https://introml.mit.edu/

6.390 Intro to Machine Learning

Lecture 1: Intro and Linear Regression

Shen Shen

Feb 2, 2026

3pm, Room 10-250

Slides and Lecture Recording

https://www.youtube.com/embed/j9688VaVKeo?enablejsapi=1

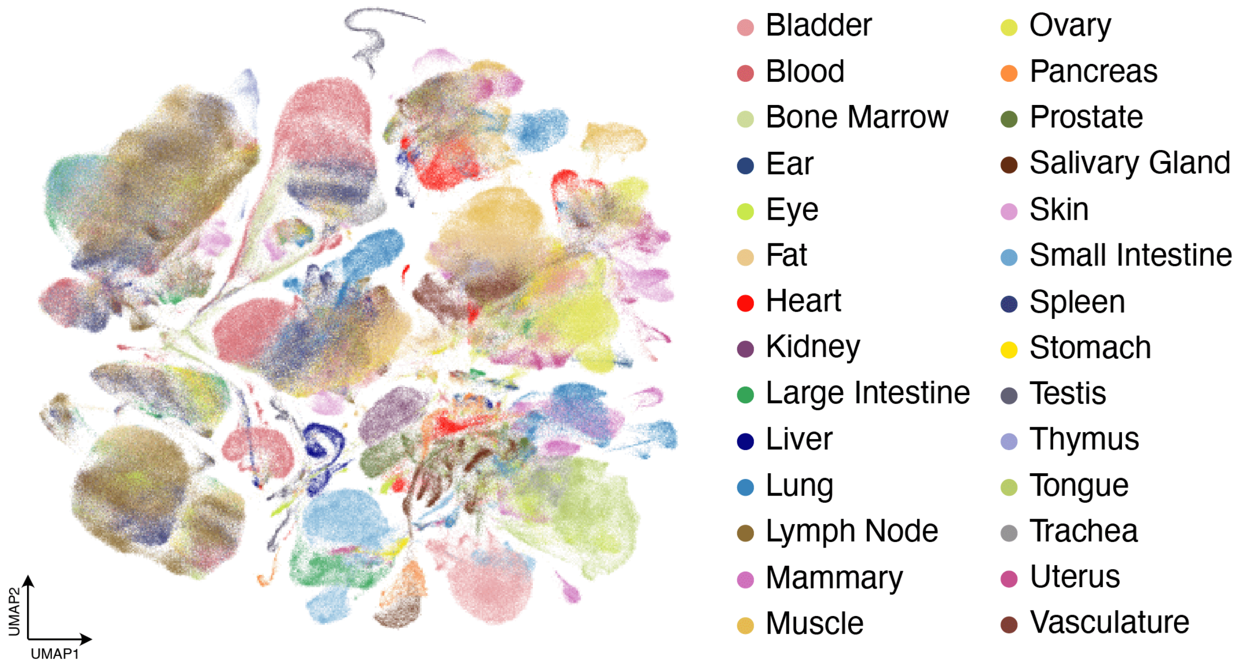
Medicine

CheXNet (Stanford, 2017)

- An ML system trained on 112K *labeled* chest X-rays
- Detects pneumonia at radiologist-level accuracy
- Heatmap shows where the model "looks"

Input: X-ray image & label (disease / no disease) →
Output: diagnosis + localization

Science



Tabula Sapiens (CZI, 2022)

- 500K+ human cells profiled by gene expression
- Clustering reveals cell types across 24 organs — *no predefined labels*
- Each dot = one cell; color = organ of origin

Input: gene expression vectors
→ Output: discovered groupings

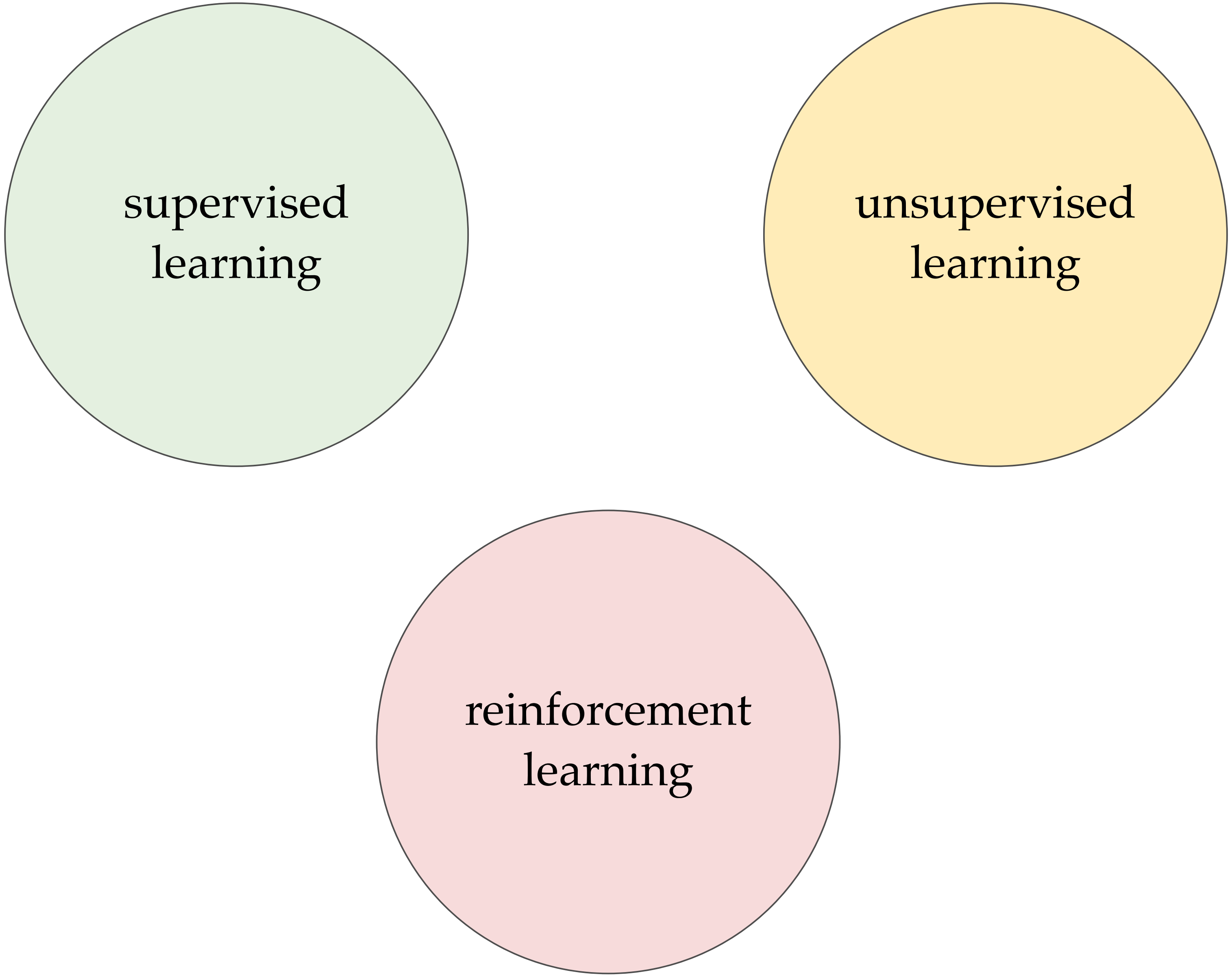
Robotics

https://www.pi.website/blog/pistar06

π^*0.6 (Physical Intelligence, 2025)

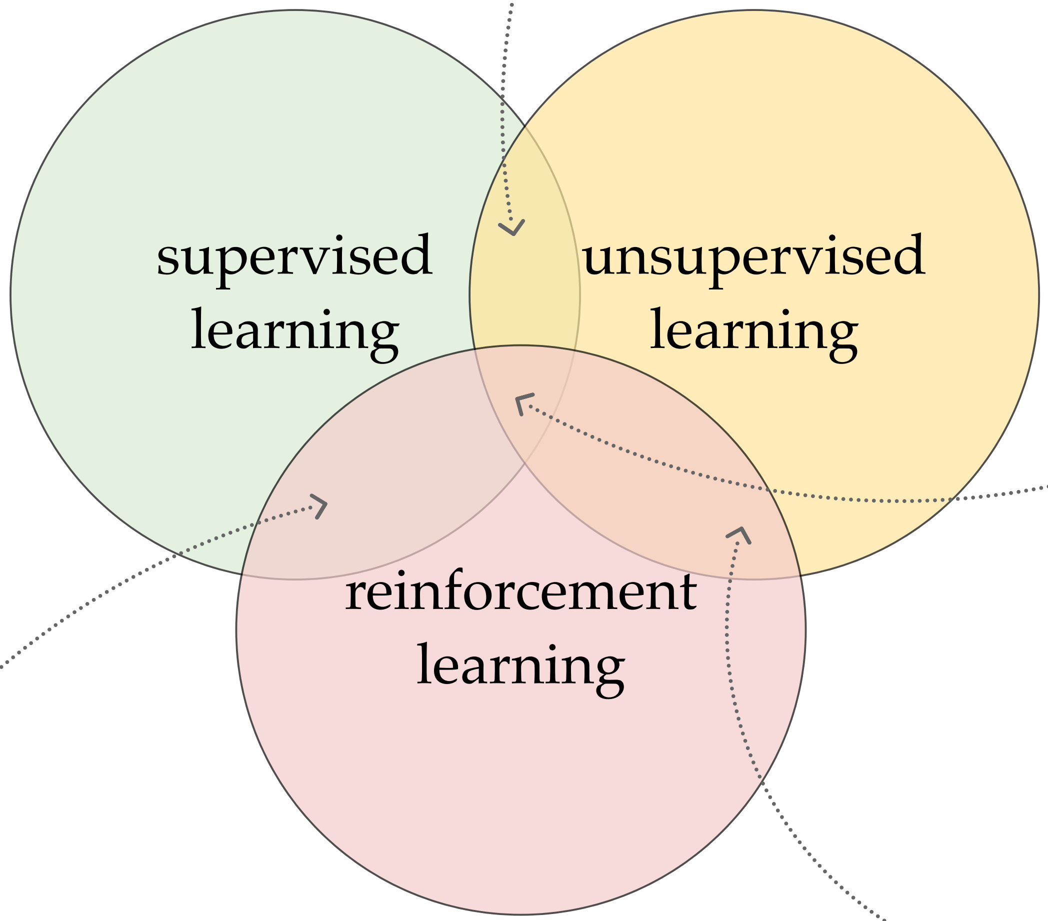
- Vision-language-action model learns from demos + practice
- Makes espresso, folds laundry, assembles boxes
- No task-specific programming — learns by trial and reward

State: camera image → Action: motor commands → Reward: task completion traditionally



nowadays

- self-supervised
- contrastive learning (DALL-E)

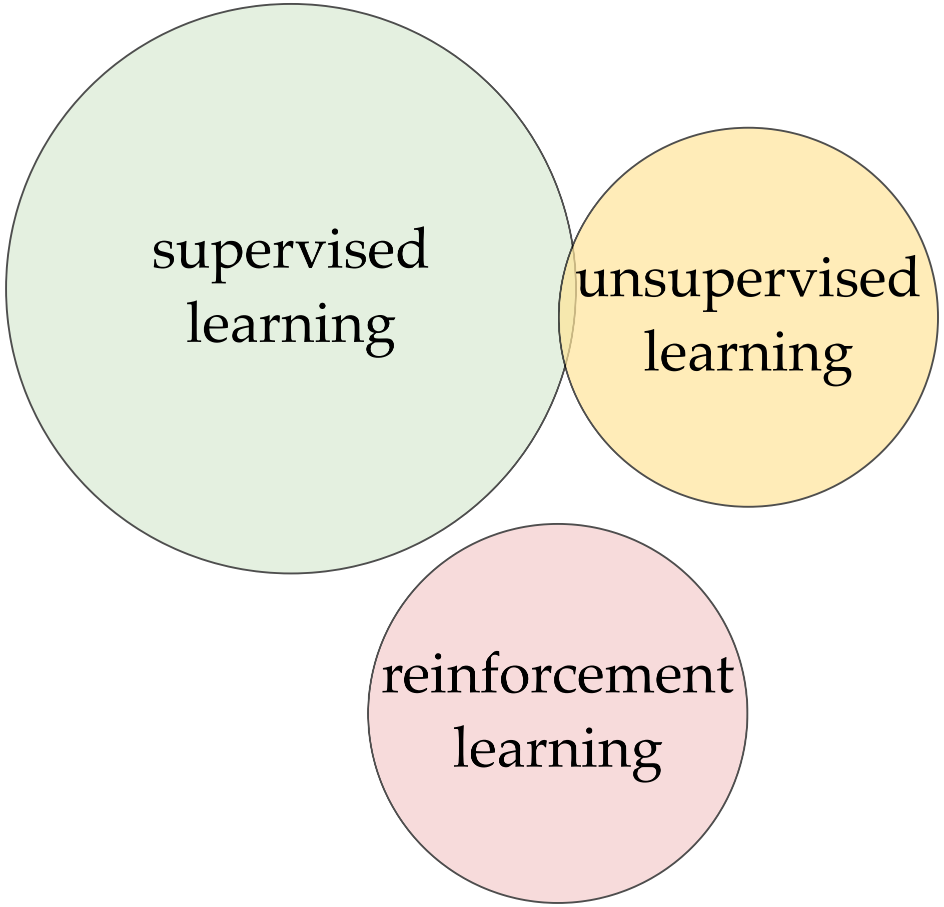


Reinforcement Learning with Human Feedback (ChatGPT etc.)

behavior cloning

Unsupervised skill discovery

In 6.390:


supervised
learning
unsupervised
learning
reinforcement
learning

Outline

- Course overview
- Supervised learning terminologies
- Ordinary least squares regression

Topics in order:

- Intro to ML
- Regression and Regularization
- Gradient Descent
- Linear Classification
- Features, Neural Networks I
- Neural Networks II (Backprop)
- Convolutional Neural Networks
- Representation Learning
- Transformers
- Markov Decision Processes
- Reinforcement Learning
- Non-parametric Models

Many other ways to dissect

Model class:

- linear models
- linear model on non-linear features
- fully connected feed-forward nets
- convolutional nets
- transformers
- Q-table
- tree, k-nearest neighbor, k-means

Modeling choices:

- Supervised:
  - regression
  - classification
- Unsupervised / self-supervised
- Reinforcement / sequential

Learning process:

- training / validation / testing
- overfitting / underfitting
- regularization
- hyper parameters

Optimization:

- analytical solutions
- gradient descent
- back propagation
- value iteration, Q-learning
- non-parametric methods

[These lists are neither exhaustive nor exclusive.]

After 6.390, you can…

- **Frame** ML problems: problem class, assumptions, evaluation.
- **Build** baselines and measure generalization (train vs. test).
- **Implement** and reason about regression and classification.
- **Optimize** models with gradients (SGD) and regularization.
- **Work** with representations and neural networks.
- **Understand** modern LLM mechanisms (transformers) and MDP/RL basics.

generalization · loss · train/test · regularization · SGD · backprop · representations · CNNs · attention · transformers · MDP · reinforcement learning

A typical content week in 6.390:

Week #	Monday	Tuesday	Wednesday	Thursday	Friday
N	Exercise *N*	Homework *N*	Exercise *N*	Recitation *N*	
	Lecture *N*		Lab *N*		
N+1				Homework *N*	

Class meetings

released

due

Hours:

Lec: 1.5 hr

Rec + Lab: 3 hr

Notes + exercise: 2 hr

Homework: 6-7 hr

Grading

- Our objective (and we hope yours) is for you to learn about machine learning — take responsibility for your understanding; we are here to help!
- Grades formula: exercises 5% + homework 20% + labs 15% + midterms 30% + final 30%
- Lateness: 20% penalty per day, applied linearly (so 1 hour late is -0.83%)
- 20 one-day extensions, applied *automatically on May 13* to maximize your benefit

- Midterm 1: Thursday, March 12, 7:30–9pm
- Midterm 2: Wednesday, April 15, 7:30–9pm
- Final: scheduled by Registrar (posted in 3rd week). ⚠️ – might be as late as May 21!

Collaboration and How to Get Help

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

- Office hours: lots! (Starting this Thursday)
- See Google Calendar for holiday/schedule shift
- Make use of Piazza and Pset-partners
- Logistic, personal issues, reach out to

6.390-personal@mit.edu **(**looping in ***S^3 and/or DAS)***

Outline

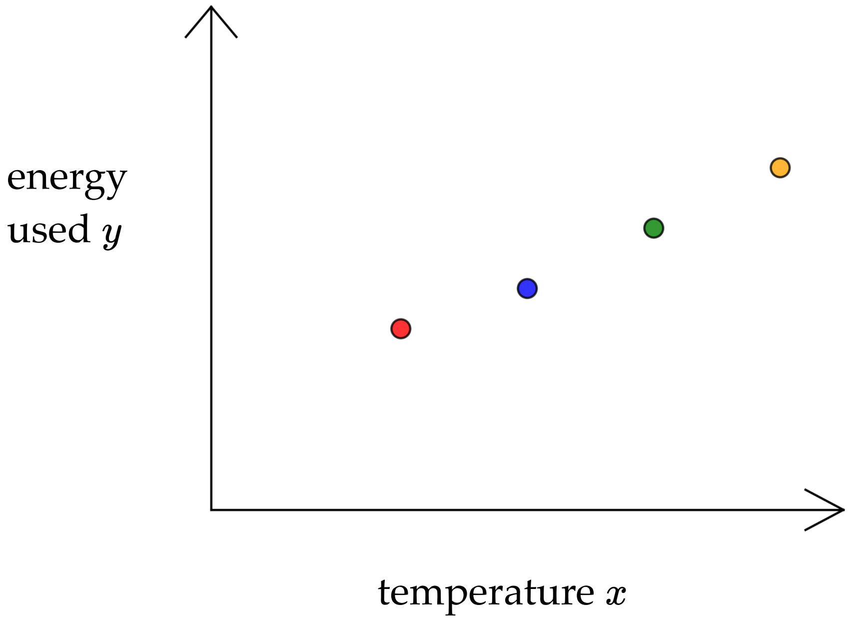
- Course overview
- Supervised learning terminologies
- Ordinary least squares regression

an instance *supervised learning* known as *regression:* predicting a continuous number

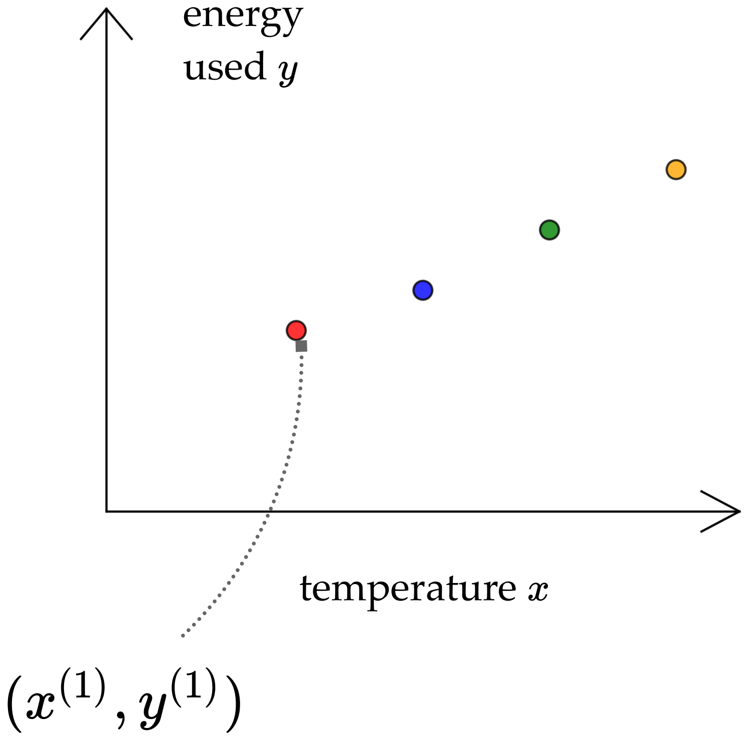
e.g., want to predict a city's energy usage

go collect some data in various cities

City	Feature	Label
	Temperature	**Energy Used**
Chicago	90	45
New York	20	32
Boston	35	99
San Diego	18	39



toy data, for illustration only



Training data:

$$\mathcal{D}_{\text{train}} := \left\{ \left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(4)}, y^{(4)}\right) \right\}$$

feature

label

$x^{(1)} \in \mathbb{R}$

$y^{(1)} \in \mathbb{R}$

Training data:

$$\mathcal{D}_{\text{train}} := \left\{ \left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(4)}, y^{(4)}\right) \right\}$$

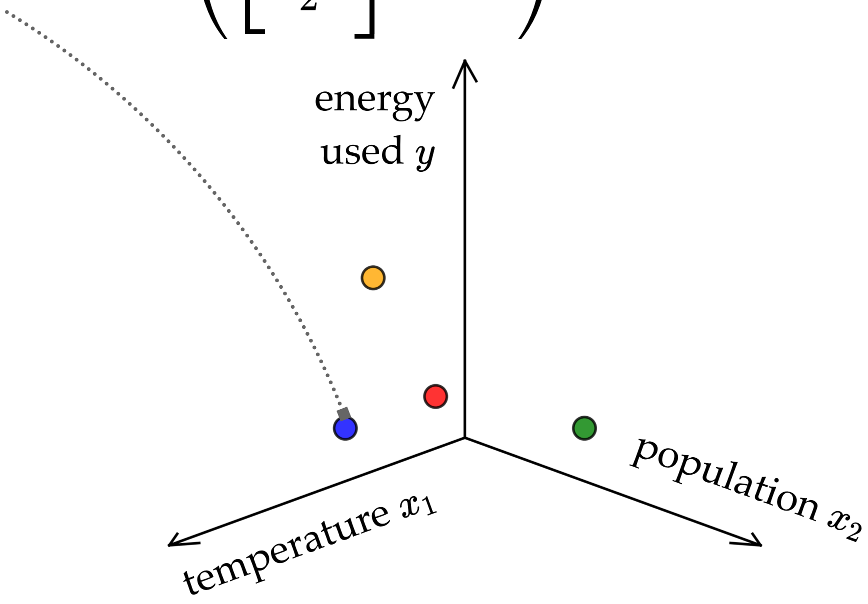
feature vector

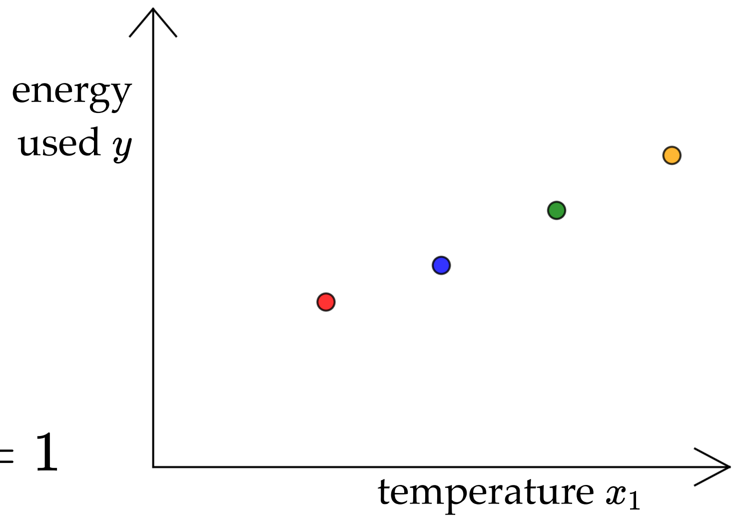
$$x^{(1)} = \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} \in \mathbb{R}^2$$

label

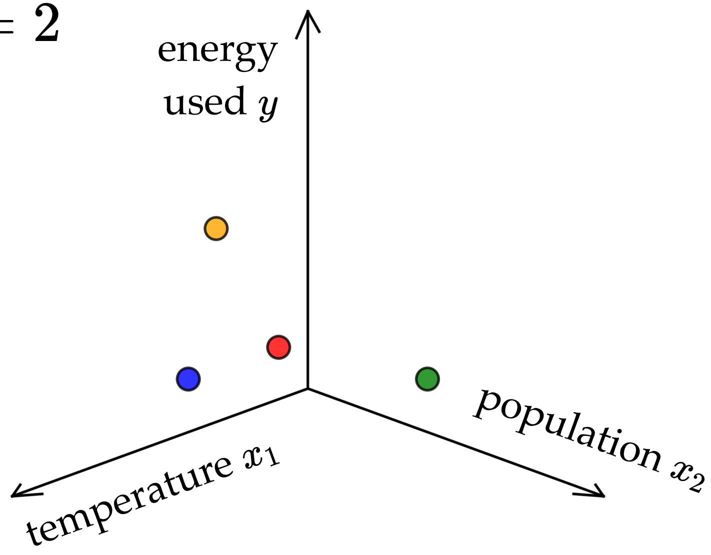
$$y^{(1)} \in \mathbb{R}$$

$$\left(x^{(1)}, y^{(1)}\right) = \left(\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix}, y^{(1)} \right)$$





$n = 4, d = 1$



$n = 4, d = 2$

Training data:

$$\mathcal{D}_{\text{train}} := \left\{ \left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(n)}, y^{(n)}\right) \right\}$$

feature vector

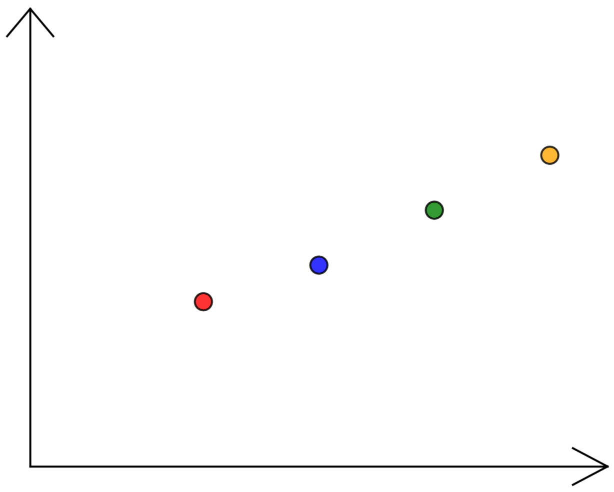
label

$$x^{(1)} = \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \vdots \\ x_d^{(1)} \end{bmatrix} \in \mathbb{R}^d$$

$$y^{(1)} \in \mathbb{R}$$

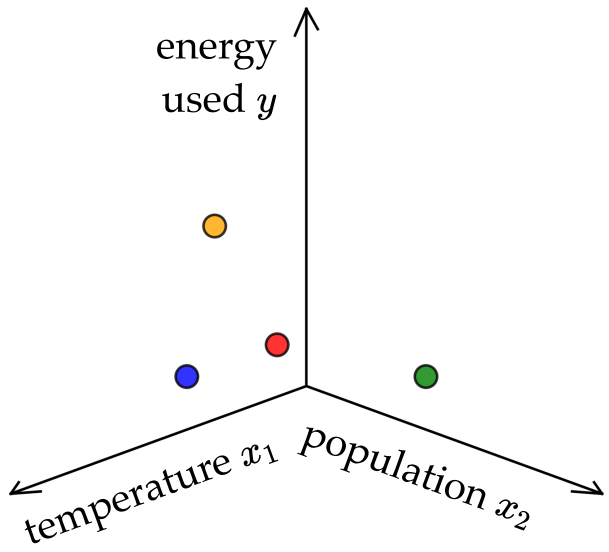
n data points, each with d-dimensional features and scalar label

Training data in matrix-vector form:



City	Feature	Label
	Temperature	Energy Used
Chicago	90	45
New York	20	32
Boston	35	99
San Diego	18	39

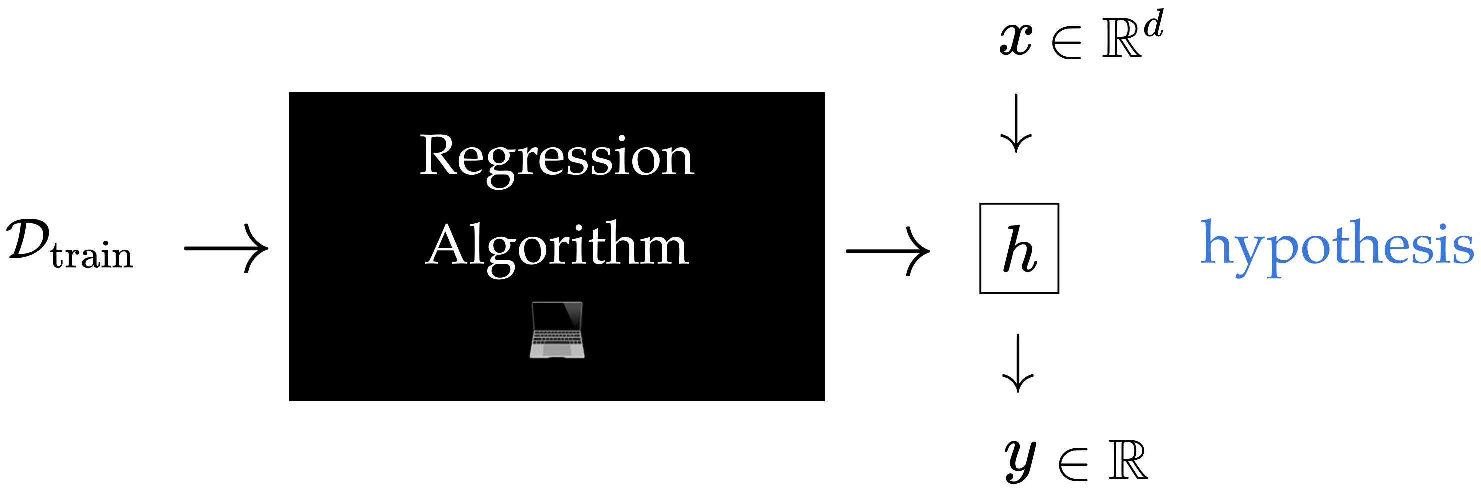
$$X = \begin{bmatrix} 90 \\ 20 \\ 35 \\ 18 \end{bmatrix} \in \mathbb{R}^{4\times 1} \qquad Y = \begin{bmatrix} 45 \\ 32 \\ 99 \\ 39 \end{bmatrix} \in \mathbb{R}^{4\times 1}$$



City	Features		Label
	Temperature	Population	Energy Used
Chicago	90	7.2	45
New York	20	9.5	32
Boston	35	8.4	99
San Diego	18	4.3	39

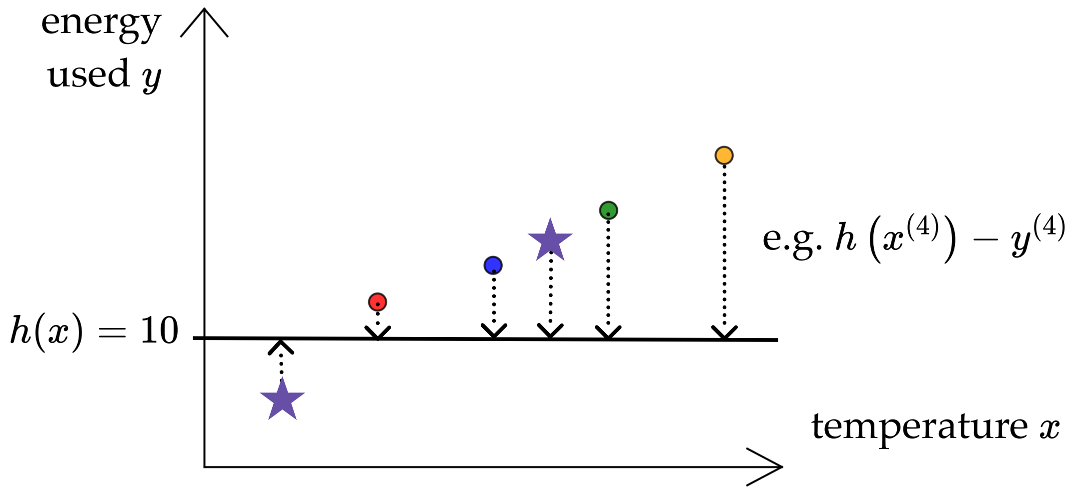
$$X = \begin{bmatrix} 90 & 7.2 \\ 20 & 9.5 \\ 35 & 8.4 \\ 18 & 4.3 \end{bmatrix} \in \mathbb{R}^{4\times 2} \qquad Y = \begin{bmatrix} 45 \\ 32 \\ 99 \\ 39 \end{bmatrix} \in \mathbb{R}^{4\times 1}$$

Learning algorithm spits out a hypothesis



What do we want from the regression algortim?

A good way to label *new* features, i.e. a *good* hypothesis.



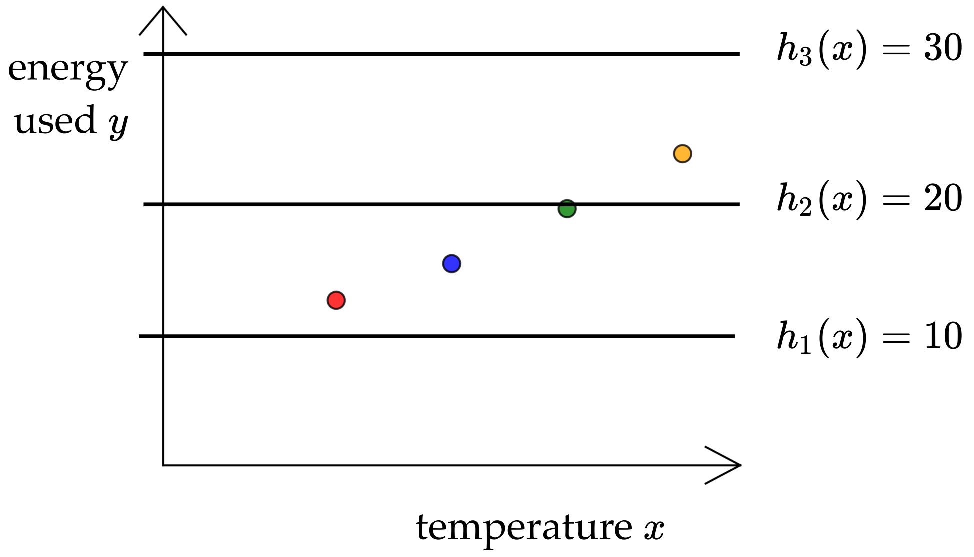
- Loss $\mathcal{L}\left(h\left(x^{(i)}\right), y^{(i)}\right)$

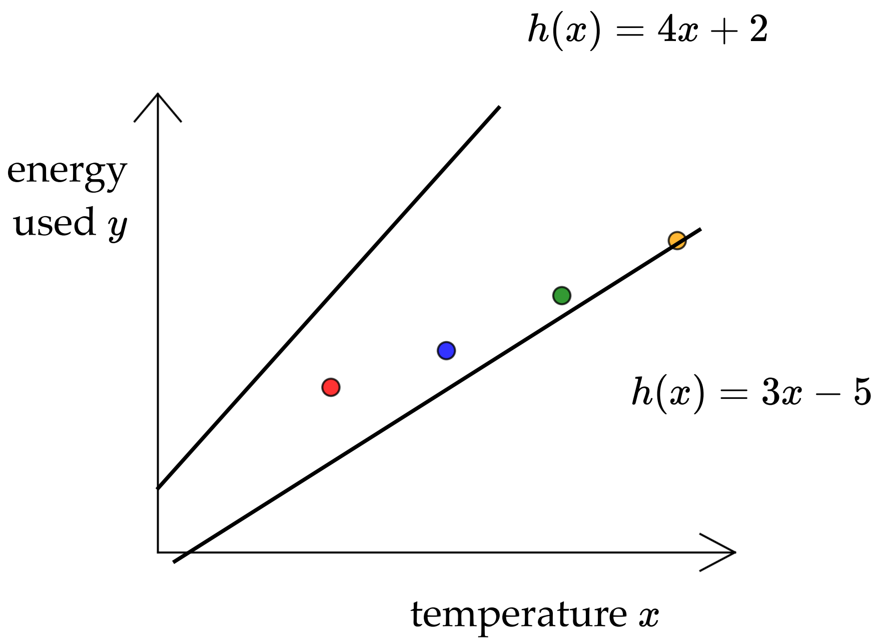
- Training error $\mathcal{E}_{\text{train}}(h) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\left(h\left(x^{(i)}\right), y^{(i)}\right)$

- Test error $\mathcal{E}_{\text{test}}(h) = \frac{1}{n'}\sum_{i=n+1}^{n+n'} \mathcal{L}\left(h\left(x^{(i)}\right), y^{(i)}\right)$

i.e. average loss on n' unseen test data points

Hypothesis class $\mathcal{H}$: set of h we ask the algorithm to search over



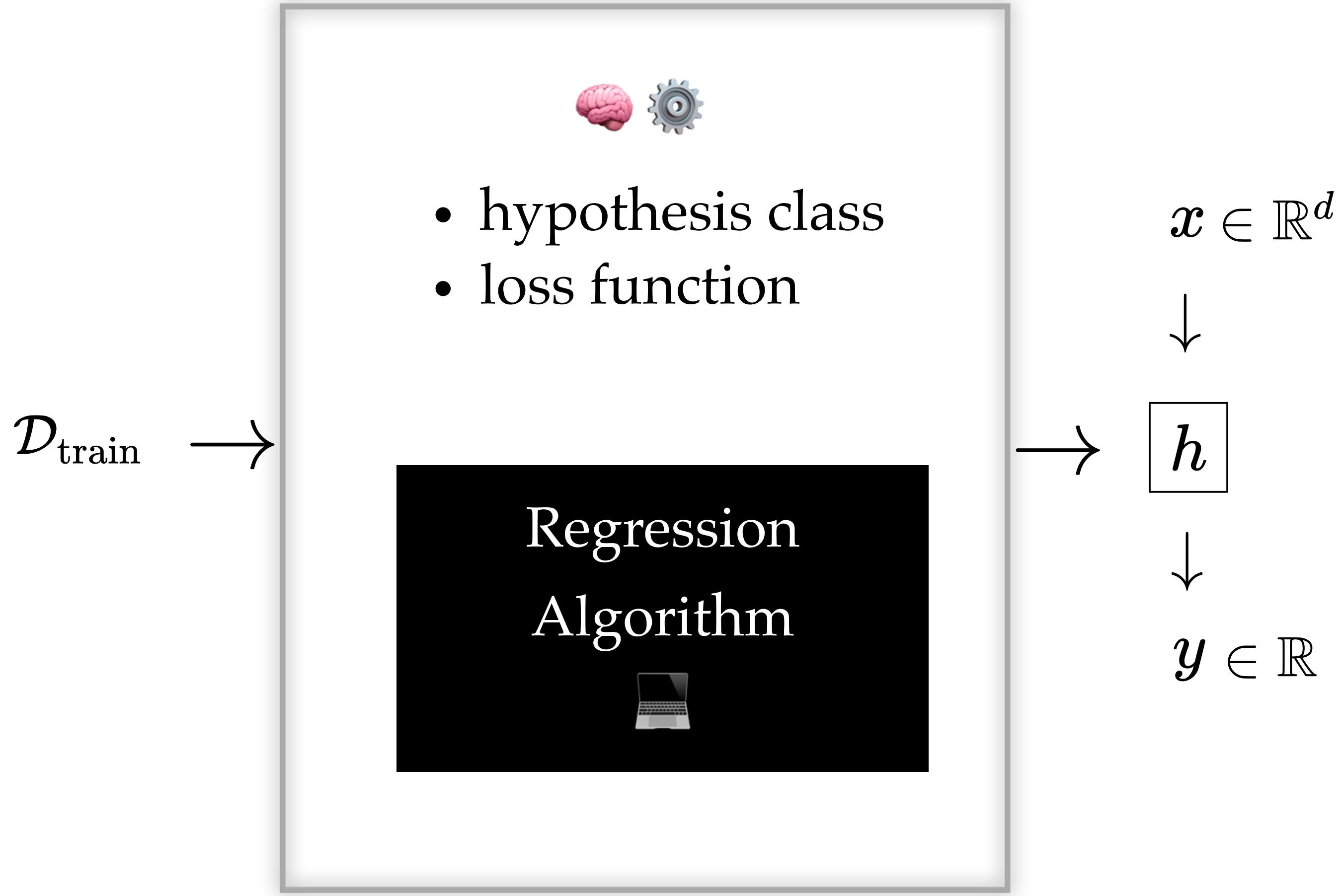


{constant functions} $\subset$ {linear functions}[1]

less expressive — more expressive

1. technically, affine functions. ML community tends to be flexible about this terminology.

Quick summary:



supervised learning | regression | training data | test data | features | label | loss function

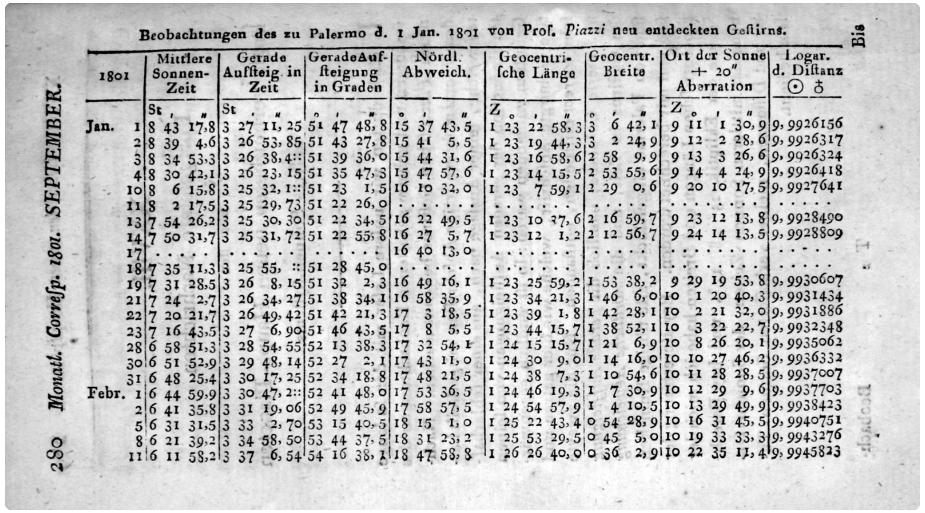
training error | test error | hypothesis | hypothesis class

Outline

- Course overview
- Supervised learning terminologies
- Ordinary least squares regression

Why least squares?

- Problem: infer orbit parameters from noisy, partial measurements.
- Idea: pick parameters that make predictions match observations as closely as possible, by minimize the sum of squared residuals.
- Today: still a fast, reliable, interpretable baseline.



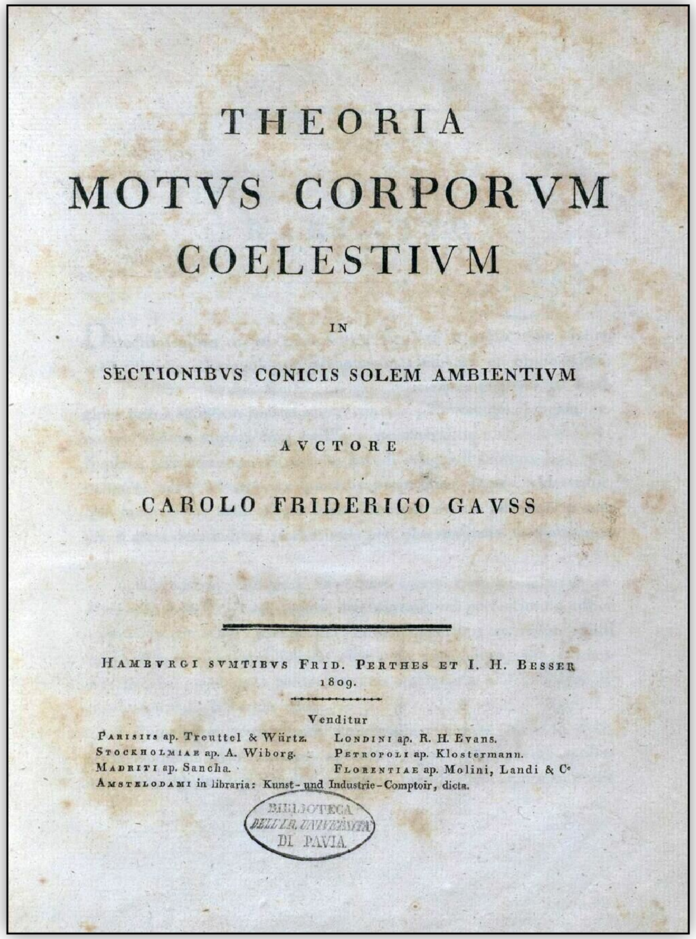
280 *Monatl. Corresp. 1801.* SEPTEMBER.

Beobachtungen des zu Palermo d. 1 Jan. 1801 von Prof. *Piazzi* neu entdeckten Gestirns.

1801	Mittlere Sonnen-Zeit	Gerade Aufsteig. in Zeit	Gerade Aufsteigung in Graden	Nördl. Abweich.	Geocentrische Länge	Geocentr. Breite	Ort der Sonne + 20″ Aberration	Logar. d. Distanz ⊙ ♁
	St ′ ″	St ′ ″	° ′ ″	° ′ ″	Z ° ′ ″	° ′ ″	Z ° ′ ″	
Jan. 1	8 43 17,8	3 27 11,25	51 47 48,8	15 37 43,5	1 23 22 58,3	3 6 42,1	9 11 1 30,9	9,9926156
2	8 39 4,6	3 26 53,85	51 43 27,8	15 41 5,5	1 23 19 44,3	3 2 24,9	9 12 2 28,6	9,9926317
3	8 34 53,3	3 26 38,4::	51 39 36,0	15 44 31,6	1 23 16 58,6	2 58 9,9	9 13 3 26,6	9,9926324
4	8 30 42,1	3 26 23,15	51 35 47,3	15 47 57,6	1 23 14 15,5	2 53 55,6	9 14 4 24,9	9,9926418
10	8 6 15,8	3 25 32,1::	51 23 1,5	16 10 32,0	1 23 7 59,1	2 29 0,6	9 20 10 17,5	9,9927641
11	8 2 17,5	3 25 29,73	51 22 26,0	......	......	......	......	......
13	7 54 26,2	3 25 30,30	51 22 34,5	16 22 49,5	1 23 10 37,6	2 16 59,7	9 23 12 13,8	9,9928490
14	7 50 31,7	3 25 31,72	51 22 55,8	16 27 5,7	1 23 12 1,2	2 12 56,7	9 24 14 13,5	9,9928809
17	......	......	......	16 40 13,0	......	......	......	......
18	7 35 11,3	3 25 55, ::	51 28 45,0	......	......	......	......	......
19	7 31 28,5	3 26 8,15	51 32 2,3	16 49 16,1	1 23 25 59,2	1 53 38,2	9 29 19 53,8	9,9930607
21	7 24 2,7	3 26 34,27	51 38 34,1	16 58 35,9	1 23 34 21,3	1 46 6,0	10 1 20 40,3	9,9931434
22	7 20 21,7	3 26 49,42	51 42 21,3	17 3 18,5	1 23 39 1,8	1 42 28,1	10 2 21 32,0	9,9931886
23	7 16 43,5	3 27 6,90	51 46 43,5	17 8 5,5	1 23 44 15,7	1 38 52,1	10 3 22 22,7	9,9932348
28	6 58 51,3	3 28 54,55	52 13 38,3	17 32 54,1	1 24 15 15,7	1 21 6,9	10 8 26 20,1	9,9935062
30	6 51 52,9	3 29 48,14	52 27 2,1	17 43 11,0	1 24 30 9,0	1 14 16,0	10 10 27 46,2	9,9936332
31	6 48 25,4	3 30 17,25	52 34 18,8	17 48 21,5	1 24 38 7,3	1 10 54,6	10 11 28 28,5	9,9937007
Febr. 1	6 44 59,9	3 30 47,2::	52 41 48,0	17 53 36,5	1 24 46 19,3	1 7 30,9	10 12 29 9,6	9,9937703
2	6 41 35,8	3 31 19,06	52 49 45,9	17 58 57,5	1 24 54 57,9	1 4 10,5	10 13 29 49,9	9,9938423
5	6 31 31,5	3 33 2,70	53 15 40,5	18 15 1,0	1 25 22 43,4	0 54 28,9	10 16 31 45,5	9,9940751
8	6 21 39,2	3 34 58,50	53 44 37,5	18 31 23,2	1 25 53 29,5	0 45 5,0	10 19 33 33,3	9,9943276
11	6 11 58,2	3 37 6,54	54 16 38,1	18 47 58,8	1 26 26 40,0	0 36 2,9	10 22 35 11,4	9,9945823

Bis

Observation table: Piazzi's measurements of Ceres (1801).



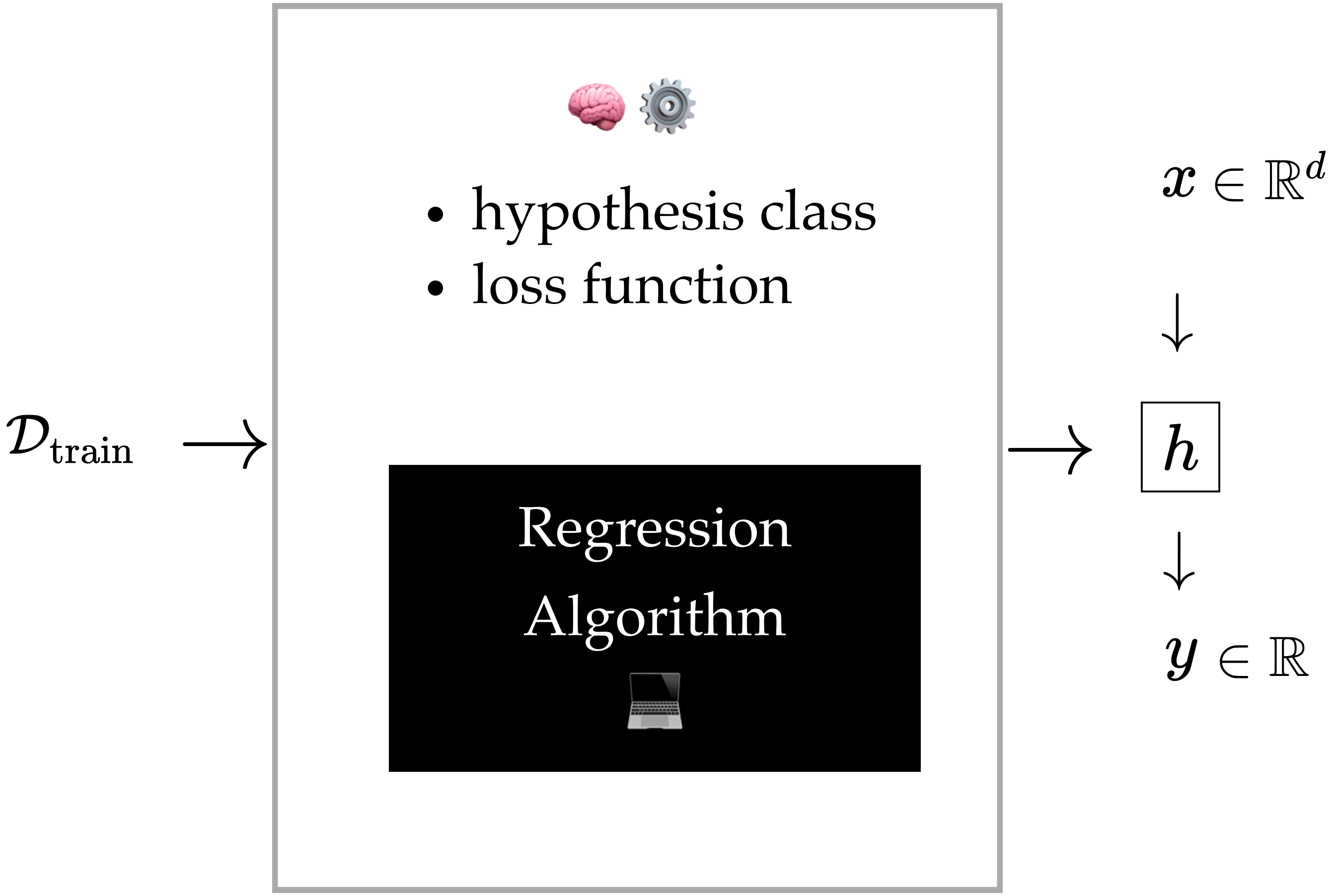
THEORIA
MOTVS CORPORVM
COELESTIVM
IN
SECTIONIBVS CONICIS SOLEM AMBIENTIVM
AVCTORE
CAROLO FRIDERICO GAVSS

HAMBVRGI SVMTIBVS FRID. PERTHES ET I. H. BESSER
1809.

Venditur
Parisiis ap. Treuttel & Würtz. Londini ap. R. H. Evans.
Stockholmiae ap. A. Wiborg. Petropoli ap. Klostermann.
Madriti ap. Sancha. Florentiae ap. Molini, Landi & Cº
Amstelodami in libraria: Kunst- und Industrie-Comptoir, dicta.

Primary source: Gauss, *Theoria motus* (1809).

General regression



Ordinary least squares regression

- linear hypothesis class
- squared loss function

Closed-form formula

$\mathcal{D}_{\text{train}} \longrightarrow$ [ ] $\longrightarrow$ $\boxed{h}$

$x \in \mathbb{R}^d$

$\downarrow$

h

$\downarrow$

$y \in \mathbb{R}$

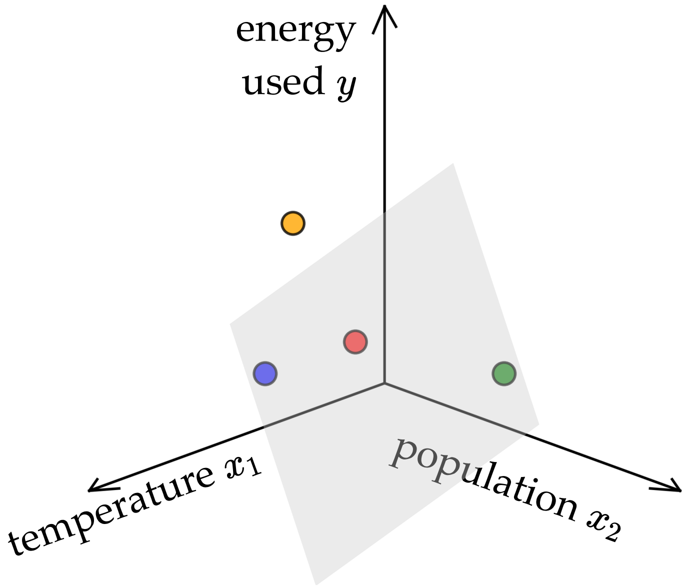
Ordinary least squares regression

Linear hypothesis class:

$$h(x;\theta) = \begin{bmatrix} \theta_1 & \theta_2 & \cdots & \theta_d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} = \theta^T x$$

parameters

features



Squared loss function: $\mathcal{L}\left(h\left(x^{(i)}\right), y^{(i)}\right) = (\theta^T x^{(i)} - y^{(i)})^2$

for now, ignoring the offset

Deriving the OLS solution

1. Write the training error $J(\theta)$ in scalar form

2. Rearrange into matrix-vector form $J(\theta) = \frac{1}{n}(X\theta - Y)^\top(X\theta - Y)$

3. Set the gradient $\nabla_\theta J$ to zero

4. Solve for the optimal parameters $\theta^* = \left(X^\top X\right)^{-1} X^\top Y$

Note: step 3 $\rightarrow$ 4 ($\nabla_\theta J(\theta) = 0 \implies \theta$ is a minimizer) isn't always true in general. We'll discuss when this implication breaks in Week 3.

1. Write out training error:

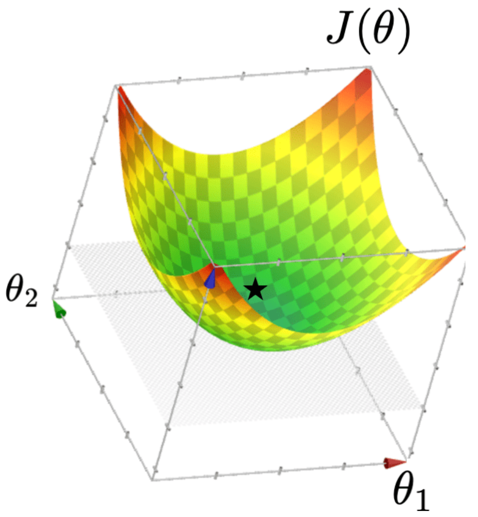
e.g.:

City	Features		Label
	Temp (°F)	Pop (M)	Energy (kWh)
Chicago	90	7.2	45
New York	20	9.5	32
Boston	35	8.4	99
San Diego	18	4.3	39

$$J(\theta) = \frac{1}{4}\left[(\theta_1 \cdot 90 + \theta_2 \cdot 7.2 - 45)^2 + (\theta_1 \cdot 20 + \theta_2 \cdot 9.5 - 32)^2 + (\theta_1 \cdot 35 + \theta_2 \cdot 8.4 - 99)^2 + (\theta_1 \cdot 18 + \theta_2 \cdot 4.3 - 39)^2\right]$$

$$\theta = \begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix} \in \mathbb{R}^{2\times 1}$$

- Q: What kind of function is $J(\theta)$?
- A: Quadratic function
- Q: What does $J(\theta)$ look like?
- A: *Typically*, looks like a "bowl"



2. Rearrange training error into matrix-vector form

City	Features		Label
	Temp (°F)	Pop (M)	Energy (kWh)
Chicago	90	7.2	45
New York	20	9.5	32
Boston	35	8.4	99
San Diego	18	4.3	39

$$X = \begin{bmatrix} 90 & 7.2 \\ 20 & 9.5 \\ 35 & 8.4 \\ 18 & 4.3 \end{bmatrix} \in \mathbb{R}^{4\times 2} \qquad Y = \begin{bmatrix} 45 \\ 32 \\ 99 \\ 39 \end{bmatrix} \in \mathbb{R}^{4\times 1} \qquad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \in \mathbb{R}^{2\times 1}$$

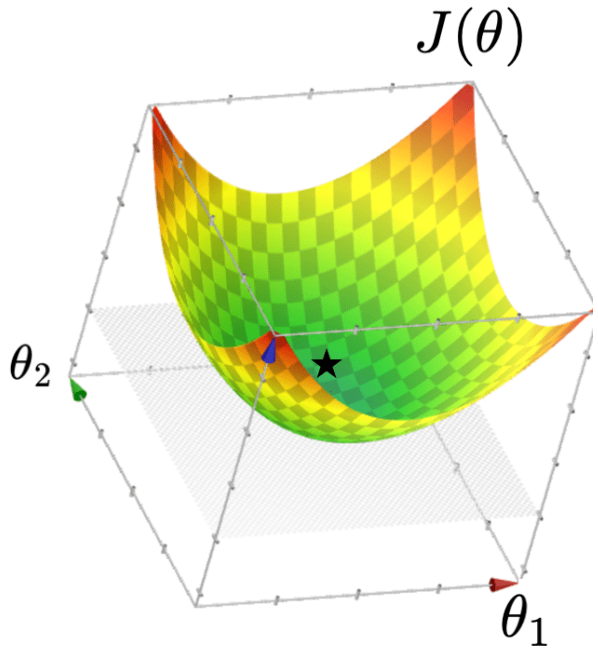
$$J(\theta) = \frac{1}{4}\left[(\theta_1 \cdot 90 + \theta_2 \cdot 7.2 - 45)^2 + (\theta_1 \cdot 20 + \theta_2 \cdot 9.5 - 32)^2 + (\theta_1 \cdot 35 + \theta_2 \cdot 8.4 - 99)^2 + (\theta_1 \cdot 18 + \theta_2 \cdot 4.3 - 39)^2\right]$$

$$\Downarrow$$

$$J(\theta) = \frac{1}{n}(X\theta - Y)^\top (X\theta - Y)$$

3. Get the gradient $\nabla_\theta J \stackrel{\text{set}}{=} 0$

$$\nabla_\theta J = \begin{bmatrix} \partial J / \partial \theta_1 \\ \vdots \\ \partial J / \partial \theta_d \end{bmatrix} = \frac{2}{n}\left(X^T X\theta - X^T Y\right)$$



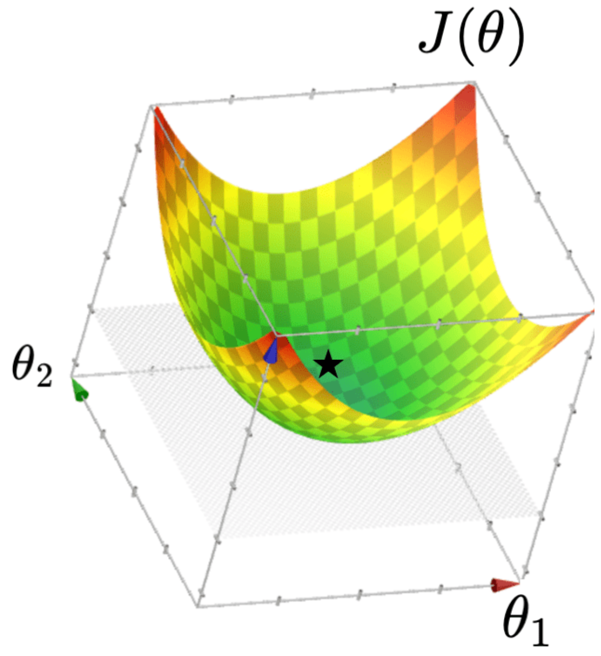
4. Set the gradient $\nabla_\theta J \stackrel{\text{set}}{=} 0$

$$\Rightarrow \quad \theta^* = \left(X^\top X\right)^{-1} X^\top Y$$

- *Typically*, $J(\theta)$ "curves up"
- The minimizer of $J(\theta)$ *necessarily* has a gradient zero.

The beauty of

$$\theta^* = \left(X^\top X\right)^{-1} X^\top Y$$



- When θ^* is well defined, it's the unique minimizer of $J(\theta)$
- Closed-form solution, does not feel like "training"
- Very rare case where we get a general and clean solution with nice theoretical guarantee.

Summary

- Terminologies:

supervised learning | regression | training data | test data | features | label | loss function | training error | test error | hypothesis | hypothesis class

- Ordinary least squares regression:
  - linear hypothesis class, squared loss, mean-squared error
  - matrix-vector form objective $J(\theta) = \frac{1}{n}(X\theta - Y)^\top(X\theta - Y)$
  - closed-form solution $\theta^* = \left(X^\top X\right)^{-1} X^\top Y$

$$\theta^* = \left(X^\top X\right)^{-1} X^\top Y$$

When θ^* is well defined, it's the unique minimizer of $J(\theta)$

Looking ahead:

- When is this θ^* not well defined?
- What can cause this "not well defined"?
- What happens if we are just "close to not well-defined", aka "ill-conditioned"?
- We'll discuss all these next week.

Reference: Gradient Vector Refresher

(5 important facts for 6.390)

For $f : \mathbb{R}^m \to \mathbb{R}$, its ***gradient*** $\nabla f : \mathbb{R}^m \to \mathbb{R}^m$ is defined at the point $p = (x_1, \ldots, x_m)$ as:

$$\nabla f(p) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(p) \\ \vdots \\ \frac{\partial f}{\partial x_m}(p) \end{bmatrix}$$

1. The gradient generalizes the concept of a derivative to multiple dimensions.
2. By construction, the gradient's dimensionality always matches the function input.

Sometimes the gradient is undefined or ill-behaved, but today it is well-behaved.

3. The gradient can be symbolic or numerical.

$$\nabla f(p) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(p) \\ \vdots \\ \frac{\partial f}{\partial x_m}(p) \end{bmatrix}$$

example: $f(x,y,z) = x^2 + y^3 + z$

its *symbolic* gradient:

$$\nabla f(x,y,z) = \begin{bmatrix} 2x \\ 3y^2 \\ 1 \end{bmatrix}$$

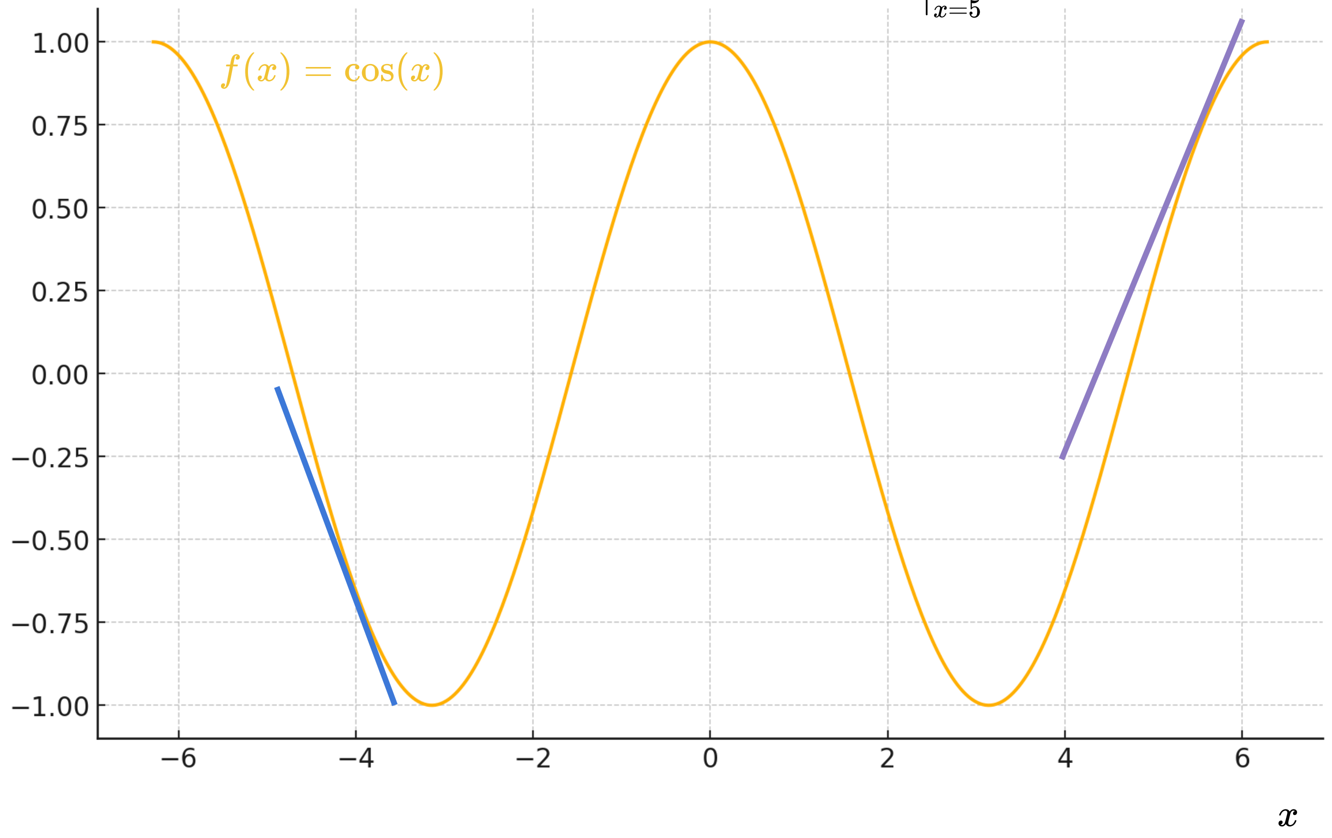
evaluating the symbolic gradient at a point gives a *numerical* gradient:

$$\nabla f(3,2,1) = \nabla f(x,y,z)\Big|_{(x,y,z)=(3,2,1)} = \begin{bmatrix} 6 \\ 12 \\ 1 \end{bmatrix}$$

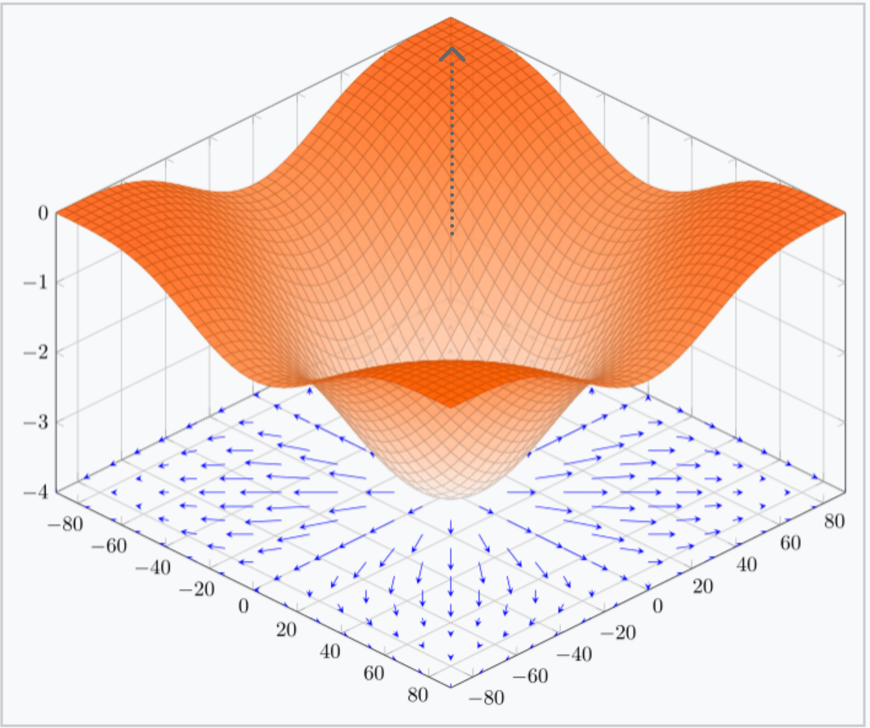
just like a derivative can be a function or a number.

4. The gradient points in the direction of the (steepest) *increase* in the function value.

$$\left.\frac{d}{dx}\cos(x)\right|_{x=5} = -\sin(5) \approx 0.9589$$

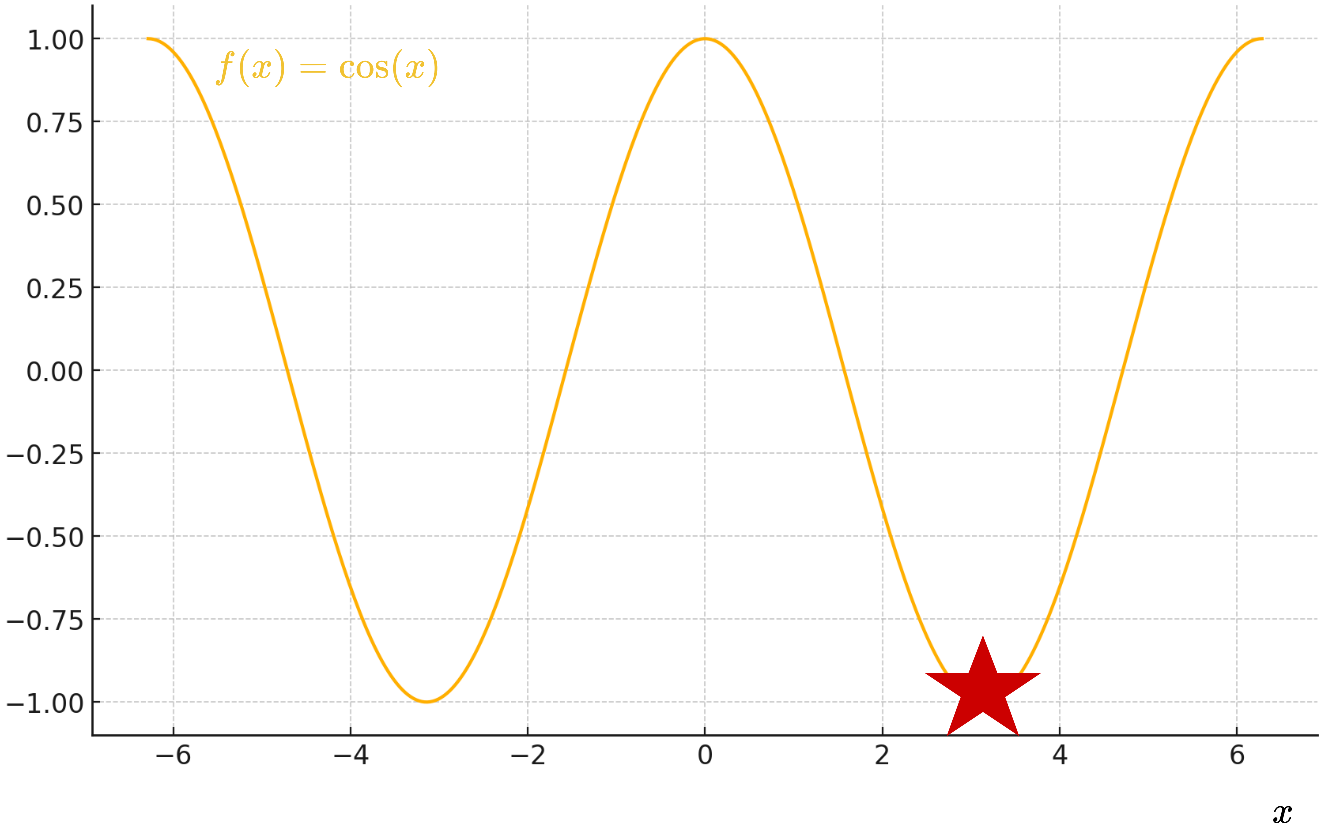


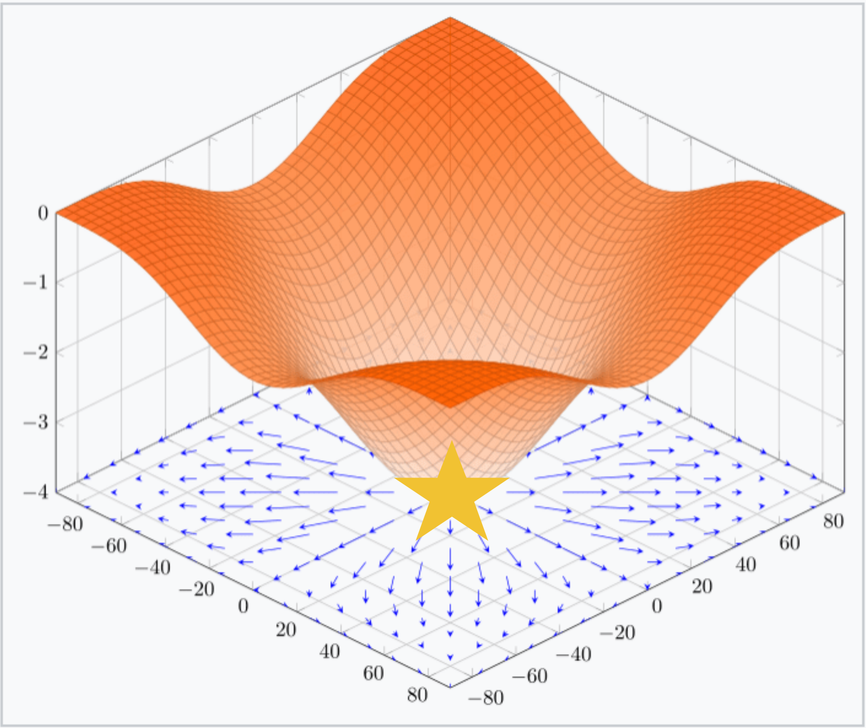
$$\left.\frac{d}{dx}\cos(x)\right|_{x=-4} = -\sin(-4) \approx -0.7568$$



The gradient of the function $f(x,y) = -(\cos^2 x + \cos^2 y)^2$

5. The gradient at the function minimizer is *necessarily* zero.





The gradient of the function $f(x,y) = -(\cos^2 x + \cos^2 y)^2$

For $f : \mathbb{R}^m \to \mathbb{R}$, its ***gradient*** $\nabla f : \mathbb{R}^m \to \mathbb{R}^m$ is defined at the point $p = (x_1, \ldots, x_m)$ as:

$$\nabla f(p) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(p) \\ \vdots \\ \frac{\partial f}{\partial x_m}(p) \end{bmatrix}$$

1. The gradient generalizes the concept of a derivative to multiple dimensions.
2. By construction, the gradient's dimensionality always matches the function input.
3. The gradient can be symbolic or numerical.
4. The gradient points in the direction of the (steepest) *increase* in the function value.
5. The gradient at the function minimizer is *necessarily* zero.

Sometimes the gradient is undefined or ill-behaved, but today it is well-behaved.

Extension: How to deal with offset θ_0

1. center the data; *or*
2. append a fake feature of 1

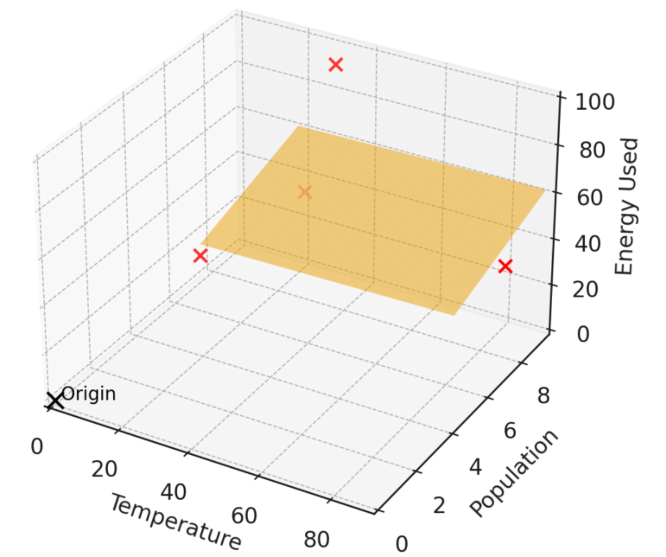
1. "center" the data

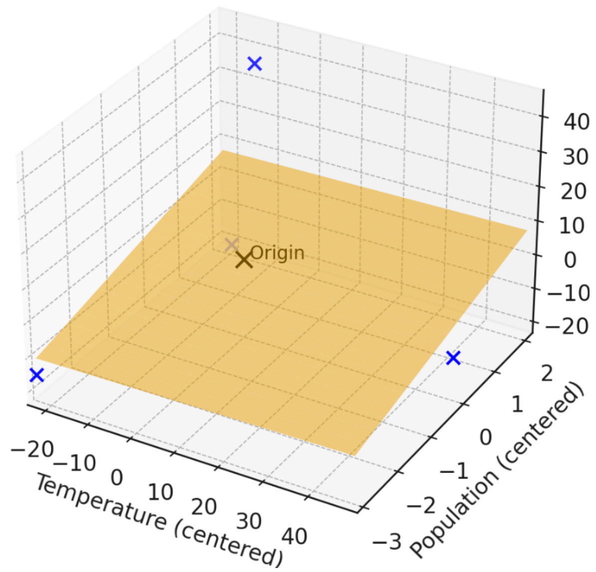
	Features		Label
City	**Temperature**	**Population**	**Energy Used**
Chicago	90	7.2	45
New York	20	9.5	32
Boston	35	8.4	100
San Diego	18	4.3	39

$\Downarrow$ centering

	Features		Label
City	**Temperature**	**Population**	**Energy Used**
Chicago	49.25	-0.15	-9.00
New York	-20.75	2.15	-22.00
Boston	-5.75	1.05	46.00
San Diego	-22.75	-3.05	-15.00

all column-wise $\Sigma = 0$





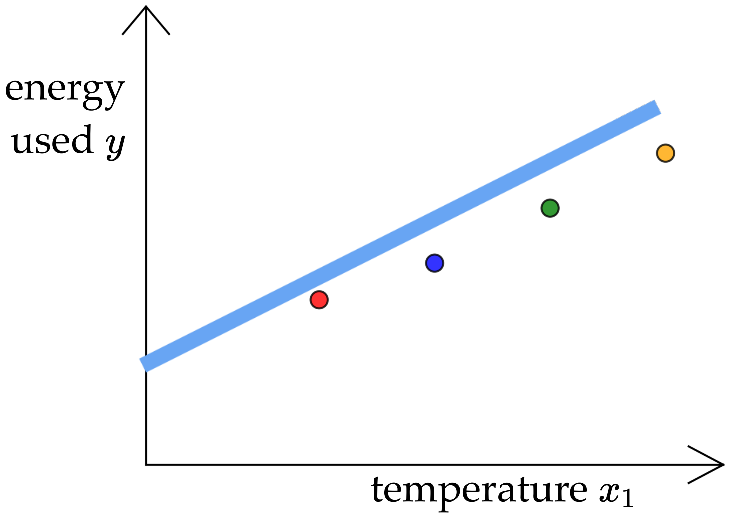
when data is centered, the optimal offset is guaranteed to be 0

2. Append a "fake" feature of 1

$$h\left(x;\theta,\theta_0\right)=\theta^T x+\theta_0 = \left[\begin{array}{llll}\theta_1 & \theta_2 & \cdots & \theta_d\end{array}\right]\left[\begin{array}{c}x_1 \\ x_2 \\ \vdots \\ x_d\end{array}\right]+\theta_0$$

$$=\left[\begin{array}{lllll}\theta_1 & \theta_2 & \cdots & \theta_d & \theta_0\end{array}\right]\left[\begin{array}{c}x_1 \\ x_2 \\ \vdots \\ x_d \\ 1\end{array}\right]$$

$$=\theta_{\text{aug}}^T x_{\text{aug}}$$



trick our model: treat the bias as just another feature, always equal to 1.
See recitation 1 for details.