General

- Everything is on dl4cv.github.io (and Moodle)
- In-person (except me)
- 4 credit points (except chemistry ☹)
- 2 hrs. lecture, ~2 hrs. tutorial (Tutorial covers new material)
- 3-4 homework assignments + Final Project
  Homework is demanding, but worth it

”If we have seen further, it is by standing on the shoulders of Giants"

- From basic to most recent SotA
- Slightly biased towards Weizmann research
- Intuition is important, but so do details
- Hands-on
- Openness

We Assume you...

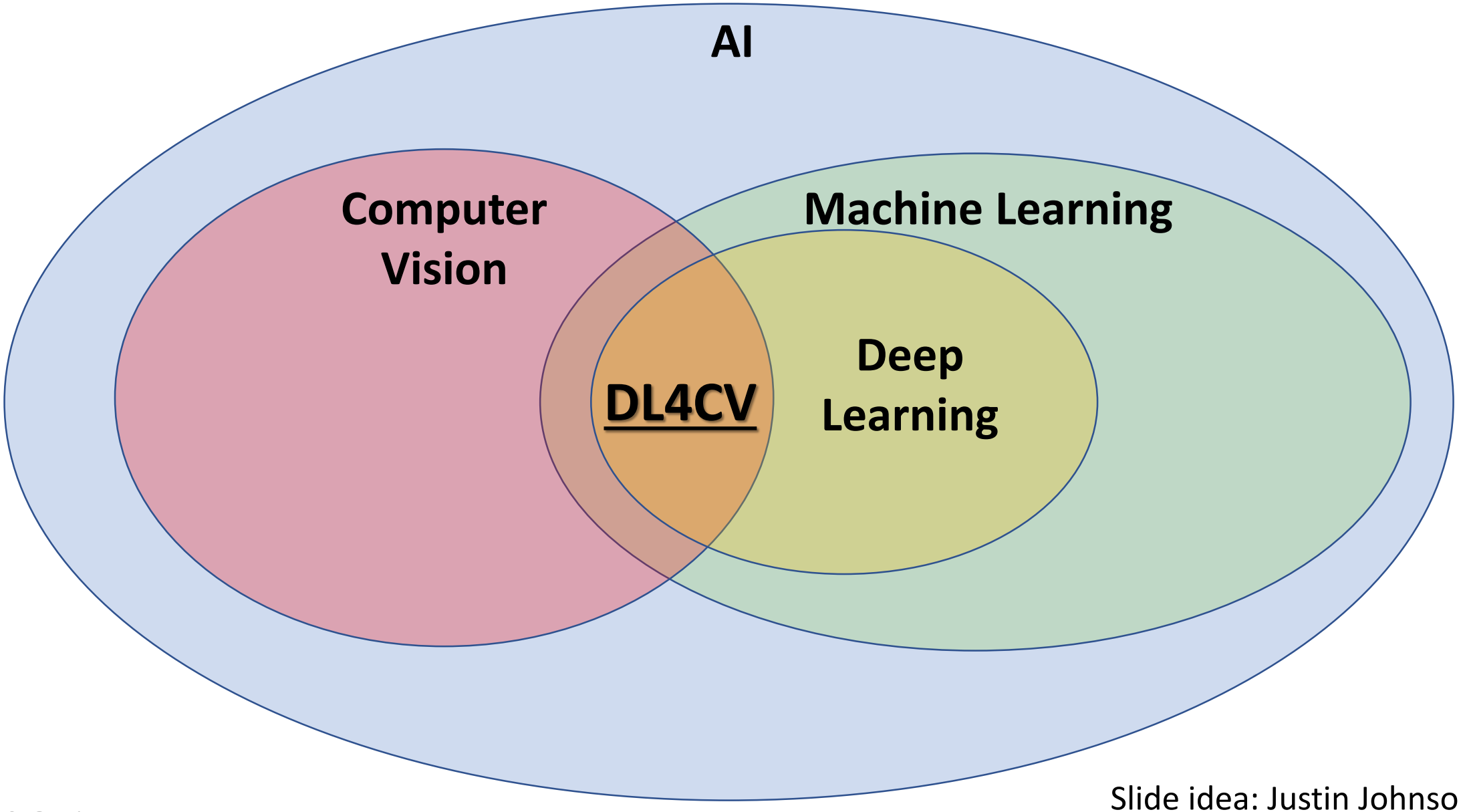
- Know Basic Calculus (e.g. know what is a Gradient).
- Know Basic Algebra (e.g. Vector spaces, Matrix multiplication, Eigen decomposition).
- Written code before (preferably Python).
- Bumped into Machine Learning (e.g. heard the term “Overfitting”).

Homework

Theory | **From Scratch** | **Applied**

HW1 is online!

Road map

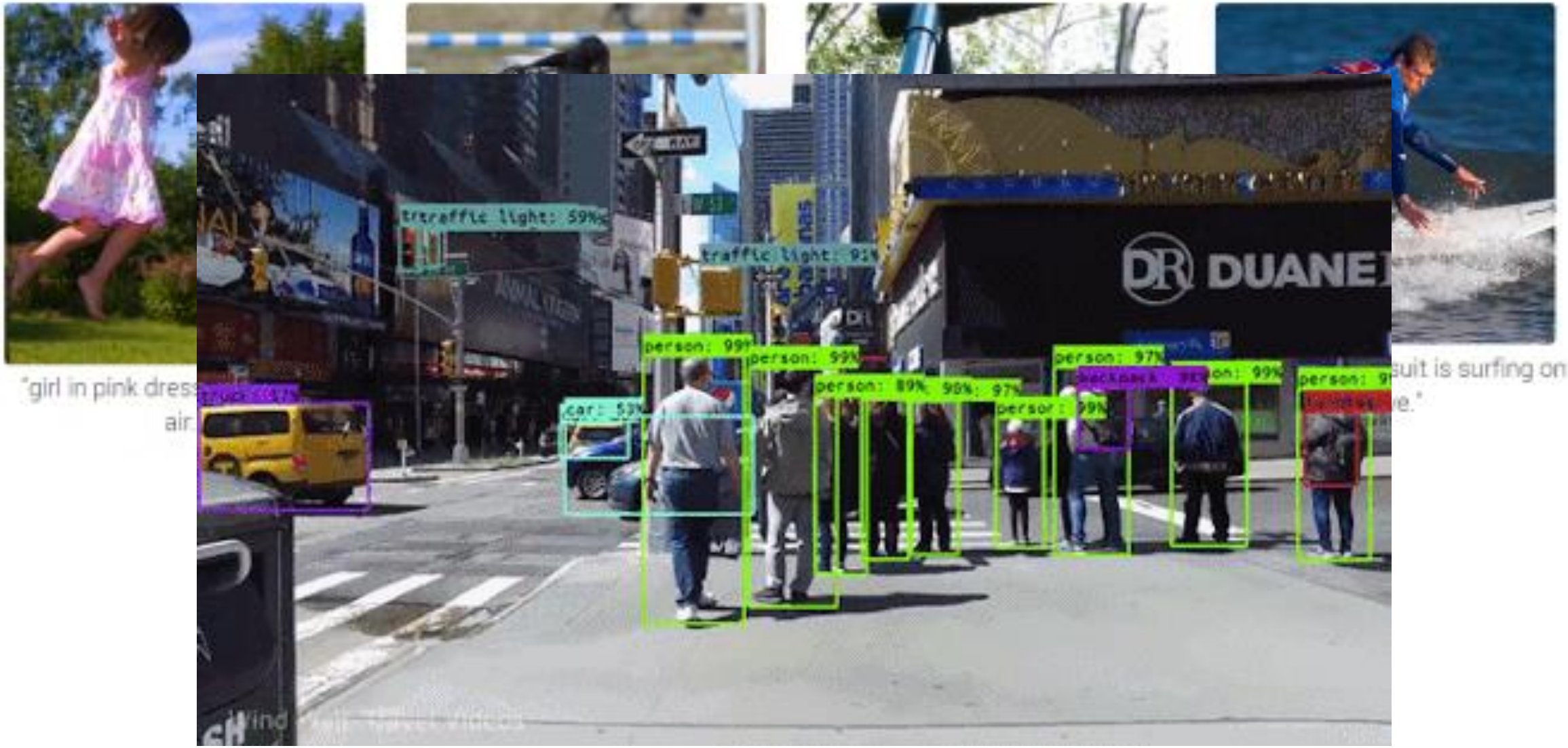


Slide idea: Justin Johnson

Today:

- Motivation and history (15%)
- Supervised learning (25%)
- Linear regression (20%)
- Gradient descent (25%)
- Feature transform (15%)

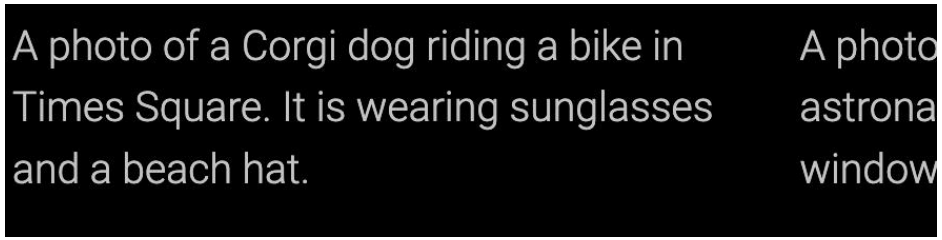
Deep Learning is powerful



Andrej Karpathy, Li Fei-Fei, CVPR 2015 Deep Visual-Semantic Alignments for Generating Image Descriptions

Abhishek Bansal- DetectMe (GitHub)

Deep Learning is powerful!



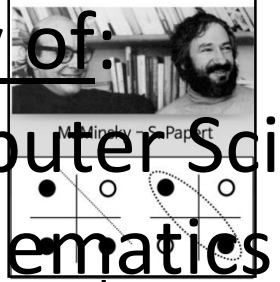
Sharia et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding ; 2022

History of:

- Computer Science
- Mathematics
- Neuroscience

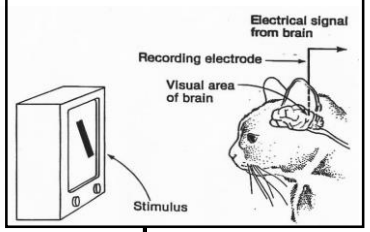
McCulloch Pitts
Non learned
Neuron

XOR kills perceptron

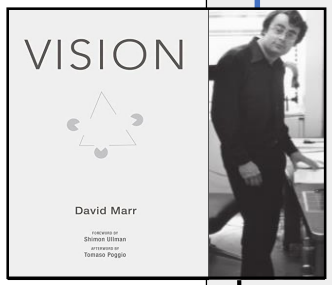


Neoc
(Fuk
Firs
Net

Hubel & Wiesel



VISION



1957

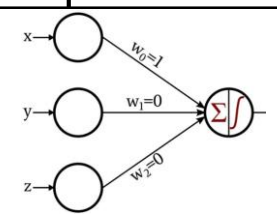
1970

1943

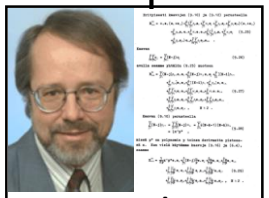
1959

1969

1979



Perceptron (Rosenblatt) learned weights

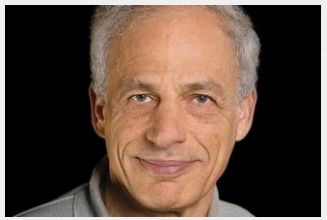


Back Propagation (Linnainmaa)

1st AI Winte

VISION

- Computer vision
- Machine learning

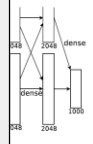


David Marr

FOREWORD BY
Shimon Ullman

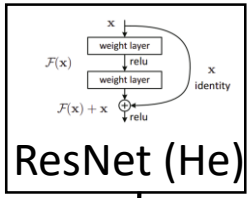
AFTERWORD BY
Tomaso Poggio

egins!
Hinton



Bengio, Hinton, LeCunn
Turing Award

ANs
ood
llow)



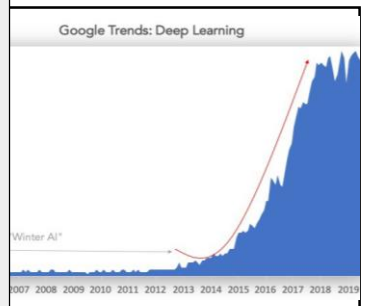
ResNet (He)

2017

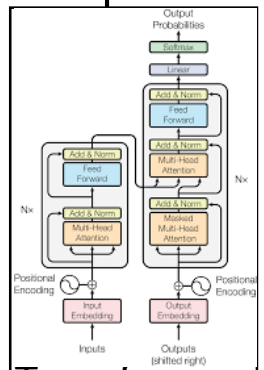
2014

2015

2018



Deep Learning revolution



Transformer (Vaswani)

Supervised Learning

Features

Labels

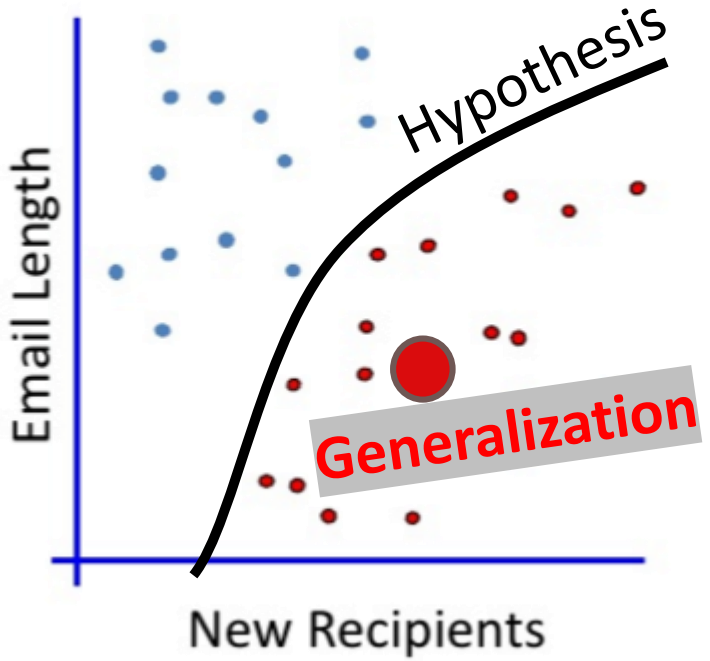
Instances

Number of new Recipients	Email Length (K)	Country (IP)	Customer Type	Email Type
0	2	Germany	Gold	Ham
1	4	Germany	Silver	Ham
5	2	Nigeria	Bronze	Spam
2	4	Russia	Bronze	Spam
3	4	Germany	Bronze	Ham
0	1	USA	Silver	Ham
4	2	USA	Silver	Spam

Numeric

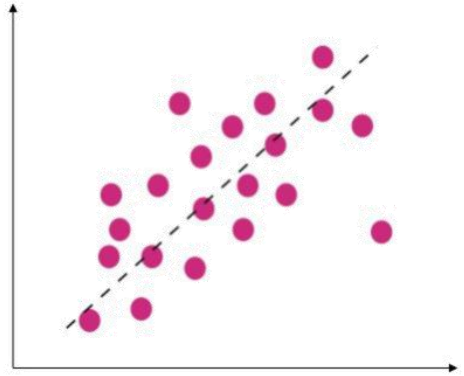
Nominal

Ordinal

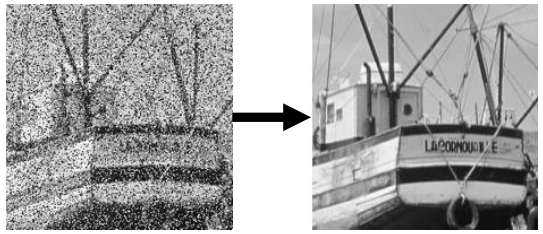


Supervised Learning

Regression

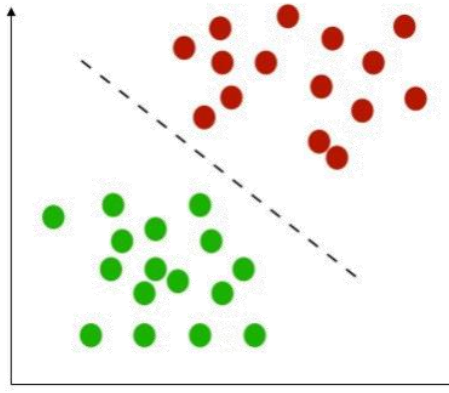


E.g. Image denoising



E.g. Object localization

Classification



E.g. Image classification

Supervised Learning

Learning Algorithm

Training set

Hypothesis

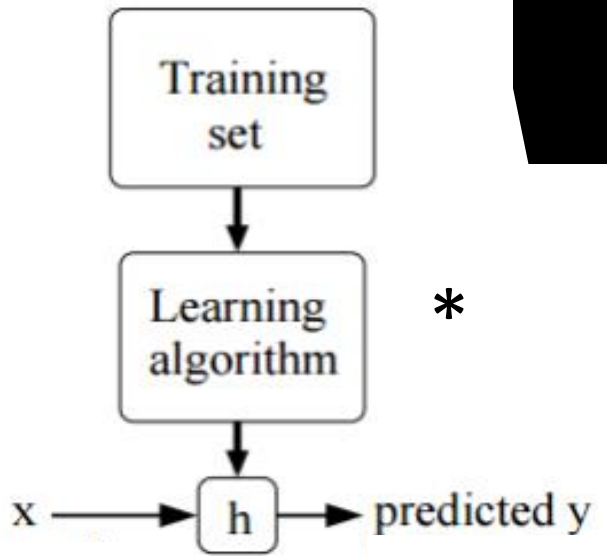
$$A(S) = h$$

Hypothesis class
$\mathcal{H} = \{h_1, h_2 \dots\}$

$h(x) \approx y$

Loss
$\mathcal{L}$

Optimization method



*

Inputs
$\boldsymbol{X}$

Labels
$\boldsymbol{Y}$

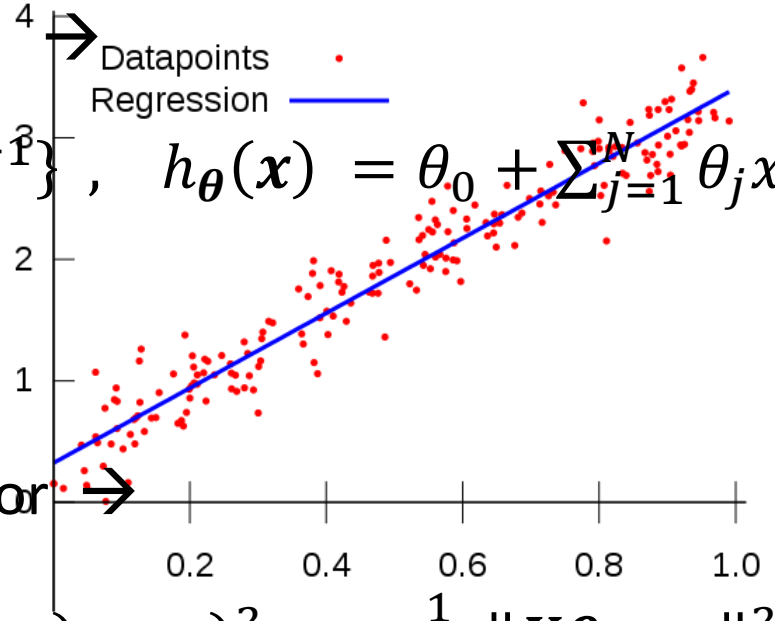
$$\begin{bmatrix} \text{---} \ \mathbf{x_1}^T \ \text{---} \\ \text{---} \ \mathbf{x_2}^T \ \text{---} \\ \vdots \\ \text{---} \ \mathbf{x_M}^T \ \text{---} \end{bmatrix}, \quad \begin{bmatrix} \text{---} \ \mathbf{y_1}^T \ \text{---} \\ \text{---} \ \mathbf{y_2}^T \ \text{---} \\ \vdots \\ \text{---} \ \mathbf{y_M}^T \ \text{---} \end{bmatrix}$$

* Diagram by Andrew NG

Linear Regression

Hypothesis class: Linear →

$$\mathcal{H} = \{h_{\boldsymbol{\theta}} | \, \boldsymbol{\theta} \epsilon \, \mathbb{R}^{N+1}\}, \quad h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \theta_0 + \sum_{j=1}^{N} \theta_j x_j = \theta_0 + \tilde{\boldsymbol{\theta}}^T \boldsymbol{x} = \boldsymbol{\theta}^T \begin{pmatrix} 1 \\ | \\ \boldsymbol{x} \\ | \end{pmatrix}$$



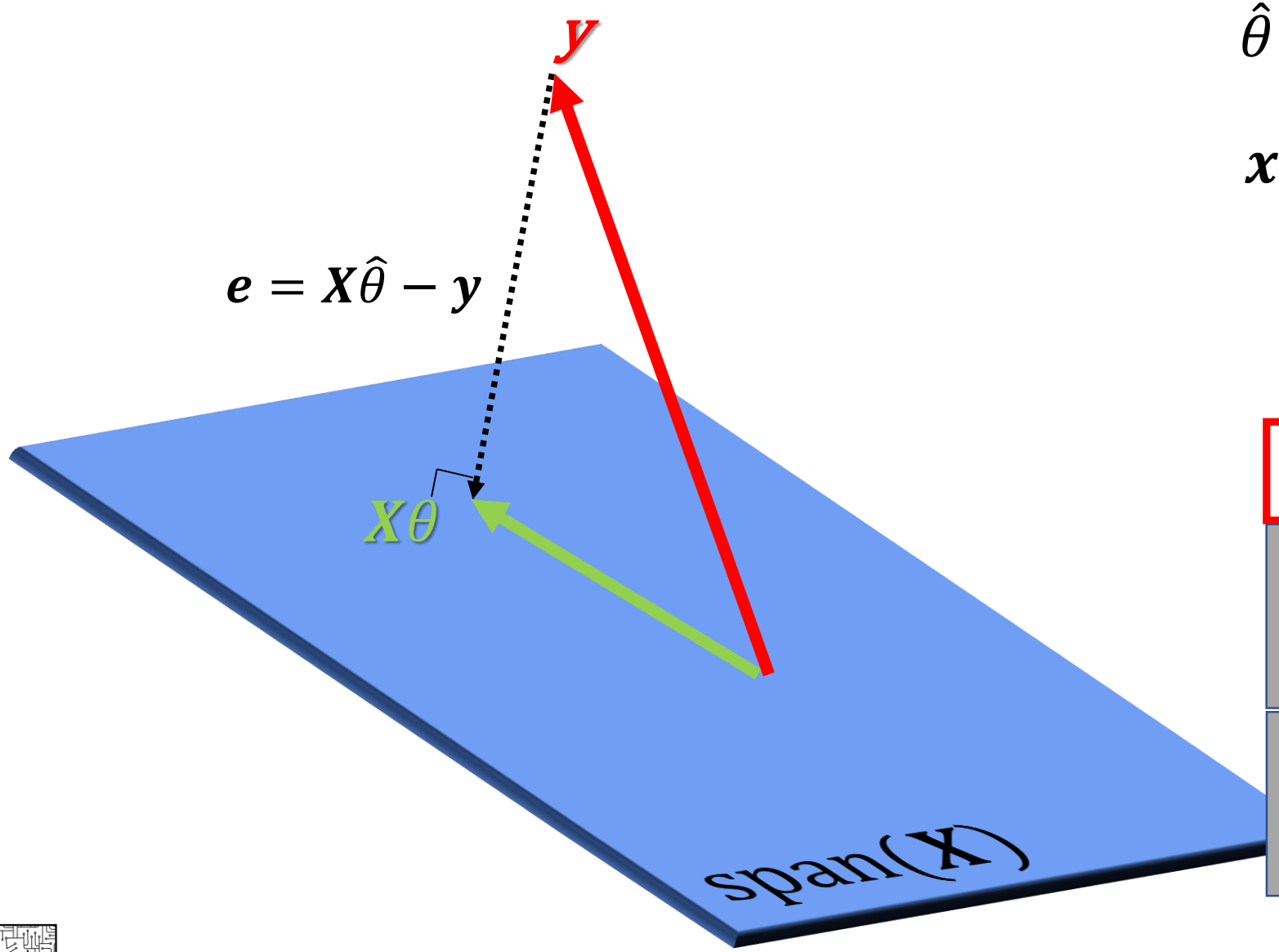
*Bias= Just add **1** at top of the input vec!*

Loss: Mean Squared Error →

$$\mathcal{L} = \frac{1}{2M} \sum_{i=1}^{M} (h_\theta(\boldsymbol{x}_i) - y_i)^2 = \frac{1}{2M} \|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|^2$$

Optimization method: Normal equations / Gradient Descent

Normal Equations (intuition)



$$\hat{\theta} = argmin_{\theta} \|y - X\theta\|^2$$

$$x \perp e \quad \forall\ x \in span(X)$$

$$\Downarrow$$

$$X^T(X\hat{\theta} - y) = 0$$

$$\Downarrow$$

$$\hat{\theta} = (X^T X)^{-1} X^T y \quad *$$

Formal proof: HW

Also in HW:
is $X^T X$ invertible?

$* \ if\ X^T X\ invertible$

Normal Equations

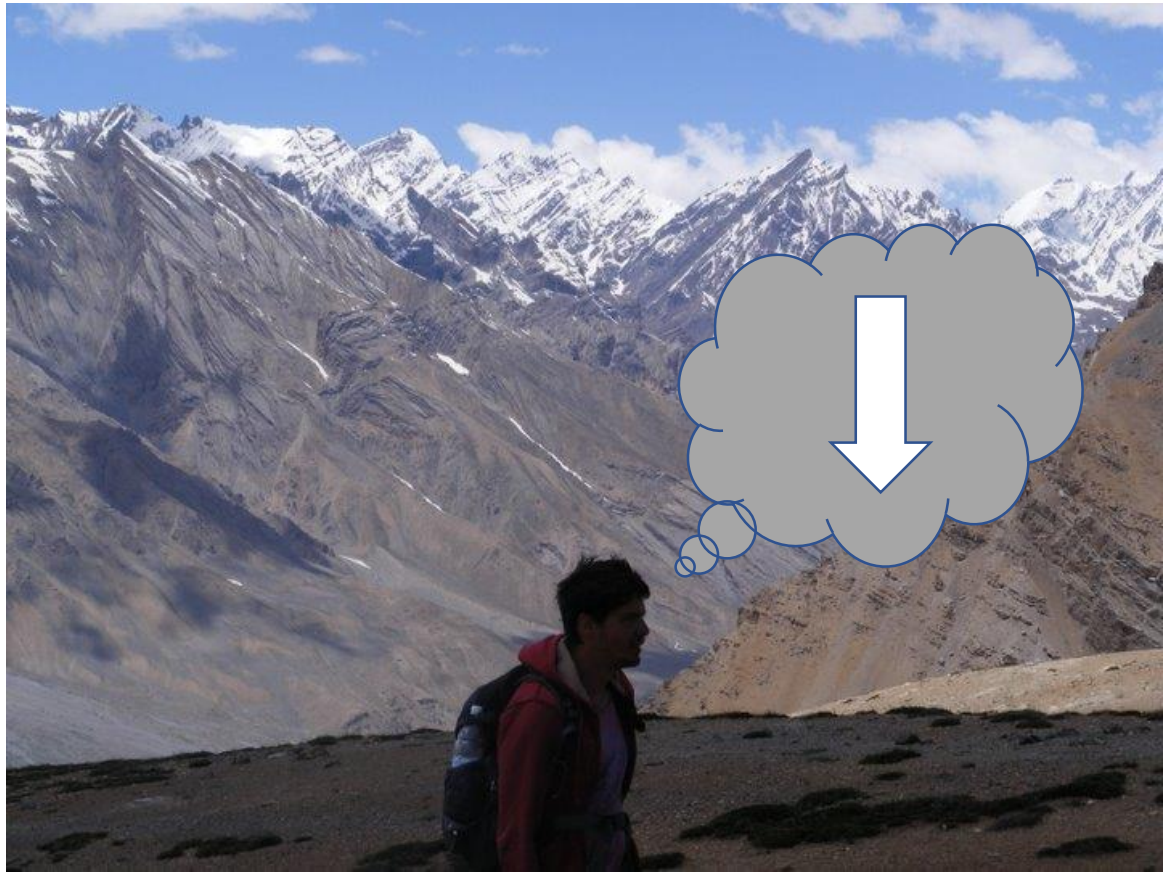
Q: Will normal equations always be practical?

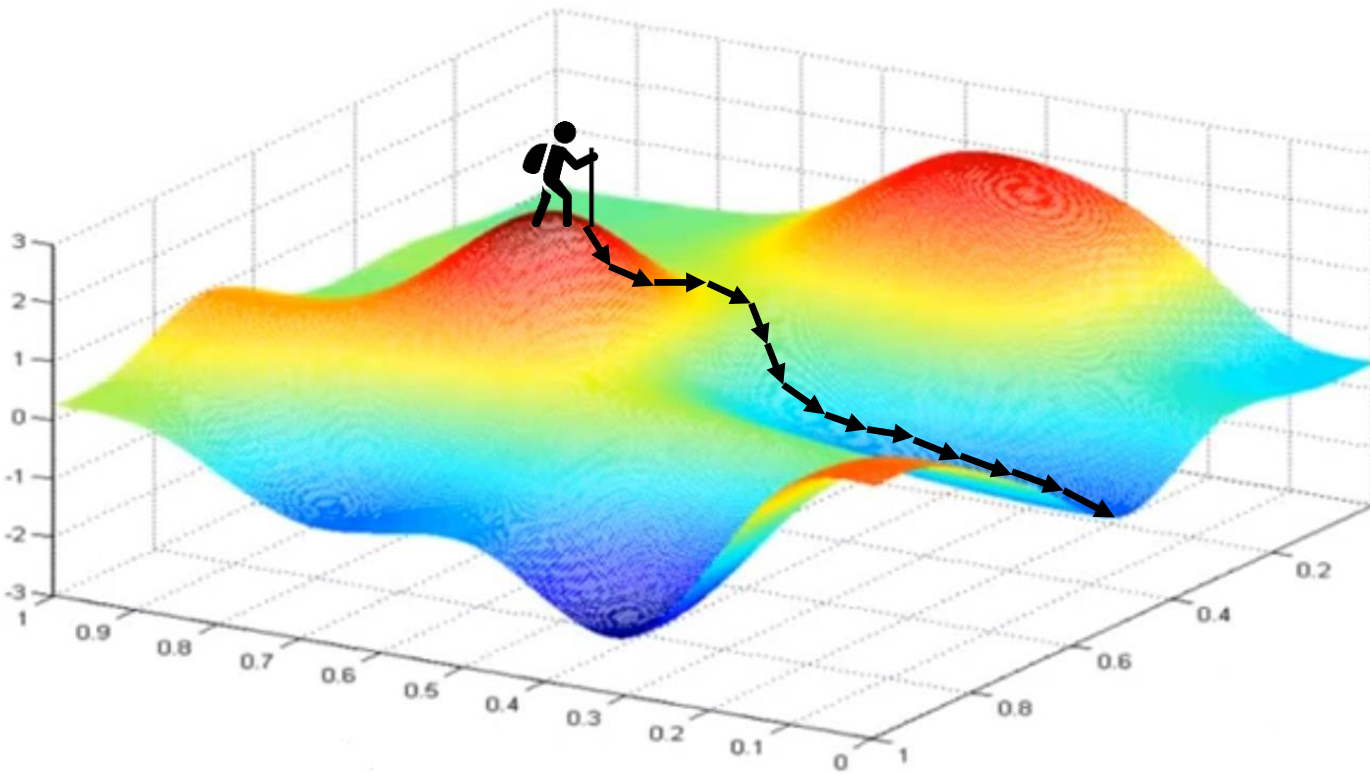
A: No;

1. Inverting $\boldsymbol{X}^T\boldsymbol{X}$ may cost unreasonable memory / time
2. Sometimes not applicable: Regularization? Different loss? More layers?

Gradient descent

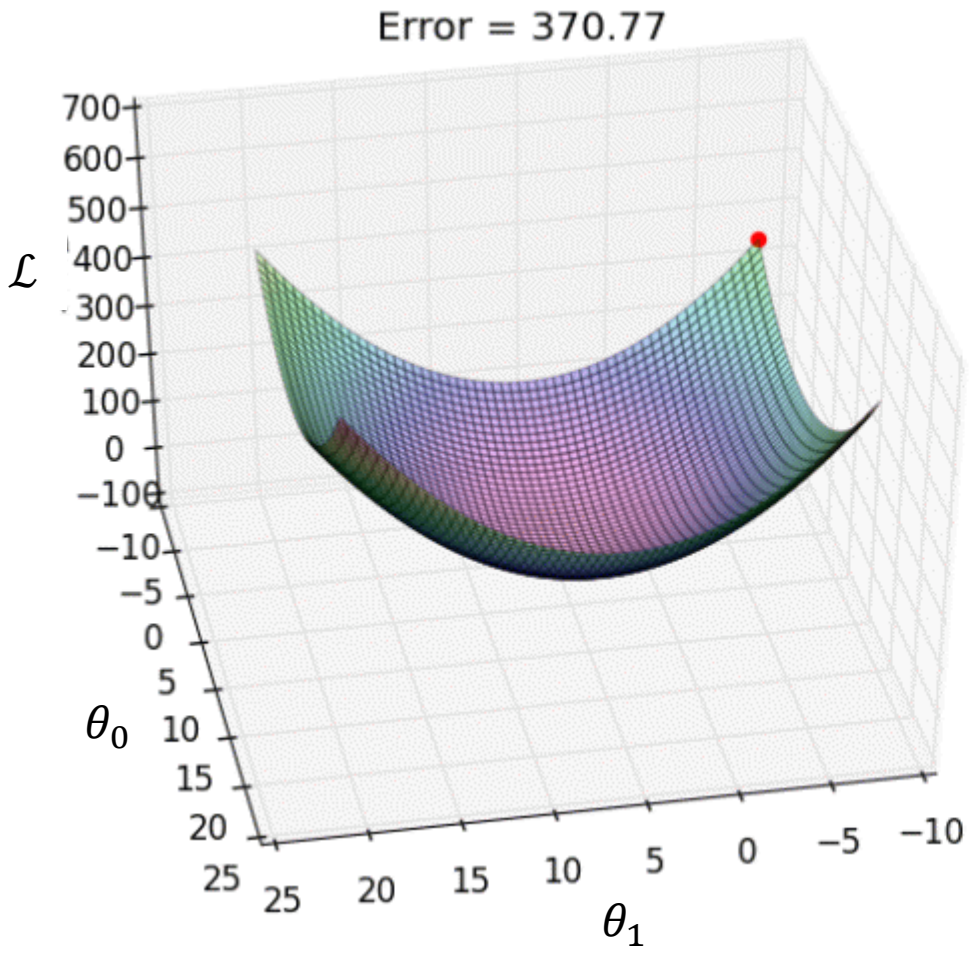
Possible solution: Iteratively reduce loss

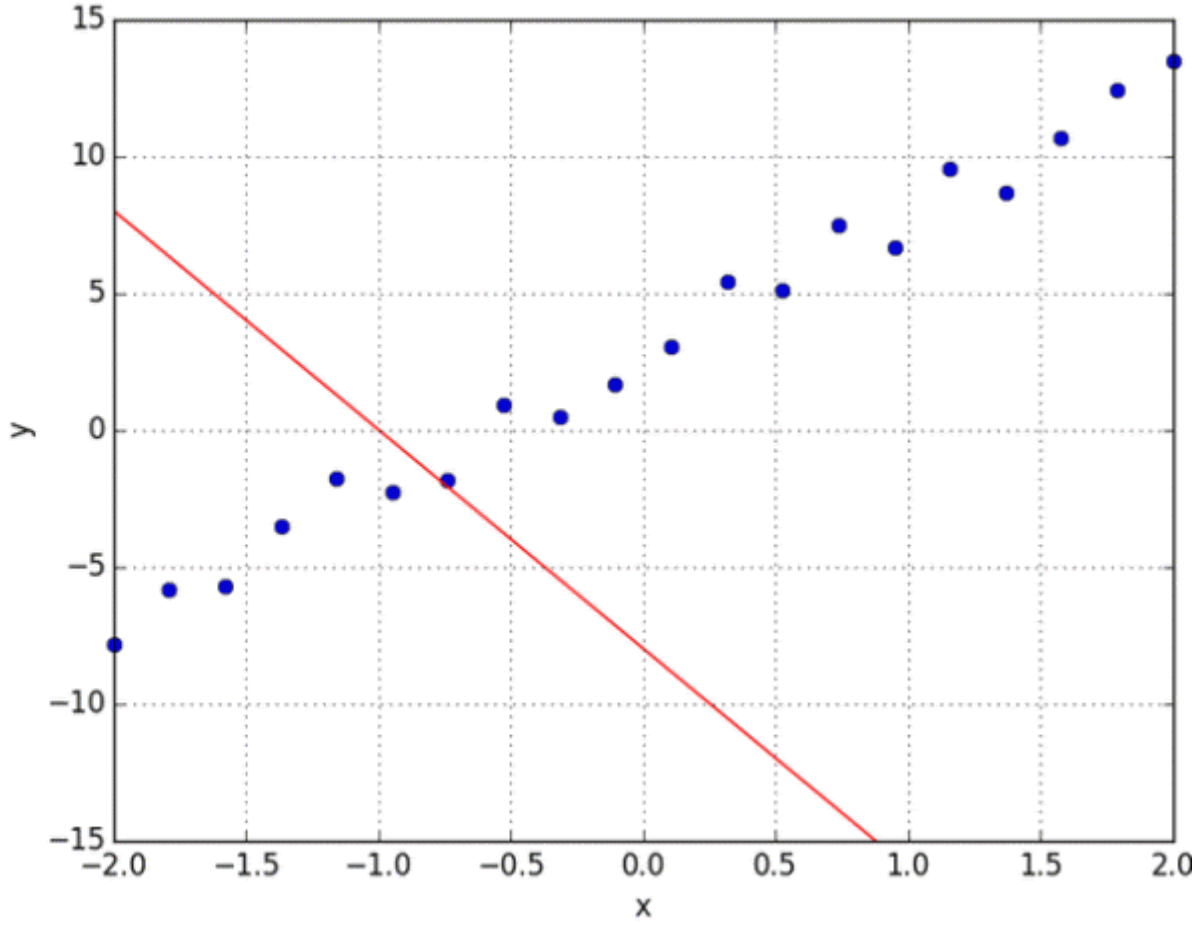




Can we guarantee global min?

Gradient descent





Parameter space: $\mathcal{L}(\boldsymbol{\theta}; S)$

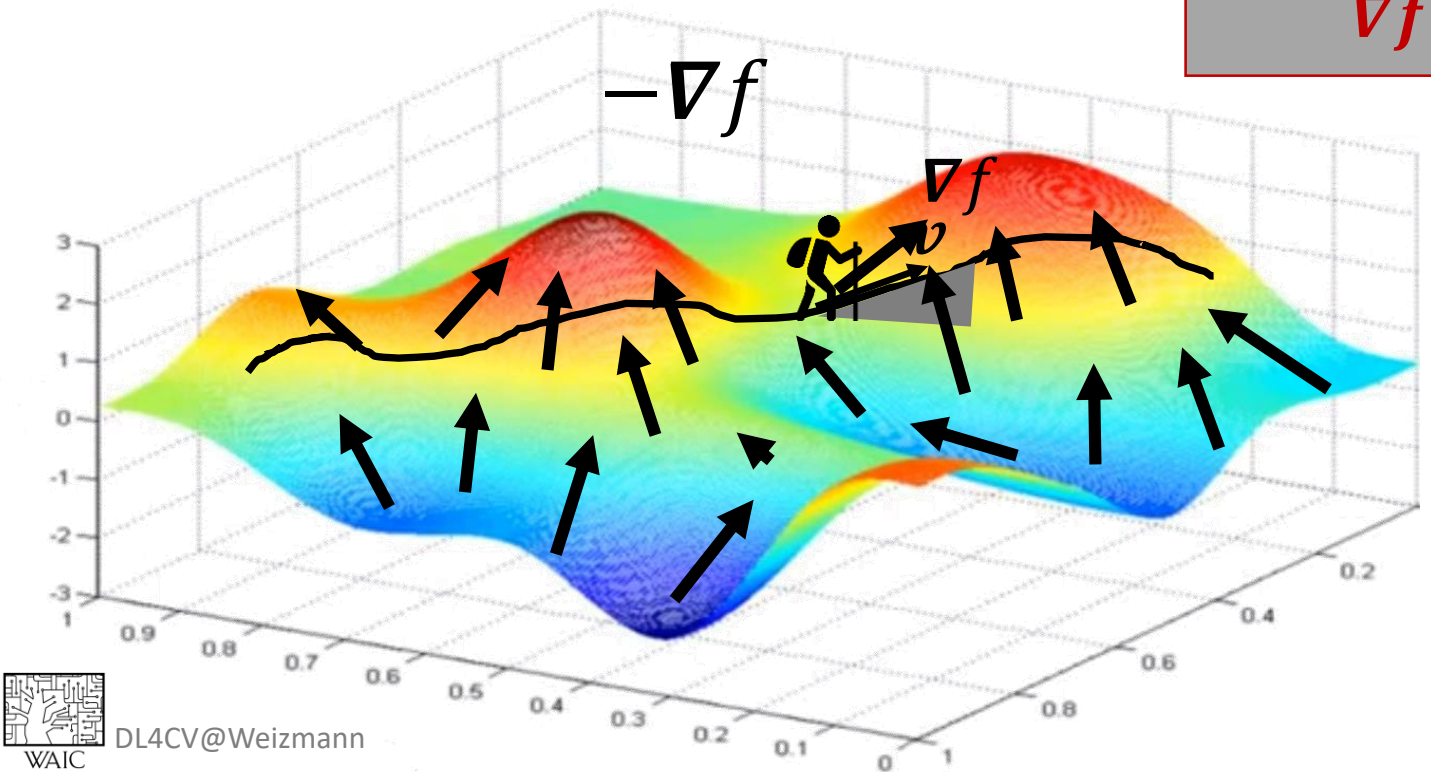
Data space: $h_{\boldsymbol{\theta}}(\boldsymbol{x})$

Calculus reminder: Directional derivative

$$\lim_{\varepsilon \to 0} \frac{f(\boldsymbol{x} + \varepsilon \boldsymbol{v}) - f(\boldsymbol{x})}{\varepsilon \|\boldsymbol{v}\|} = \frac{1}{\|\boldsymbol{v}\|} \sum v_i \frac{\partial f}{\partial x_i} = \frac{\boldsymbol{v}^T}{\|\boldsymbol{v}\|} \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix} = \left\langle \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}, \nabla f \right\rangle$$

If differentiable

Gradient!
$\vec{\nabla} f$



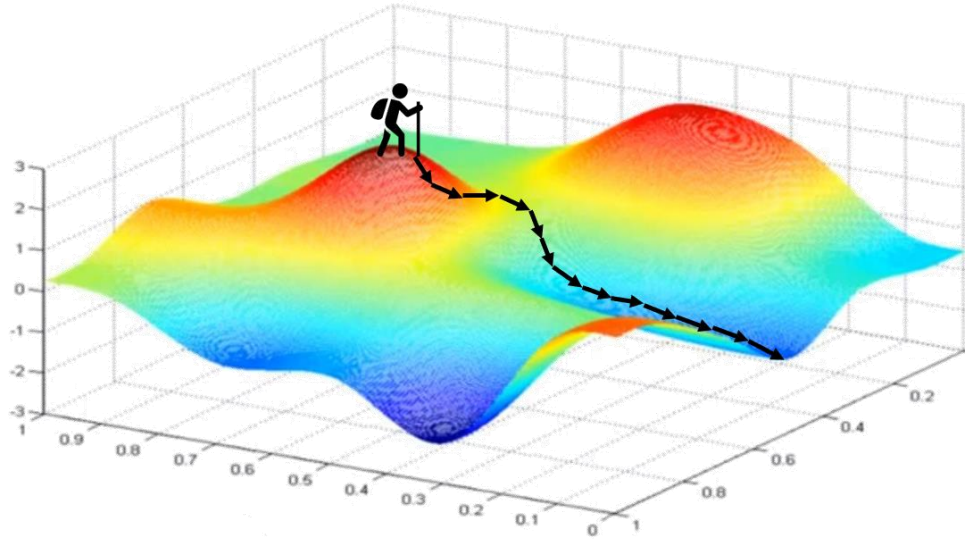
<u>According to Cauchy-Schwarz inequality</u>:

- Max value is $\|\nabla f\|$
- Obtained when $\boldsymbol{v}$ is parallel to ∇f

- Gradient directs to steepest ascent.
- It's size is the max steepness.

Gradient descent

$$\nabla\mathcal{L}(\theta_0, \theta_1 \dots \theta_N) = \begin{pmatrix} \frac{\partial\mathcal{L}}{\partial\theta_0} \\ \frac{\partial\mathcal{L}}{\partial\theta_1} \\ \vdots \\ \frac{\partial\mathcal{L}}{\partial\theta_N} \end{pmatrix}$$



Augustin Louis Cauchy

1. Initialize $\theta \sim$ Random
2. Repeat until convergence:
   {
   $$\boldsymbol{\theta} := \boldsymbol{\theta} - \alpha\nabla\mathcal{L}(\boldsymbol{\theta}; S)$$
   }

α: Learning rate

Gradient descent

Full batch Gradient Descent $\boldsymbol{\theta} := \boldsymbol{\theta} - \alpha \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\theta}; S)$

lescent
ient Descent
gradient descent

]}

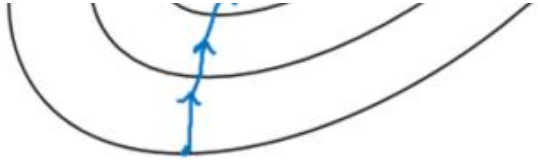


Figure by Z² Little on Medium

Gradient descent for Linear Regression

$$\mathcal{L} = \frac{1}{2m}\sum_{i=1}^{m}(\boldsymbol{\theta}^T \boldsymbol{x}_i - y_i)^2$$

$$\frac{\partial}{\partial \boldsymbol{\theta}}\mathcal{L} = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{\theta}^T \boldsymbol{x}_i - y_i)\boldsymbol{x}_i = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i(\boldsymbol{X\theta} - \boldsymbol{y})_{\boldsymbol{i}} = \boldsymbol{X}^T(\boldsymbol{X\theta} - \boldsymbol{y})$$

$\boldsymbol{e}$

Repeat until convergence:

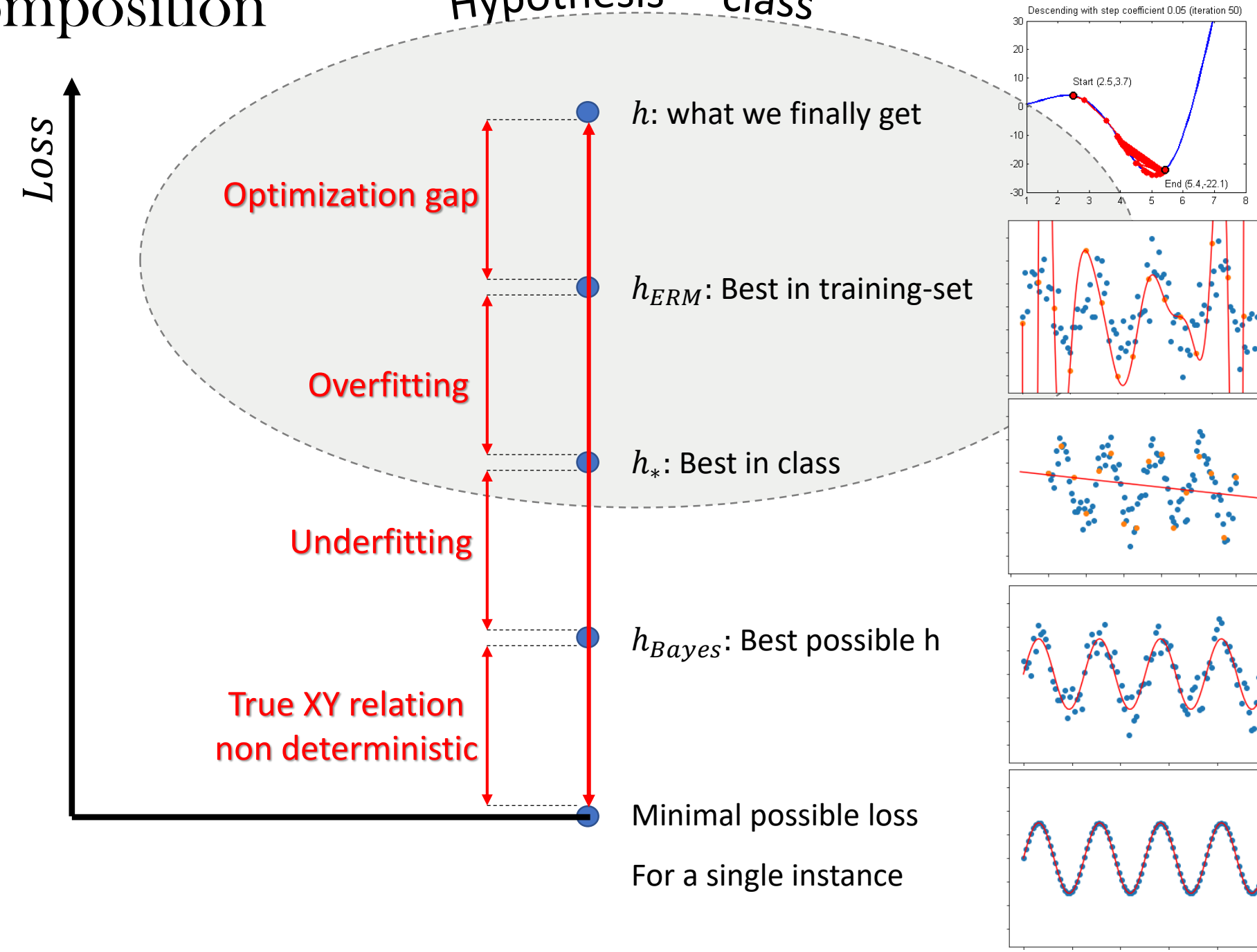
{

$$\boldsymbol{\theta} := \boldsymbol{\theta} - \frac{\alpha}{m}\boldsymbol{X}^T\boldsymbol{e}$$

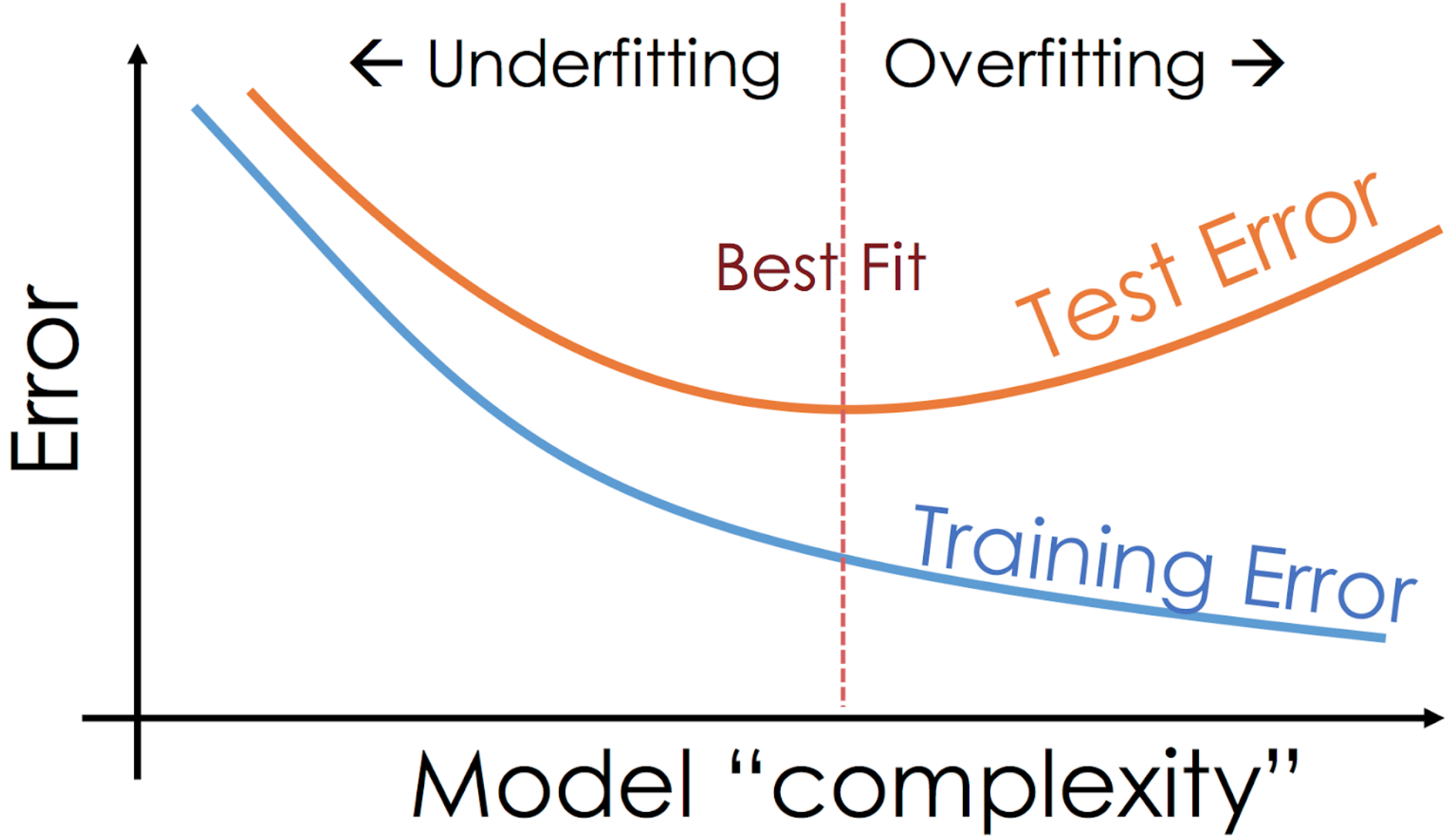
}

Q: Find the relation between convergence and Normal Equations

Error decomposition



Generalization



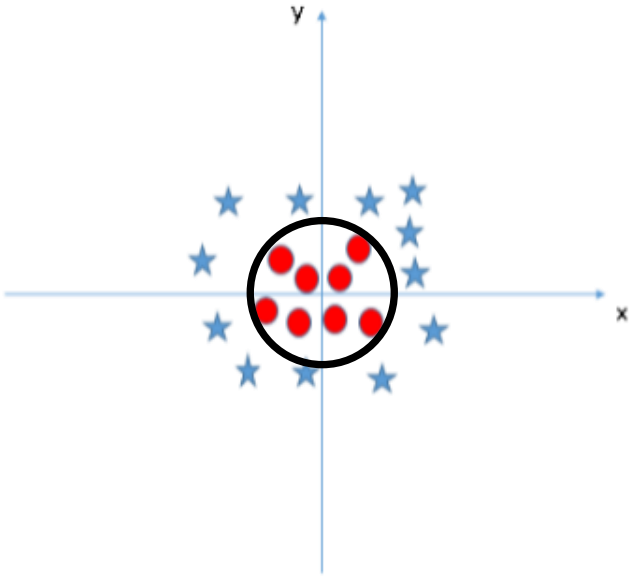
Overfitting- Data influence

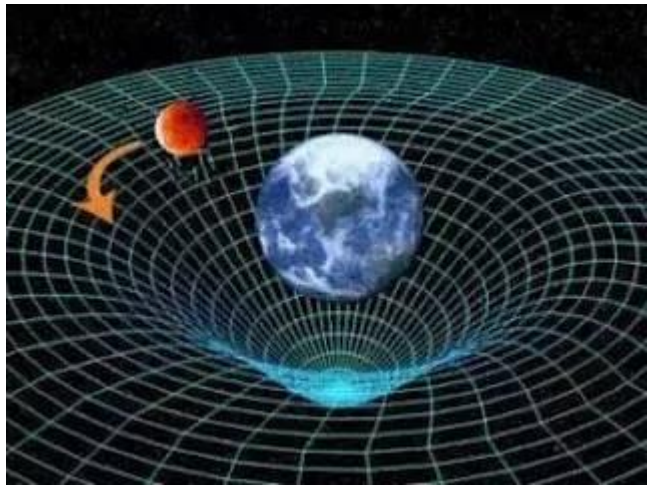
Matrix A

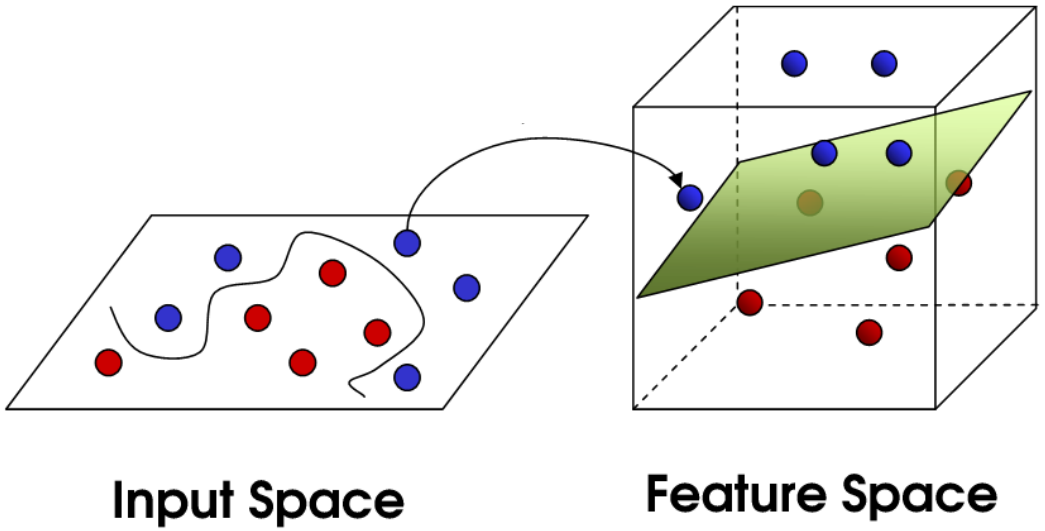
$$\begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix}$$

Feature transform

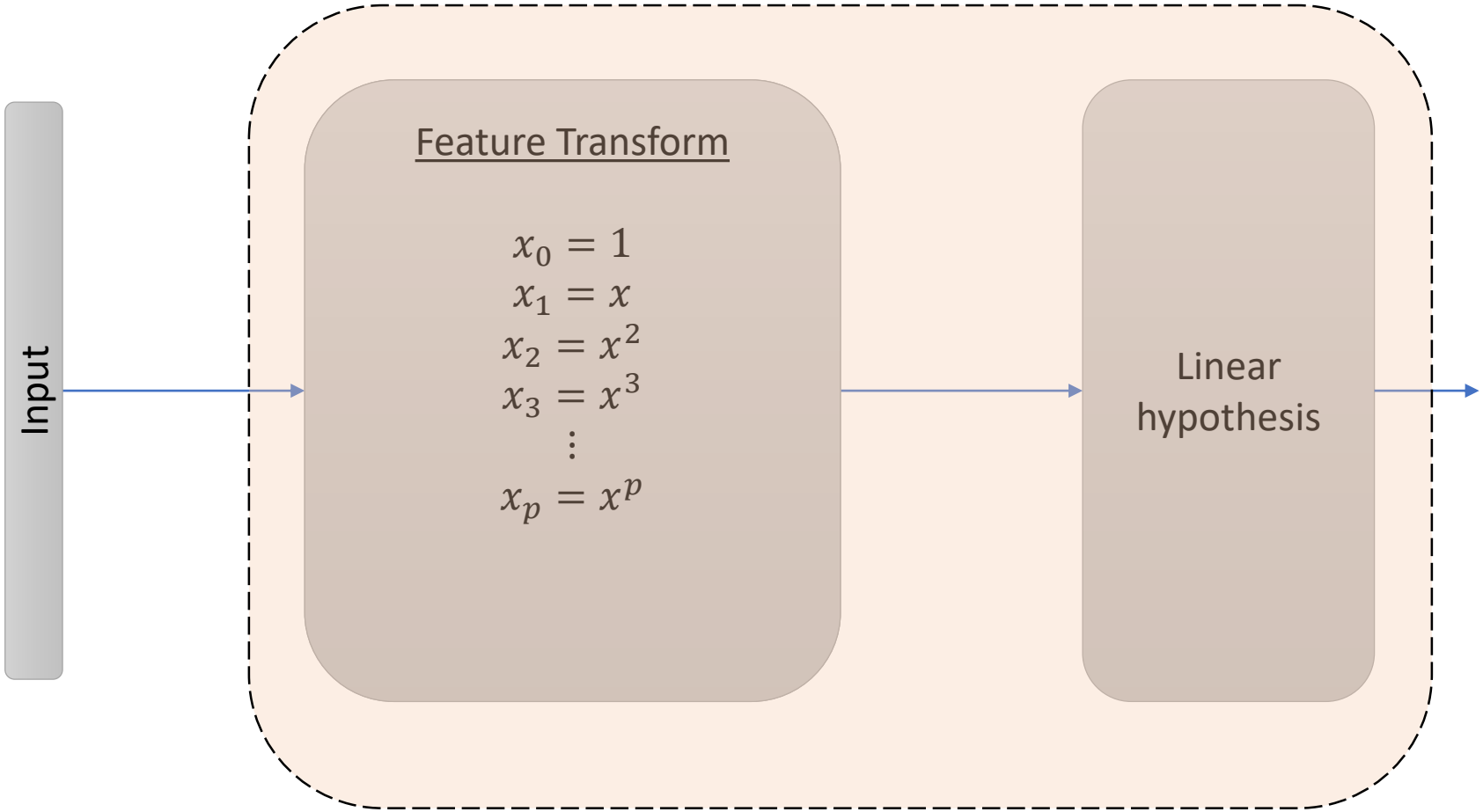
$$z = \sqrt{x^2 + y^2}$$





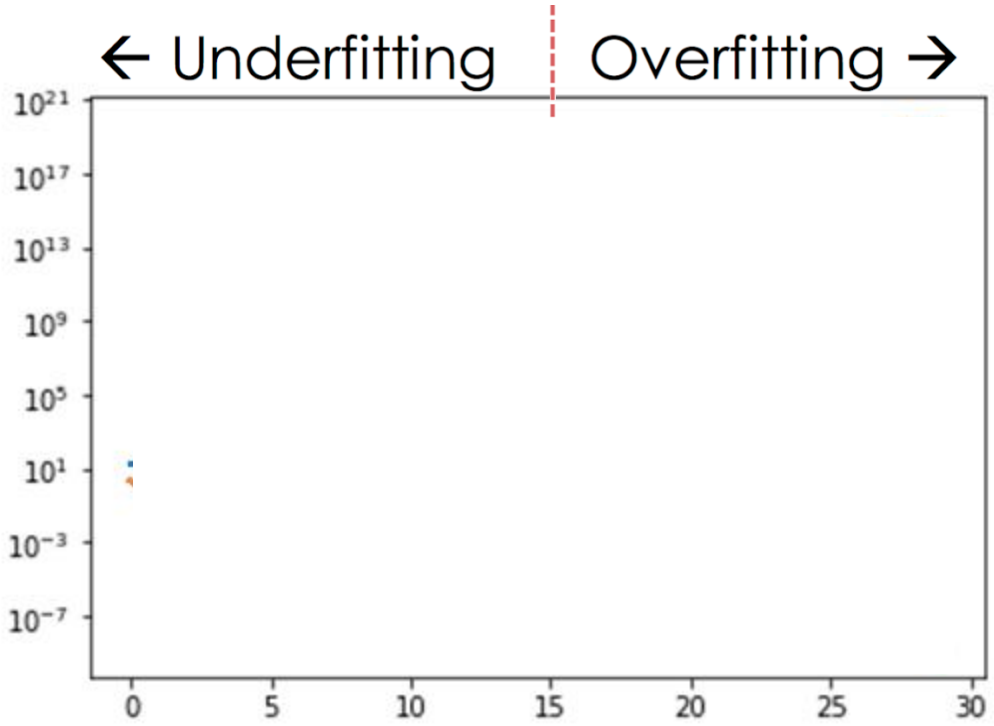


Feature transform



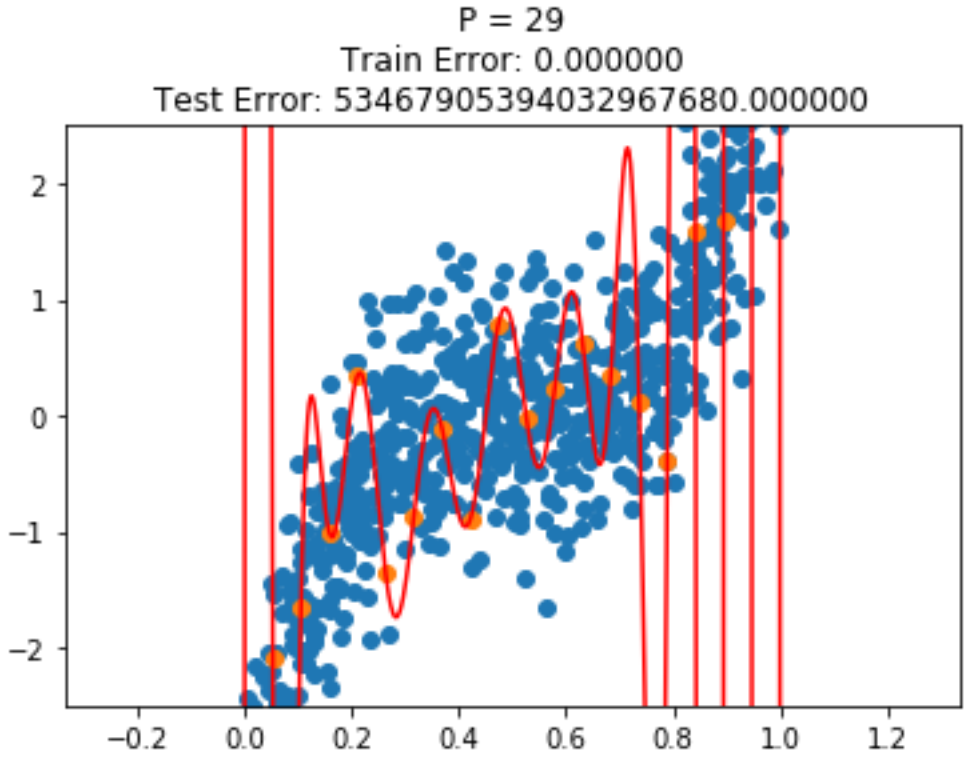
Non-linear hypothesis!
(Polynomial)

Polynomial fitting



Train

Test

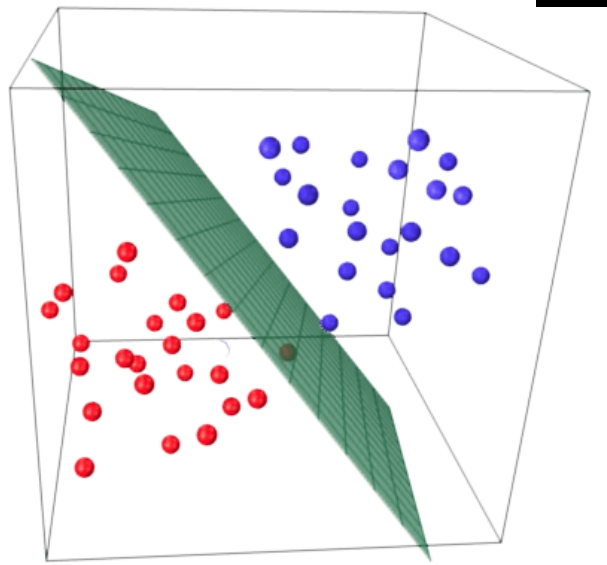


This week's tutorial:

Dana Joffe

Linear classification

+ some PyTorch



Next week's lecture:

(Me Again) # Neural Networks