Optimisation - 5

16 July 2025 20:48

(most Difficult) →

*

Agenda → Introduction

→ (Set up some Rules)

→ (Ask you questions)

→ [Some core concepts] → PCA

→ eigen values eigen vectors

→ implementation

Introduction → Rohit Jindal

→ Senior Data Scientist (Target) Retail ML

→ Ex Mastercard AI Garage → D.S work → Data Drift (ICAIF)

SSGA → D.S work → financial ML

IRCon Internation [1 year] → Excel, VBA

Civil Engineer → D.S

Masters from [Indian Institute of Sc] (IISc BLR)

B.Tech D.elw Technological Univ (DCE)

Univ of Cambridge (Fully Sponsored) → (IP, Sustainability) Data Science

→ Logistics

→ Mode of Comm → (English)

→ Raise a Hand

→ Notes & Tools →

→ Doubts → [end of session] [contact → whats app]

→ Chat

* [Gen AI]

(X) Maths For ML → [work at company]

→ (Core) [BRAIN]

GRADIENT → (derivative in higher dimension)

→ $y = x^2 + 3x$ → $\frac{dy}{dx} = 2x + 3$

→ $y = x_1^2 + 3x_2 + 9x_3$

Price of House = w_1 (no. of Rooms) + w_2 (Sq. ft) + w_3 (Balcony)

(model will learn)

Y

$\frac{\partial \text{Price}}{\partial \text{Rooms}}$ ⇒ All other params are constant

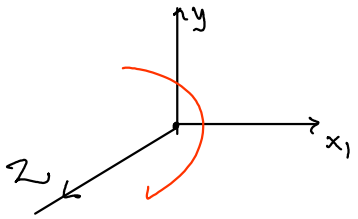
$f = 2x$

$\frac{df}{dx} = (2x)^1$

$\frac{\partial [\text{Experience}]}{\partial [\text{Pass}]}\Big|_{[f, L, P = K]}$

∇ = (grad) [Direction] = $\begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \\ \frac{\partial L}{\partial w_3} \end{bmatrix}$

Loss = $fn(w_1, w_2, w_3) = (3w_1 + 9w_2 + w_1 w_3)$



$\nabla = \begin{bmatrix} \frac{\partial L}{\partial x_1} \\ \end{bmatrix}$

Summary

Loss function = $f(w_1\ w_2 \dots w_n)$ (unknown quantities)

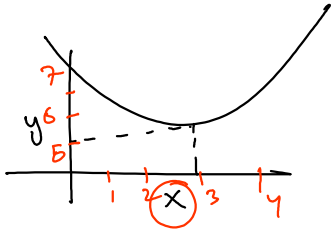
optimise

↓ max | min

optimise loss (Loss fn)

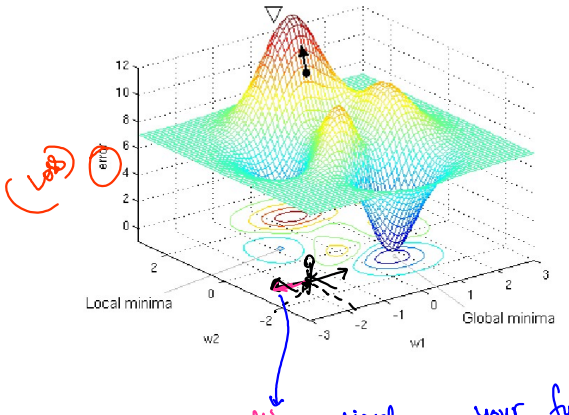
$y = x^2 + c$

(dependent)



dir^n of steepest ascent

gradient : (a) dir^n of steepest ascent
(b) dir^n of steepest descent



(Loss)

$\nabla f = \begin{bmatrix} \\ \end{bmatrix}$

if you move in this direction, your function will increase maximum

$L = f(x,y) = x^2 + y^2$ ← optimise

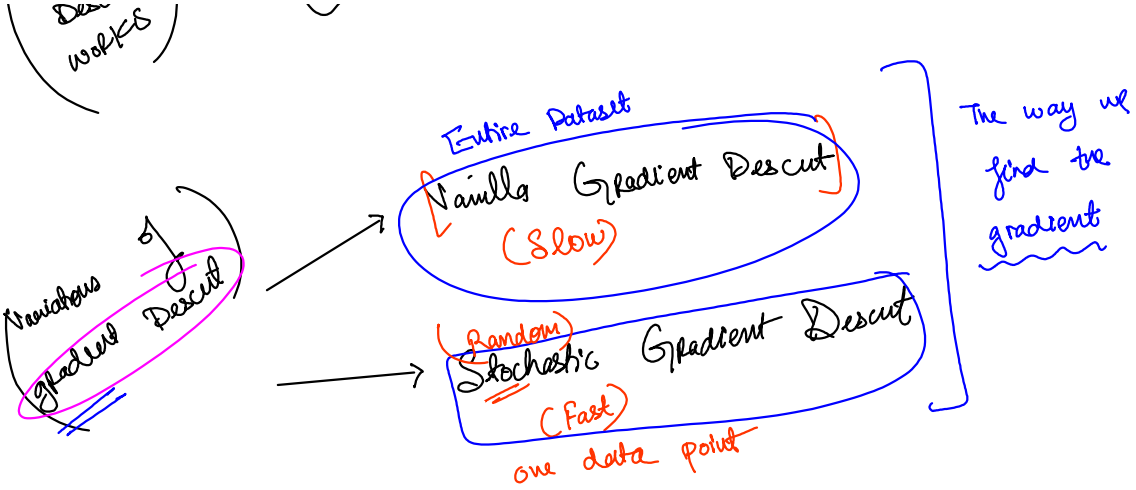
gradient $= \nabla = \begin{bmatrix} \frac{\partial L}{\partial x} \\ \frac{\partial L}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x + 0 \\ 0 + 2y \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$

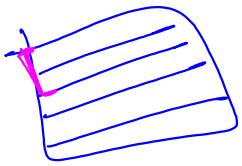
At what point do you want to find gradient?

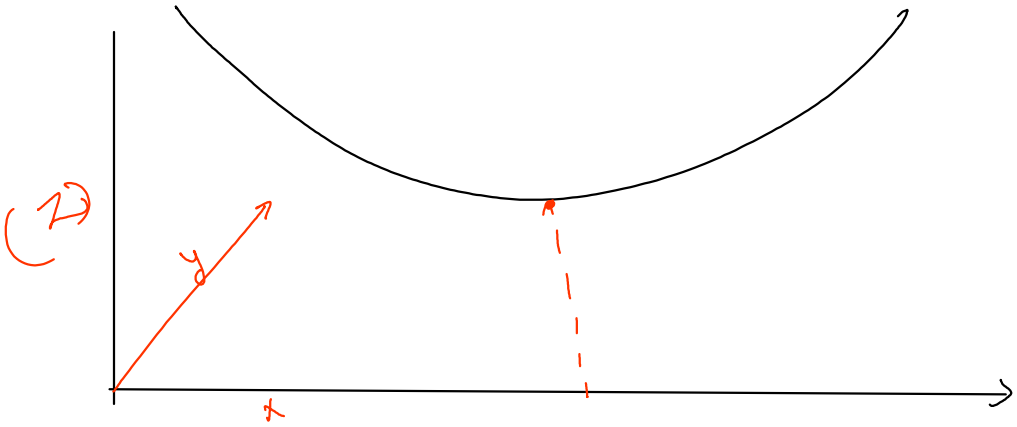
* (Newton's Magic)

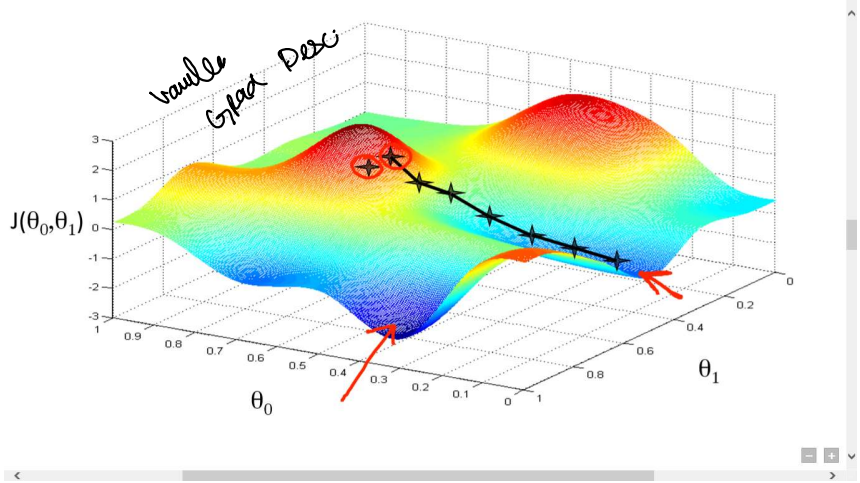
Gradient Descent works

$$\vec{W}_{new} = \vec{W}_{old} - \eta \frac{\partial L}{\partial \vec{W}}$$

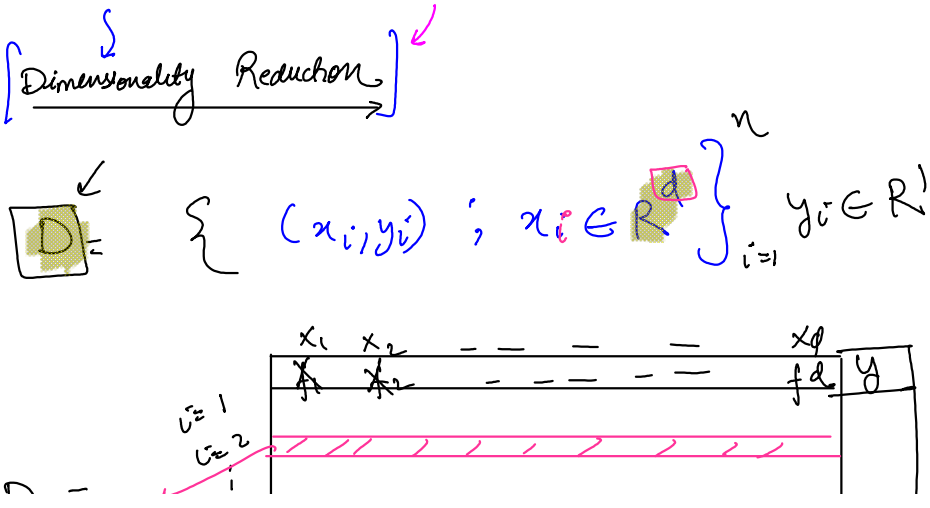

Entire Dataset
Vanilla Gradient Descent
(Slow)
Variations of Gradient Descent
(Random)
Stochastic Gradient Descent
(Fast)
one data point
The way we find the gradient

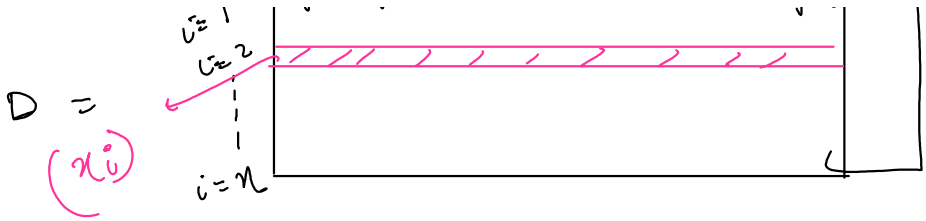



z
y
x


Vanilla Grad Desc
J(θ0,θ1)
θ0
θ1

- **Batch GD:** Smooth, direct path to the minimum, but each step is slow.
- **SGD:** Fast, but the path is noisy and jumps around.





Dimensionality = d

(Dimensionality Reduction) → d → d' such that

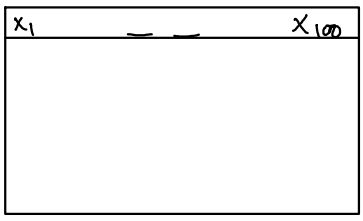
$d' < d$ *

Problem with high Dim

Problems with higher dimensional data.

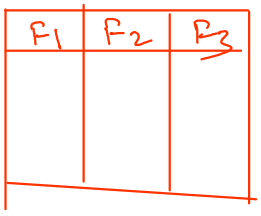
1. Visualisation is tough
2. Training time increases.
3. Computation resources requirement increase.
4. Difficult to work math around such data set.

d = 100 dim



Dimensionality Red^n

→ PCA

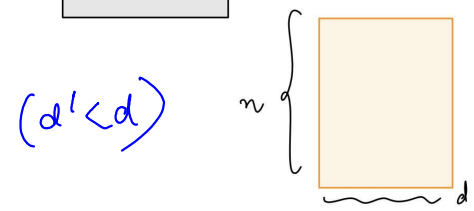


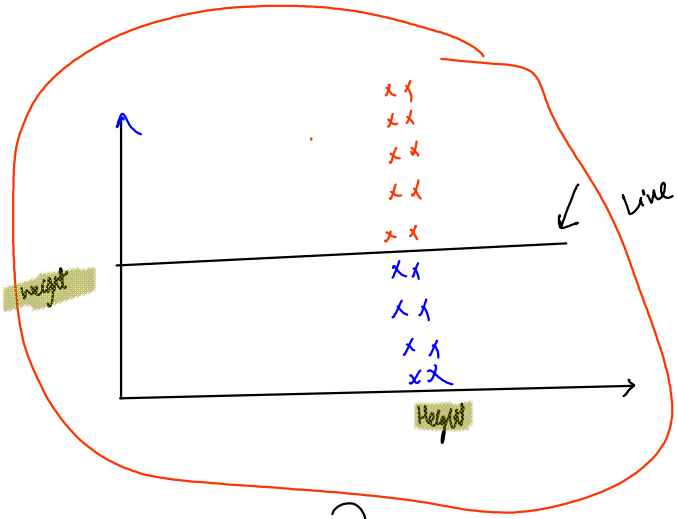
Dimensionality Reduction algorithms help us to deal with these problems

Eg: Principal Component Analysis

$\mathcal{D} = \{\bar{x}_i \in \mathbb{R}^d\}_{i=1}^{n}$ → i/p → PCA → o/p → $\mathcal{D}' = \{(\bar{x}_i \in \mathbb{R}^{d'})\}_{i=1}^{n}$

1. $d = d'$ ✗
2. $d > d'$ ✓ $(d' < d)$
3. $d < d'$ ✗





wt	ht	Diabetes (Yes/No)
—	—	
—	—	
—	—	

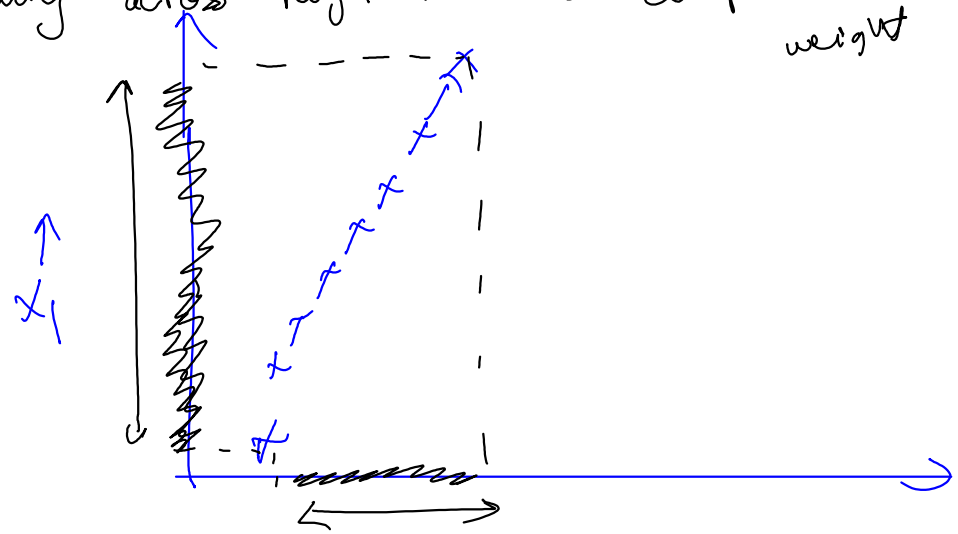
I weight

II Height

Q Are both wt & ht important to classify a person as diabetic or not?

* But why weight is Imp?

Variability across height is less compared Variability across weight



1 - 2

d = 1

x_2

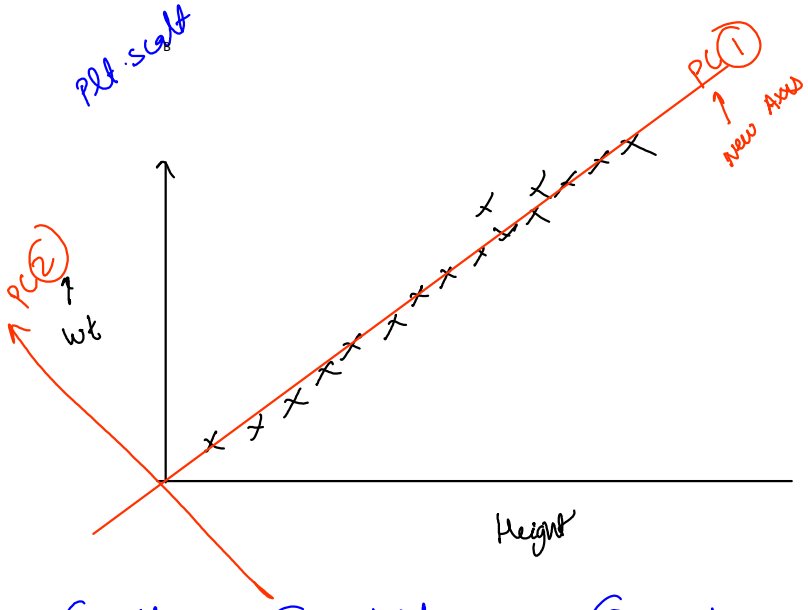
$d = 2$

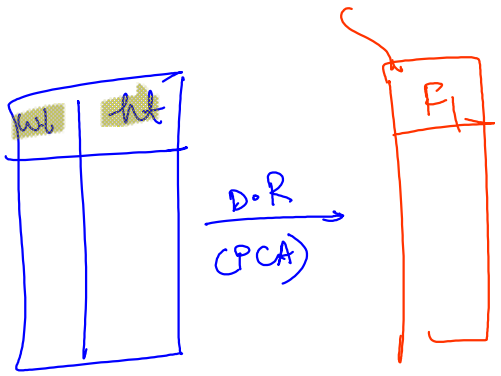
$d = 1$

x_1 (wt)	x_2 (ht)	Y (diabetes)
—	—	
—	—	
—	—	

dimensionality Reduction →

F_1	Y (diabetic)



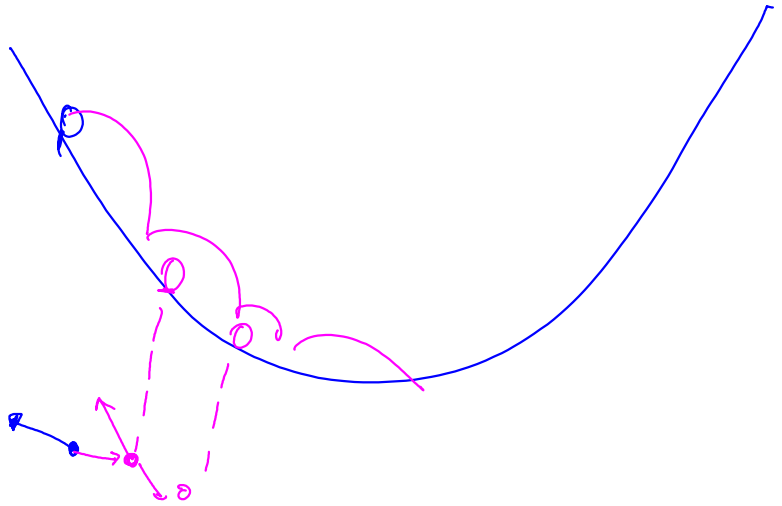


① Ht ② Wt ⑤ Ht + Wt

PCA → it finds those new set of directions, where when you project your original data, the variance of the projected data is maximised

derive this

10d $\xrightarrow[(k=4)]{\text{PCA}}$ PC_1 | PC_2 | PC_3 | PC_4



Loss fn

PCA → Eigen values
PCA → Eigen vector

100 dim → 90 dim → 80

(some info) again

Lucky

100 — 40

(inf loss) is v. less

PCA → Quant | Stat heavy Roles

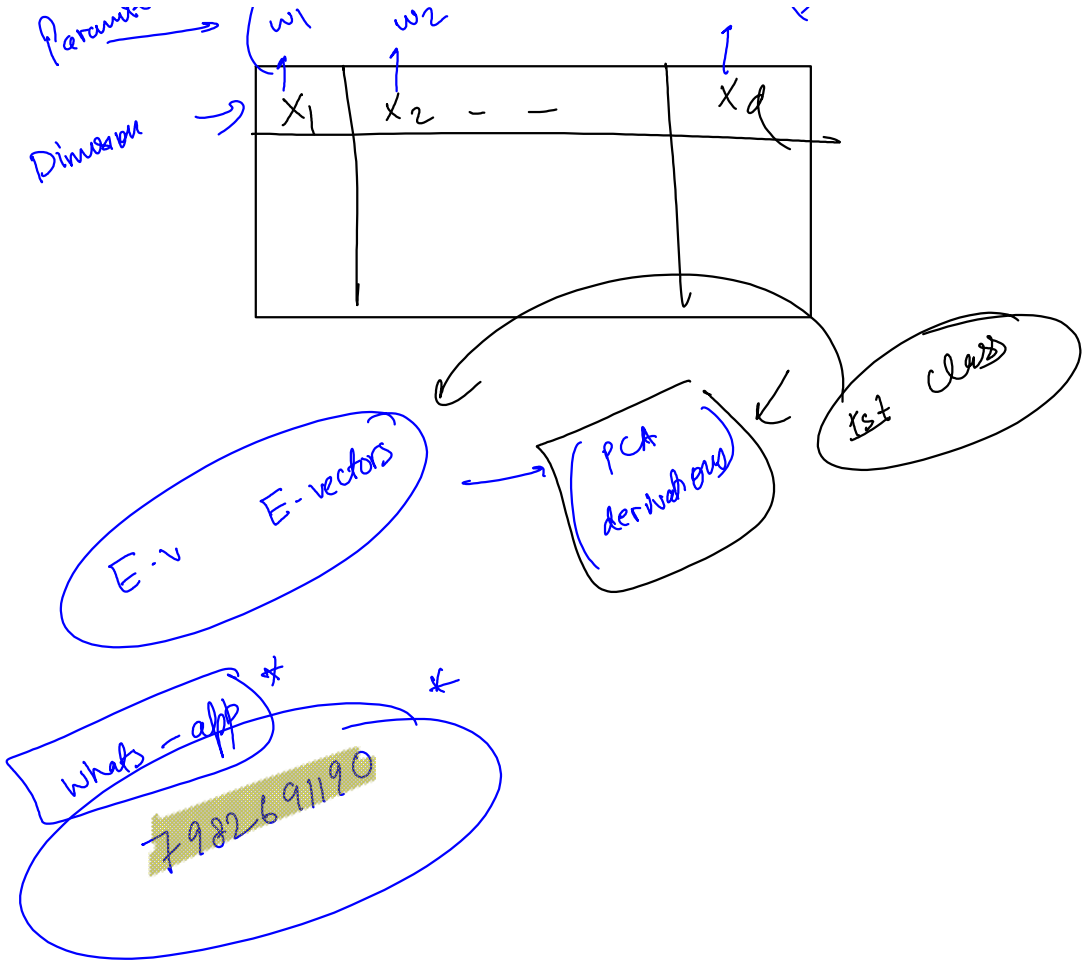
Parameter → (w_1 w_2 w_d)

Parameter → w_1 w_2

Dimension → x_1 x_2 - - x_d

1st class

E.V E-vectors → PCA derivations

whats-app *

7982691190 *