CS540 Introduction to Artificial Intelligence
Lecture 3

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

June 23, 2021

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.

Prerecorded Lectures

Admin

- If you find the Zoom lectures difficult to follow, you can watch the prerecorded lectures first.
- If you prefer learning the materials more systematically (not through examples), you can watch the prerecorded lectures after the Zoom lectures.

Additional Discussion Sessions

Admin

- I could add unofficial discussion sessions (on Zoom, recorded) on Fridays from 12 : 30 to 1 : 45 go through examples, quizzes and homework questions again more slowly (no new materials, no new questions).
- A: I am planning to attend these sessions.
- B: I am not planning to attend but I am okay with having these sessions.
- C: I am not planning to attend and I am against having these sessions.
- D: Do not choose.
- E: Do not choose.

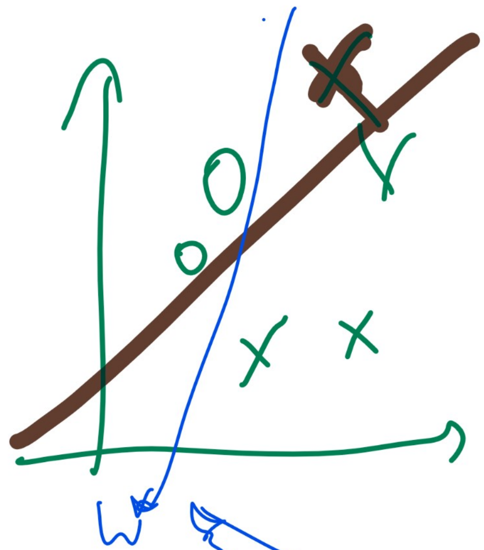
Remind Me to Start Recording

Admin

- The messages you send in chat will be recorded: you can change your Zoom name now before I start recording.

Optimization Diagram

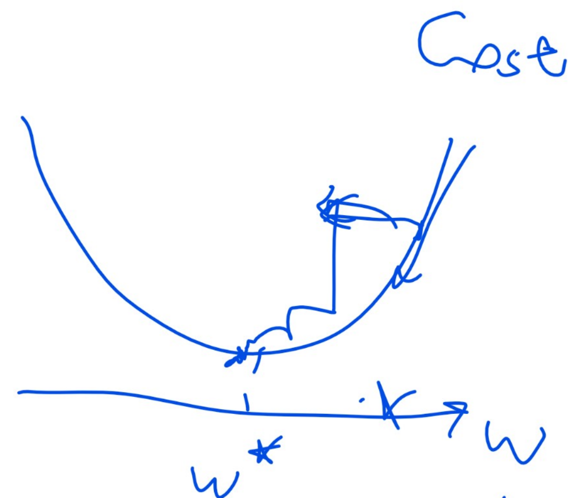
Motivation



$\text{Cost} \approx \#\text{ mistake}$

$\text{line} = \arg\min_{\text{line}} \text{Cost}$

f



Gradient Descent

Quiz

- What is the gradient descent step for w if the objective (cost) function is the squared error?

$$C = \frac{1}{2}\sum_{i=1}^{n}(a_i - y_i)^2, a_i = g\left(\underbrace{w^T x_i + b}_{z}\right), g'(z) = \underbrace{g(z) \cdot (1 - g(z))}_{\text{logistic}}$$

Cost = square loss

- A: $w = w - \alpha \sum (a_i - y_i)$
- B: $w = w - \alpha \sum (a_i - y_i)\, x_i$ $\nabla_w C$
- C: $w = w - \alpha \sum (a_i - y_i)\, a_i x_i$
- ~~D~~: $w = w - \alpha \sum (a_i - y_i)\, (1 - a_i)\, x_i$
- E: $w = w - \alpha \boxed{\sum (a_i - y_i)\, a_i\, (1 - a_i)\, x_i}$

Cost = Cross entropy
Activation = logistic.

$a_i - y_i \quad a_i(1-a_i) \quad x_{ij}$

$$\frac{\partial C}{\partial w_j} = \sum_{i=1}^{n} \frac{\partial C}{\partial a_i} \frac{\partial a_i}{\partial z} \frac{\partial z}{\partial w_j}$$

$w = w - \alpha \frac{\partial C}{\partial w}$ $\nabla_w C$

Gradient Descent, Answer

Quiz

Gradient Descent, Another One

Quiz

Q3

- What is the gradient descent step for w if the activation function is the identity function?

$$C = \frac{1}{2}\sum_{i=1}^{n}\left(a_i - y_i\right)^2, a_i = \underbrace{w^T x_i + b}_{z}, \quad a_i = g(z) = z$$

$z \rightarrow$ linear regression

- A: $w = w - \alpha \sum \left(a_i - y_i\right)$
- B: $w = w - \alpha \sum \left(a_i - y_i\right) x_i$ $\rightarrow \nabla_w C$
- C: $w = w - \alpha \sum \left(a_i - y_i\right) a_i x_i$
- D: $w = w - \alpha \sum \left(a_i - y_i\right) \left(1 - a_i\right) x_i$
- E: $w = w - \alpha \sum \left(a_i - y_i\right) a_i \left(1 - a_i\right) x_i$

z

linear part $= w^T x_i$

$$\frac{\partial C}{\partial w_j} = \sum_{i=1}^{n} \frac{\partial C}{\partial a_i} \frac{\partial a_i}{\partial z} \frac{\partial z}{\partial w_j}$$

$$(a_i - y_i) \cdot 1 \cdot x_{ij}$$

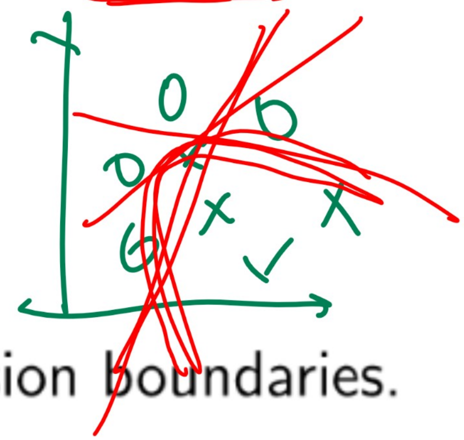
Gradient Descent, Another One, Answer

Quiz

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.

Multi-Layer Perceptron

Motivation

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

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

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

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

$$\hat{y} = \mathbb{1}_{\{a''>0\}}$$

← input layer

← hidden layer

← output layer

Neural Network Biology

Motivation

- 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.

$n \to \infty$

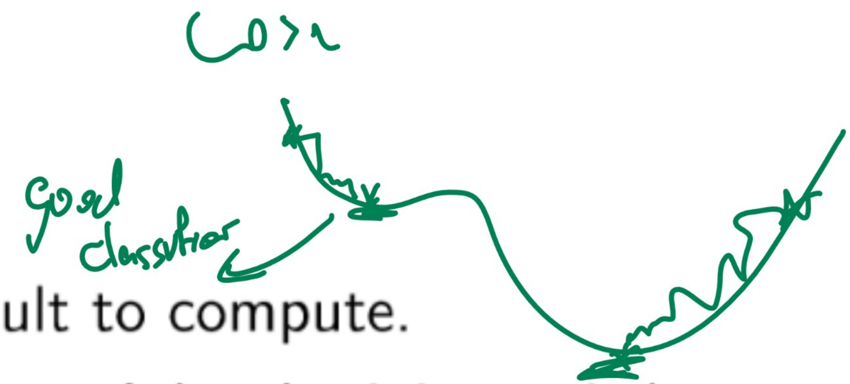
- In practice:

deep

  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 longer guaranteed to be a global minimum.
- Need to use chain rule between layers called backpropagation.

Backpropagation

Description

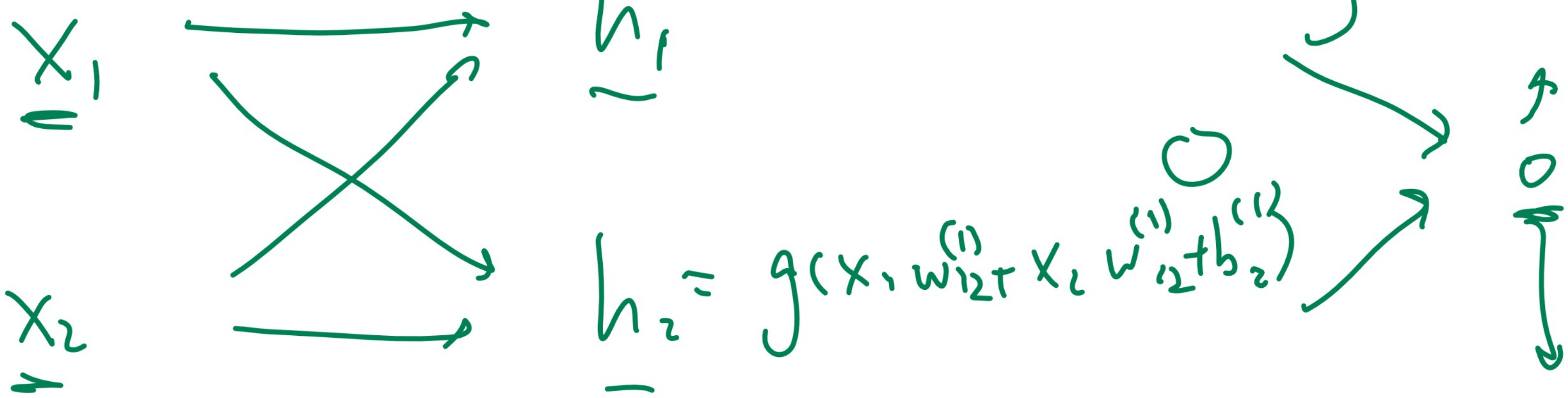
- Initialize random weights.
- (Feedforward Step) Evaluate the activation functions.
- (Backpropagation Step) Compute the gradient of the cost function with respect to each weight and bias using the chain rule.
- Update the weights and biases using gradient descent.
- Repeat until convergent.

Neural Network Demo

Motivation

Two-Layer Neural Network Weights Diagram 1

Motivation



$$h_1 = g(x_1 w^{(1)}_{11} + x_2 w^{(1)}_{21} + b^{(1)}_1)$$

$$h_2 = g(x_1 w^{(1)}_{12} + x_2 w^{(1)}_{12} + b^{(1)}_2)$$

$$g(h_1 w^{(2)}_1 + h_2 w^{(2)}_2 + b^{(2)})$$

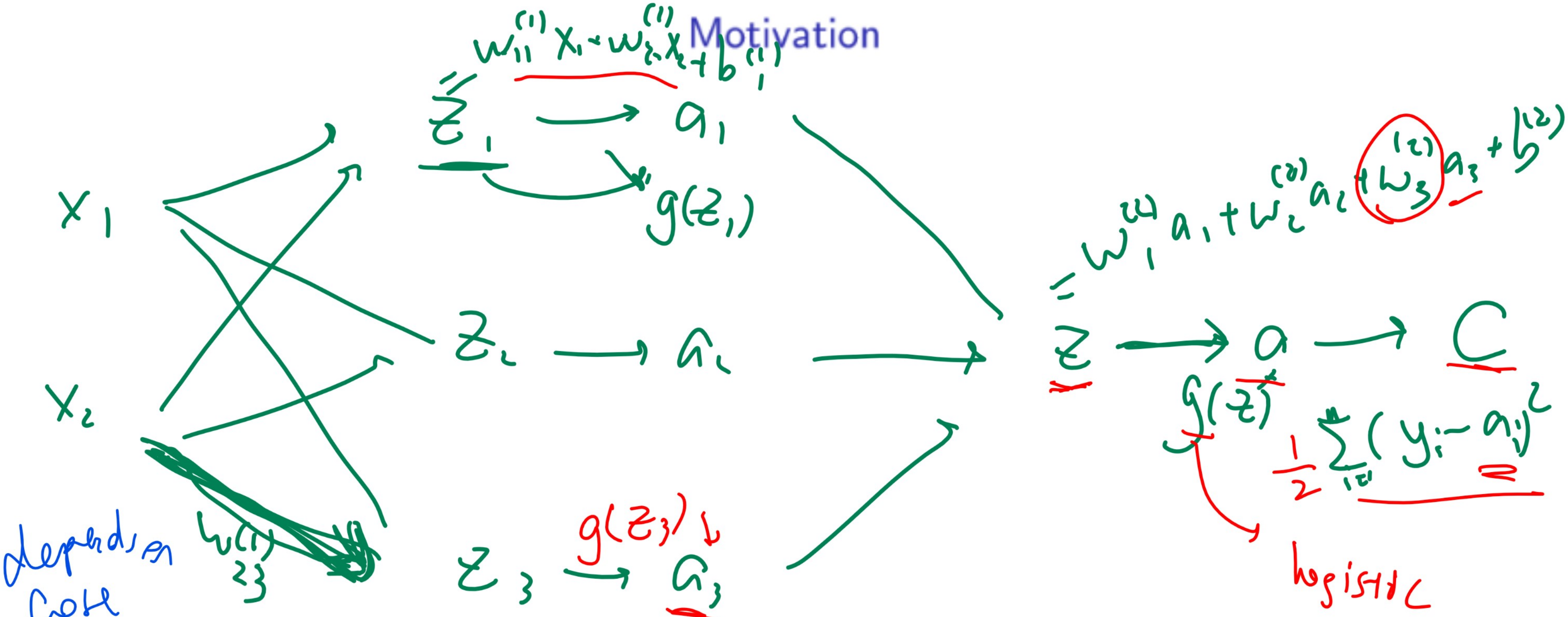
$$W^{(layer)}_{from \to to}$$

$$Cost_i = (o - y)^2$$

$$Cost = \sum_{i=1}^{n} Cost_i$$

Two-Layer Neural Network Weights Diagram 2

Motivation



$$\frac{\partial C}{\partial w_{23}^{(1)}} = \sum_{i=1}^{n} \frac{\partial C}{\partial a}\cdot\frac{\partial a}{\partial z}\cdot\frac{\partial z}{\partial a_3}\cdot\frac{\partial a_3}{\partial z_3}\cdot\frac{\partial z_3}{\partial w_{23}^{(1)}}$$

$$= (y_i - a)\cdot \boxed{a(1-a)}\cdot w_3^{(2)}\cdot \boxed{a_3(1-a_3)}\cdot x_2$$

logistic activation

$$w_{23}^{(1)} = w_{23}^{(1)} - \alpha \frac{\partial C}{\partial w_{23}^{(1)}}$$

Two-Layer Neural Network Weights Diagram 3

Motivation

Gradient Step, Combined

Definition

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

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

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

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

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

Gradient Descent Step

Definition

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

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

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

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

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

Remind Me to Stop Recording

Admin

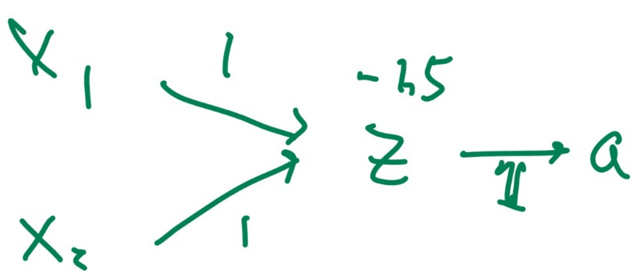
- If you accidentally selected an obviously incorrect answer earlier, you can enter the question name and the correct answer here.

Learning Logical Operators 1

Quiz

- What function does the single layer LTU perceptron with $w_1^{(1)} = 1, w_2^{(1)} = 1, b^{(1)} = -1.5$ compute?

a	z	x_1	x_2	y_A	y_B	y_C	y_D	y_E
0	-1.5	0	0	0	0	1	1	0
0	-0.5	0	1	0	1	1	0	1
0	-0.5	1	0	0	1	1	0	1
1	0.5	1	1	1	1	0	1	0

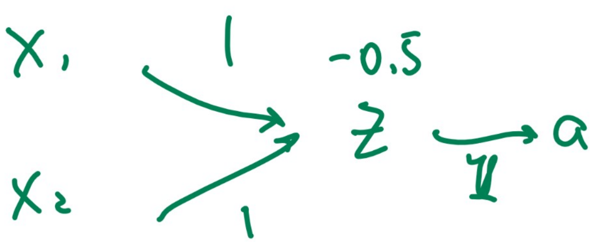


Learning Logical Operators 2

Quiz

- What function does the single layer LTU perceptron with $w_1^{(1)} = 1, w_2^{(1)} = 1, b^{(1)} = -0.5$ compute?

a	z	x_1	x_2	y_A	y_B	y_C	y_D	y_E
0	-0.5	0	0	0	0	1	1	0
1	0.5	0	1	0	1	1	0	1
1	0.5	1	0	0	1	1	0	1
1	1.5	1	1	1	1	0	1	0



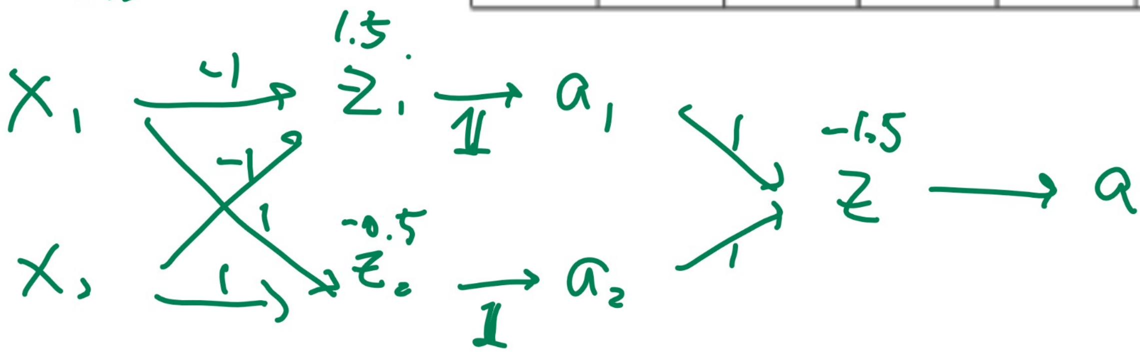
Learning Logical Operators 3

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?

a	z	a_2	a_1
0	-0.5	0	1
1	0.5	1	1
1	0.5	1	1
0	-0.5	1	0

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 3, Answer

Quiz

Learning Logical Operators 4

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

Solve on Thursday lecture.

Learning Logical Operators 4, Answer

Quiz