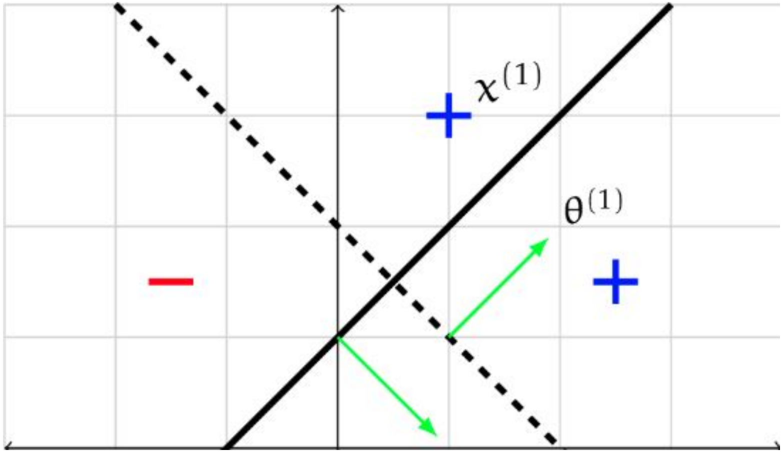
Fall 2024!

Introduction to Machine Learning

https://introml.mit.edu



6.390 Fall 2024 Team
6.390-personal@mit.edu

Outline for today

1. Quick intros to the teaching team
2. Course logistics
3. What we're teaching: Machine Learning!
4. Regression in a nutshell

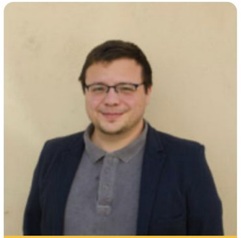
Outline for today

1. Quick intros to the teaching team
2. Course logistics
3. What we're teaching: Machine Learning!
4. Regression in a nutshell

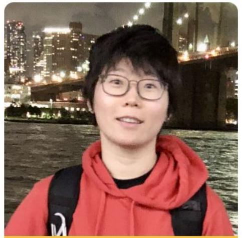
Instructor Team


Ike Chuang

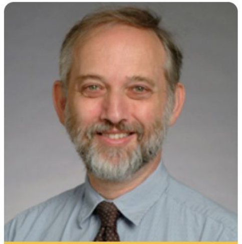

Alexandre Megretski

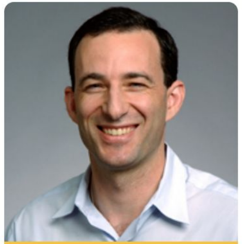

Vince Monardo


Mardavij Roozbehani


Shen Shen


Tess Smidt


Pete Szolovits


Bruce Tidor

Course Assistant


Taylor Braun

Logistical issues? Personal concerns? We'd love to help out at 6.390-personal@mit.edu


Mauricio Barba


Abhay Basireddy

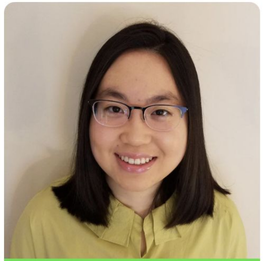

Kevin Bunn


Shaunticlair Ruiz


Yogi Sragow


Yan Wu

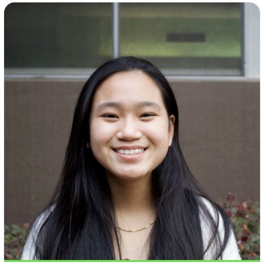

Audrey Douglas


Song Kim


Kartikesh Mishra


Elisa Xia


Haley Nakamura


Anh Nguyen

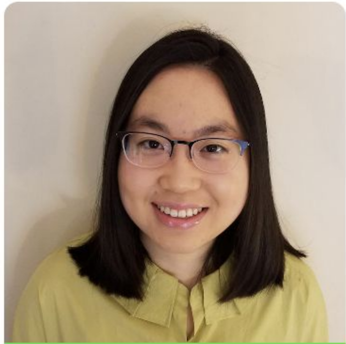

Linh Nguyen

and ~40 awesome LAs

Section 1 staff

Ike Chuang

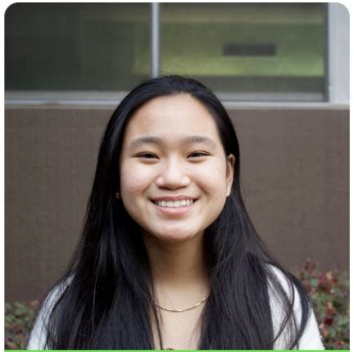
Recitation + Lab



Song Kim

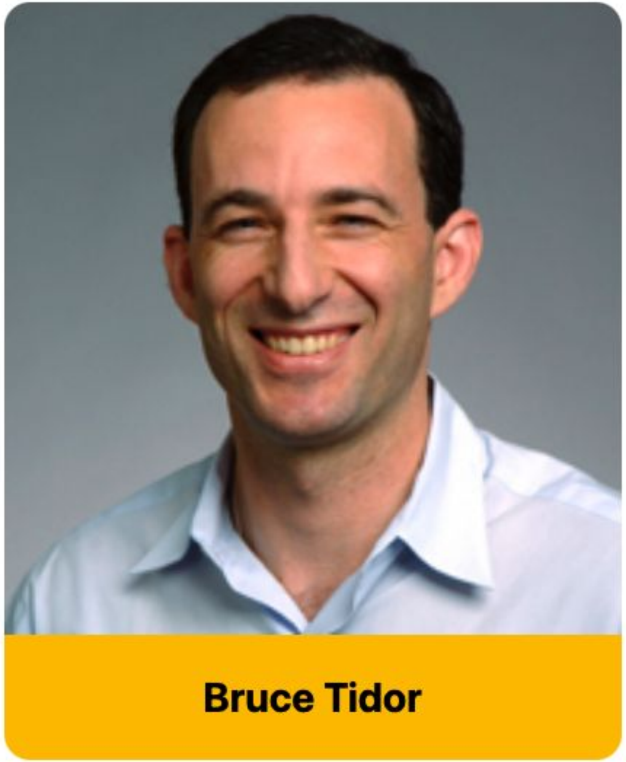
Lab plus ~8 awesome (TAs + LAs)

Yan Wu

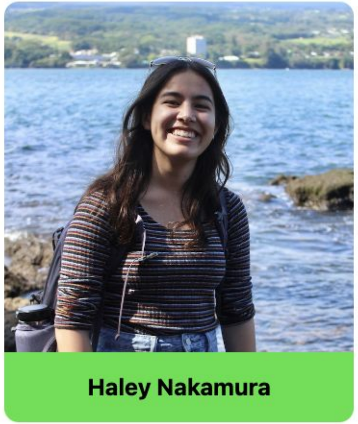


Anh Nguyen

Section 2 staff

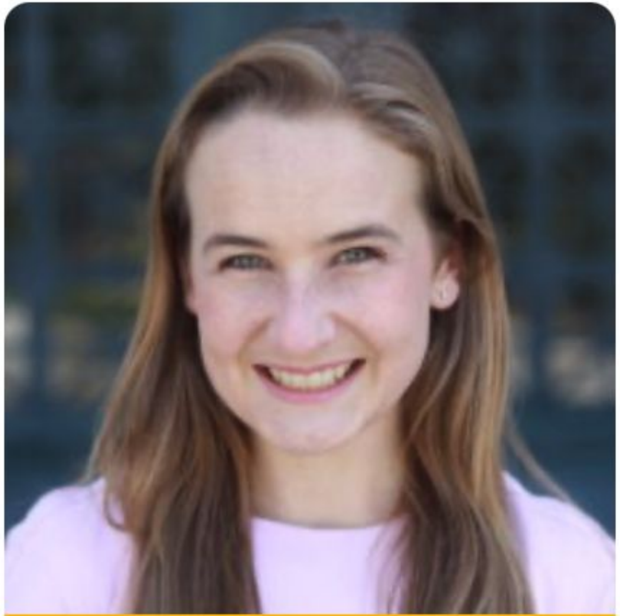


Recitation + Lab



Lab plus ~8 awesome (TAs + LAs)

Section 3 staff

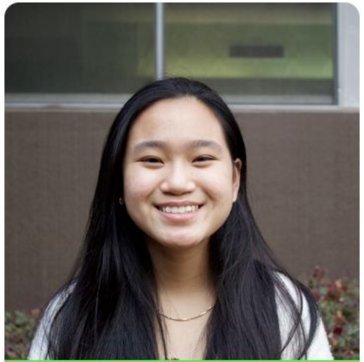


Tess Smidt

Recitation + Lab

Yan Wu

Lab plus ~8 awesome (TAs + LAs)



Anh Nguyen

Kartikesh Mishra

Section 4 staff

Mardavij Roozbehani

Recitation + Lab

Linh Nguyen

Abhay Basireddy

Lab plus ~ 7 awesome (TAs + LAs)

Elisa Xia

Yogi Sragow

Section 5 staff

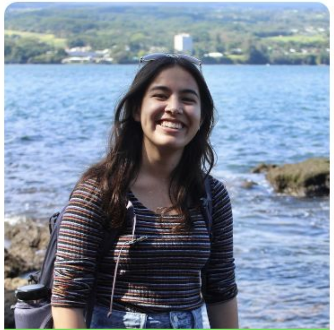
Shen Shen

Recitation + Lab

Kartikesh Mishra

Lab

Mauricio Barba



Haley Nakamura

plus ~ 7 awesome LAs

Section 6 staff

Alexandre Megretski

Recitation + Lab

Yogi Sragow

Lab plus ~ 7 awesome (TAs + LAs)

Audrey Douglas

Kevin Bunn

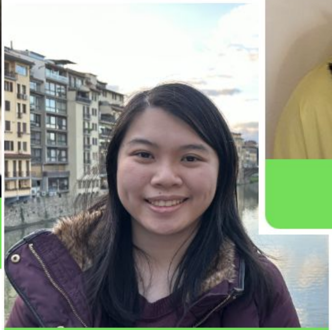
Section 7 staff

Pete Szolovits

Recitation + Lab

Kevin Bunn

Lab plus ~8 awesome (TAs + LAs)



Elisa Xia

Song Kim

Audrey Douglas

Outline for today

1. Quick intros to the teaching team
2. Course logistics
3. What we're teaching: Machine Learning!
4. Regression in a nutshell

Course pedagogy:

A nominal week – mix of theory, concepts, and application to problems!

- **Exercises:** Releases on Wed. 5pm, due the following Mon. 9am

  Relatively easy questions based on that week's notes reading (and viewing lecture)
- **Lecture:** Fri., no attendance check-in. noon-1pm in 45-230. Will be live-streamed.

  Overview the technical contents, and tie together the high-level motivations, concepts, and stories.
- **Recitation**: Mon., with attendance check-in

  Assumes you have read and done exercises; start on homework.

  (Solutions posted on class day at 5pm.)
- **Homework:** Releases Mon. 9am; due Wed. (9 days later) at 11pm

  Harder questions: concepts, mechanics, implementations
- **Lab:** Wednesday, with attendance check-in *(not today)*

  In-class empirical exploration of concepts, Work with partner(s) on lab assignment

  Check-off conversation with staff member, due the following Mon. 11pm

Grading and collaboration (details on web)

Our **objective** (and we hope yours) is **for you to learn about machine learning**

- take responsibility for your understanding
- we will help!

Formula:

exercises 5% + attendance 5% + homework 15% + labs 15% + midterm 25% + final 35%

Lateness: 20% penalty per day, applied linearly (so 1 hour late is -0.83%)

Extensions:

- **20 one-day extensions** (extend one assignment's deadline by one full day)
- will be **applied automatically** at the end of the term in a way that is maximally helpful
- for medical or personal difficulties see S^3 & contact us at `6.390-personal@mit.edu`

Collaboration: don't cheat!

- Understand everything you turn in
- Coding and detailed derivations must be done by you
- See collaboration policy/examples on course web site

How to get help

- Office hours: lots! (Starting Sun. Sep 8)
- Schedule details on OHs page (Instructors OHs schedule TBD).
- See Calendar page for holiday/schedule shift.
- Make use of Piazza and Pset-partners!

Exams

- Midterm: Wednesday, October 23: 7:30pm-9:30 pm.
- Final: scheduled by Registrar (posted in 3rd week). ***ALERT – might be as late as Dec 20!***

Expected prerequisite background

Things we expect you to know (we use these constantly, but don't teach them explicitly):

Programming (e.g. as in 6.101[009] or 6.121[006])

- Intermediate Python, including classes
- Exposure to algorithms – ability to understand & discuss pseudo-code, and implement in Python

Linear Algebra (e.g. as in 18.06, 18.C06, 18.03, or 18.700)

- Matrix manipulations: transpose, multiplication, inverse etc.
- Points and planes in high-dimensional space
- (Together with calculus): taking gradients, matrix calculus

Useful background

Things it helps to have prior exposure to, but we don’t expect (we use these in 6.390, but will discuss as we go):

- numpy (Python package for matrix/linear algebra)
- pytorch (python package for modern ml models like deep neural networks)
- Basic discrete probability: random variables, independence, conditioning

Heads-up for Monday

- Starting Monday Sep 9, attend only your <u>assigned section</u>
- If you need to change your permanent section assignment, you will be able to self-switch, starting 5pm today; details on introml homepage

Rest of Today

- Start our ML journey with an overview
- Work through your first lab
- Ask questions by putting yourself in the help queue
- No worries if no introml access yet; great chance to know your neighbor (ask them to put you in the queue)

Outline for today

1. Quick intros to the teaching team
2. Course logistics
3. What we're teaching: Machine Learning!
4. Regression in a nutshell

What we're teaching: Machine Learning!

Given:

- a **collection of examples** (gene sequences, documents, …)
- an **encoding of those examples** in a computer (as vectors)

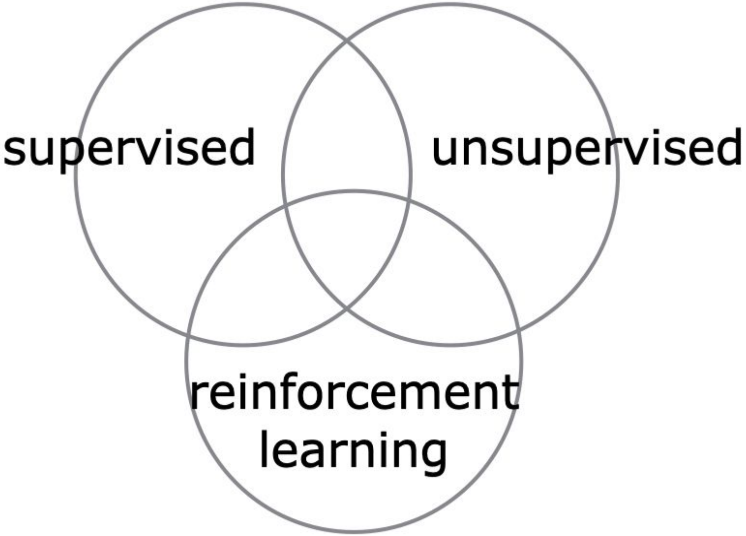
Derive:

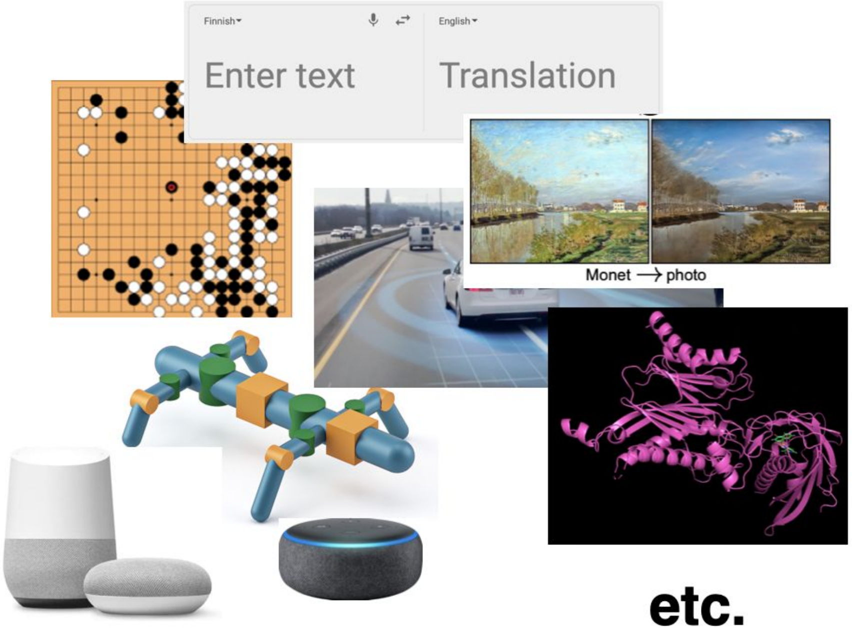
- a **computational model** (called a hypothesis) that describes relationships within and among the examples that is expected to characterize well new examples from that same population, to make good predictions or decisions

A model might:

- **classify images** of cells as to whether they're cancerous
- **specify groupings (clusters)** of documents that address similar topics
- **steer** a car appropriately given lidar images of the surroundings

Very roughly, ML can be categorized into





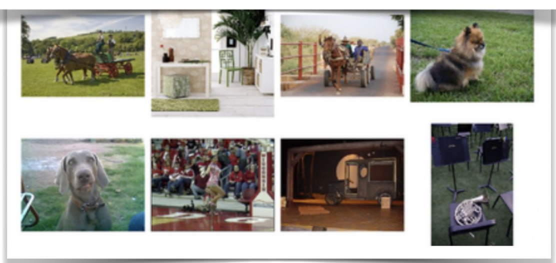
etc.

(the categorization can be refined, e.g. there are active learning, semi-supervised, selective, contrastive, few-shot, inverse reinforcement learning…)

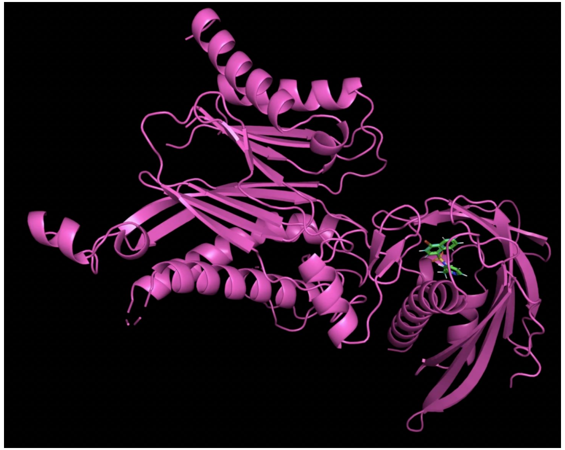
[Slides adapted from 6.790]

Supervised learning

Goal: correctly classify so far unseen test images



Goal: predict to what degree a drug candidate binds to the intended target protein (based on a dataset of already-screened molecules against the target)



- Learning a machine translation system from pairs of sentences

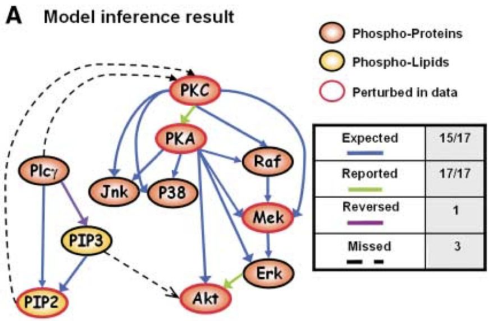
<u>**Spanish (input)**</u>	<u>**English (output)**</u>
Aquí tienes un bolígrafo	Here’s a pen
Las conferencias de ML son divertidas	ML conferences are fun
Todo el mundo debería estudiar AI	Everyone should study AI
…	…

[Slides adapted from 6.790]

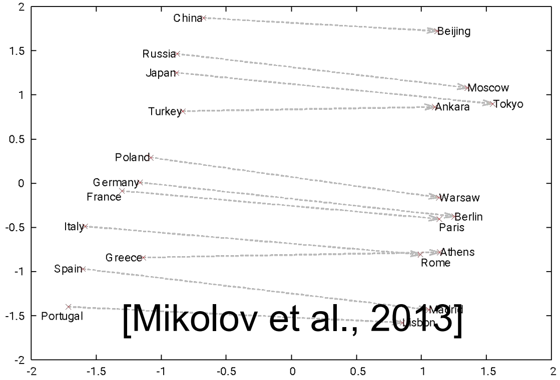
Unsupervised learning

dimensionality reduction, embedding

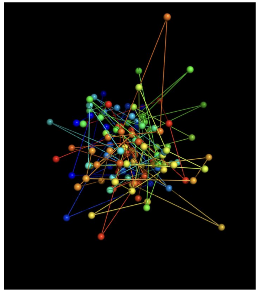
dependency /causal structure

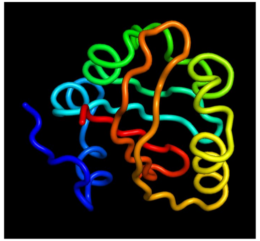


[Sachs et al 05]



[Mikolov et al., 2013]



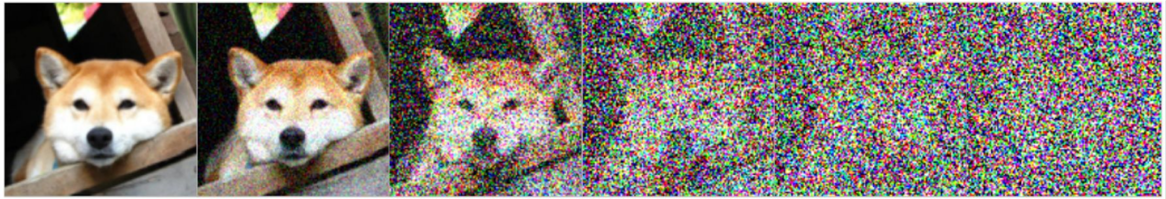


[courtesy of Jason Yim]

Over 3D protein structures, etc.

+Self-Supervised paradigm

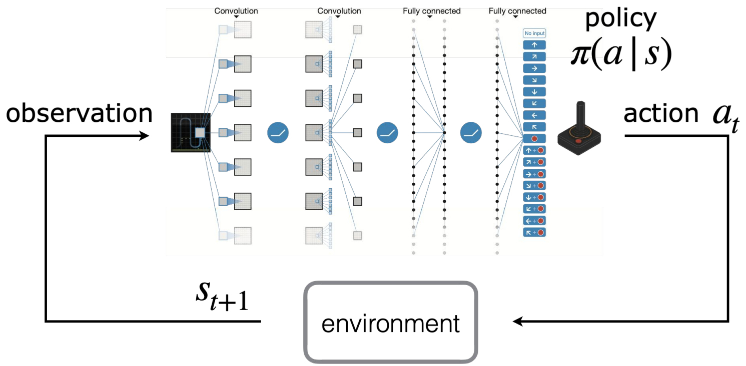
de-noising diffusion models over images

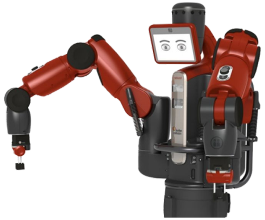


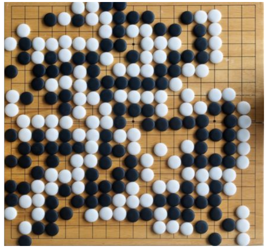
[image from Rissanen et al 2022]

[Slides adapted from 6.790]

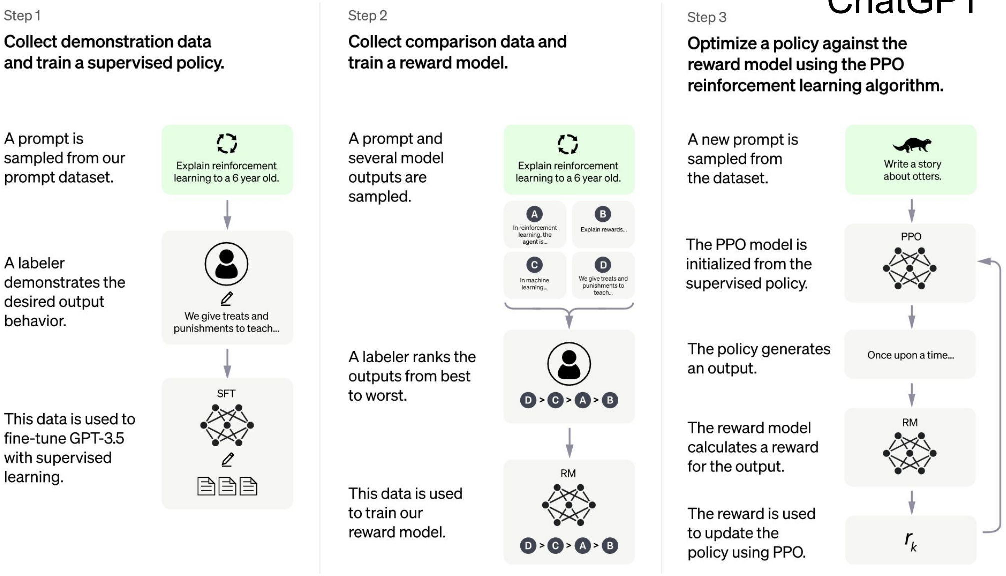
Reinforcement learning

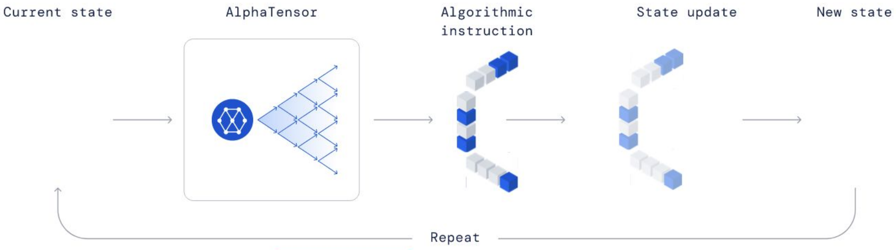






ChatGPT





Single-player game played by AlphaTensor, where the goal is to find a correct matrix multiplication algorithm. The state of the game is a cubic array of numbers (shown as grey for 0, blue for 1, and green for –1), representing the remaining work to be done.

[Slides adapted from 6.790]

Machine learning (ML): why & what

- **What is ML?**
  - Roughly, a set of methods for making predictions and decisions from data.
- **Why study ML?**
  - To apply; to understand; to evaluate; to create
- **What do we have?**
  - Data! And computation!
- **What do we want?**
  - To make predictions on new data!
- **How do we learn to make those decisions?**
  - The topic of this course!

Outline for today

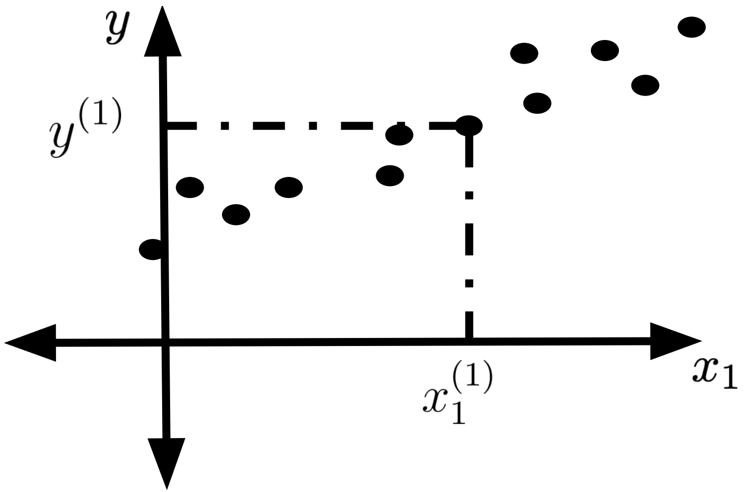
1. Quick intros to the teaching team
2. Course logistics
3. What we're teaching: Machine Learning!
4. Regression in a nutshell

What do we have?

- There are many different problem classes in ML
  - We will first focus on an instance of supervised learning known as **regression**.

(Training) data

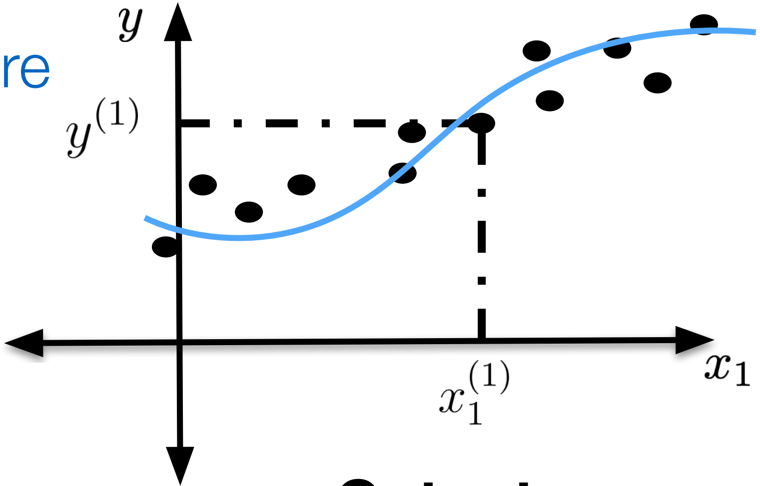
- n training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector
    $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

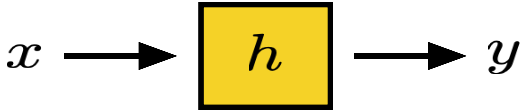


What do we want?

We want a "good" way to label new feature vectors

- How to label? Learn a hypothesis
- We typically consider a class of possible hypotheses



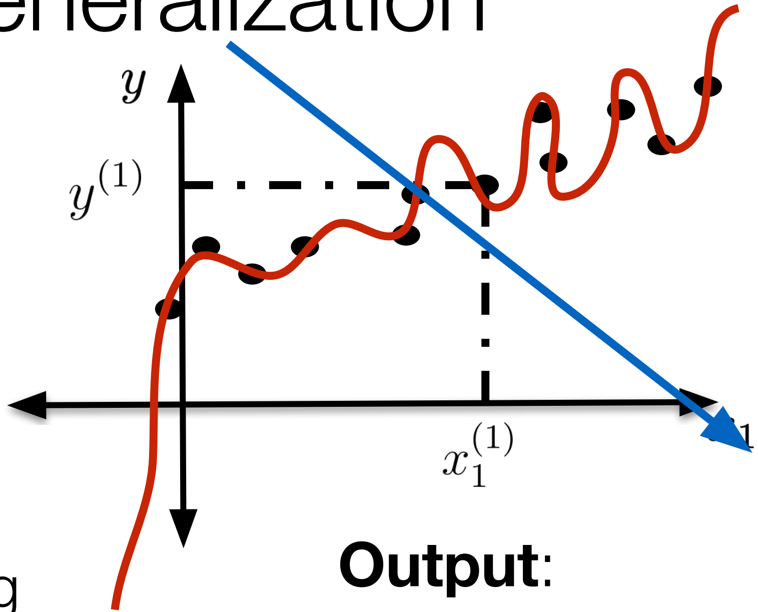


how well our hypothesis labels new feature vectors depends largely on how expressive the hypothesis class is

Warning: Overfitting vs. Generalization

What we really want is to generalize to **future data**!

- What we don't want:
  - Model does not capture the input-output relationship $\rightarrow$ **Underfitting**
  - Model too specialized to training data $\rightarrow$ **Overfitting**



Output:
Label

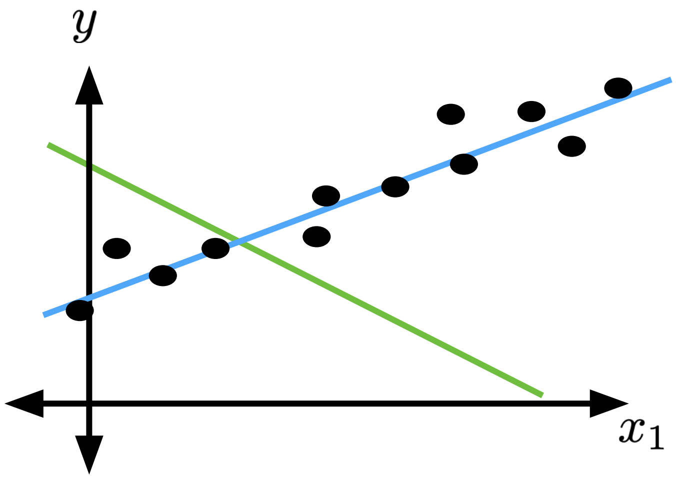
What do we want?

We may consider the class of linear regressors:

- Hypotheses take the form:

$$h(x; \underbrace{\theta, \theta_0}) = \theta^\top x + \theta_0$$

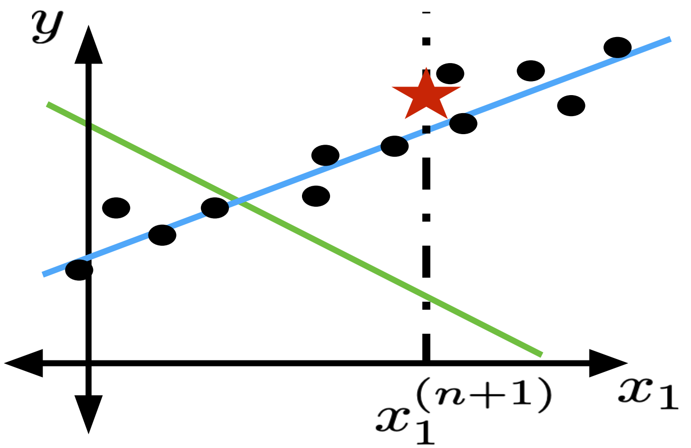
Generally, we might refer to the set of all learned parameters as $\boldsymbol{\Theta}$ (capital θ)



How **good** is a hypothesis?

Hopefully predict well on *future data*

How good is a regressor at one point?



- Quantify the error using a ***loss function***, $\mathcal{L}(g, a)$

g: guess
a: actual

- Common choice: squared loss:

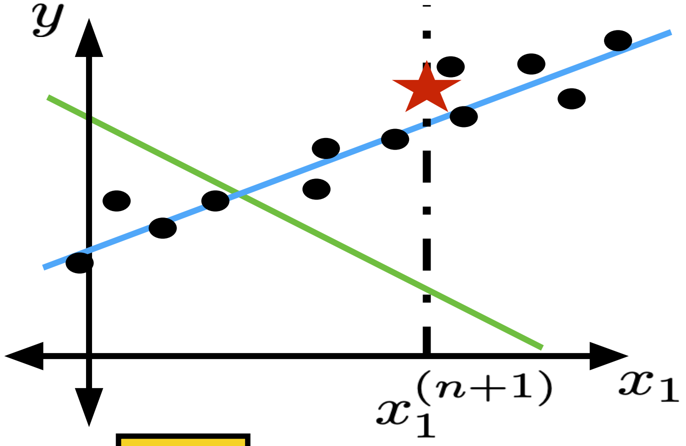
$$\mathcal{L}(g, a) = (g - a)^2$$

h: hypothesis function (outputs *g*)
x: input, θ: parameters, *y*: actual

- **Training error**: $\mathcal{E}_n(h; \Theta) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(h(x^{(i)}; \Theta), y^{(i)}\right)$

- **Validation or Test error** (*n'* new points): $\mathcal{E}(h) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} \mathcal{L}\left(h(x^{(i)}), y^{(i)}\right)$

How do we learn?



- Have data; have hypothesis class
- Want to choose (learn) a good hypothesis (a set of parameters)

What we want: $x \rightarrow \boxed{h} \rightarrow y$

How to get it:
(Next time!)

$$\boxed{\mathcal{D}_n \rightarrow \boxed{\text{learning algorithm}} \rightarrow h}$$

Outline for today

1. Quick intros to the teaching team
2. Course logistics
3. What we're teaching: Machine Learning!
4. Regression in a nutshell

Your turn:

Let’s do a quick lab

Heads-up for Monday

- Starting Monday Sep 9, attend only your assigned section
- If you need to change your permanent section assignment, you will be able to self-switch, starting 5pm today; details on introml homepage

Rest of Today

Goto: https://introml.mit.edu

- Start our ML journey with an overview
- Work through your first lab
- Ask questions by putting yourself in the help queue
- No worries if no introml access yet; great chance to know your neighbor (ask them to put you in the queue)

Rest of Today

Goto: https://introml.mit.edu

1. Lab attendance check: enter today's section passcode (see board)
2. Create / join a group
3. Work through lab

- Ask questions by putting yourself in the help queue
- No worries if no introml access yet; great chance to know your neighbor (ask them to put you in the queue)