# CS540 Introduction to Artificial Intelligence

## Lecture 3

Young Wu

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

June 1, 2022

# Two-thirds of the Average Game

## Quiz

- 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.

# Homework Due Dates

## Admin

- $M, P$ are due on Mondays at midnight, after that you have another one week or so to submit or resubmit: no penalty except if you need to submit a regrade request (no official documentation required).
- Sharing solutions to $M2$ questions are due on the same day $M2$: late posts are not accepted.
- Sharing solutions to $X1$ questions are due the week before the midterm: late posts are not accepted.
- Piazza discussions: preferrably before the due date on Canvas (that's when I update), but late ones are okay too: no need regrade request.

# Homework Grades

## Admin

- $M, Q$ : 1 point each, lowest 2 dropped.
- $D$ : 0.5 points for each post: you can do more than 2 during some weeks, by preferrably less than 4.
- $P$ : 8 points each, use project to replace one of them: I do not have to approve the topic.

# AND Operator Data

## Quiz

- 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

# 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

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

# Learning XOR Operator Network

## Quiz

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

# Single Layer Perceptron

## Motivation

- 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

# Multi-Layer Perceptron

## Motivation

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

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

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

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

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

# 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.

- 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

# Two-Layer Neural Network Weights Diagram 2

## Motivation

# 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.

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

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

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

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

# 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