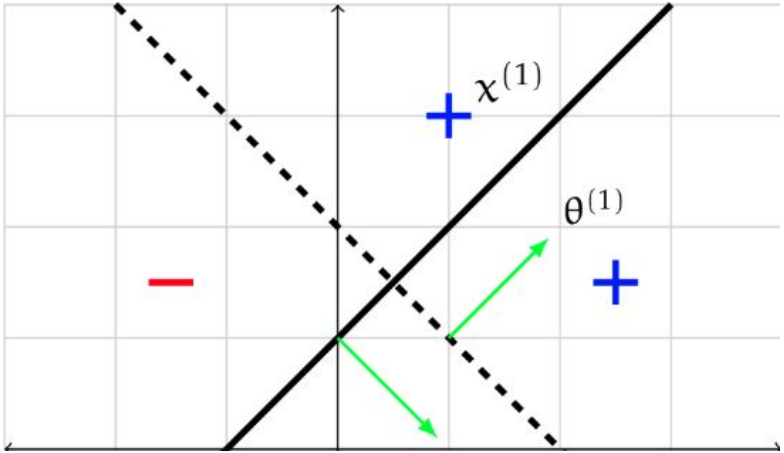
Fall 2023!

6.390

# Introduction to Machine Learning

https://introml.mit.edu




The 6.390 Instructor Team
6.390-personal@mit.edu

## Instructors

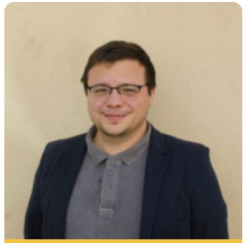
Duane Boning

Ike Chuang

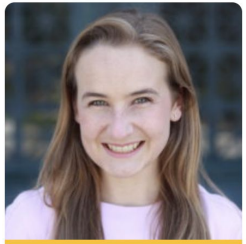
Kyle Keane

Wojciech Matusik

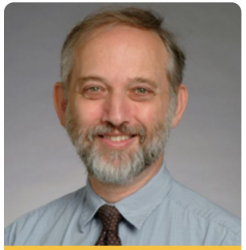
Alexandre Megretski



Vince Monardo



Tess Smidt



Peter Szolovits

**Course Assistant**

**Taylor Braun**

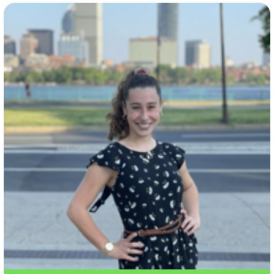
Logistical issues? Personal concerns?
We'd love to help out at 6.390-personal@mit.edu

George Bian

Kevin Bunn

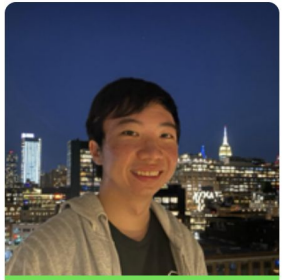
Jessica Ding

Linh Nguyen



Sage Simhon

Yogi Sragow



Dewei Feng

Emily Jiang

Prabhakar Kafle

George Tang

Warren Wang

DongHun Ryu

Mimi Lohanimit

Helen Merker

Haley Nakamura

and ~50 awesome LAs!

# Course pedagogy:

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

- **Exercises:** Released on Wed., due following Mon. 9AM
  Easy questions based on that week's notes reading (and viewing optional recorded lecture)
- **Recitation**: Monday, with attendance check-in
  Assumes you have read and done exercises; start on homework
- **Homework:** Releases Monday morning; due Wednesday (9 days later) at 11PM
  Harder questions: concepts, mechanics, implementations
- **Lab:** Wednesday, with attendance check-in (starting today)
  In-class empirical exploration of concepts
  Work with partner on lab assignment
  **Check-off** conversation with staff member

Office hours: **lots!** posted on website. Also use Piazza forum for help!

Exams:

- Midterm: *Wed, 25 Oct at 7:30pm-9:30pm*
- Final: scheduled by Registrar (posted in 3rd week). **Alert – might be as late as Dec 22!**

# 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** (move one assignment's deadline forward by one 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

# Expected prerequisite background

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

**Programming (e.g. as in 6.1010[6.009] or 6.1210[6.006])**

- 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 Next Week

- Attend your **assigned section** only starting Monday Sept. 11
- 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
- Form a group with those around you to complete the lab
- Ask questions by putting yourself in the help queue
- No worries if no introml.mit.edu access yet; great chance to know your neighbor (ask them to put you in the queue)

# What we're teaching: Machine Learning!

## Given:

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

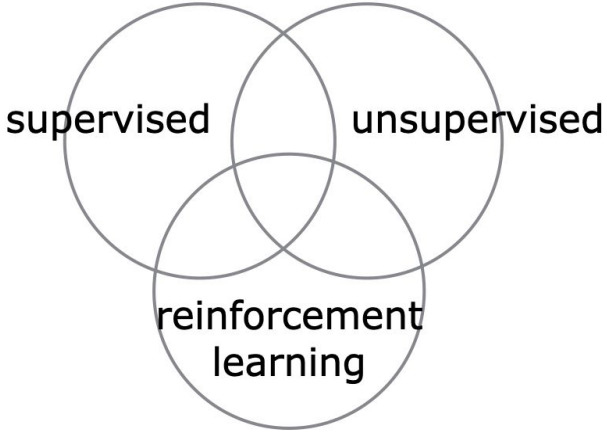
## Derive:

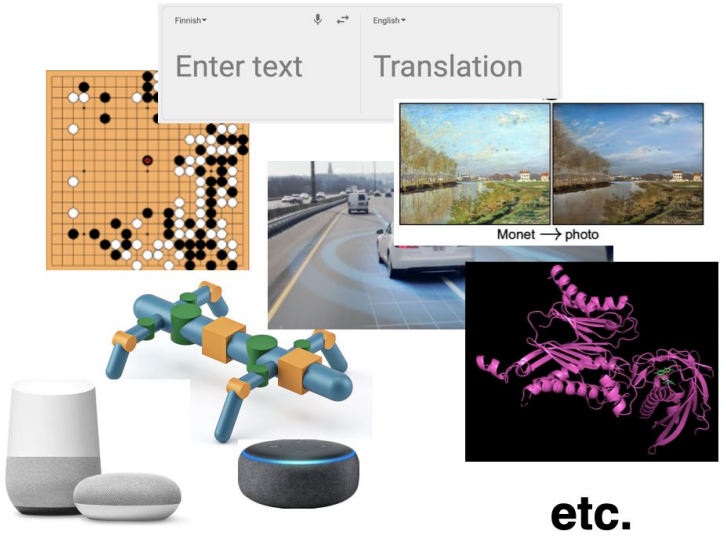
- a **computational model** (called a hypothesis) that describes relationships within and among the examples that is expected to characterize well new examples from that same population, to make good predictions or decisions

## 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



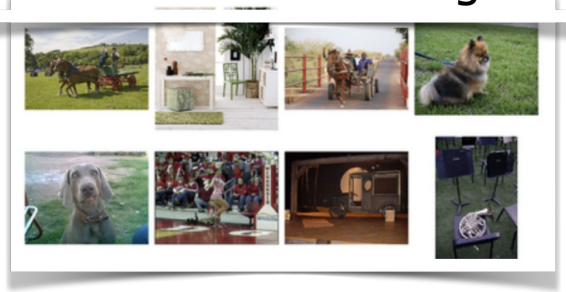




(the categorization can be refined, e.g. there are active learning, semi-supervised, selective, contrastive, few-shot, inverse reinforcement learning… )

[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

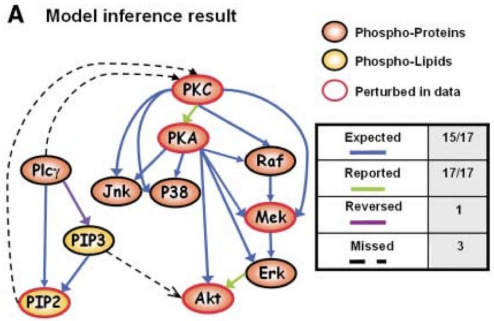
| **Spanish (input)** | **English (output)** |
|---|---|
| Aquí tienes un bolígrafo | Here's a pen |
| Las conferencias de ML son divertidas | ML conferences are fun |
| Todo el mundo debería estudiar AI | Everyone should study AI |
| ... | ... |

[Slides adapted from 6.790]

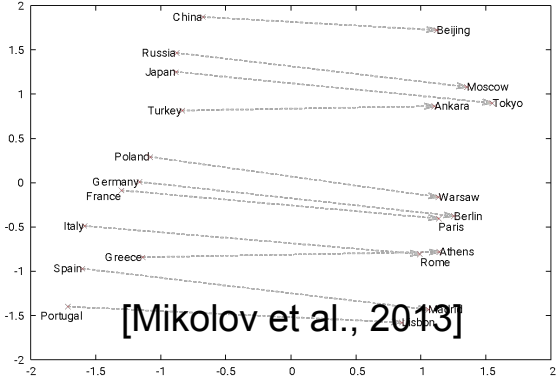
# Unsupervised learning

dimensionality reduction, embedding

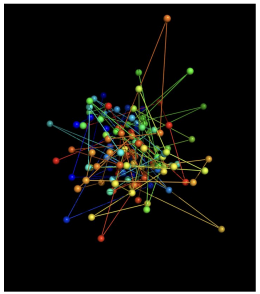
dependency /causal structure

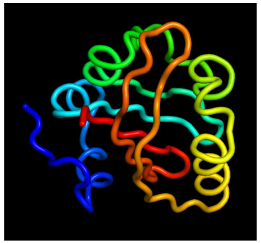



[Sachs et al 05]




[Mikolov et al., 2013]

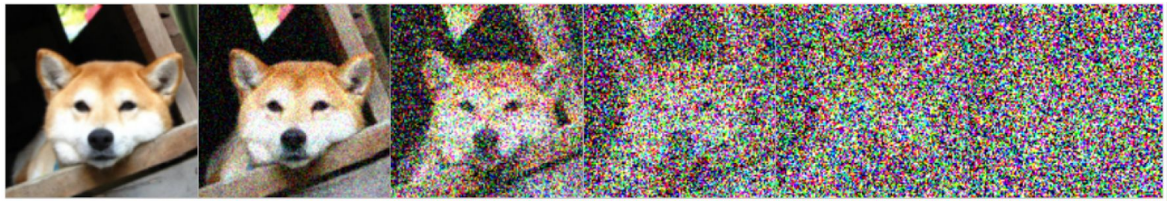




[courtesy of Jason Yim]

Over 3D protein structures, etc.

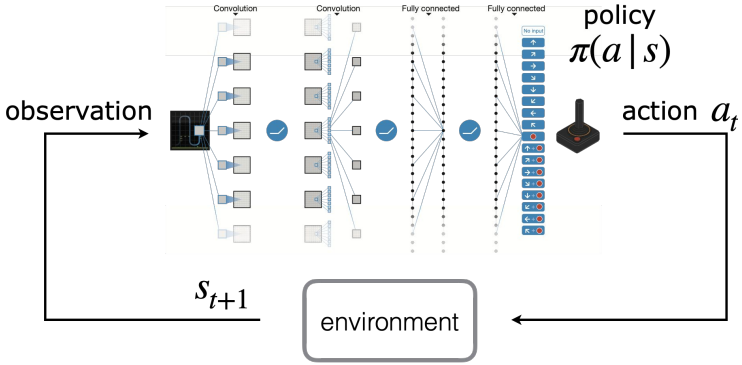
de-noising diffusion models over images

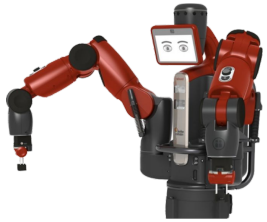


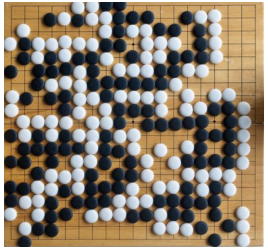
[image from Rissanen et al 2022]

[Slides adapted from 6.790]

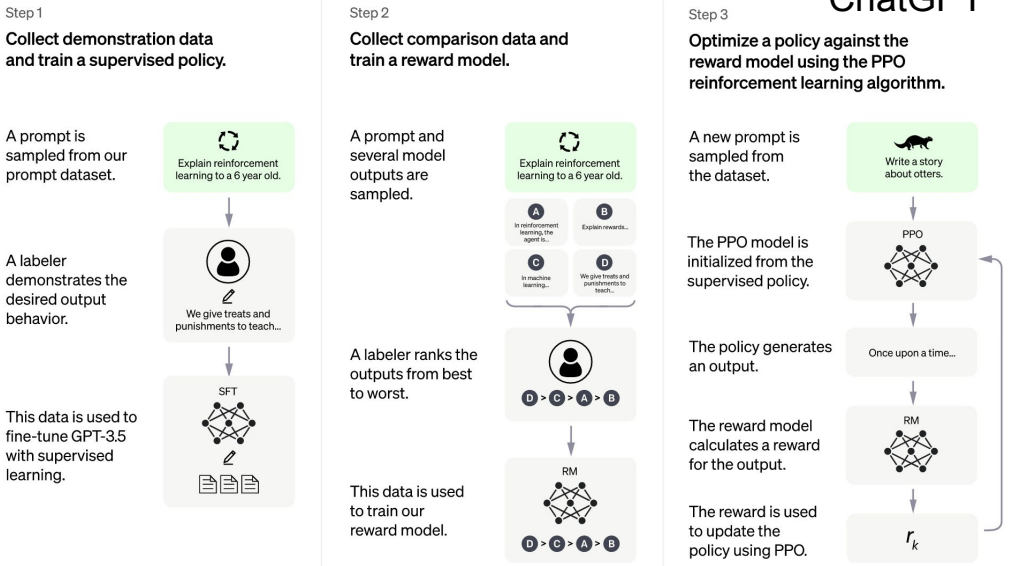
# Reinforcement learning

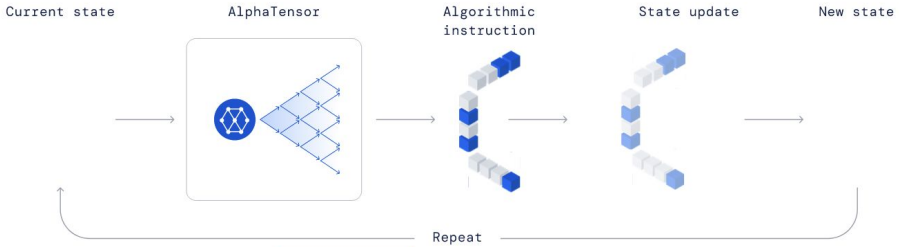




ChatGPT




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!
- **Notes:** ML is a tool with pros & cons

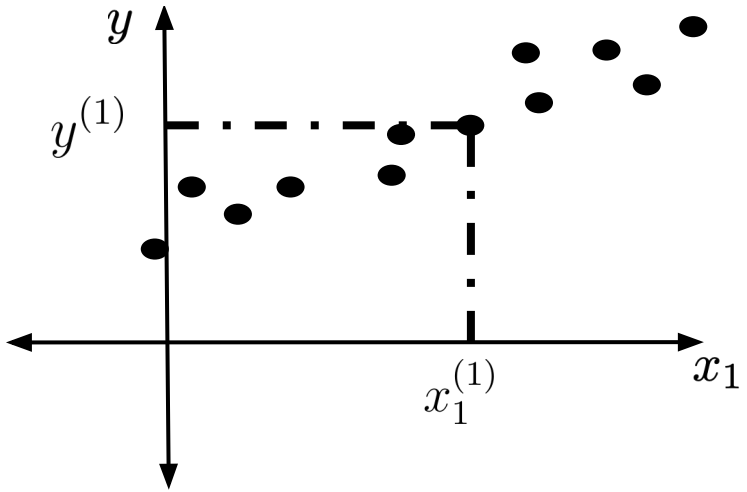
- **What do we have?** Data! And Computation!
- **What do we want?** To make predictions on new data!
- **How do we learn to make those decisions?**
  - The topic of this course!

# 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**

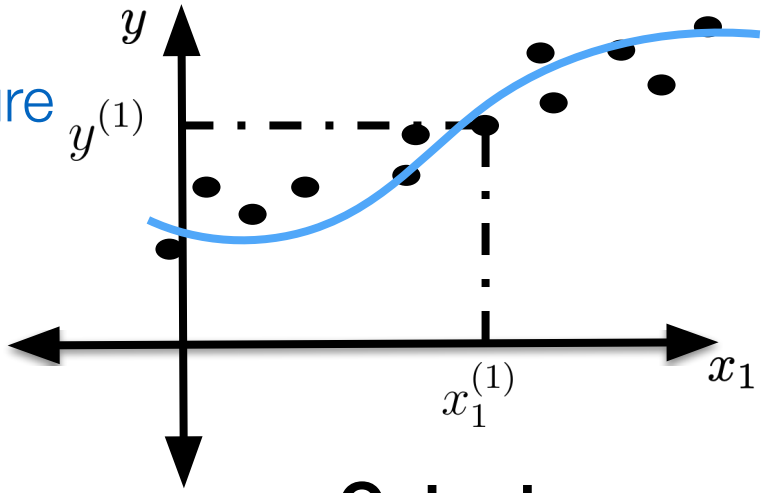
- *n* training data points
- For data point $i \in \{1, \dots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \dots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$



# 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




**Input**:
Feature vector


**Output**:
Label

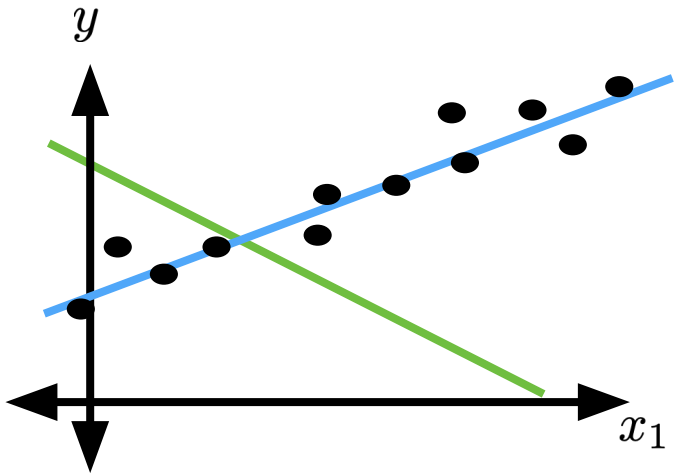
how well our hypothesis labels new feature vectors depends largely on how expressive the hypothesis class is

# What do we want?

We may consider the class of linear regressors:

- Hypotheses take the form:

$$h(x; \underbrace{\theta, \theta_0}_{\text{parameters to learn } \Theta}) = \theta^\top x + \theta_0$$

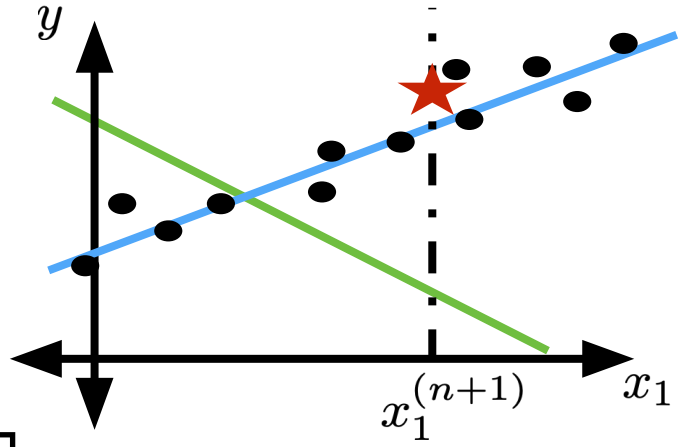



- What we really want is to generalize to **future data**!
- What we don't want:
  - Model does not capture the input-output relationship (e.g., not enough data) —> Underfitting
  - Model too specific to training data —> Overfitting

# 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, $\mathcal{L}(g, a)$
  - Common choice: squared loss:




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

g: guess,
a: actual

- Training error:

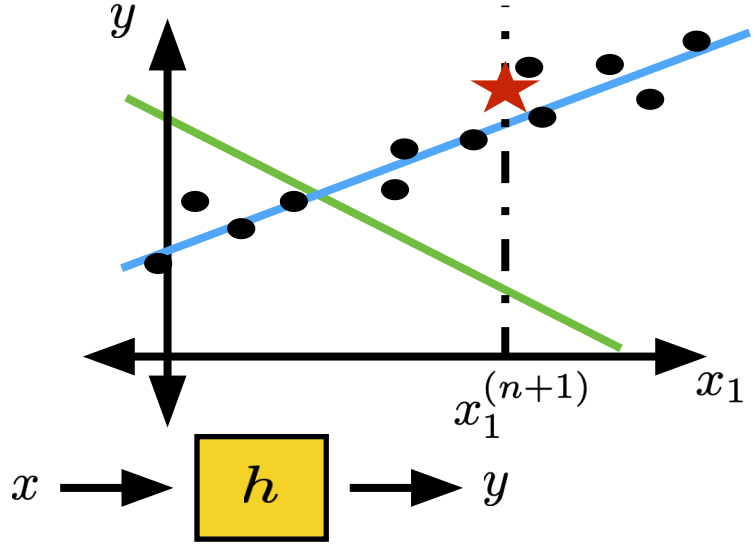
$$\mathcal{E}_n(h; \Theta) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(h(x^{(i)}; \Theta), y^{(i)}\right)$$

- 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:

How to get it:
(Next week!)