Practical Application in Machine Learning

Rina BUOY, PhD

ChatGPT 4.0

Disclaimer

Adopted from

6.390

Introduction to Machine Learning
(Fall 2024)

https://introml.mit.edu/fall24

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

What we're teaching: Machine Learning!

Given:

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

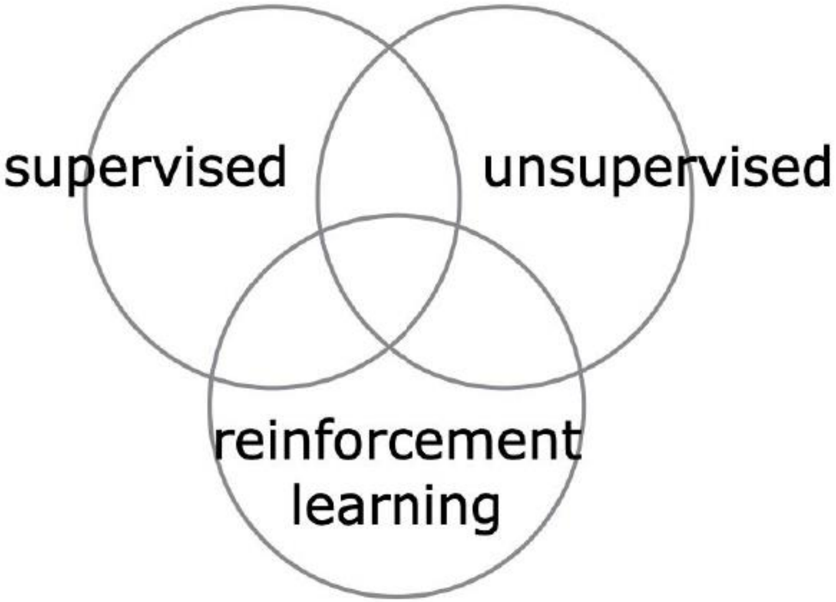
Derive:

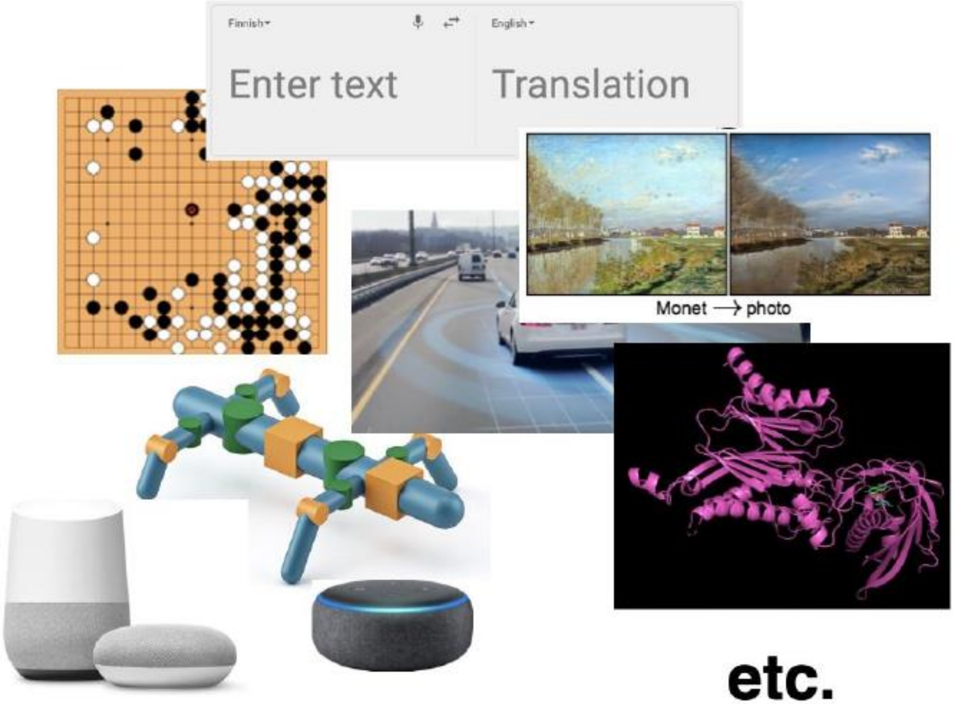
- a **computational model** (called a hypothesis) that describes relationships within and among the examples that is expected to characterize well new examples from that same population, to make good predictions or decisions

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]

Adopted from : https://introml.mit.edu

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)

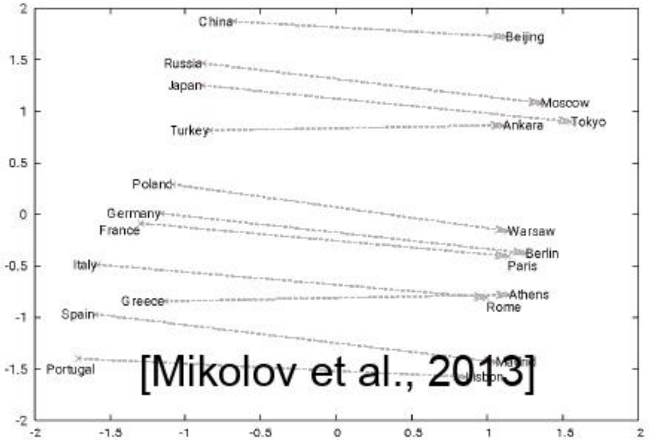
- Learning a machine translation system from pairs of sentences

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]

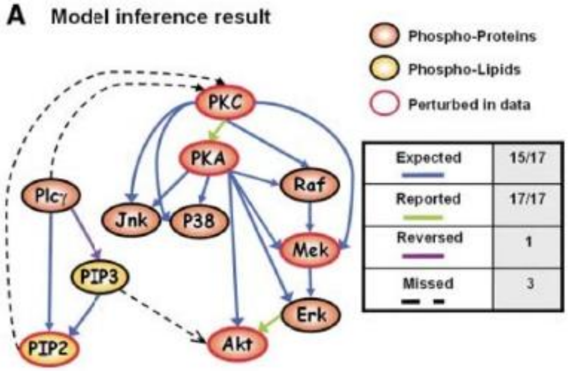
Unsupervised learning

dimensionality reduction, embedding

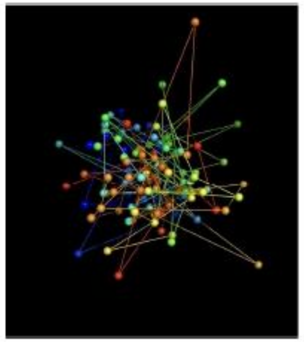


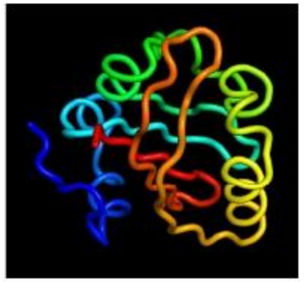
[Mikolov et al., 2013]

dependency /causal structure



[Sachs et al 05]



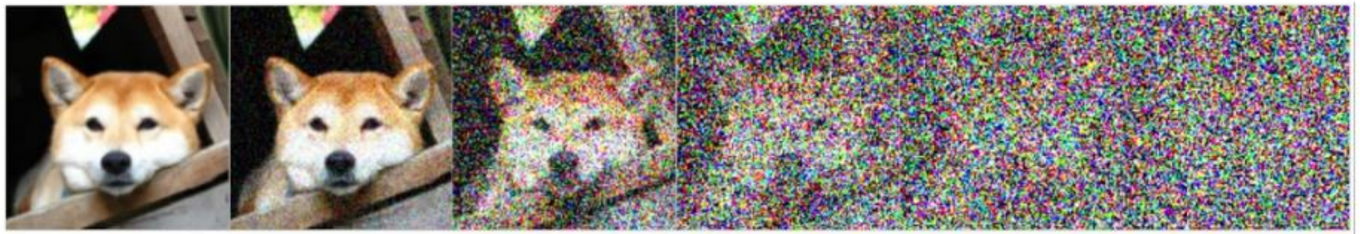


[courtesy of Jason Yim]

Over 3D protein structures, etc.

+Self-Supervised paradigm

de-noising diffusion models over images

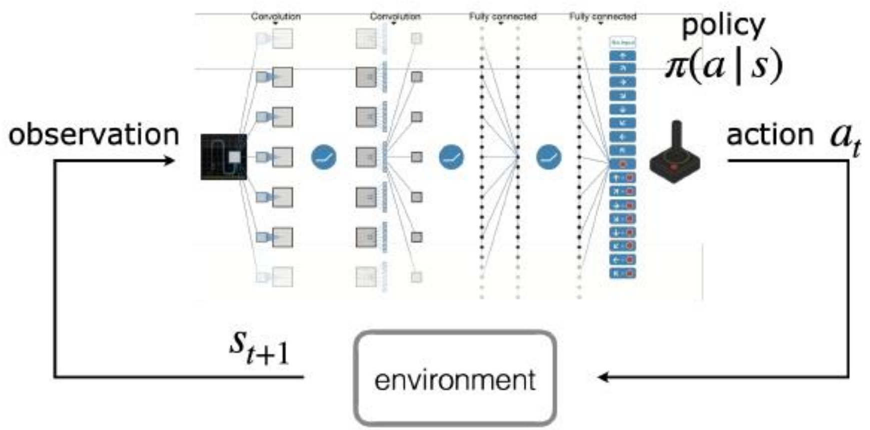


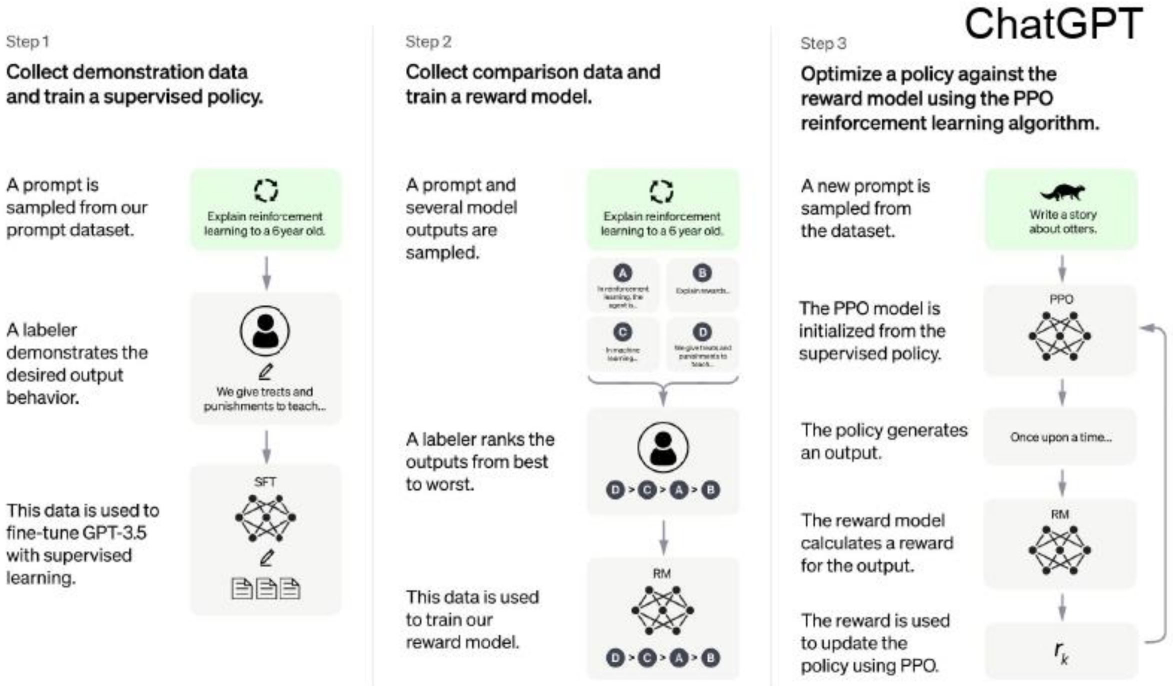
[image from Rissanen et al 2022]

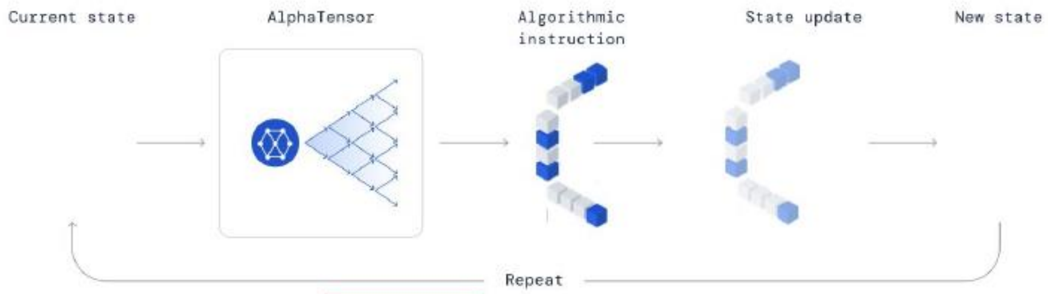
[Slides adapted from 6.790]

Adopted from : https://introml.mit.edu

Reinforcement learning







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]

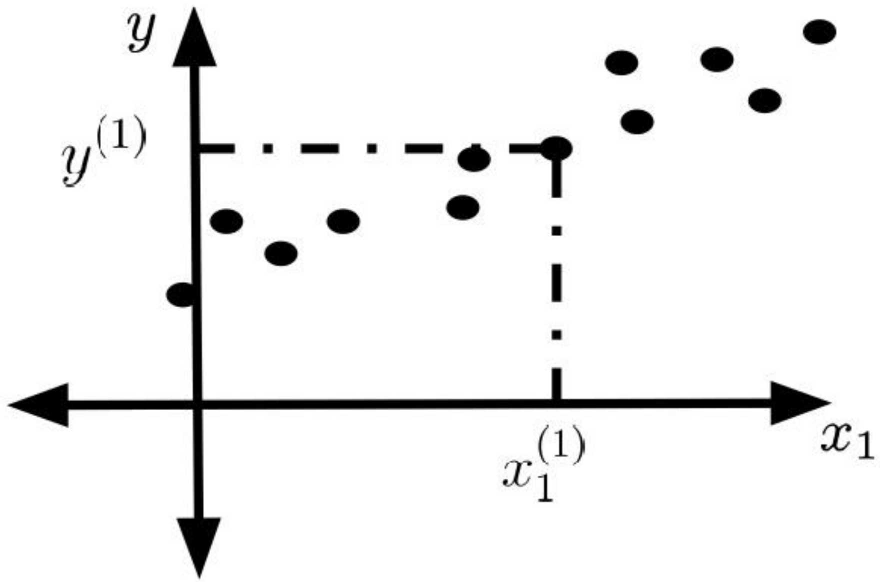
Machine learning (ML): why & what

- **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!

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

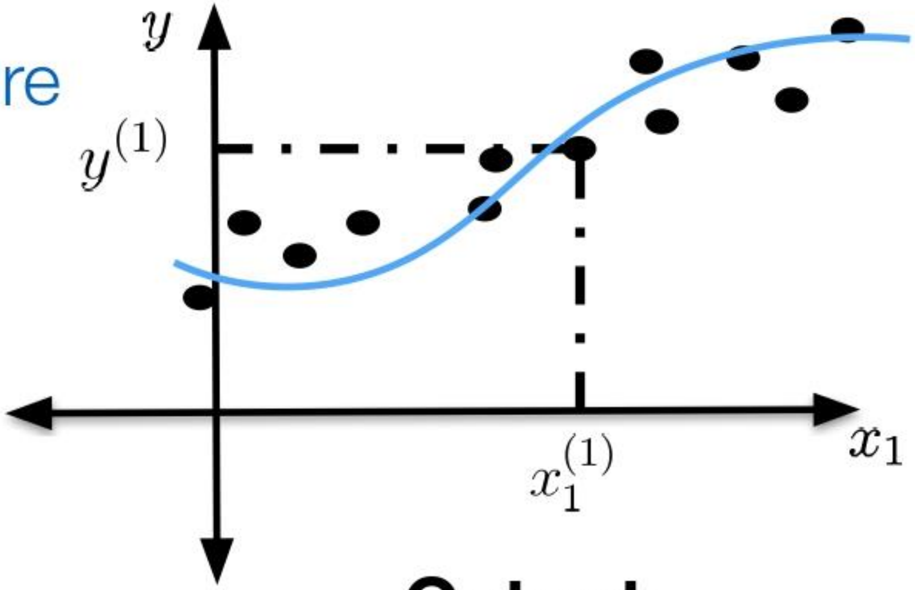


- *n* training data points
- For data point $i \in \{1, \ldots, n\}$
  - Feature vector $x^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})^\top \in \mathbb{R}^d$
  - Label $y^{(i)} \in \mathbb{R}$
- Training data $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

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



Input:
Feature vector

$$x \longrightarrow \boxed{h} \longrightarrow y$$

Output:
Label

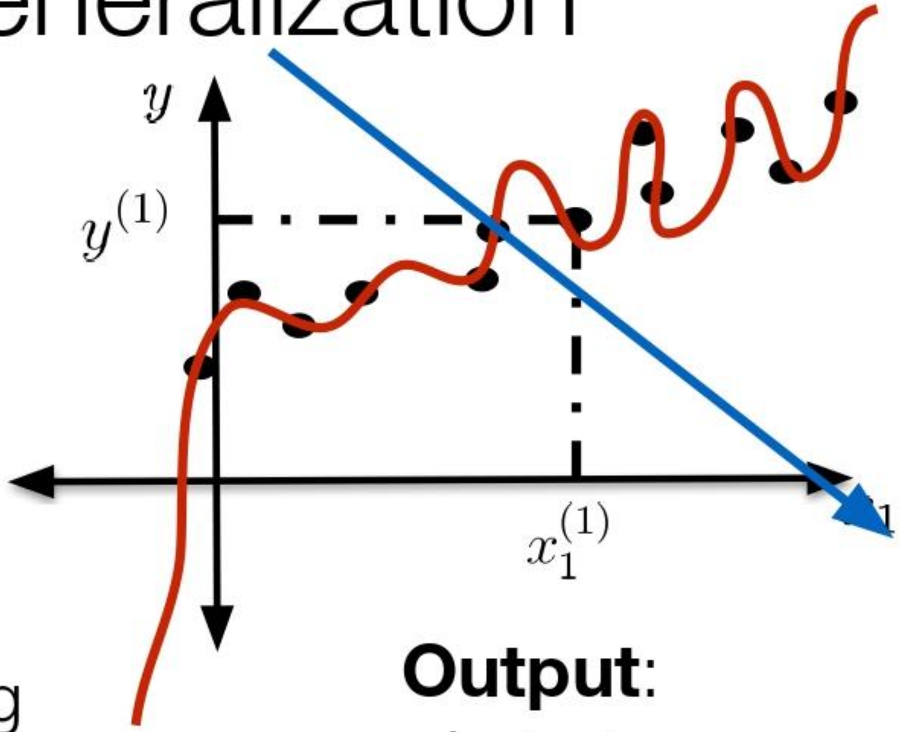
how well our hypothesis labels new feature vectors depends largely on how expressive the hypothesis class is

Adopted from : https://introml.mit.edu

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 → **Underfitting**
  - Model too specialized to training data → **Overfitting**



Output: Label

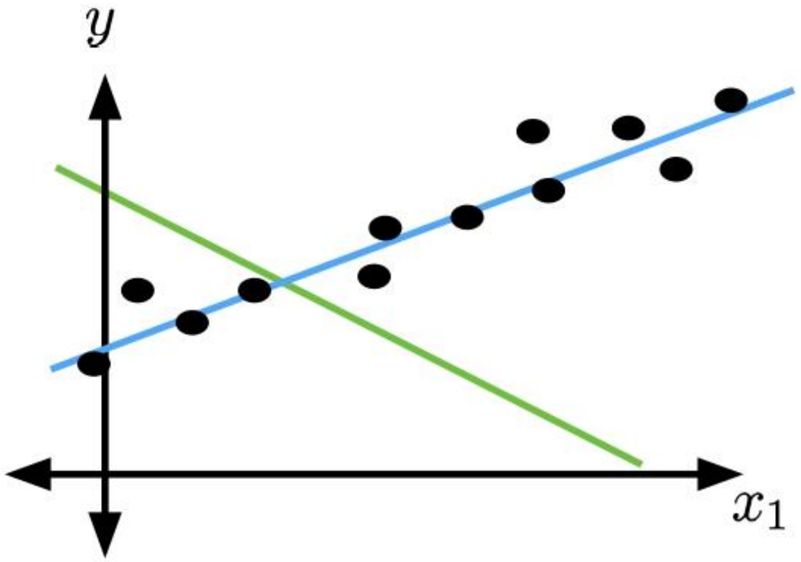
What do we want?

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)



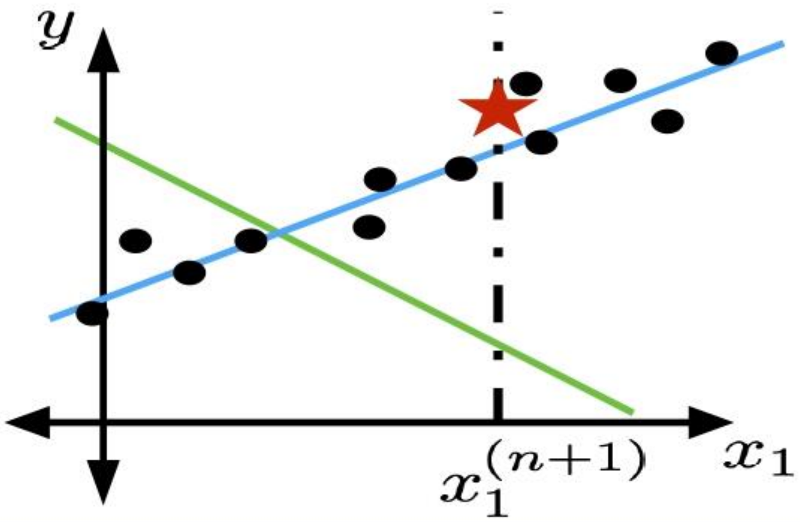
How **good** is a hypothesis?

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:

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

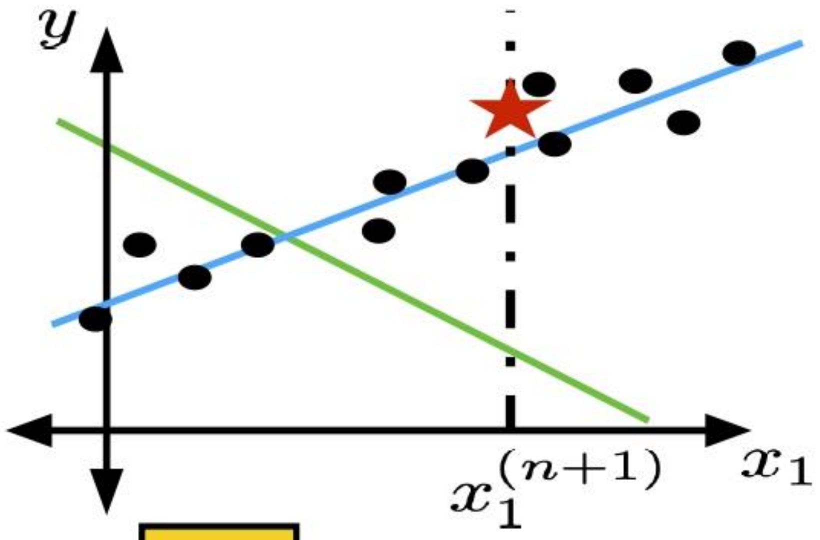
h: hypothesis function (outputs g)
x: input, θ: parameters, y: actual

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

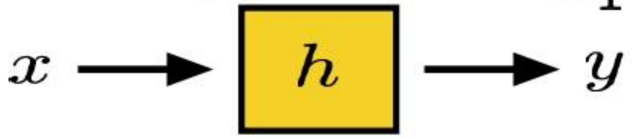
Adopted from : https://introml.mit.edu

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 time!)