Lecture 1: ECE C147/C247, Neural Networks & Deep Learning

General info:

ECE C147/C247, Neural Networks & Deep Learning
Winter quarter, UCLA AY 2024-25

Instructor:

Prof. Jonathan Kao

TAs:

Shawn Dong
Kaifeng Pang
Shreyas Rajesh
Tonmoy Monsoor (Volunteer)
Bruce Qu
Xu Yan

Some high-level thoughts about this class:

- AI is mainstream. This class is about the fundamental tools underlying modern AI and deep learning.
- This class will (hopefully) be fun, but also a lot of work.
- Ask questions!

Zoom online lectures

We will open a Zoom room at `https://ucla.zoom.us/j/91366715461` during live lecture so MSOL students can participate remotely.

By default, all students will be muted in the Zoom room. We will not pause to take verbal questions over Zoom. When possible, we will try to have at least one TA will monitor the chat on Zoom and answer questions in the chat.

Deep Learning is ubiquitous

- **AI is a part of everyday part of life.**

- It has resulted in key breakthroughs in many areas; in many different fields, many groups are looking towards deep learning to achieve better performance. Deep learning may be useful to research in your area / future work.

But AI is also complex

- AI is also complex. In this course, I don't want these things to be magic to you.
- A lot of the architectures and techniques that underlie many of the AIs we've presented (including ChatGPT, generative models, transformers, diffusion, image segmentation, game playing) rely on a considerable foundation.
- It is true that with tools today, you could e.g., go and implement a diffusion model off-the-shelf and generate images. But this course, and its sequel, will teach you all the details of how they work so they aren't magic.
- For this reason, last year we created for the first time a sequel course, and I want to lay out the relationship of these courses.

ECE C147/C247 course sequence

ECE C147/C247 (soon: C147A/C247A)
Winter 2025 (this course)

- Foundations of deep learning:
  - loss functions, softmax layer, neural network layers.
  - backpropagation, gradient descent, optimization (Adam and others)
  - regularizations (batch normalization, dropout)
- Application to computer vision
  - CNNs (e.g., ResNets), computer vision, object detection & segmentation
- Recurrent neural networks

ECE C147B/C247B
Spring 2025 (next quarter, pre-req: C147A/C247B)

- Generative models
  - GANs, VAEs and evidence lower bounds
  - Diffusion models
- Transformers
  - Attention
  - Applications to language (GPT) and robotics
- Deep reinforcement learning (deep Q networks, PPO)

Deep learning foundations through computer vision

https://www.youtube.com/watch?v=40riCqvRoMs

Deep learning foundations through computer vision

- Neural network research dates back to 1943 (McCullough and Pitts, a logical calculus of nervous activity) and 1958 (Rosenblatt's perceptron)

- Example video from 1989 of using a multi-layer perceptron (fully connected neural network) to do classification of digits.

- Fukushima's NeoCognitron (1982), Yann LeCun's "LeNet" (1998) which is the modern convolutional architecture, existed well before today's deep learning renaissance.

Tackling computer vision

Human Perf error rate: 5%

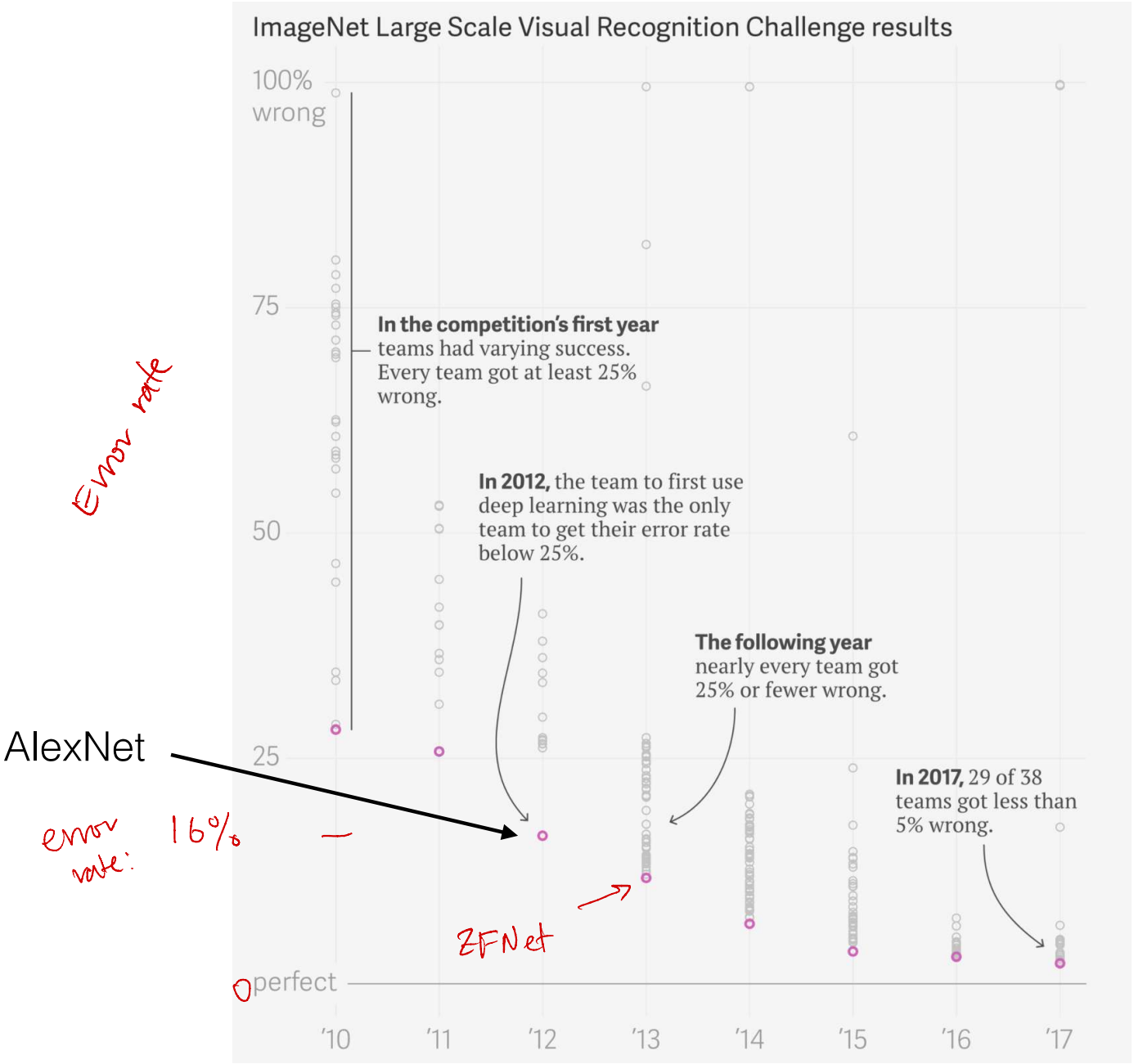
- IMAGENET

Overall

- Total number of non-empty synsets: 21841
- Total number of images: 14,197,122
- Number of images with bounding box annotations: 1,034,908

http://image-net.org/about-stats

A big splash


ImageNet Large Scale Visual Recognition Challenge results
100% wrong
75
50
25
0 perfect
'10
'11
'12
'13
'14
'15
'16
'17
In the competition's first year teams had varying success. Every team got at least 25% wrong.
In 2012, the team to first use deep learning was the only team to get their error rate below 25%.
The following year nearly every team got 25% or fewer wrong.
In 2017, 29 of 38 teams got less than 5% wrong.
Error rate
AlexNet
error rate: 16%
ZFNet

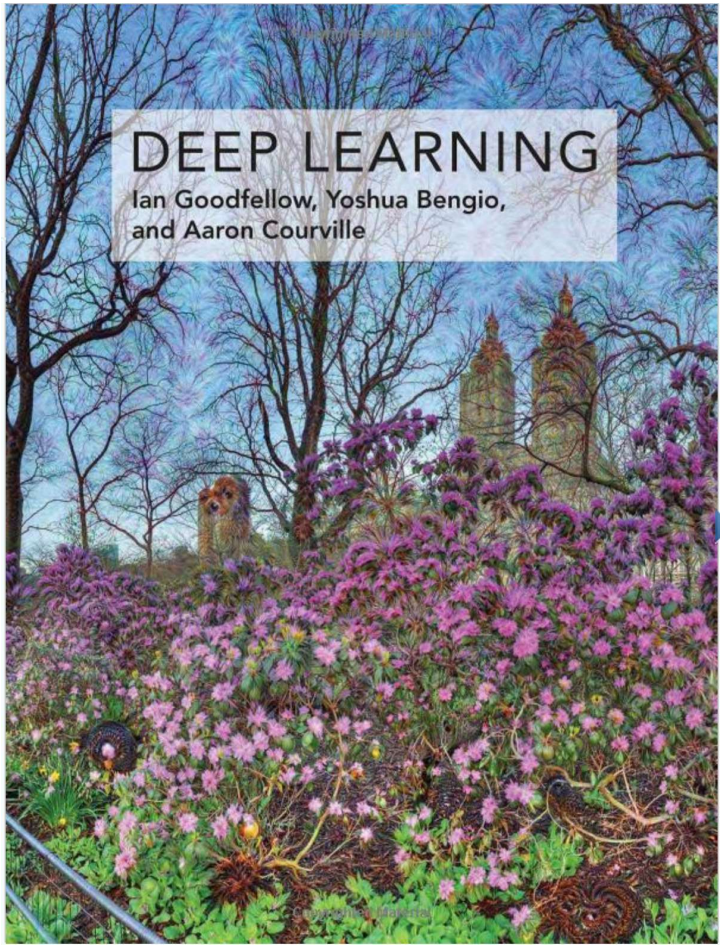
Data and technology

- Part of what has driven deep learning today is the amount of data we have and the computational power we have to process it.
- This includes larger models as computing infrastructure improves.

Fortunately, the amount of skill required [to get good performance from a deep learning algorithm] reduces as the amount of training data increases.
(Goodfellow, p. 18)

Rough class schedule

Date (2025)	Lecture	Content
06 Jan	1	Overview to deep learning
08 Jan	2	Machine learning refresher I
13 Jan	3	Supervised Classification & Gradient descent principles I
		HW #1 released
15 Jan	4	Supervised Classification & Gradient descent principles II
20 Jan	-	No class (MLK holiday)
		HW #2 released
22 Jan	5	Fully connected neural networks
27 Jan	6	Backpropagation
		HW #3 released
29 Jan	7	Regularizations for training neural networks
03 Feb	8	Optimization for training neural networks
		HW #4 released
05 Feb	9	Convolutional Neural Networks I
10 Feb	10	Convolutional Neural Networks II
		HW #5 released
12 Feb	11	Convolutional Networks III
17 Feb	-	President's day holiday
19 Feb	M	Midterm
		Project released
24 Feb	12	Recurrent neural networks I
26 Feb	13	Recurrent neural networks II
03 Mar	14	Deep learning libraries
05 Mar	15	Object detection & segmentation
10 Mar	16	Adversarial attacks
12 Mar	17	Overview



http://www.deeplearningbook.org/

For MSOL students: we will organize a remote exam on Saturday, Feb 22, from 2-3:50p.

On discussion sections

We will hold DIS 1A and DIS 1B, and at least one will be recorded and uploaded to ~~Zoom~~. Bruin Learn.

DIS 1A is in Fowler A103B, which seats 320.
DIS 1B is in Boelter 5280, which seats 40.

Grading

You will be graded on three components:

1. **Homework (40%)**. Homework will contain both written components as well as `Python` components.
   - Assignments are due (i.e., submitted to Gradescope) by 11:59pm on the day they are due.
   - To accomodate for unexpected or unforeseen circumstances, we will give *three late days* to every student. These late days should only be used in extenuating circumstances. We will not grant additional late days beyond these. 72 hours.
   - You may use **at most** 2 late days on any given assignment.
   - Any assignment more than two days late receives a grade of **zero**.
2. **Midterm exam (30%)**, in class.
3. **Final project (30%)**, details to be released.

Grading (cont.)

The class is graded on an absolute scale. The scale may be relaxed but it will not be made more stringent. These scales will be calculated and applied separately for undergraduate and graduate students.

Final score	Letter grade
93 – 100	A
90 – 93	A-
86 – 90	B+
83 – 86	B
80 – 83	B-
76 – 80	C+
73 – 76	C
70 – 73	C-
66 – 70	D+
63 – 66	D
60 – 63	D-
< 60	F

- We award A+ to the class's top students.

Grading (cont.)

In addition to these grading scales, we will also award bonuses on top of your final grade as follows.

Bonuses (2 points)

- (Feedback) You earn a bonus of +0.5% for filling out the class evaluation at the end of class.
- (Piazza) You receive a bonus of at most +1.5% for participating on Piazza. While your answers to others will be anonymous, they will be known to the instructors, who will determine an appropriate number of points for instructor-approved student replies. Your bonus will be based on your participation on Piazza, which will be curved.
- (Piazza, cont.) Please do not conspire to post and answer questions for extra credit. We will be able to detect this. We do not want the Piazza forums to be spammed; this makes it more difficult for all students to find helpful questions.

Grading

P/NP and S/U grading

Note that per the UCLA registrar:

> *The grade P is assigned for a letter grade of C or better. Units earned this way count toward degree requirements but do not affect the GPA.*
>
> *The grade S is assigned for a letter grade of B or better, but units earned in this manner are not counted in computing the GPA.*

This is University policy. If take this class P/NP and earn a C-, by definition that is a NP. If you take this class S/U and earn a B-, by definition that is a U.

A class this large has strict rules

- This class has over 500 students. To keep things fair in a class this large, and to be able to run this course smoothly, we are strict and firm on all deadlines and dates.

- If you have CAE exam accommodations, you must take the exam with CAE. We cannot provide the accommodations ourselves.

Grading (cont.)

- I will not change your final grade unless I made a calculation error, in accordance with UCLA Academic Senate Regulation A-313 and strict rules governing the integrity of the grading process.
- In HW # 3, at least one question will be optional for undergraduates.
- Unless I made an error in calculating your final grade, do not send me any requests to change your final grade from what you earned according to the syllabus. When you send a request like this, you are asking me to break UCLA Academic Senate rules. I also view these requests as fundamentally unfair to your peers who worked hard to earn their grades. **I will not entertain any such requests and I will reply to any request of this nature by sending a screenshot of this page in the syllabus.**

HW

Regrades will not be adjudicated after regrade request deadlines

- We use Gradescope to grade every HW and exam. After grades for an assignment are released, we give a one week window to submit regrade requests. These regrade requests will be reviewed by the person who graded the question across the whole class.
- Please be aware that when considering regrade requests, we usually regrade the entire question. You may therefore lose points in your regrade request. Please be thoughtful about these requests before you submit.
- In general, the TAs and I will defer to the grader's judgment in these regrade requests, even if we may have graded it differently. This is done out of fairness; if the grader applied one standard to the class, it should be applied to everyone the same way.
- We **will not consider any regrade requests** after the one week regrade request deadline. For example, a student may have earned an A- but be close to an A, and request regrades on HW assignments early in the quarter after final grades were assigned. These regrade requests would be denied because we do not adjudicate any regrades after the one week regrade request window has ended.

No HW extensions beyond the late deadline

- In general, we will not allow any HW extensions beyond the late deadline. Any assignment submitted after the late deadline receives a zero.

No alternate exam times

There are no alternate exam dates.

The midterm for this course will be held on February 19, 2025 (February 22, 2025 for MSOL students). **We are firm on this date and there are no other alternate exam dates.** If you do not take the midterm exam at the scheduled time, you will receive a zero on the midterm. I will respond to any requests to take the midterm at another time by sending a screenshot of this page in the syllabus.

Final project due date is set

The final project will be due Friday, March 14, 2025.

The final project will be due Friday, March 14, 2025. I will not extend this deadline for any reason. I will respond to any requests by sending a screenshot of this page in the syllabus.

Some notes on the class

Course material questions

- The primary ways to get questions answered are to (1) post questions to Piazza, and (2) come to our office hours.

- **Regarding course material questions**: Please post course material questions to Piazza instead of emailing us individually. This benefits other students who will see the posted question and answer. If you e-mail us, we will likely respond by asking you to post your question to Piazza unless it is of a unique or private nature.

- **Regarding Piazza**: We aim for Piazza will be student-driven. This means answers will be primarily provided by other students. **Why?** We believe this leads to (1) increased student engagement with course material, (2) quicker answers, and (3) increased understanding, because explaining or teaching something solidifies understanding. To encourage this, we assign bonus grades for participating on Piazza. The teaching staff will regularly monitor Piazza to approve student answers and answer any questions without a student answer. Because we assign Piazza bonuses, you may post anonymously to your classmates, but your identity will be known to the teaching staff.

Piazza and e-mail

Course material questions (cont.)

- **Regarding email**: We cannot always answer e-mails immediately. For example, if the HW is due in less than 24 hours and you e-mail us, there is a chance we will not get back to you before the HW is due.

- **Regarding personal inquiries**: For any questions regarding personal matters (e.g., questions about grades, discussing an extenuating circumstance), please e-mail Prof. Kao directly.

- **All other email inquiries**: All other email inquiries should be sent to the TAs and myself. Please e-mail all of us concurrently, so that no one instructor is overloaded.

Some notes on the class

- 5 HW assignments.
- Assignments will contain both written components as well as coding components.
- Code in the HW will walk you through Jupyter Notebooks, and usually it'll be clear if you're doing things correctly or not.
    - Note, we will release solutions for written HW exams, but we will **not** release solution code for HW's.
    - These assignments are planned to be used in future years.
- HW will be submitted via Gradescope; they are due at 11:59PM Pacific Time on the stated due date.
- Any late assignment will receive a grade of **zero**.
- We understand that life happens. We are giving **three late days to every student**, with the intent that these are to be used only in extenuating circumstances. You may use at most 2 late days on any given assignment.

Some notes on the class

- C147 and C247 students will be graded on different scales.
    - C147 students will also have one question of one homework be optional.
    - The classes are otherwise the same.
- All HWs are to be submitted to Gradescope.
- In general, I will post the non-annotated versions of the slide prior to lecture, so you can take notes if you want. I will always post the annotated version of the slides as well as the lecture videos (unless there are technical failures).
- We will list readings in the textbook at the start of each different lecture topic.

Some notes on the class

- This class will focus on implementation of neural networks and on algorithms to train them.
- It will **not** cover theory of deep learning.
- It will **not** be a theoretical class in general.
    - We aim to be rigorous in our use of notation and math.
    - But we won't be doing proofs.
    - We're more focused on application and equipping you with tools that may be helpful in your research or future industry positions.
- This is not a statement on the importance of theory. It merely reflects the priorities for this class alone.

Academic integrity

UCLA embraces the core values of integrity, excellence, accountability, respect, and service through the True Bruin program

`http://www.truebruin.ucla.edu`

I take academic integrity very seriously; students caught cheating or violating these principles will face disciplinary action. Please refer to the UCLA student conduct code:

`https://deanofstudents.ucla.edu/student-conduct-code`

In this class, unacceptable behavior includes plagiarizing the work of others, plagiarizing code, and copying another person's exam. In accordance with UCLA policy, any instance of suspected academic dishonesty will be immediately reported to the Dean of Students Office and zero credit will be given for any work determined to be dishonest.

What I assume you know

Pre-requisites

This class requires a solid understanding in probability (131A) and linear algebra (133A or 205A). It will also help if you have had prior exposure to machine learning (such as through M146). We will spend two lectures doing machine learning review to ensure we are familiar with concepts we will expand upon in machine learning.

It also requires coding experience. The class will be taught entirely in `Python`. If you have only had exposure to `MATLAB`, there will be ramp up time to familiarize yourself with `Python`. You should factor this into your course load.

Pre-requisite topics I will **assume** you know.

- Probability: independence, conditional probability, Bayes rule, marginalization, expectation, variance
- Linear algebra: basic matrix operations, span, rank, range and null space, eigenvalue decomposition, pseudoinverse

A few last notes about this class

Last notes

- Common student feedback is that, even if they were familiar with `MATLAB`, it was still time-consuming to transition to `Python`. Please consider this seriously as you plan your schedule for assignments. `Python` is the standard language for machine learning research today, and the best deep learning packages are specifically designed for `Python`.
- We know, and consistently receive feedback, that this class is a lot of work and is time-consuming. I want to state this up front so you can plan accordingly. We will aim to keep the stated HW schedule, following the assignments and schedule used last year.

About me

- This is my eighth time teaching this class at UCLA.
- Though this class focuses on deep learning for engineering purposes, I'm interested in deep learning because of its ties to the brain.
    - I am primarily a computational neuroscientist and neural engineer.
    - My work includes brain-machine interfaces and applying machine learning to understand how neurons in the brain communicate.
    - We use deep learning as a model to understand the brain and build brain-machine interfaces.

Lecture 2: Reviewing the basics of machine learning

Announcements:

- We sent out an Announcement on Bruin Learn on how to sign up for Gradescope and Piazza. Please do so ASAP.
  - Piazza: https://piazza.com/ucla/winter2025/c147c247. Please be sure to use your full name (first + last name) in setting up your account, so we can assign bonuses properly. **If you do not put in your full name, you will not receive any Piazza bonus.**
  - Gradescope: go to https://gradescope.com and use our course code: PY2VVB. Please use your UCLA email. If you create two accounts, please remove the account you will not use. **Please also enter your ID #, which will help us facilitate grading significantly. Also, please format your name as "[First Name] [Last Name]".**
- We uploaded midterm exams dating back to 2020. This is 5 midterm exams.
- We understand that MSOL students are generally not in LA, and therefore the exam is held remotely.

MSOL also graded on separate scale.

Any syllabus questions?

https://piazza.com/class/m5ngohnf8yi4ek/post/6

For coding help, please see TAs!

OFFICE HOURS:

Monday:
Kaifeng: 10a-12p, Eng 4, Room 67-112
Jonathan: 4-5p, Eng 4, Room 56-147H + https://ucla.zoom.us/j/91366715461

Tuesday:
Shawn: 12-2p, Eng 4, Room 67-112
Shreyas: 2-4p, Eng 4, Room 67-112

Wednesday:
Shawn: 11a-1p, Eng 4, Room 67-112
Jonathan: 4-5p, Eng 4, Room 56-147H + https://ucla.zoom.us/j/91366715461
Kaifeng: 4-6p, Eng 4, Room 67-112

Thursday:
Bruce: 9-11a, Eng 4, Room 67-112
Kaifeng: 12-2p, Eng 4, Room 67-112
Shreyas: 2-4p, Eng 4, Room 67-112
Xu Yan: 7-9p, on Zoom @ https://ucla.zoom.us/my/xuyan (MSOL priority)

Friday:
Bruce: 1-3p, Eng 4, Room 67-112

Saturday:
Xu Yan: 9-11a, on Zoom @ https://ucla.zoom.us/my/xuyan (MSOL priority)

Sunday:
Tonmoy: 12-1p, on Zoom @ https://ucla.zoom.us/j/8855338382

DISCUSSION (All Friday):

Tonmoy (DIS 1A): 11:00A-11:50A FOWLER A103B + https://ucla.zoom.us/j/8855338382
Tonmoy (DIS 1B): 12:00P-12:50P BOELTER 5420 + https://ucla.zoom.us/j/8855338382

Lecture 2: Machine learning refresher

This lecture gives a refresher on key concepts from machine learning.

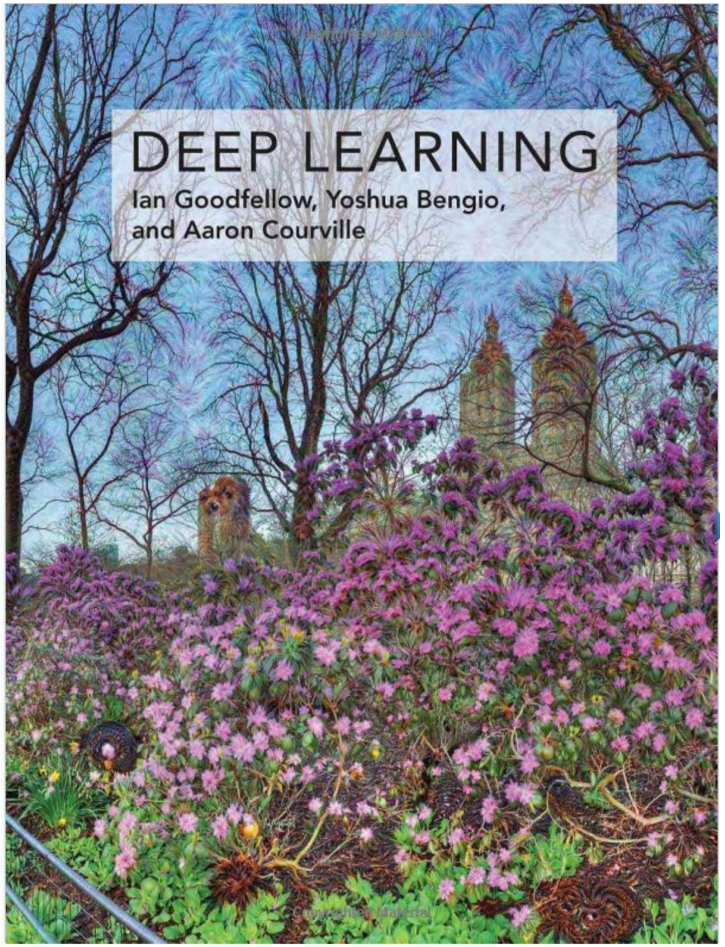
- Introduction to concepts in machine learning
- Cost functions
- Example: linear and polynomial regression
- Model complexity and overfitting
- Training set, validation set, test set
- Dealing with probabilistic cost functions and models
- Example: maximum-likelihood classification

Lecture 2: Basics of machine learning

Reading:

Deep Learning, chapter 5 (up to and including section 5.5).

To refresh your linear algebra and probability, look at chapters 2 and 3 respectively.



Focus of this class:

N examples

$x^{(i)}$ $y^{(i)}$

- This class will focus on **supervised** learning problems.
- This class will also focus on **classification** largely (e.g., when considering convolutional neural networks) as well as some instances of **regression** (e.g., when considering recurrent neural networks)

$y^{(i)}$

Classification

CIFAR-10

$y^{(4)} \leftarrow x^{(4)}$

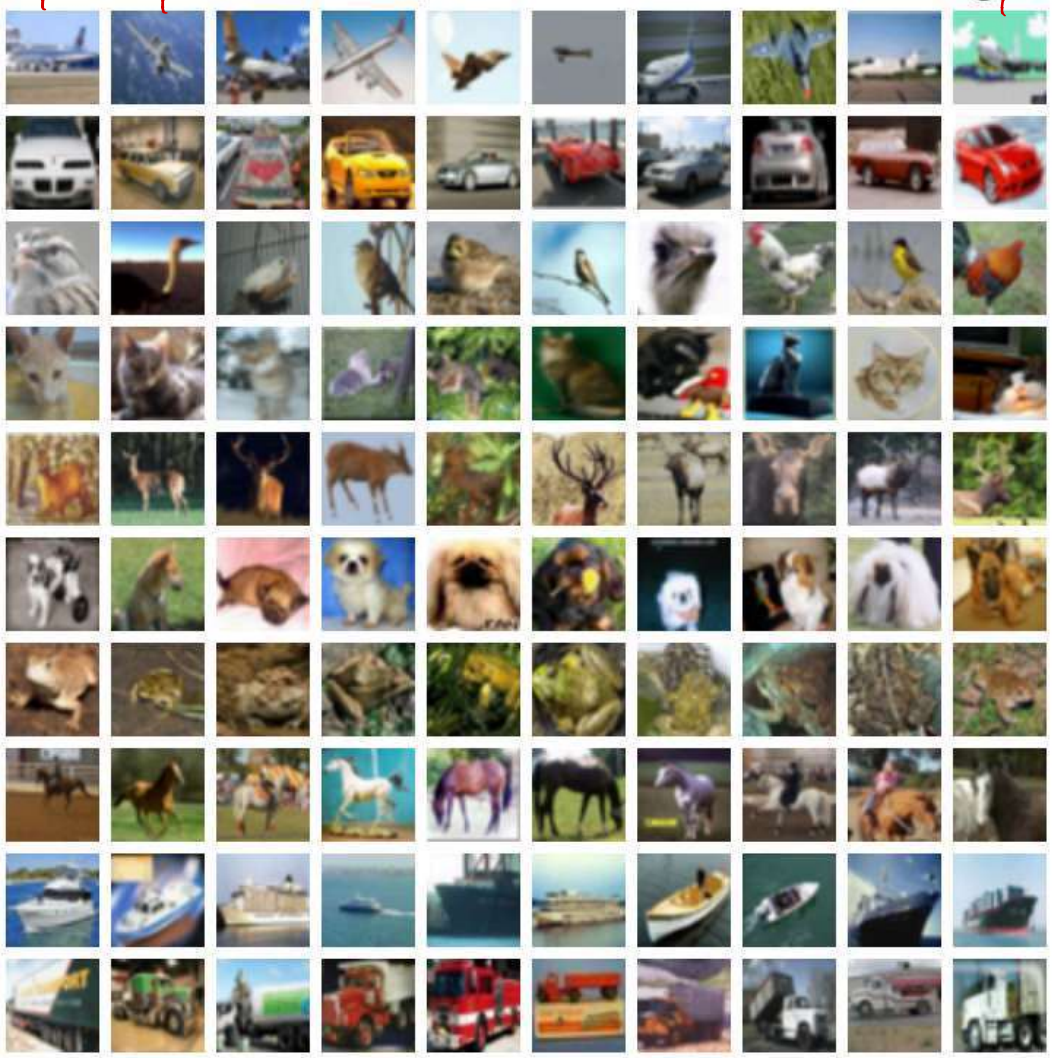
color: r, g, b

$x^{(1)}$ $x^{(2)}$ $x^{(4)}$ $x^{(10)}$ [0, 255]

$y^{(i)} = 1$

airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

10 classes



x

$32 \times 32 \times 3$

3072

$x \in \mathbb{R}$

=

classification

$f(x^{(i)})$

class

what class the image belongs to

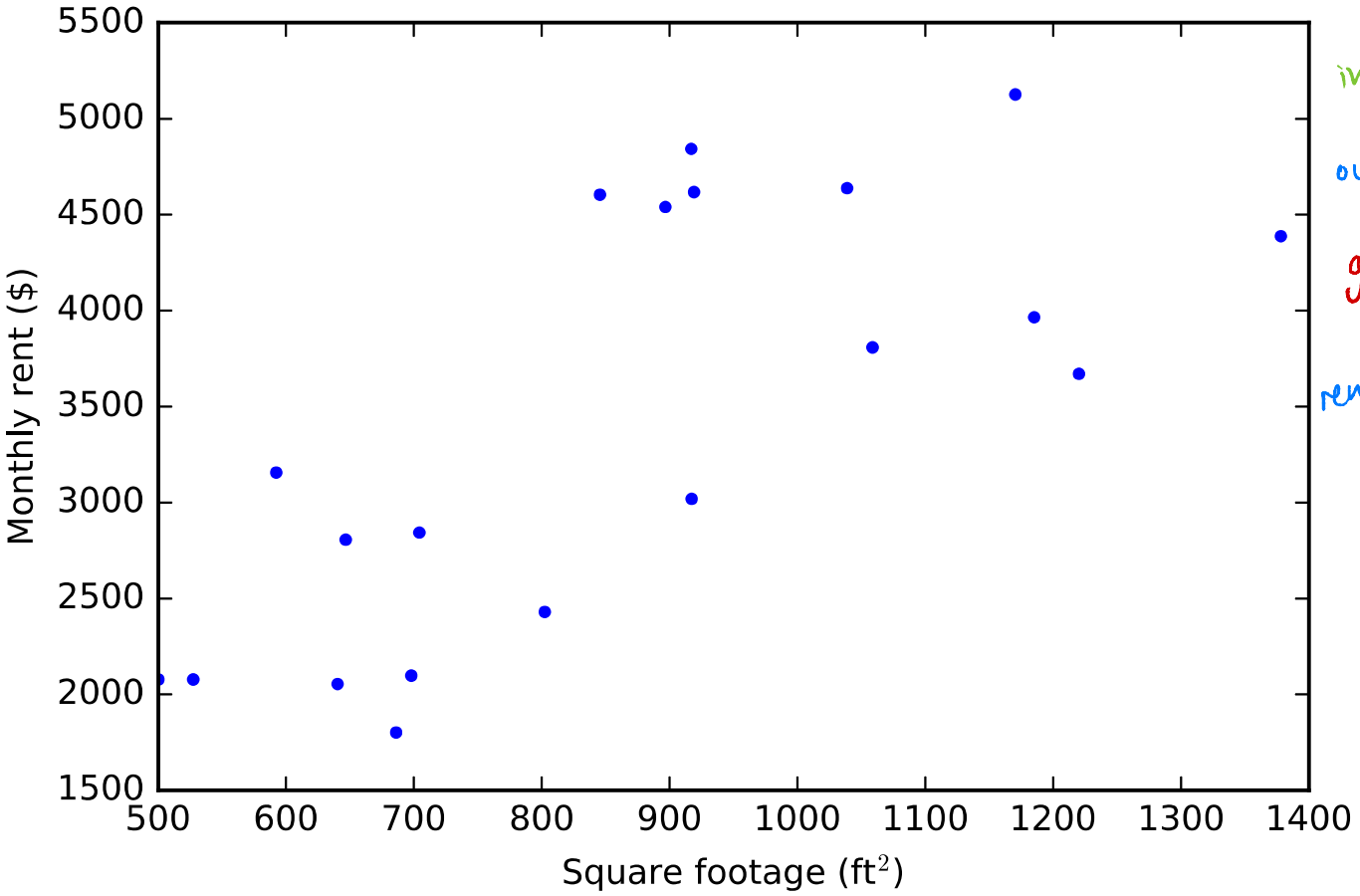
1 out of 10

CIFAR-10 dataset, https://www.cs.toronto.edu/~kriz/cifar.html

Regression

An example of supervised learning

Let's say we want to rent a home in Westwood, and we wanted to know if we were getting a good deal. **(Warning: this data is synthetic!)**



input: sq. ft

output: rent

goal: learn f

rent ← f(sq. ft.)

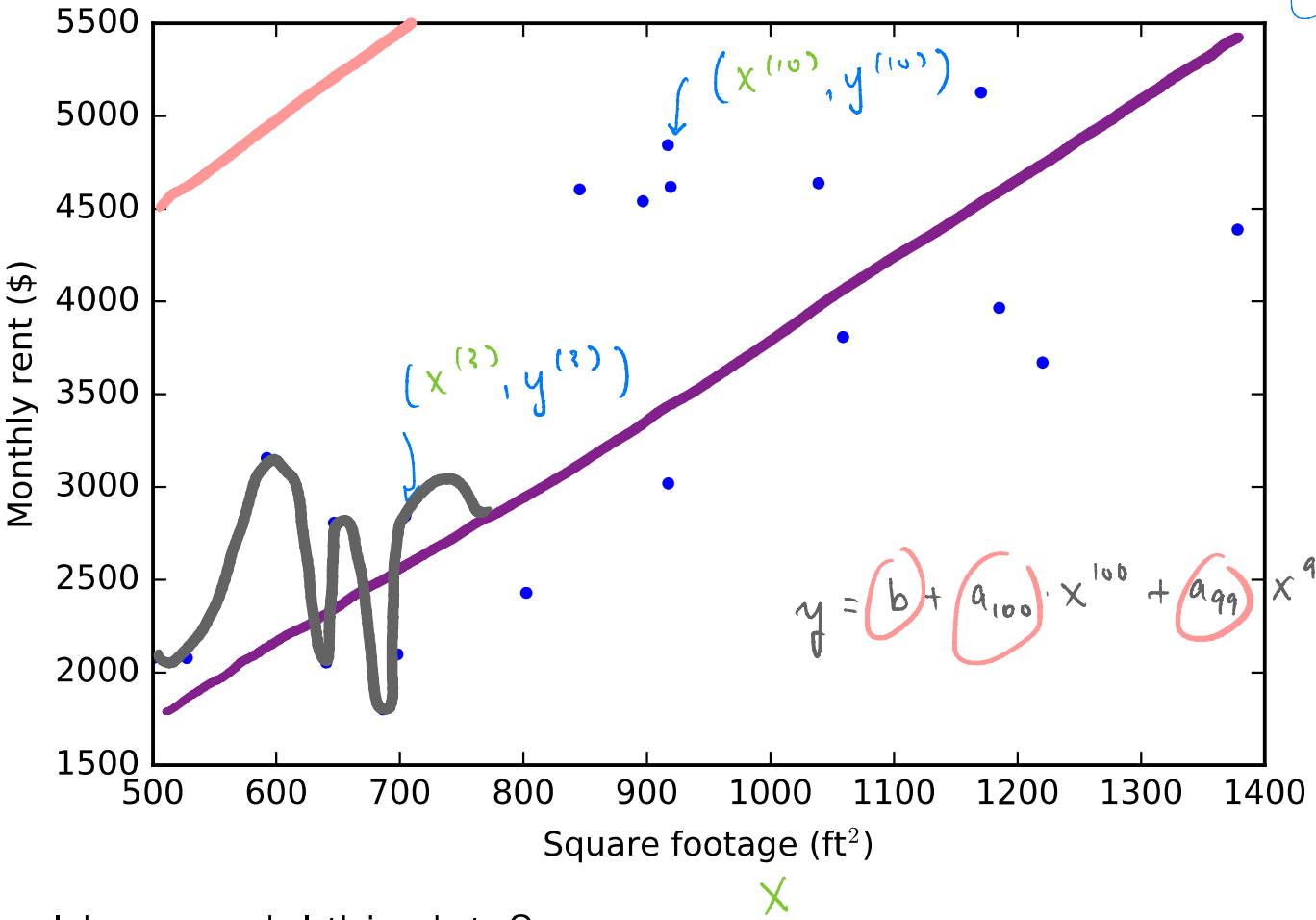
An example of supervised learning

$f(x) = a \cdot x + b$

data given to me

$y = a x + b$

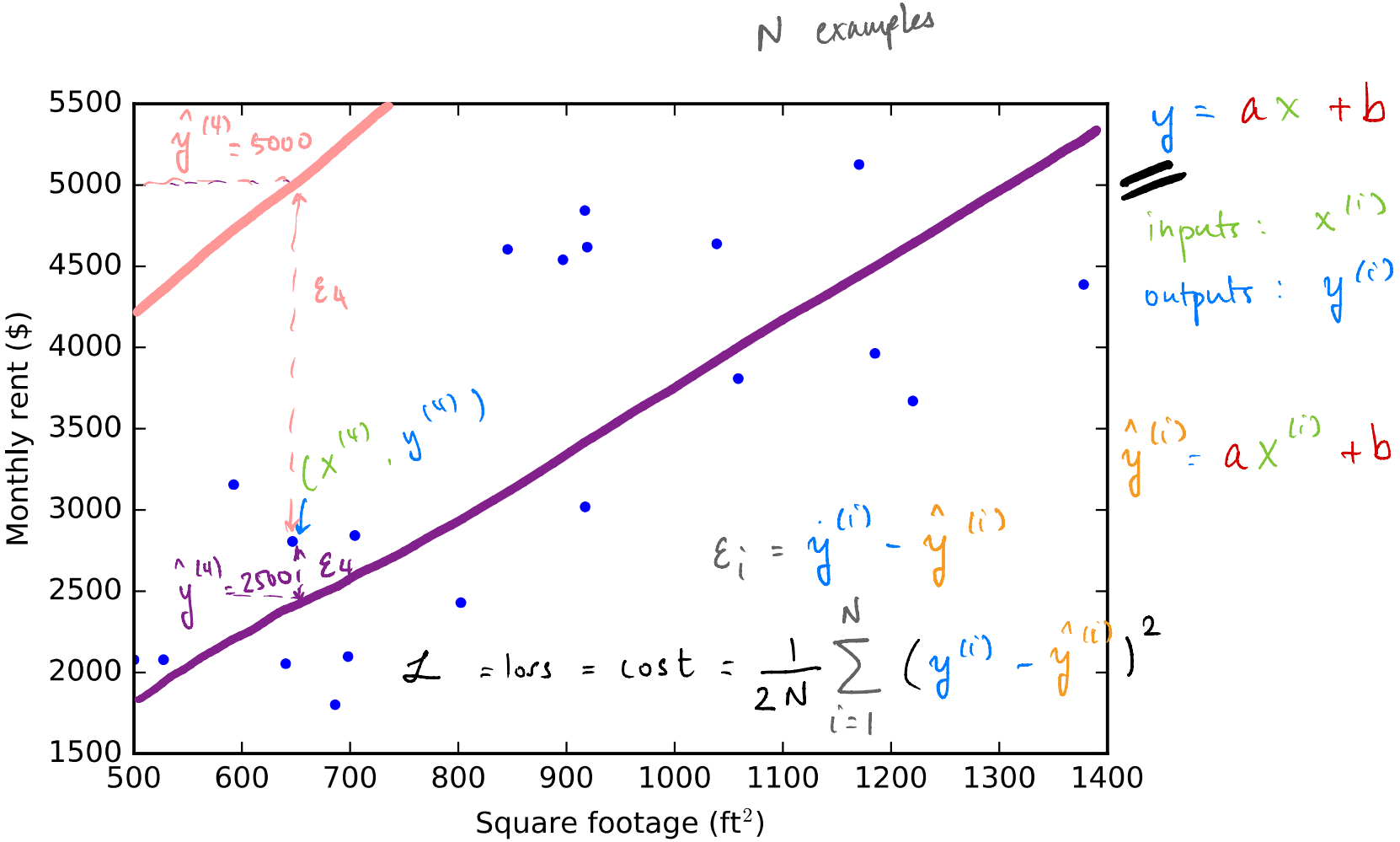
parameters



How should we model this data?

- Inputs, **x**? Outputs, **y**? Data
- What model should we use? f to be NN.
- How do we assess how good our model is?

An example of supervised learning



How should we model this data?

- Inputs, **x**? Outputs, **y**? Data
- What model should we use? f
- How do we assess how good our model is? $\mathcal{L}$

An example of supervised learning

$\hat{y} = ax^2 + bx + c$

$\theta = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \hat{x} = \begin{bmatrix} x^2 \\ x \\ 1 \end{bmatrix}$

A simple linear example:

- Model:

$$\begin{aligned} \hat{y} &= ax + b \\ &= \theta^T \hat{\mathbf{x}} \end{aligned}$$

y: data given to us

parameters: I get to choose a, b to make my loss as small as possible (this makes the model as good as possible.)

- Cost function:

$\theta = \begin{bmatrix} a \\ b \end{bmatrix} \quad \hat{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}$

$\mathcal{L}(a, b)$

$$\begin{aligned} \mathcal{L}(\theta) &= \frac{1}{2N} \sum_{i=1}^{N} (y^{(i)} - \hat{y}^{(i)})^2 \\ &= \frac{1}{2N} \sum_{i=1}^{N} (y^{(i)} - \theta^T \hat{\mathbf{x}}^{(i)})^2 \end{aligned}$$

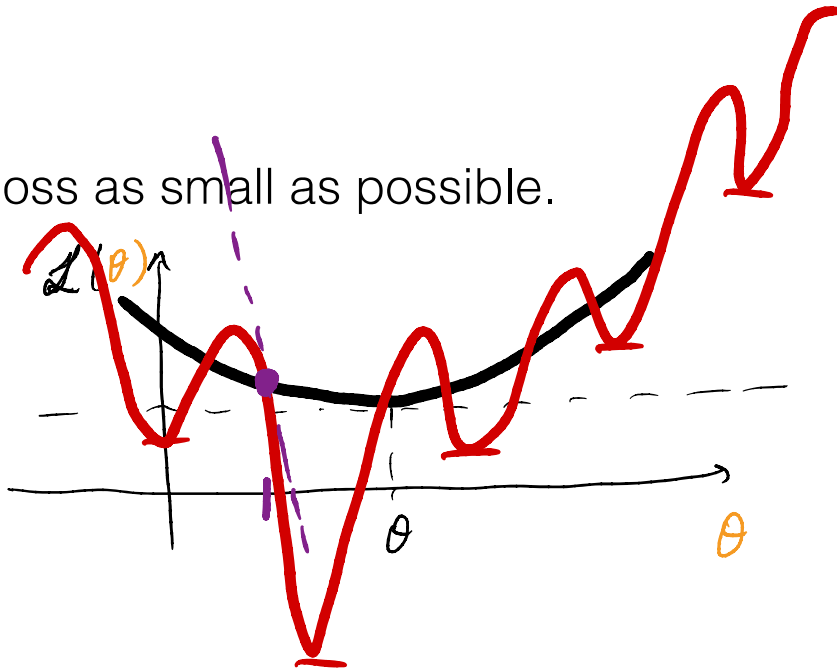
How do we learn the parameters of this model?

An example of supervised learning

Construct it as an optimization problem.

Our goal is to choose the parameters to make the loss as small as possible.

$$\frac{\partial \mathcal{L}}{\partial \theta} = 0$$



Thus our strategy to find the best θ is to:

- Calculate

$$\frac{d\mathcal{L}}{d\theta}$$

- Solve for θ such that

$$\frac{\partial \mathcal{L}}{\partial \theta} = 0$$

However, θ is a vector, so how do we take derivatives with respect to it?

Aside: vector and matrix derivatives

In machine learning, we often take derivatives with respect to vectors and matrices.

These are typically called *gradients*, and in this class we'll use the following notation to denote derivatives with respect to vectors and matrices.

Diff. a scalar y w.r.t. a vector $\vec{x}$

In this class, we will use both the differentiation operator ∂ and ∇ to denote derivatives. If y is a scalar, and $\mathbf{x}$ is a vector, then the gradient of y with respect to $\mathbf{x}$ is denoted as both:

$$\frac{\partial y}{\partial \mathbf{x}} \quad \text{and} \quad \nabla_{\mathbf{x}} y$$

The gradient of y with respect to $\mathbf{x}$ is itself a vector with the same dimensionality as $\mathbf{x}$.

Aside: vector and matrix derivatives

y scalar

x vector $\in \mathbb{R}^n$

$\frac{\partial y}{\partial \vec{x}}$ or $\nabla_{\vec{x}} y \in \mathbb{R}^{n \times 1}$

The gradient

The gradient generalizes the scalar derivative to multiple dimensions. Suppose $f : \mathbb{R}^n \to \mathbb{R}$ transforms a vector $\mathbf{x} \in \mathbb{R}^n$ to a scalar. If $y = f(\mathbf{x})$, then the gradient is:

$$\nabla_{\mathbf{x}} y = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\frac{\partial y}{\partial x} = \nabla_x y = \begin{bmatrix} 1 \\ 0.5 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$\frac{\partial y}{\partial x_1} = 1$

$\frac{\partial y}{\partial x_2} = 0.5$

Δy due to Δx_1 is $\approx \frac{\partial y}{\partial x_1} \cdot \Delta x_1$

Aside: vector and matrix derivatives

$\in \mathbb{R}^{1 \times n}$ $\quad x \in \mathbb{R}^{n \times 1}$

$y = [\quad][\quad]$

In other words, the gradient is:

- A *vector* that is the same size as $\mathbf{x}$, i.e., if $\mathbf{x} \in \mathbb{R}^n$ then $\nabla_{\mathbf{x}} y \in \mathbb{R}^n$.
- Each dimension of $\nabla_{\mathbf{x}} y$ tells us how small changes in $\mathbf{x}$ in that dimension affect y. i.e., changing the ith dimension of $\mathbf{x}$ by a small amount, Δx_i, will change y by

$$\frac{\partial y}{\partial x_i} \Delta x_i$$

We may also denote this as:

$$(\nabla_{\mathbf{x}} y)_i \, \Delta x_i$$

$$\Delta x = \begin{bmatrix} 0.05 \\ 0.01 \\ 0.02 \\ \vdots \\ \cdot \end{bmatrix} \rightarrow \text{how much does } y \text{ change? } (\Delta y)$$

$$\Delta y \approx \frac{\partial y}{\partial x_1} \Delta x_1 + \frac{\partial y}{\partial x_2} \Delta x_2 + \frac{\partial y}{\partial x_3} \Delta x_3 + \cdots$$

$$\Delta y \approx \sum_{i=1}^{n} \frac{\partial y}{\partial x_i} \Delta x_i$$

$$\Delta y \approx (\nabla_x y)^T \Delta x$$

Aside: vector and matrix derivatives

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Example: derivative with respect to a vector

Let $f(\mathbf{x}) = \theta^T \mathbf{x}$. What is $\nabla_{\mathbf{x}} f(\mathbf{x})$?

$\theta, x \in \mathbb{R}^n$

$y = \theta^T x \qquad \nabla_x y \qquad y \in \mathbb{R} \qquad \nabla_x y \in \mathbb{R}^n$

$$y = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_n x_n$$

$$\nabla_x y = \begin{bmatrix} \partial y / \partial x_1 \\ \partial y / \partial x_2 \\ \vdots \\ \partial y / \partial x_n \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} = \theta$$

$$\nabla_x y = \theta$$

$$\nabla_\theta y = x$$

Aside: vector and matrix derivatives

Example: derivative with respect to a vector

Let $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A}\mathbf{x}$. What is $\nabla_{\mathbf{x}} f(\mathbf{x})$?

$(1\times n)\,(n\times n)\,(n\times 1)$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdots & & a_{nn} \end{bmatrix}$$

$x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$

$f(x) \in \mathbb{R}$, $\nabla_x f(x) \in \mathbb{R}^n$

$$f(x) = x^T A x = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i \cdot a_{ij} \cdot x_j$$

$$\frac{\partial f(x)}{\partial x_1} = 2a_{11}x_1 + \sum_{j=2}^{n} a_{1j}x_j + \sum_{i=2}^{n} a_{i1}x_i$$

$$= \sum_{j=1}^{n} a_{1j}x_j + \sum_{i=1}^{n} a_{i1}x_i$$

$i=1,\ j=1$

$$\frac{\partial\, a_{11}x_1^2}{\partial x_1} = 2a_{11}x_1$$

$i=1,\ j\neq 1$

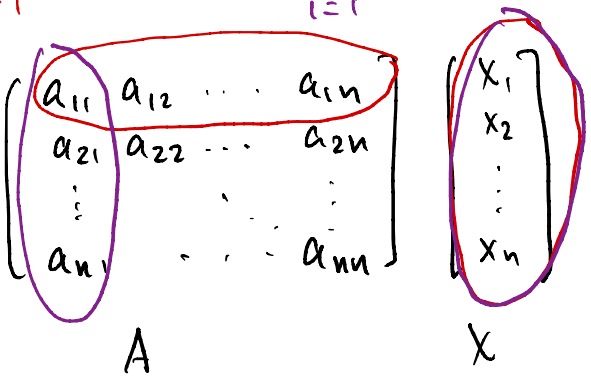
$$\frac{\partial \sum_{j=2}^{n} x_1 \cdot a_{1j} \cdot x_j}{\partial x_1} = \sum_{j=2}^{n} a_{1j} \cdot x_j$$

$i\neq 1,\ j=1$

$$\frac{\partial \sum_{i=2}^{n} x_i \cdot a_{i1} x_1}{\partial x_1} = \sum_{i=2}^{n} a_{i1} \cdot x_i$$

Aside: vector and matrix derivatives

$$\frac{\partial f(x)}{\partial x_1} = \sum_{j=1}^{n} a_{1j} x_j + \sum_{i=1}^{n} a_{i1} \cdot x_i$$



$$(Ax)_1 + (A^T x)_1$$

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix} = \begin{bmatrix} (Ax)_1 + (A^T x)_1 \\ (Ax)_2 + (A^T x)_2 \\ \vdots \\ (Ax)_n + (A^T x)_n \end{bmatrix}$$

$\frac{\partial f(x)}{\partial x_2}$, $\frac{\partial f(x)}{\partial x_3}$, ..., all follow the same pattern

$$\nabla_x f(x) = Ax + A^T x$$

$$= (A + A^T) x$$

if A is symmetric

$$= 2Ax$$

$(n \times n)(n \times 1)$

when $n = 1$

$$f(x) = x \cdot a \cdot x = ax^2$$

$$\frac{\partial f(x)}{\partial x} = 2ax$$

Aside: vector and matrix derivatives

First, we note that $\mathbf{x}^T\mathbf{A}\mathbf{x} = \sum_i \sum_j a_{ij} x_i x_j$. Then, we have

$$\begin{aligned}
\nabla_{\mathbf{x}} f(\mathbf{x}) &= \begin{bmatrix} 2a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + a_{21}x_2 + \cdots + a_{n1}x_n \\ 2a_{22}x_2 + a_{21}x_1 + \cdots + a_{2n}x_n + a_{12}x_1 + \cdots + a_{n2}x_n \\ \vdots \\ 2a_{nn}x_n + a_{n1}x_1 + \cdots + a_{n,n-1}x_{n-1} + a_{1n}x_1 + \cdots + a_{n-1,n}x_{n-1} \end{bmatrix} \\
&= \mathbf{A}\mathbf{x} + \mathbf{A}^T\mathbf{x}
\end{aligned}$$

Matrix derivatives

$y = z^T A x \qquad z \in \mathbb{R}^m \qquad x \in \mathbb{R}^n \qquad A \in \mathbb{R}^{m \times n}$

$$\frac{\partial y}{\partial A} = \begin{bmatrix} \frac{\partial y}{\partial a_{11}} & \frac{\partial y}{\partial a_{12}} & \cdots & \frac{\partial y}{\partial a_{1n}} \\ \frac{\partial y}{\partial a_{21}} & \frac{\partial y}{\partial a_{22}} & \cdots & \frac{\partial y}{\partial a_{2n}} \\ \vdots & & \ddots & \vdots \\ \frac{\partial y}{\partial a_{m1}} & \cdots & & \frac{\partial y}{\partial a_{mn}} \end{bmatrix}$$

$\nabla_A y \in \mathbb{R}^{m \times n}$

$z \in \mathbb{R}^p \quad A \in \mathbb{R}^{m \times n} \quad \frac{\partial z}{\partial A} \in \mathbb{R}^{m \times n \times p}$

"Denominator Layout"

$x \in \mathbb{R}^{n \times 1} \qquad \nabla_x y \in \mathbb{R}^{n \times 1}$

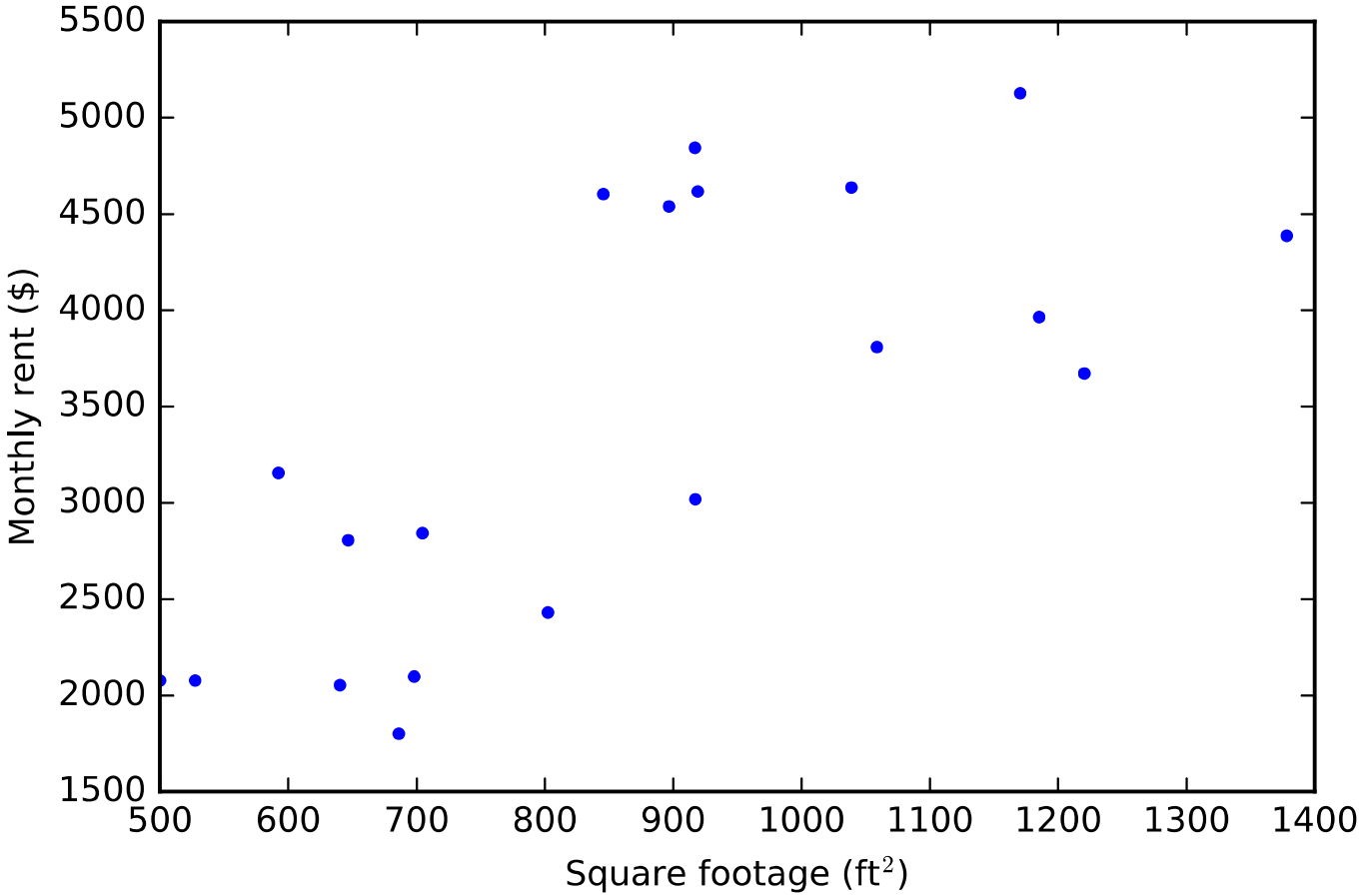
$A \in \mathbb{R}^{m \times n} \qquad \nabla_A y \in \mathbb{R}^{m \times n}$

"Numerator Layout"

$x \in \mathbb{R}^{n \times 1} \qquad \nabla_x y \in \mathbb{R}^{1 \times n}$

$A \in \mathbb{R}^{m \times n} \qquad \nabla_A y \in \mathbb{R}^{n \times m}$

Back to our supervised learning example



Thus our strategy to find the best θ is to:

- Calculate

$$\frac{d\mathcal{L}}{d\theta}$$

- Solve for θ such that

$$\frac{\partial \mathcal{L}}{\partial \theta} = 0$$