CS540 Introduction to Artificial Intelligence
Lecture 1

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

June 21, 2021

Lecture Format

Admin

- Pre-recorded lectures will be posted on the course website.
- University-assigned lecture time will be used to go over examples and quizzes.
- The hour after the lecture will be used as office hours.

Lecture Recording

Admin

- The Zoom lectures will be recorded and can be accessed on Canvas.
- All course materials will be covered in the pre-recorded lectures on YouTube.
- The annotated lecture notes (with quiz questions) will be posted on the Course Website.

Zoom Login

Admin

- You can log in Zoom anonymously using any name you like (preferably one that can be pronounced easily and nothing offensive please).
- You have to log in Socrative with you real wisc ID.

Grading

Admin

- Quizzes: best 20 of 24, daily, 0 or 0.5 points each.
- Math homework: best 10 of $10 + 2$, weekly, 0 or 1 point each.
- Programming homework: best 5 of $5 + 1$, weekly, 8 points each.
- Exams: one midterm and one final, 30 or 25 or 20 points each.

Quizzes

Admin

- Download Socrative, the room number is CS540.
- Default login for Socrative is your wisc email ID.
- If someone else tries to hack your account, please email or private post on Piazza.
- Quiz questions can show up any time during the lecture.
- Missing one or two questions due to technical difficulty is okay.
- If you select obviously false answers, you will lose points.

Blank Quiz

Quiz

Math Homework

Admin

- Please do not start before I announce it on Canvas and Piazza.
- Officially: due in 1 week Sunday.
- Unofficially: any time before the midterm or the final.
- Solution: please volunteer to share your answers on Piazza.
- Auto-graded: unlimited number of times, I will not see your submission as long as you do not click the "Submit" button.

Programming Homework

Admin

- Please do not start before I announce it on Canvas and Piazza.
- Officially: due in 2 weeks Sunday.
- Unofficially: any time before the final.
- Solution: posted in 1 week Sunday.
- Auto-graded: use the "Submit" button AND submit the output and code on Canvas.
- Code: any language, Java and Python are recommended, MATLAB, R, JavaScript okay too.

Midterm and Final

Admin

- Synchronous exams: two parts, $12:30$ PM and $12:30$ AM versions, choose any two to take.
- 30 Questions: ~ 10 from homework, ~ 10 from homework or quizzes, ~ 10 new.

Textbook

Admin

- Lecture slides and videos should be sufficient.
- RN is a good background reading, does not cover everything.
- SS is very theoretical, useful if you are planning to take $760, 761, 861$.

Blank Quiz

Quiz

Generative Adversarial Network

Motivation

- Generative Adversarial Network (GAN):

1. Generative part: input random noise and output fake images.
2. Discriminative part: input real and fake images and output labels real or fake.
3. The two parts compete with each other.

Supervised Learning Example 1

Motivation

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

Supervised Learning Example 2

Motivation

Data	medical records
Features (Input)	scan, blood, and other test results
Output	disease or no

Data	patient information
Features (Input)	age, pre-existing conditions, ...
Output	likelihood of death

Supervised Learning Example 3

Motivation

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

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

Supervised Learning Example 4

Motivation

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

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

Supervised Learning Example 5

Motivation

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

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

Supervised Learning Example 6

Motivation

Data	handwritten letters
Features (Input)	pixel, stroke
Output	δ or σ, φ or ψ

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

Supervised Learning Example 7

Motivation

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

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

Supervised Learning

Motivation

- Supervised learning:

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

Training and Test Sets

Motivation

- Supervised learning:

Data	Features (Input)	Output	-
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	-

Simple $2D$ Example Diagram

Motivation

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.

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

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.

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

Equation of a Line

Motivation

- In $1D$, $w_1 x_1 + b \geqslant 0$ is just a threshold rule: $x_1 \geqslant -\frac{b}{w_1}$ implies $\hat{y} = 1$.
- In $2D$, $w_1 x_1 + w_2 x_2 + b \geqslant 0$ can be written as $\begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b \geqslant 0$. Note that $w_1 x_1 + w_2 x_2 + b = 0$ is the equation of a line, usually written as $x_2 = -\frac{w_1}{w_2} x_1 - \frac{b}{w_2}$.

Equation of a Hyperplane

Motivation

- In $3D$, $w_1x_1 + w_2x_2 + w_3x_3 + b \geqslant 0$ can be written as $\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + b \geqslant 0$. Note that $w_1x_1 + w_2x_2 + w_3x_3 + b = 0$ is the equation of a plane, and $\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$ is normal vector of the plane.
- The normal vector is perpendicular to all vectors on the plane.

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.

Brute Force LTU Learning

Motivation

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 Diagram, 0 Example

Description

Perceptron Algorithm Diagram, 1 Example

Description

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.

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

- 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.