CS540 Introduction to Artificial Intelligence

Lecture 3

Young Wu

Based on lecture slides by Jerry Zhu, Yingyu Liang, and Charles Dyer

June 29, 2022

Two-thirds of the Average Game

Quiz

A1

- Pick an integer between 0 and 100 (including 0 and 100) that is the closest to two-thirds of the average of the numbers other people picked.
- The results from the previous lecture is posted on the $Q1$ page of the course website.

wrong stats need to update

AND Operator Data

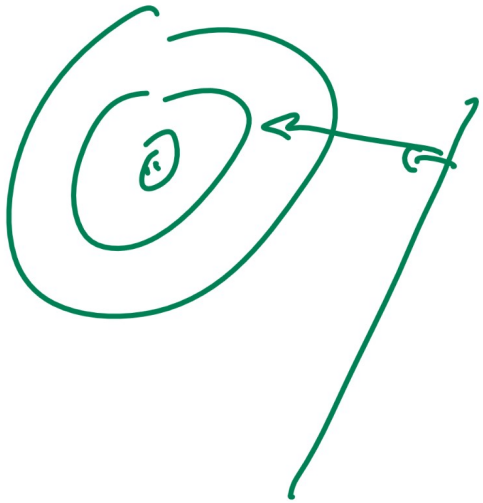
Quiz

$2 \cdot \text{Rand} - 1$

$(0,1) \rightarrow (-1,1)$

- Sample data for AND

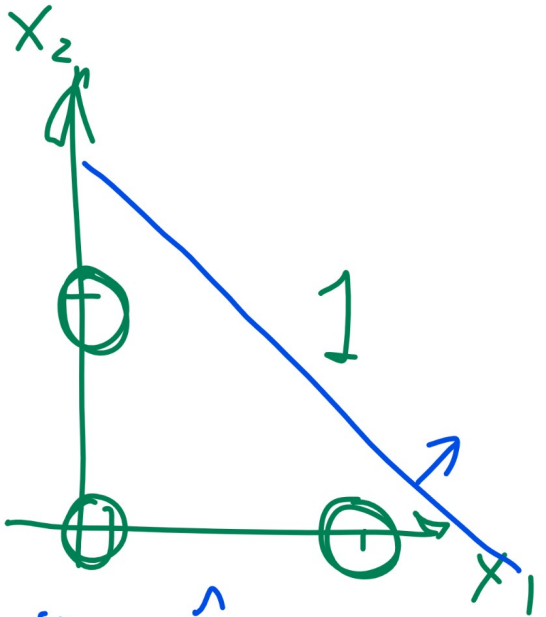
x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1



Learning AND Operator

Quiz

- Which one of the following is AND?
- $A : \hat{y} = \mathbb{1}_{\{1x_1+1x_2-1.5\geqslant 0\}}$
- $B : \hat{y} = \mathbb{1}_{\{1x_1+1x_2-0.5\geqslant 0\}}$
- $C : \hat{y} = \mathbb{1}_{\{-1x_1+0.5\geqslant 0\}}$
- $D : \hat{y} = \mathbb{1}_{\{-1x_1-1x_2+0.5\geqslant 0\}}$
- E : None of the above



x_{i1}	x_{i2}	y_i	$\hat{y}_A$
0	0	0	$\mathbb{1}_{\{-1.5\geqslant 0\}} = 0$
0	1	0	$\mathbb{1}_{\{-0.5\geqslant 0\}} = 0$
1	0	0	0
1	1	1	$\mathbb{1}_{\{0.5\geqslant 0\}} = 1$

OR Operator Data

Quiz

- Sample data for OR

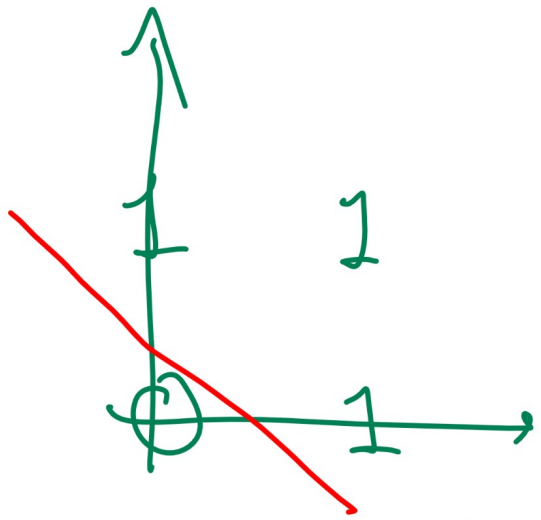
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

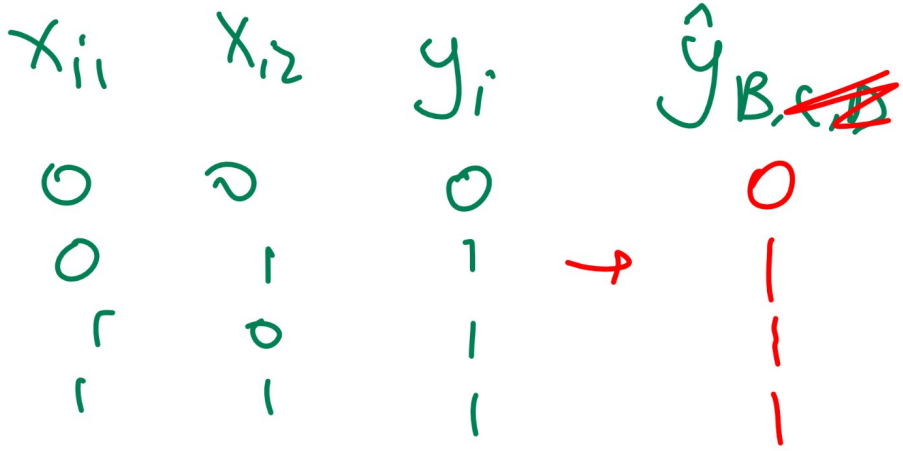
Learning OR Operator

Quiz

Q2

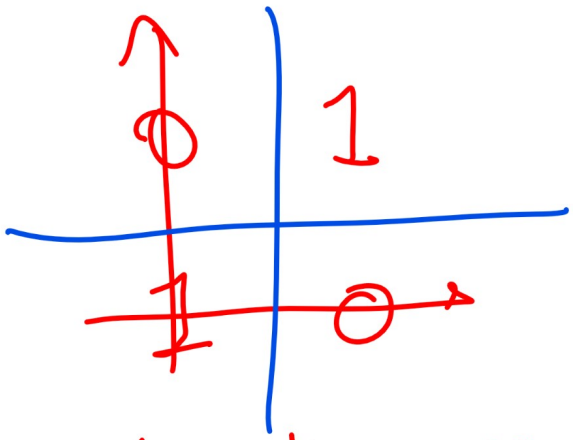
- Which one of the following is OR?
- ~~A~~ : $\hat{y} = \mathbb{1}_{\{1x_1+1x_2-1.5\geqslant 0\}}$
- B : $\hat{y} = \mathbb{1}_{\{1x_1+1x_2-0.5\geqslant 0\}}$
- ~~C~~ : $\hat{y} = \mathbb{1}_{\{-1x_1+0.5\geqslant 0\}}$
- D : $\hat{y} = \mathbb{1}_{\{-1x_1-1x_2+0.5\geqslant 0\}}$
- E : None of the above





Learning XOR Operator

Quiz



- Which one of the following is XOR?
- $A : \hat{y} = \mathbb{1}_{\{1x_1+1x_2-1.5\geqslant 0\}}$
- $B : \hat{y} = \mathbb{1}_{\{1x_1+1x_2-0.5\geqslant 0\}}$
- $C : \hat{y} = \mathbb{1}_{\{-1x_1+0.5\geqslant 0\}}$
- $D : \hat{y} = \mathbb{1}_{\{-1x_1-1x_2+0.5\geqslant 0\}}$
- E : None of the above

X_{i1}	X_{i2}	y_i
0	0	0
0	1	1
1	0	1
1	1	0

Learning XOR Operator Network

Quiz

- $y = x_1$ XOR x_2 is the same as
  $y = (x_1$ OR $x_2)$ AND $(x_1$ NAND $x_2)$

Single Layer Perceptron

Motivation

LTU, logistic

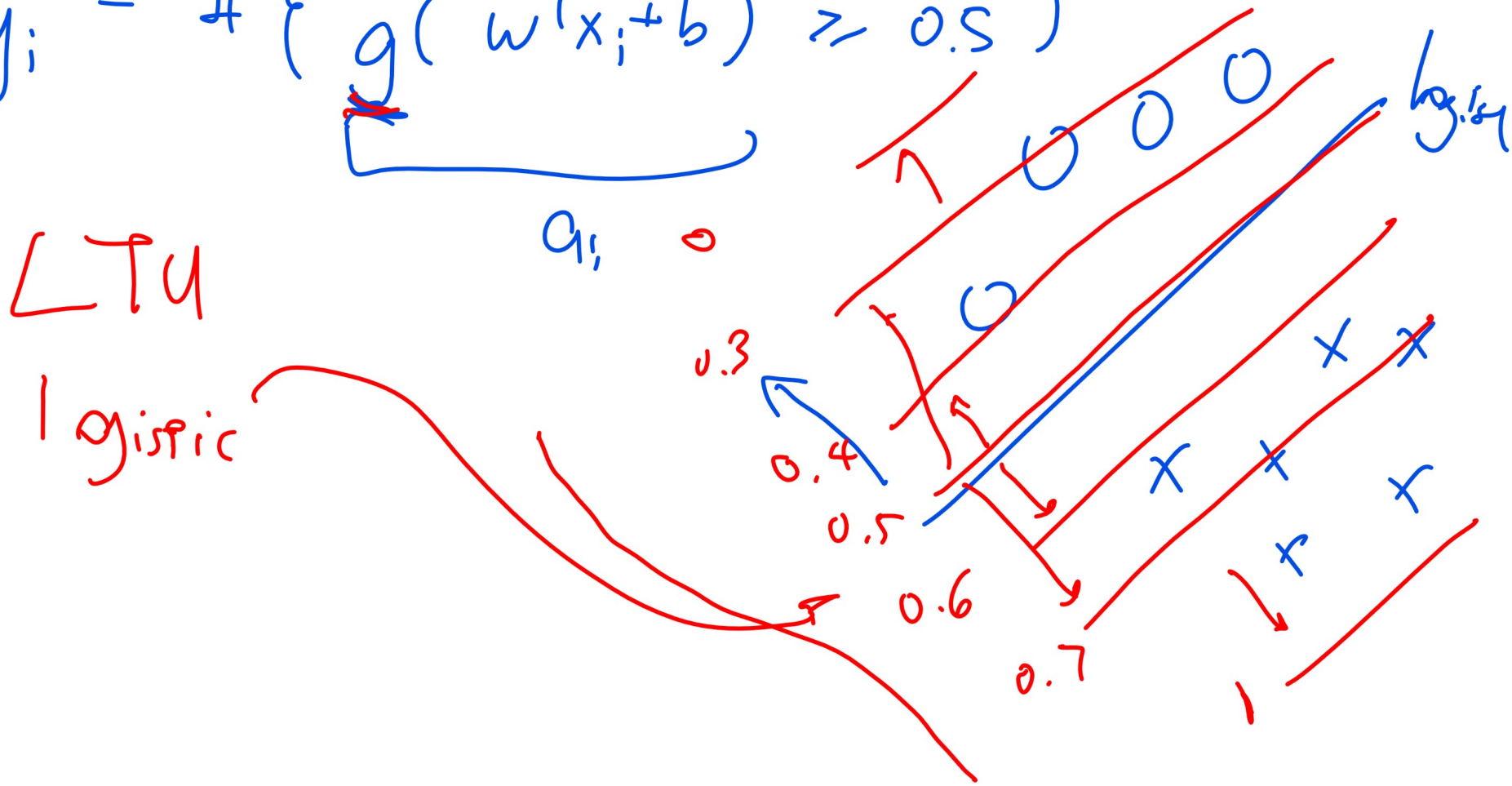
- Perceptrons can only learn linear decision boundaries.
- Many problems have non-linear boundaries.
- One solution is to connect perceptrons to form a network.

Decision Boundary Diagram

Motivation

$$\hat{y}_i = \mathbb{1}\{g(w^T x_i + b) \geq 0.5\}$$

$a_i = 0$



Multi-Layer Perceptron

Motivation

- The output of a perceptron can be the input of another.

$$a = g\left(w^T x + b\right)$$

$$a' = g\left(w'^T a + b'\right)$$

$$a'' = g\left(w''^T a' + b''\right)$$

$$\hat{y} = \mathbb{1}_{\{a'' > 0.5\}}$$

Relu

logistic.

fixed

$$\hat{y} = g(w''^T g(w'^T g(w^T x + b)) + b')$$

Neural Network Biology

Motivation

$\operatorname{argmin}_{h \in H} C(y_i, g_i)$

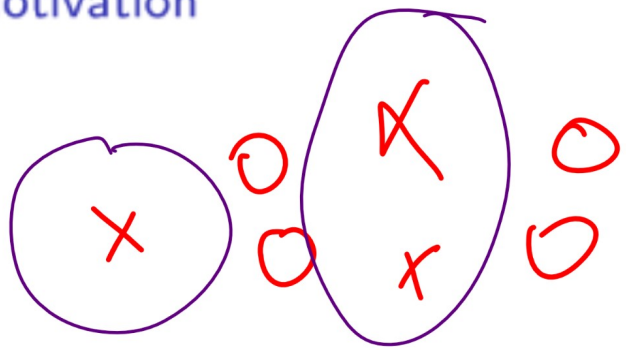
g_i

$g = g_3 \circ g_2 \circ g_1$

- Human brain: 100,000,000,000 neurons.
- Each neuron receives input from 1,000 others.
- An impulse can either increase or decrease the possibility of nerve pulse firing.
- If sufficiently strong, a nerve pulse is generated.
- The pulse forms the input to other neurons.

Theory of Neural Network

Motivation

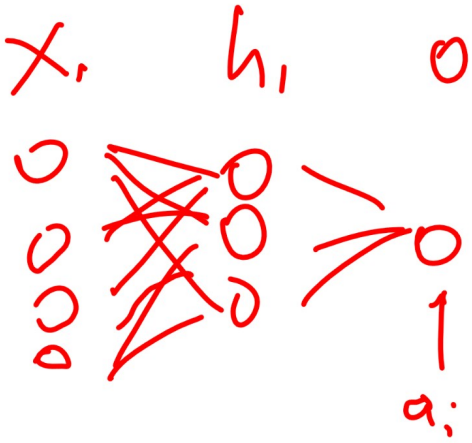


- In theory:

1. 1 Hidden-layer with enough hidden units can represent any continuous function of the inputs with arbitrary accuracy.
2. 2 Hidden-layer can represent discontinuous functions.

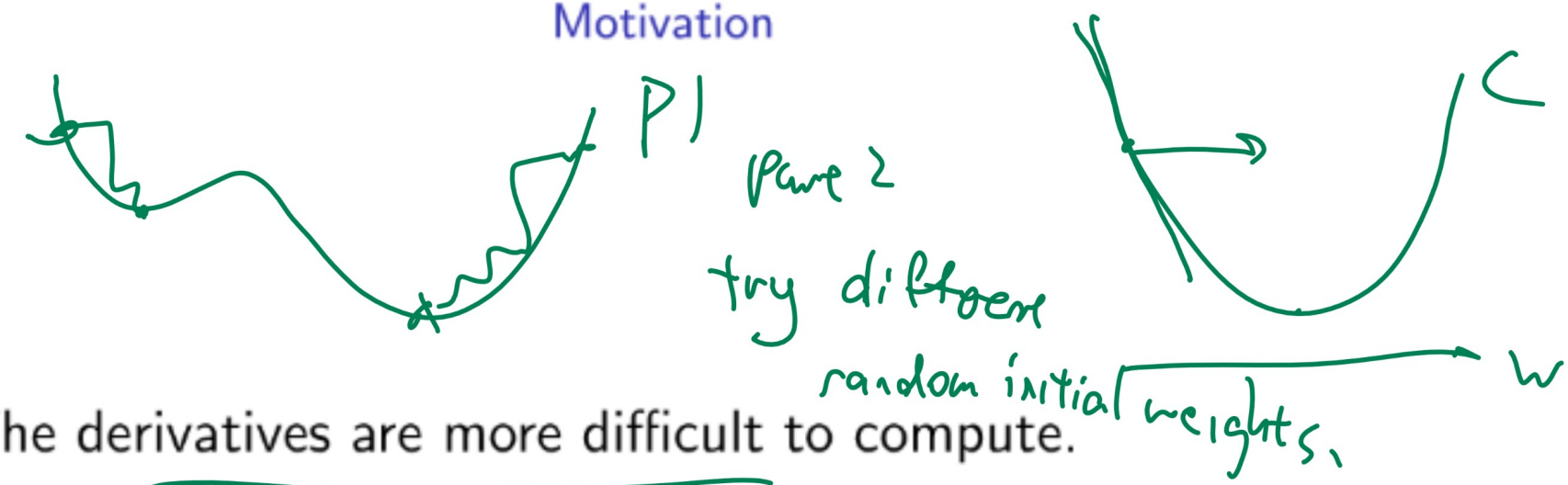
- In practice:

1. AlexNet: 8 layers.
2. GoogLeNet: 27 layers (or 22 + pooling).
3. ResNet: 152 layers.



Gradient Descent

Motivation



- The derivatives are more difficult to compute.
- The problem is no longer convex. A local minimum is no longer guaranteed to be a global minimum.
- Need to use chain rule between layers called backpropagation.

Backpropagation

Description

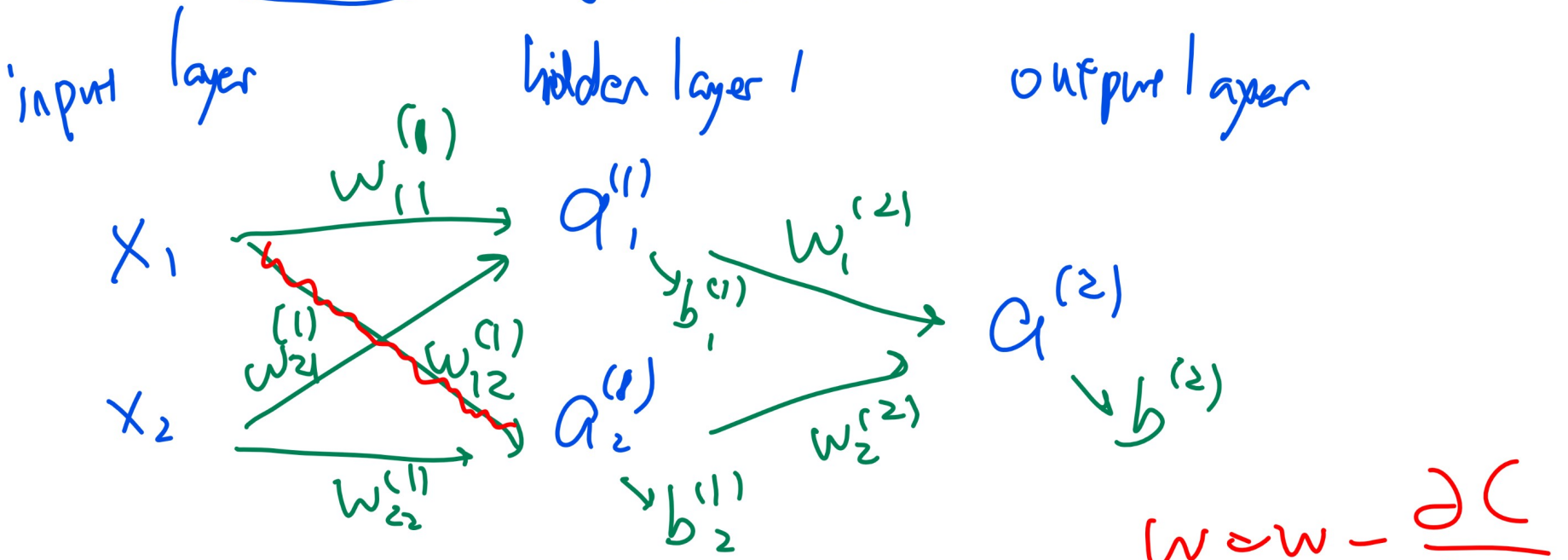
- Initialize random weights.
- (Feedforward Step) Evaluate the activation functions.
- (Backpropagation Step) Compute the gradient of the cost function with respect to each weight and bias using the chain rule.
- Update the weights and biases using gradient descent.
- Repeat until convergent.

Neural Network Demo

Motivation

Two-Layer Neural Network Weights Diagram 1

Motivation



$$w \to w - \frac{\partial C}{\partial w}$$

$$a_1^{(1)} = g\left(w_{11}^{(1)} x_1 + w_{21}^{(1)} x_2 + b_1^{(1)}\right)$$

$$a_2^{(1)} = g\left(w_{12}^{(1)} x_1 + w_{22}^{(1)} x_2 + b_2^{(1)}\right)$$

$$a^{(2)} = g\left(w_1^{(2)} a_1^{(1)} + w_2^{(2)} a_2^{(1)} + b^{(2)}\right)$$

$$C = \frac{1}{2}\sum_{i=1}^{n} \left(y_i - a_i^{(2)}\right)^2$$

Two-Layer Neural Network Weights Diagram 2

Motivation

$$\frac{\partial C}{\partial w_{12}^{(1)}} = \sum_{i=1}^{n} \frac{\partial C}{\partial a_i^{(2)}} \cdot \frac{\partial a_1^{(2)}}{\partial a_2^{(1)}} \cdot \frac{\partial a_2^{(1)}}{\partial w_{12}^{(1)}}$$

$$= (a_1^{(2)} - y_i)\left[a^{(2)}(1 - a^{(2)}) \cdot w_2^{(2)}\right]$$

$$a_2^{(1)} \cdot (1 - a_2^{(1)})\, x_1$$

$g' = g_i(1 - g)$
from logistic regression.

Two-Layer Neural Network Weights Diagram 3

Motivation

Gradient Step, Combined

Definition

squared loss + logistic activation

- Put everything back into the chain rule formula. (Please check for typos!)

$$\frac{\partial C}{\partial w_{j'j}^{(1)}} = \sum_{i=1}^{n} (a_i - y_i)\, a_i\, (1 - a_i)\, w_j^{(2)}\, a_{ij}^{(1)} \left(1 - a_{ij}^{(1)}\right) x_{ij'}$$

$$\frac{\partial C}{\partial b_j^{(1)}} = \sum_{i=1}^{n} (a_i - y_i)\, a_i\, (1 - a_i)\, w_j^{(2)}\, a_{ij}^{(1)} \left(1 - a_{ij}^{(1)}\right)$$

$$\frac{\partial C}{\partial w_j^{(2)}} = \sum_{i=1}^{n} (a_i - y_i)\, a_i\, (1 - a_i)\, a_{ij}^{(1)}$$

$$\frac{\partial C}{\partial b^{(2)}} = \sum_{i=1}^{n} (a_i - y_i)\, a_i\, (1 - a_i)$$

Gradient Descent Step

Definition

- The gradient descent step is the same as the one for logistic regression.

$$w_j^{(2)} \leftarrow w_j^{(2)} - \alpha \frac{\partial C}{\partial w_j^{(2)}}, j = 1, 2, ..., m^{(1)}$$

$$b^{(2)} \leftarrow b^{(2)} - \alpha \frac{\partial C}{\partial b^{(2)}},$$

$$w_{j'j}^{(1)} \leftarrow w_{j'j}^{(1)} - \alpha \frac{\partial C}{\partial w_{j'j}^{(1)}}, j' = 1, 2, ..., m, j = 1, 2, ..., m^{(1)}$$

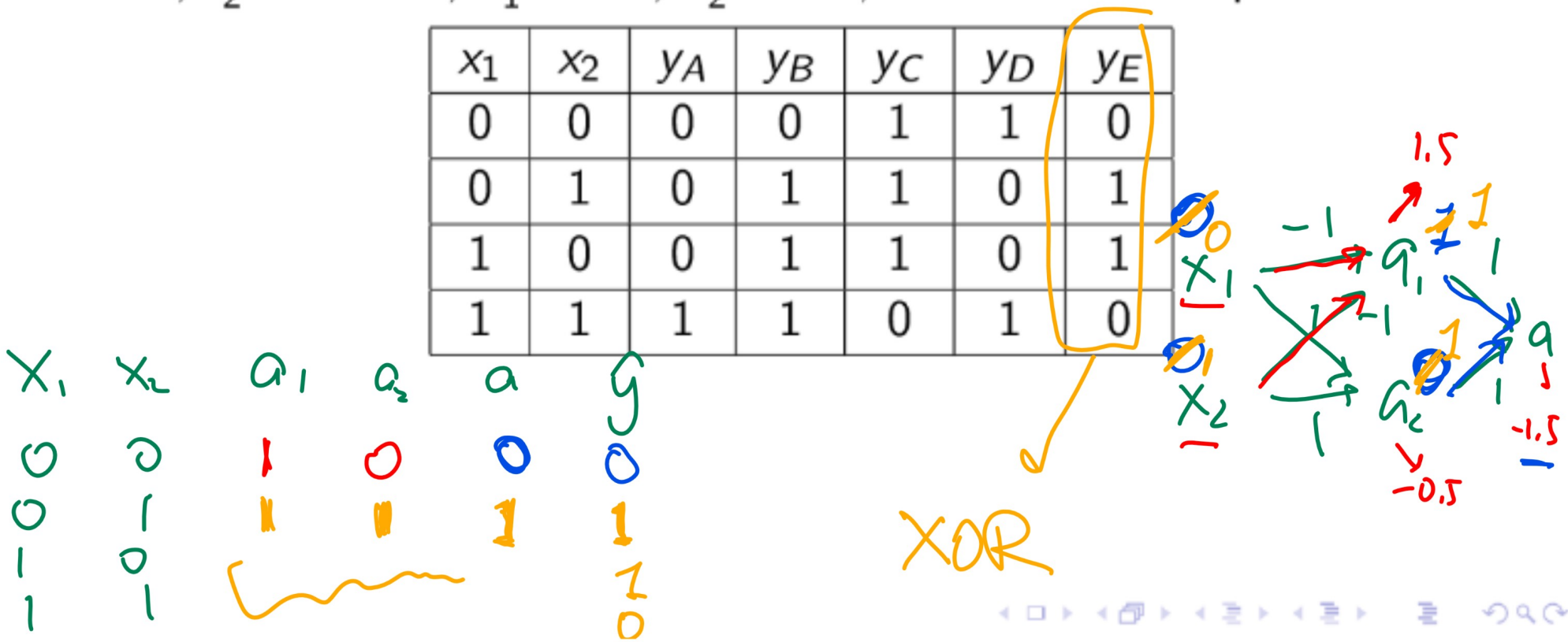
$$b_j^{(1)} \leftarrow b_j^{(1)} - \alpha \frac{\partial C}{\partial b_j^{(1)}}, j = 1, 2, ..., m^{(1)}$$

Learning Logical Operators, XOR

Quiz

- What function does the multi-layer LTU perceptron network with $w_{11}^{(1)} = -1, w_{21}^{(1)} = -1, b_1^{(1)} = 1.5, w_{12}^{(1)} = 1, w_{22}^{(1)} = 1, b_2^{(1)} = -0.5, w_1^{(2)} = 1, w_2^{(2)} = 1, b^{(2)} = -1.5$ compute?

x_1	x_2	y_A	y_B	y_C	y_D	y_E
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	1	0



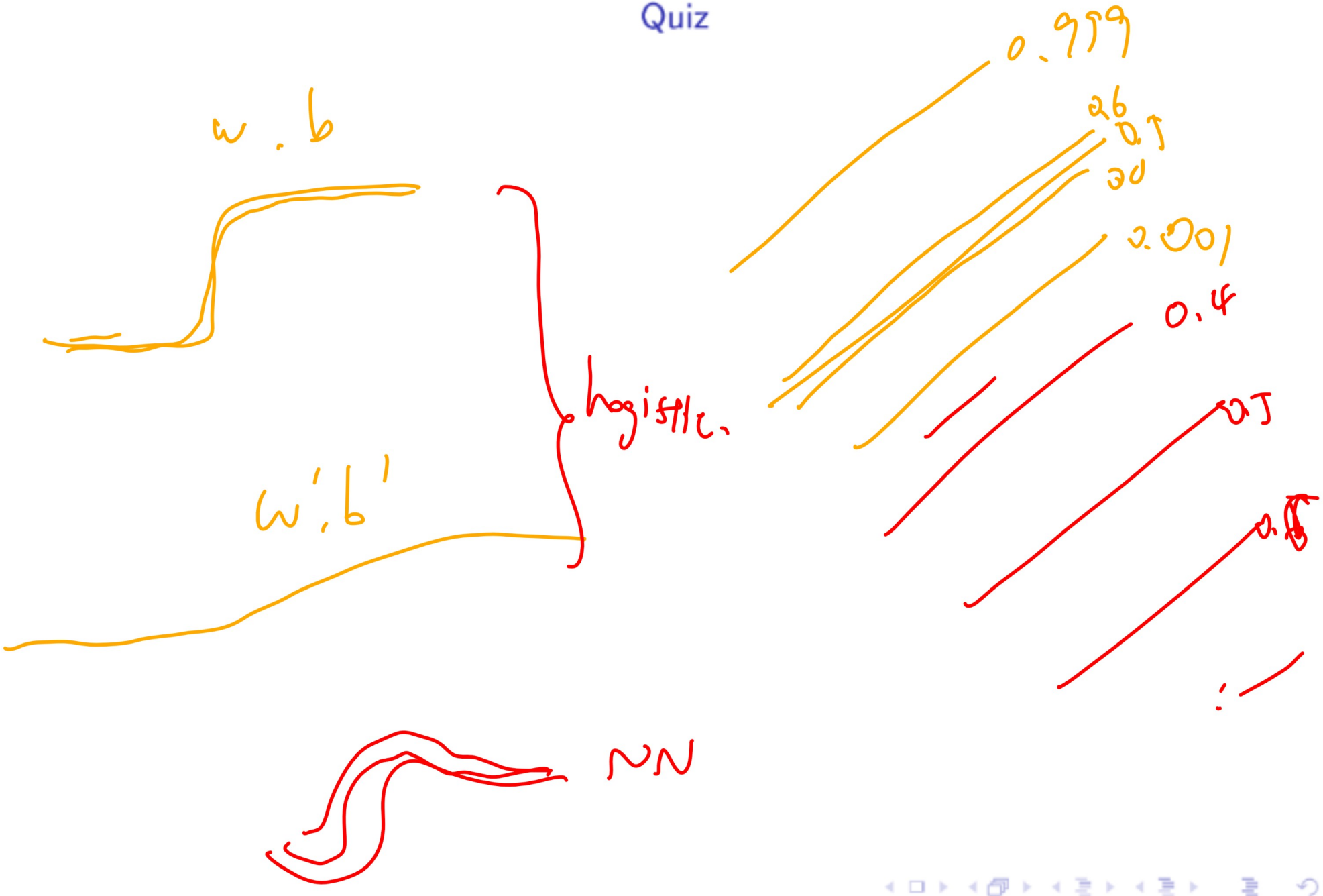
Learning Logical Operators, XOR, Diagram

Quiz

x_1	x_2	y_A	y_B	y_C	y_D	y_E
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	1	0

Learning Logical Operators, XOR, Answer

Quiz



Three-Layer Neural Network Weights Diagram

Motivation