe, Yoonsuck
Epoch
002,353
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?
1 HIDDEN LAYER
2 neurons
This is the output from one neuron. Hover to see it larger.
OUTPUT
Test loss 0.219
Training loss 0.166
Colors shows data, neuron and weight values.
Show test data
Discretize output
-1 +1
+1 -1
Um, What Is a Neural Network?
It's a technique for building a computer program that learns from data. It is based very loosely on how we think the human brain works. First, a
Right. So so to solve this
You are screen sharing
Stop Share
Type here to search
12:04 PM
11/8/2022


22 FALL CSCE 420 501: ARTIFICIA
A Neural Network Playground
playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=xor&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
Who can see what you share here? Recording On
Choe, Yoonsuck
Epoch
002,076
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?
1 HIDDEN LAYER
8 neurons
OUTPUT
Test loss 0.006
Training loss 0.001
Colors shows data, neuron and weight values.
Show test data
Discretize output
This is the output from one neuron. Hover to see it larger.
Um, What Is a Neural Network?
If you're using a singlely perceptron
It's a technique for building a computer program that learns from data.
You are screen sharing
Stop Share
Type here to search
12:06 PM
11/8/2022

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

The Central Nervous System: Facts

Facts: human brain

- Neocortex Thickness: 1.6mm
- Neocortex Area: 2,500 cm^2 (about 2.5 ft^2: Peters and Jones, 1984)
- Neurons: 86 billion (86×10^9: Frederico Azevedo el al. 2009), 10 billion in the neocortex (Shephered, 1998).
- Connections: 150 trillion (150^{12}: Pakkenberg et al. 1997, 2003). 60 trillion in the neocortex (Shephered, 1998).
- Connections per neuron: $\sim 1,800$ (from above)
- Energy usage per operation: 10^{-16}J (compare to 10^{-6}J in modern computers)

Skip

Unless specified, the numbers are from *Neural Networks: a comprehensive foundation* by Simon Haykin (1994) and *Foundations of Vision* by Brian Wandell (1995).

Where is the area of your cortex? If you spread it out on a flat sheet, and so on.

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Overview

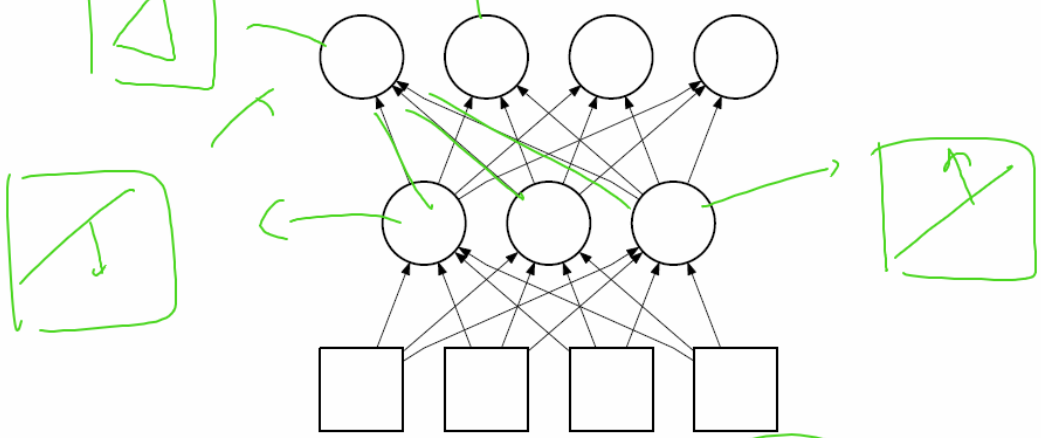
- Multilayer feed-forward networks
- Gradient descent search
- Backpropagation learning rule
- Evaluation of backpropagation
- Applications of backpropagation

gradient descent, which is the basis for the the main learning rule for these multilay annual networks, and then we look at some

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Multilayer Feed-Forward Networks

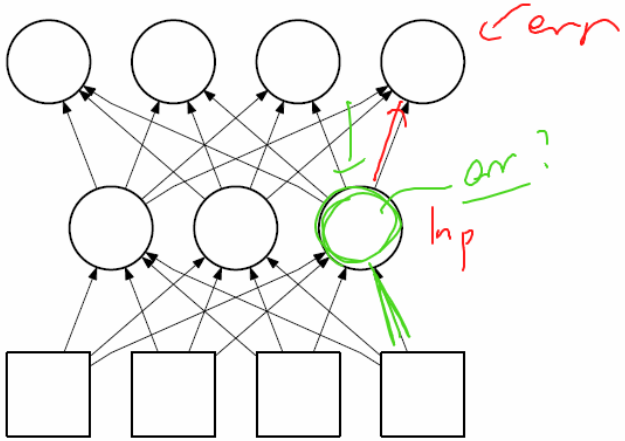


- Proposed in the 1950s
- Proper procedure for training the network came later (1969) and became popular in the 1980s: **back-propagation**

So they found a really cool technique, but it turns out that there were
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

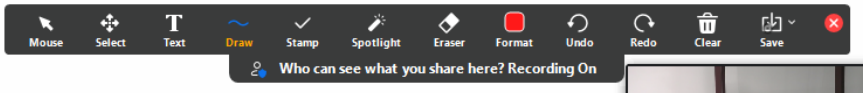
Back-Propagation Learning Rule



- Back-prop is basically a **gradient descent** algorithm.
- The tough problem: output layer has explicit error measure, so finding the error surface is trivial. However, for the hidden layers, how much error each connection eventually cause at the output nodes is hard to determine.
- Backpropagation determines how to distribute the **blame** to each connection.

Unit that is detached from the output side. So this was the tough part

You are screen sharing Stop Share



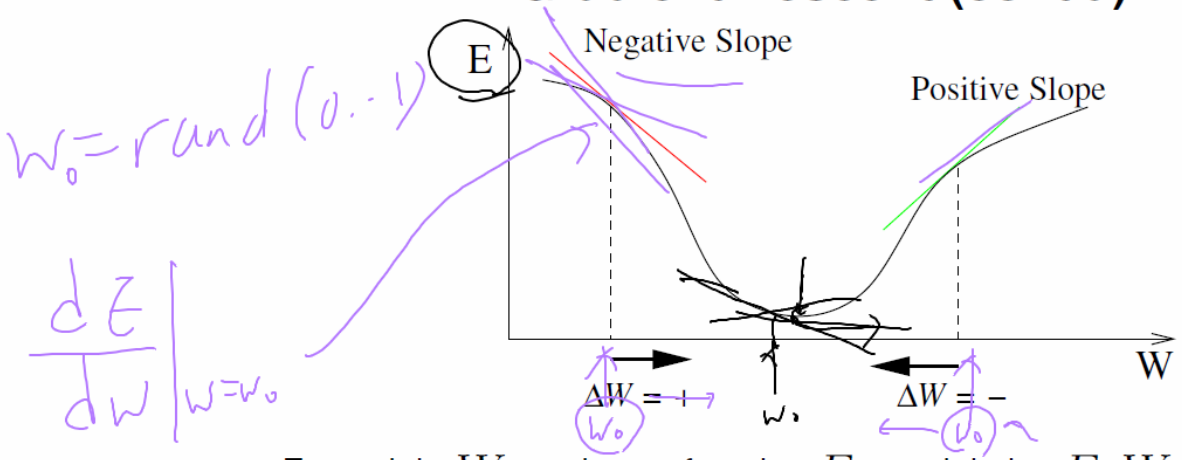
Gradient Descent

- We want to minimize the total error E by tweaking the network weights.
- E depends on W_i, thus by adjusting W_i, you can reduce E.
- Figuring out how to simultaneously adjust weights W_i for all i at once is practically impossible, so use an iterative approach.
- A sensible way is to reduce E with respect to one weight W_i at a time, proportional to the gradient (or slope) at that point.

And then the answer is gradient descent. The idea is to basically find this partial derivative for a particular connection, and then use that to update the question, Wait

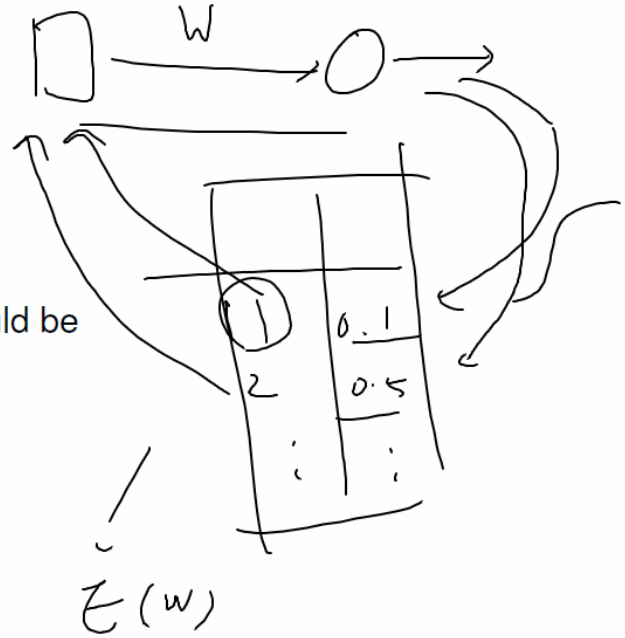
Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Gradient Descent (cont'd)



$W_0 = rand(0..1)$

$\left.\frac{dE}{dW}\right|_{W=W_0}$



- For weight W_i and error function E, to minimize E, W_i should be changed according to $W_i \leftarrow W_i + \Delta W_i$:

$$\Delta W_i = \alpha \times \left(-\frac{\partial E}{\partial W_i}\right),$$

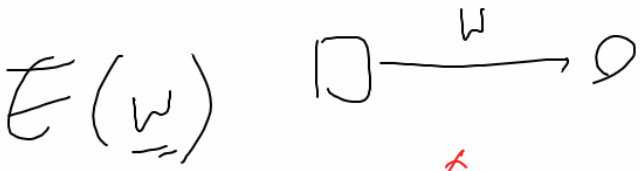
where α is the learning rate parameter.

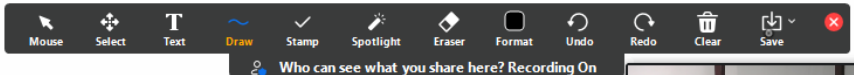
- E can be a function of many weights, thus the partial derivative is used in the above:

$$E(W_1, W_2, \ldots, W_i, \ldots W_n, \ldots)$$

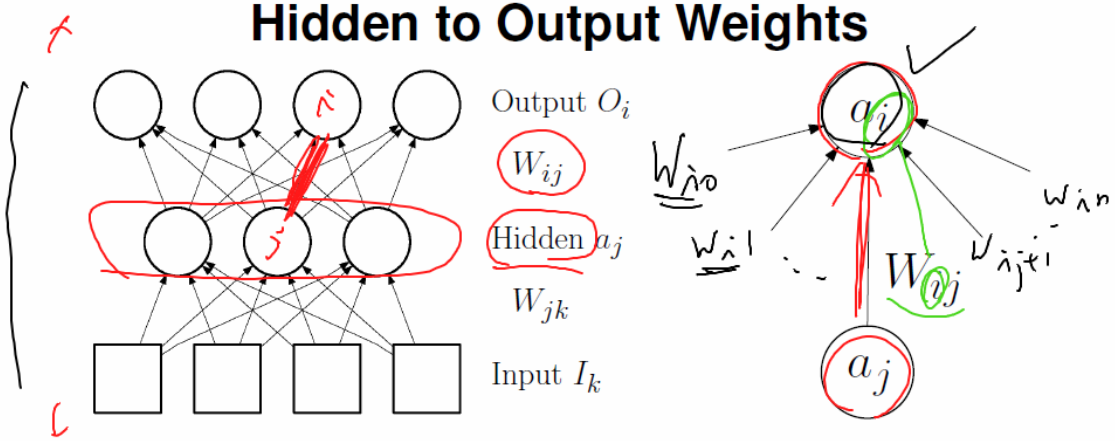
This will become 0 at that point, and your weight will remain unchanged.

You are screen sharing Stop Share





Hidden to Output Weights



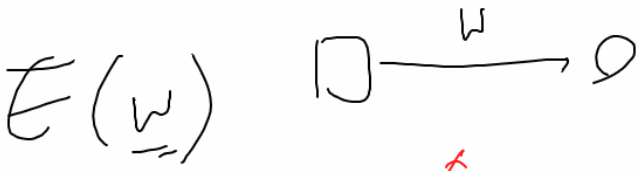
- Error function

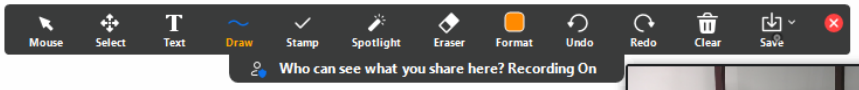
$$
\begin{aligned}
E &= \frac{1}{2}\sum_i (E_i)^2 \\
E_i &= T_i - O_i \\
&= T_i - g\left(\sum_j W_{ij} a_j\right),
\end{aligned}
$$

where $g(\cdot)$ is the sigmoid activation function, and T_i the target.

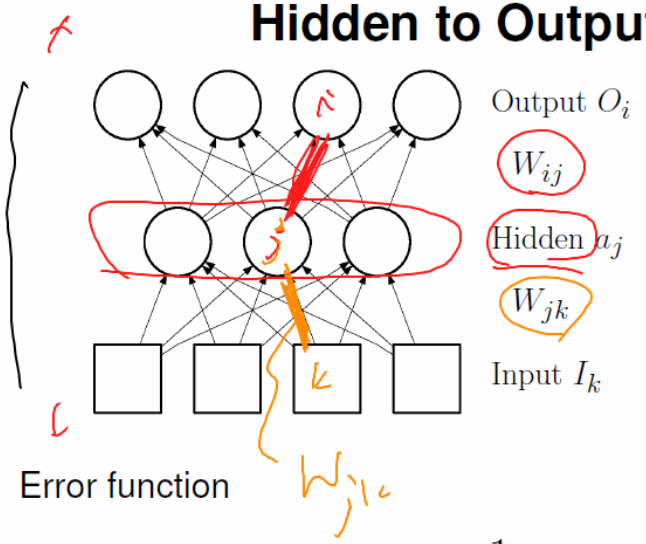
N Hi. So when you look look at this first subscript it tells you that it's I the index I is for the output side of the connection

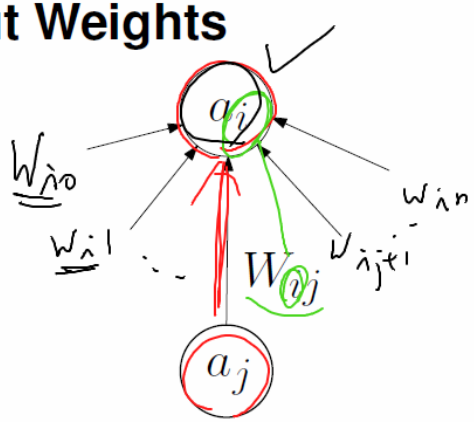
You are screen sharing
Stop Share





Hidden to Output Weights





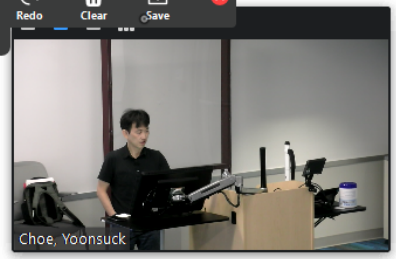
- Error function

$$
\begin{aligned}
E &= \frac{1}{2}\sum_i (E_i)^2 \\
E_i &= T_i - O_i \\
&= T_i - g\left(\sum_j W_{ij} a_j\right),
\end{aligned}
$$

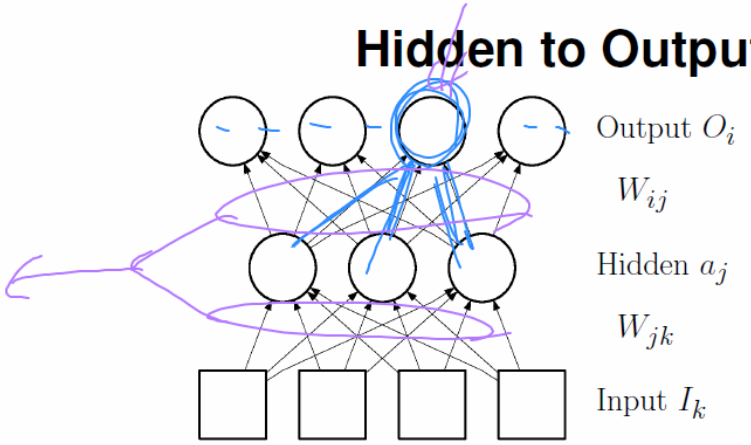
where $g(\cdot)$ is the sigmoid activation function, and T_i the target.

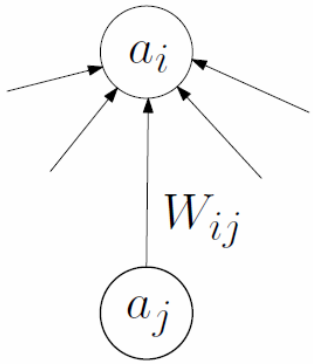
We just assume that there is this kind of nice shaped
You are screen sharing
Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On



Hidden to Output Weights





- Error function

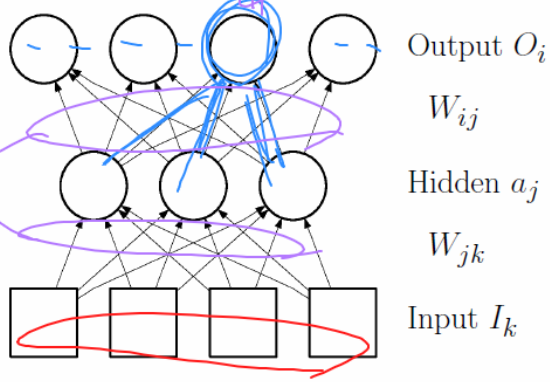
$$
\begin{aligned}
E &= \frac{1}{2}\sum_{i \in D} (E_i)^2 \\
E_i &= T_i - O_i \\
&= T_i - g\left(\sum_j W_{ij} a_j\right),
\end{aligned}
$$

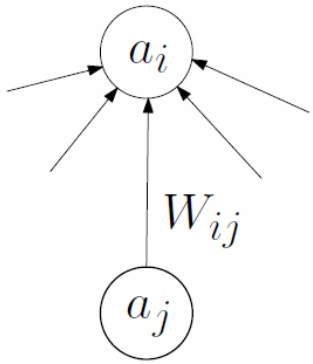
where $g(\cdot)$ is the sigmoid activation function, and T_i the target.

Okay.
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Hidden to Output Weights





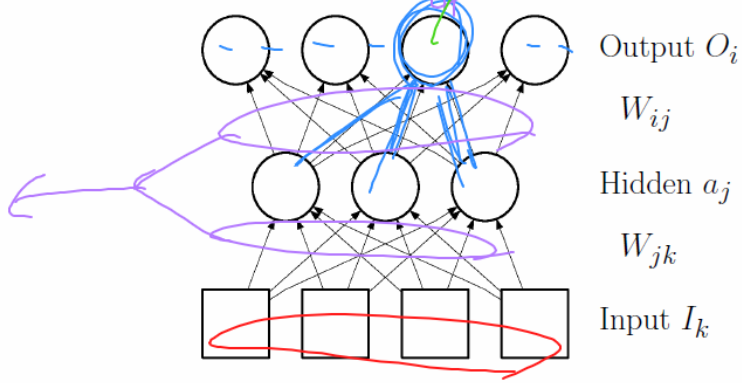
- Error function

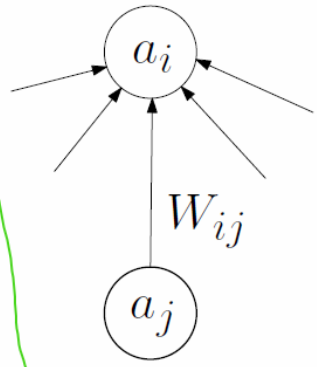
$$
\begin{aligned}
E &= \frac{1}{2}\sum_i (E_i)^2 \\
E_i &= T_i - O_i \\
&= T_i - g\left(\sum_j W_{ij} a_j\right),
\end{aligned}
$$

where $g(\cdot)$ is the sigmoid activation function, and T_i the target.

Yeah, let's talk and then
You are screen sharing Stop Share

Hidden to Output Weights





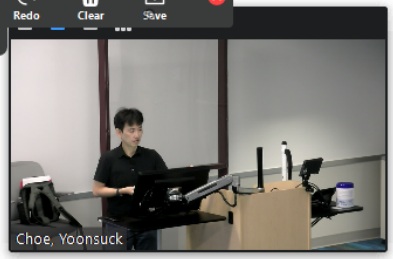
- Error function

$$
\begin{aligned}
E &= \frac{1}{2}\sum_i (E_i)^2 \\
E_i &= T_i - O_i \\
&= T_i - g\left(\sum_j W_{ij} a_j\right),
\end{aligned}
$$

where $g(\cdot)$ is the sigmoid activation function, and T_i the target.

So this is plugging the input. Get output and then look up the target

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On



How the Brain Differs from Computers

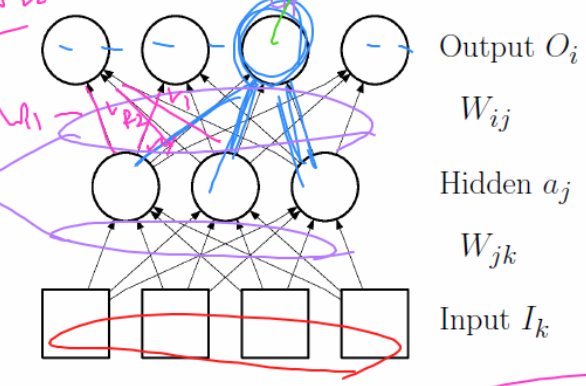
- Densely connected.
- Massively parallel.
- Highly nonlinear.
- Asynchronous: no central clock.
- Fault tolerant.
- Highly adaptable.
- Creative.

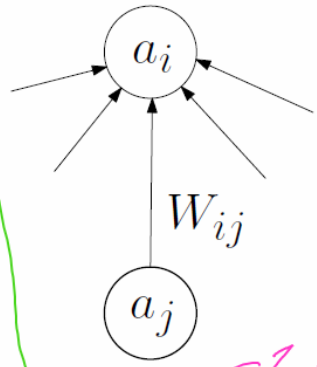
Why are these crucial?

skip

And here's how the brain differs from computers so again I'm gonna just skip this
You are screen sharing Stop Share

Hidden to Output Weights





- Error function

$$E = \frac{1}{2}\sum_i (E_i)^2$$

$$E_i = T_i - O_i$$

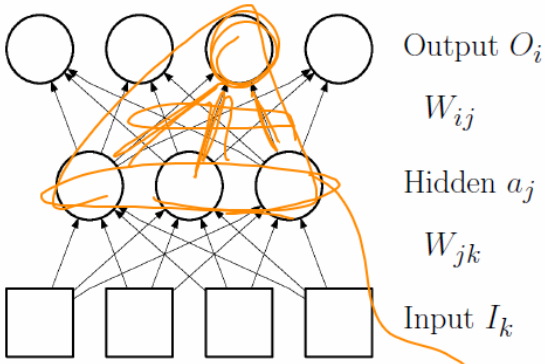
$$= T_i - g\left(\sum_j W_{ij} a_j\right),$$

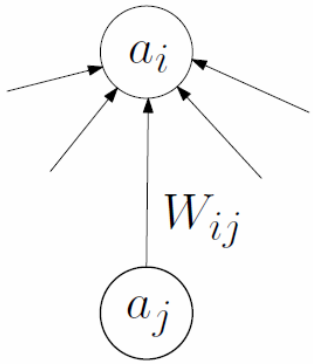
where $g(\cdot)$ is the sigmoid activation function, and T_i the target.

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On
Choe, Yoonsuck
2, one, sorry W. 2, one W. 2, and so on.
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Hidden to Output Weights





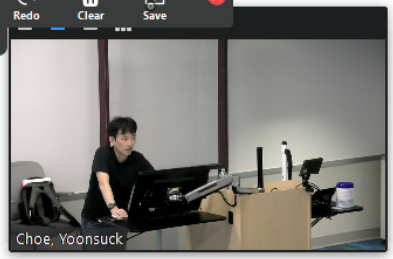
- Error function

$$
\begin{aligned}
E &= \frac{1}{2}\sum_i (E_i)^2 \\
E_i &= T_i - O_i \\
&= T_i - g\left(\sum_j W_{ij} a_j\right),
\end{aligned}
$$

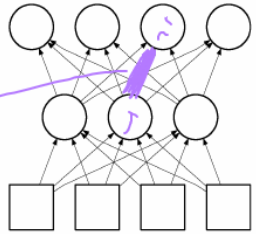
where $g(\cdot)$ is the sigmoid activation function, and T_i the target.

And then the input so here we're just looking at 1 one of these, then you pass it through the activation function
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On



Hidden to Output Weights (cont'd)



$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial \left(\frac{1}{2} \sum_i \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}}$$

$$= \frac{\partial \left(\frac{1}{2} \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}}$$

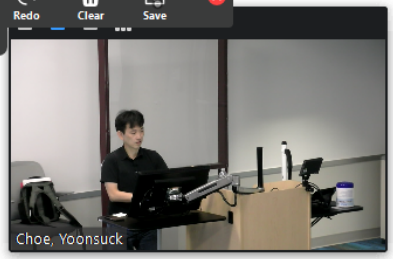
$$= (T_i - O_i) \times \left(-\frac{\partial g(\sum_j W_{ij} a_j)}{\partial W_{ij}} \right)$$

$$= -(T_i - O_i) \times g'(\sum_j W_{ij} a_j) \times a_j$$

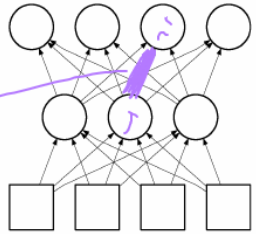
Then this would be w. I. J. So you find the partial derivative of the error function, and this is from the previous page

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On



Hidden to Output Weights (cont'd)



from the prev. page

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial \left(\frac{1}{2} \sum_i \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}}$$

$$= \frac{\partial \left(\frac{1}{2} \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}}$$

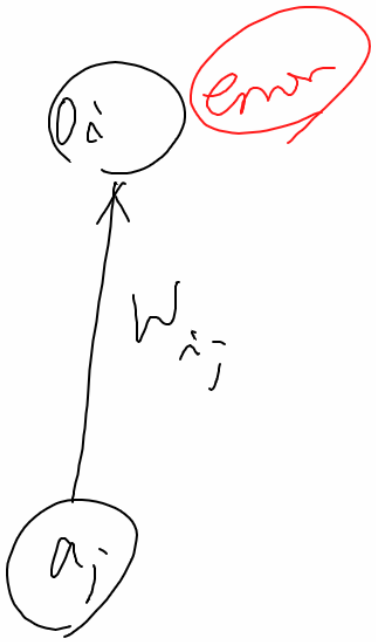
$$= (T_i - O_i) \times \left(-\frac{\partial g(\sum_j W_{ij} a_j)}{\partial W_{ij}} \right)$$

$$= -(T_i - O_i) \times g'(\sum_j W_{ij} a_j) \times a_j$$

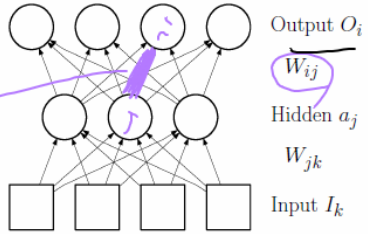
derivation skipped!

So the derivation skipped

You are screen sharing Stop Share



Hidden to Output Weights (cont'd)



$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial \left(\frac{1}{2} \sum_i \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}}$$

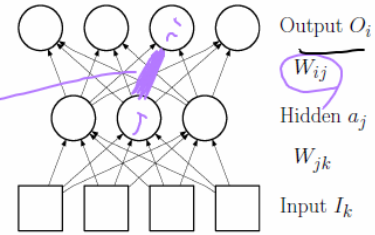
$$= \frac{\partial \left(\frac{1}{2} \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}}$$

$$= (T_i - O_i) \times \left(-\frac{\partial g(\sum_j W_{ij} a_j)}{\partial W_{ij}} \right)$$

$$= -(T_i - O_i) \times g'(\sum_j W_{ij} a_j) \times a_j$$

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On
Choe, Yoonsuck
That you want to update. So you compute the error so that's
You are screen sharing Stop Share

Hidden to Output Weights (cont'd)



$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial \left(\frac{1}{2} \sum_i \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}}$$

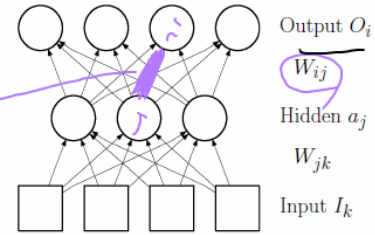
$$= \frac{\partial \left(\frac{1}{2} \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}}$$

$$= (T_i - O_i) \times \left(- \frac{\partial g(\sum_j W_{ij} a_j)}{\partial W_{ij}} \right)$$

$$= -(T_i - O_i) \times g'\left(\sum_j W_{ij} a_j \right) \times a_j$$

So the weight is updated based on delta of I, and then the input of J.

Hidden to Output Weights (cont'd)



$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial \left(\frac{1}{2} \sum_i \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}}$$

$$= \frac{\partial \left(\frac{1}{2} \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}}$$

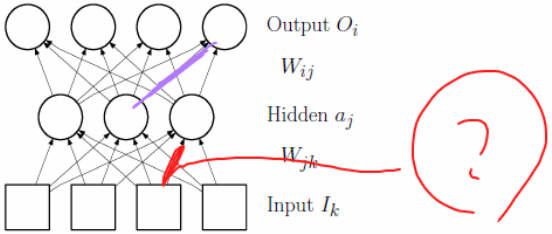
$$= (T_i - O_i) \times \left(-\frac{\partial g(\sum_j W_{ij} a_j)}{\partial W_{ij}} \right)$$

$$= -(T_i - O_i) \times g'\left(\sum_j W_{ij} a_j\right) \times a_j$$

Right. The Arrow times, the input today

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Hidden to Output Weights (cont'd)



$$= -\alpha \frac{\partial E}{\partial W_{ij}} = \frac{\partial \left(\frac{1}{2} \sum_i \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}}$$

$$= \frac{\partial \left(\frac{1}{2} \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}}$$

$$= (T_i - O_i) \times \left(-\frac{\partial g(\sum_j W_{ij} a_j)}{\partial W_{ij}} \right)$$

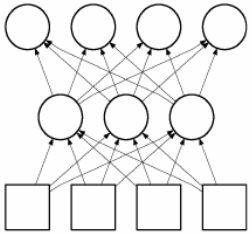
$$= -(T_i - O_i) \times g'(\sum_j W_{ij} a_j) \times a_j$$

But the real question is, what about these connections and we'll look at that in the next lecture right so

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

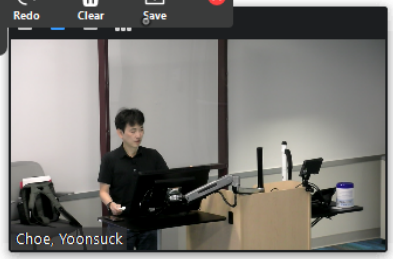
Hidden to Output Weights (cont'd)



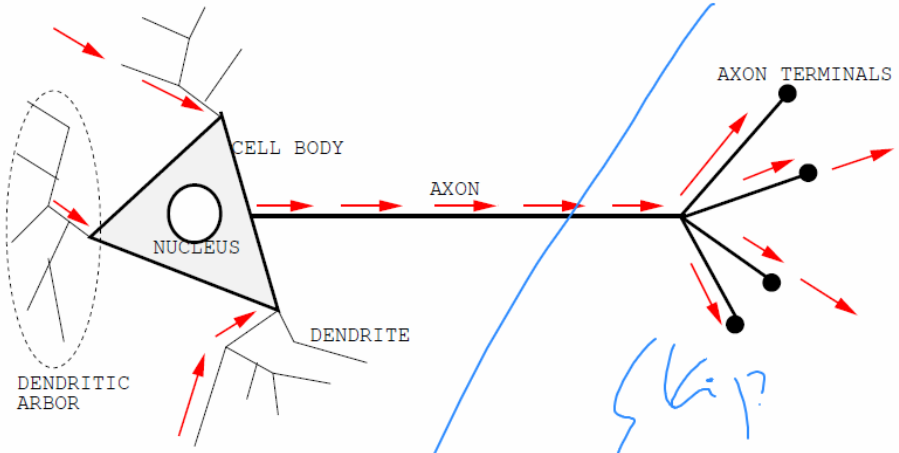
$$
\begin{aligned}
\frac{\partial E}{\partial W_{ij}} &= \frac{\partial \left(\frac{1}{2} \sum_i \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}} \\
&= \frac{\partial \left(\frac{1}{2} \left(T_i - g\left(\sum_j W_{ij} a_j \right) \right)^2 \right)}{\partial W_{ij}} \\
&= (T_i - O_i) \times \left(- \frac{\partial g(\sum_j W_{ij} a_j)}{\partial W_{ij}} \right) \\
&= -(T_i - O_i) \times g'(\sum_j W_{ij} a_j) \times a_j
\end{aligned}
$$

Please sign it and bring it to the podium. Please
You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On



Neurons: Basic Functional Unit of the Brain



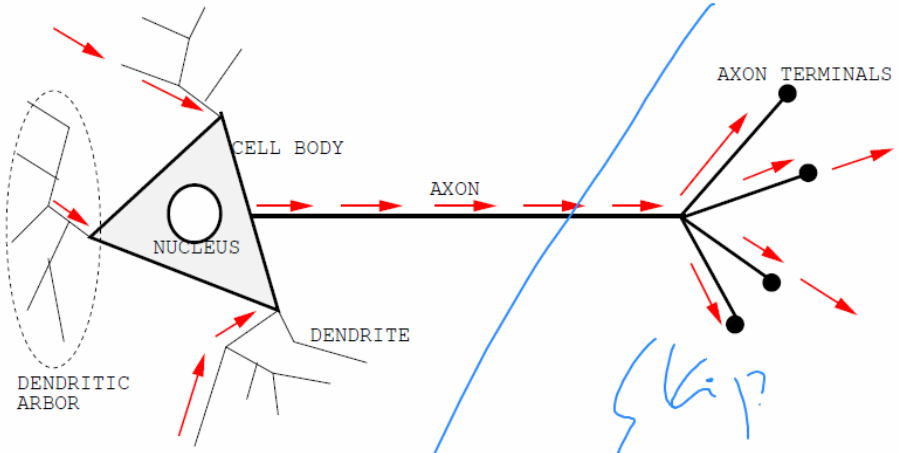
- Dendrites receive input from upstream neurons.
- Ions flow in to make the cell positively charged.
- Once a firing threshold is reached, a spike is generated and transmitted along the axon.
- Axon terminals release neurotransmitters to relay the signal to the downstream neurons.

So the next one is about how the neuron functions. So you can skip

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Neurons: Basic Functional Unit of the Brain



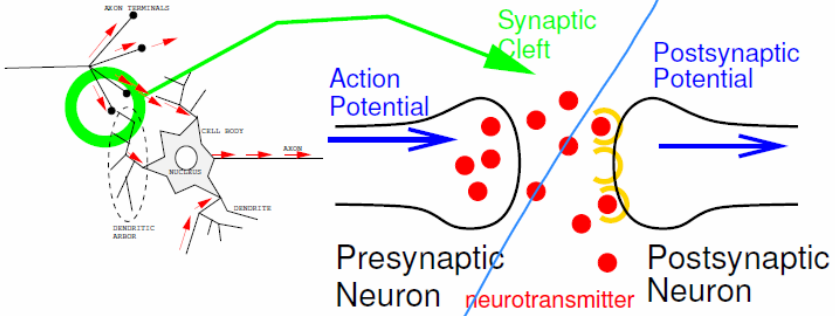
- Dendrites receive input from upstream neurons.
- Ions flow in to make the cell positively charged.
- Once a firing threshold is reached, a spike is generated and transmitted along the axon.
- Axon terminals release neurotransmitters to relay the signal to the downstream neurons.

Then this neuron or unit.

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Propagation of Activation Across the Synapse



1. Action potential reaches axon terminal.
2. Neurotransmitters are released into synaptic cleft and bind to postsynaptic cell's receptors.
3. Binding allows ion channels to open (Na^+), and Na^+ ions flows in and makes the postsynaptic cell depolarize.
4. Once the membrane voltage reaches the threshold, an action potential is generated.

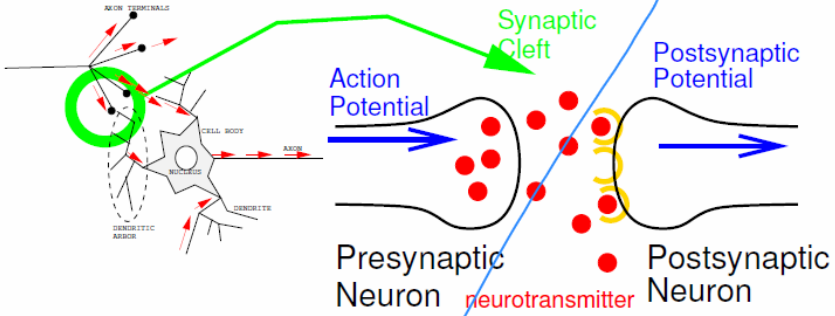
Lesson: neural activity propagation has a very complex cellular/molecular mechanism.

And again here's some more good detail

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On

Propagation of Activation Across the Synapse



1. Action potential reaches axon terminal.
2. Neurotransmitters are released into synaptic cleft and bind to postsynaptic cell's receptors.
3. Binding allows ion channels to open (Na^+), and Na^+ ions flows in and makes the postsynaptic cell depolarize.
4. Once the membrane voltage reaches the threshold, an action potential is generated.

Lesson: neural activity propagation has a very complex cellular/molecular mechanism.

Send them rain, and it's all 10 bananas and so on.
You are screen sharing Stop Share