CS540 Introduction to Artificial Intelligence

Lecture 1

Young Wu

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

June 20, 2023

Lecture Format

Admin

- TWRF synchronous lecture from 1 : 00 to 2 : 15, and office hours from 2 : 15 to 3 : 00.
- Monday (optional) math review session from 1 : 00 to 3 : 00.
- Saturday (optional) programming session from 1 : 00 to 3 : 00 (Java and Python seperate rooms).
- Exam review sessions and exams during lecture times.

Lecture Recording

Admin

- The Zoom lectures will be recorded and can be accessed on Canvas.
- You can log in Zoom anonymously using any name you like (preferably one that can be pronounced easily and nothing offensive please).
- Blank lecture slides usually posted on Monday, annotated slides will NOT be posted: please copy down the notes yourself, either during the lectures or use the recordings.

Grading

Admin

- Programming homework: best 5 of 6, weekly, 8 points each.
- Exams: one midterm and one final, 30 points each, OR:

1. Math homework: best 10 of 12, two per week, 1 point each.
2. Quizzes: best 20 of 24, daily, 0.5 points each.
3. Discussions: total of 20, and 0.5 points each each.

TopHat Quiz

Quiz

Course Coverage

Admin

- The course is mostly theoretical, and not concerned with specific applications and implementations.
- The course requires basic calculus, linear algebra, and statistics, but not details about the proofs and algorithm analysis.

ChatGPT

Admin

- GPT stands for Generative Pre-trained Transformer.
- Natural languge processing: Week 3
- Unsupervised learning: Week 5
- Supervised learning: Week 1, Week 2, Week 3
- Reinforcement learning: Week 5

DALL-*E*

Admin

- Uses GPT3 for language processing.
- Computer vision: Week 2
- Generative supervised learning: Week 3
- Discriminative supervised learning: Week 1, Week 2
- Game theory: Week 7, Week 8

Tesla Autopilot

Admin

- Self-driving cars and drones (unmanned aerial vehicle).
- Search algorithm: Week 6
- Computer vision: Week 2
- Reinforcement learning: Week 5

AlphaGO

Admin

- Computer program to play the board game GO.
- Adversarial search: Week 6, Week 7
- Computer vision: Week 2
- Multi-agent reinforcement learning: Week 5, Week 7, Week 8

Supervised Learning Example 1

Motivation

Data	images of cats and dogs
Features (Input)	height, length, eye color, ...
Labels (Output)	cat or dog

Data	images of 1000 object classes
Features (Input)	pixel information ...
Labels (Output)	turtle or rifle

Supervised Learning Example 2

Motivation

Data	handwritten characters
Features (Input)	pixel intensity, stroke, ...
Labels (Output)	δ or σ, φ or ψ

Data	voice recording
Features (Input)	signal, sound (phoneme), ...
Labels (Output)	recognize speech or wreck a nice beach

Supervised Learning Example 3

Motivation

Data	medical records
Features (Input)	scan, blood, and test results, ...
Labels (Output)	cancer or no cancer

Data	patient information
Features (Input)	age, pre-existing conditions, ...
Labels (Output)	cancer or no cancer

Supervised Learning Example 4

Motivation

Data	emails
Features (Input)	word count, capitalization, ...
Labels (Output)	spam or ham

Data	comments
Features (Input)	word count, capitalization, ...
Labels (Output)	offensive or not

Supervised Learning Example 5

Motivation

Data	face images
Features (Input)	edges, corners, ...
Labels (Output)	face or non-face

Data	self-driving car data
Features (Input)	color, distance (depth), movement, ...
Labels (Output)	road or car or pedestrian

Supervised Learning Example 6

Motivation

Data	book or movie reviews
Features (Input)	word count, capitalization, ...
Labels (Output)	positive or negative

Data	financial transactions
Features (Input)	amount, frequency, ...
Labels (Output)	fraud or not

Supervised Learning Example 7

Motivation

Data	painting
Features (Input)	appearance, price, ...
Labels (Output)	art or garbage

Data	essay
Features (Input)	length, key words, ...
Labels (Output)	A+ or F

Supervised Learning

Motivation

- Supervised learning:

Data	Features	Labels	-
Sample	$\{(x_{i1}, ..., x_{im})\}_{i=1}^{n}$	$\{y_i\}_{i=1}^{n}$	find "best" $\hat{f}$
-	observable	known	-
New	$(x'_1, ..., x'_m)$	y'	guess $\hat{y} = \hat{f}(x')$
-	observable	unknown	-

Training and Test Sets

Motivation

- Supervised learning:

Data	Features	Labels	-
Training	$\{(x_{i1}, ..., x_{im})\}_{i=1}^{n'}$	$\{y_i\}_{i=1}^{n'}$	find "good" $\hat{f}$
-	observable	known	-
Validation	$\{(x_{i1}, ..., x_{im})\}_{i=n'}^{n}$	$\{y_i\}_{i=n'}^{n}$	find "best" $\hat{f}$
-	observable	known	-
Test	$(x'_1, ..., x'_m)$	y'	guess $\hat{y} = \hat{f}(x')$
-	observable	unknown	-

Linear Classifier

Motivation

- One possible guess is in the form of a linear classifier.

$$\begin{aligned}\hat{y} &= \mathbb{1}_{\{w_1x_1+w_2x_2+\ldots+w_mx_m+b\geqslant 0\}} \\ &= \mathbb{1}_{\{w^Tx+b\geqslant 0\}}\end{aligned}$$

- The $\mathbb{1}$ (open number 1) is the indicator function.

$$\mathbb{1}_E = \begin{cases} 1 & \text{if } E \text{ is true} \\ 0 & \text{if } E \text{ is false} \end{cases}$$

Linear Threshold Unit

Motivation

- This simple linear classifier is also called a Linear Threshold Unit (LTU) Perceptron.
- $w_1, w_2, ..., w_m$ are called the weights, and b is called the bias.
- The function that makes the prediction based on $w^T x + b$ is called the activation function.
- For an LTU Perceptron, the activation function is the indicator function.

$$g\left(\boxed{\cdot}\right) = \mathbb{1}_{\left\{\boxed{\cdot} \geqslant 0\right\}}$$

LTU Perceptron Training

Motivation

- Given the training set $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, the process of figuring out the weights and the bias is called training an LTU Perceptron.
- A training data point x_i is also called an instance.
- One algorithm to train an LTU Perceptron is called the Perceptron Algorithm.

Perceptron Algorithm

Description

- Initialize random weights.
- Evaluate the activation function at one instance x_i to get $\hat{y}_i$.
- If the prediction $\hat{y}_i$ is 0 and actual y_i is 1, increase the weights by x_i.
- If the prediction $\hat{y}_i$ is 1 and actual y_i is 0, decrease the weights by x_i.
- Repeat for all data points and until convergent.

Perceptron Algorithm, Part 1

Algorithm

- Inputs: instances: $\{x_i\}_{i=1}^{n}$ and $\{y_i\}_{i=1}^{n.}$
- Outputs: weights and biases: $w_1, ..., w_m$, and b.
- Initialize the weights.

$$w_1, ..., w_m, b \sim \text{ Unif } [-1, 1]$$

Unif $[l, u]$ means picking a random number between l and u.

- Evaluate the activation function at a single data point $x_{i.}$

$$a_i = \mathbb{1}_{\{w^T x_i + b \geqslant 0\}}$$

Perceptron Algorithm, Part 2

Algorithm

- Update weights using the following rule.

$$w = w - \alpha (a_i - y_i) x_i$$
$$b = b - \alpha (a_i - y_i)$$

- Repeat the process for every $x_i, i = 1, 2, ..., n$.
- Repeat until $a_i = y_i$ for every $i = 1, 2, ..., n$.

Learning Rate

Discussion

- The learning rate α controls how fast the weights are updated.
- They can be constant for each update or they can change (usually decrease) for each update.
- For perceptron learning, it is typically set to 1.

Summary

Discussion

- Supervised learning:
- Linear threshold unit: Perceptron algorithm.
- Logistic regression (next time).