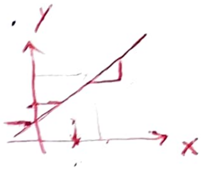
→ Numerical optimization   Session 1   12/1/2025

- Skills (program, based on numerical optimization)
  - Building training neural networks.
  - supervised machine learning algorithms.
  - Reinforcement learning.
  - Choosing Algorithms, measuring its relevance



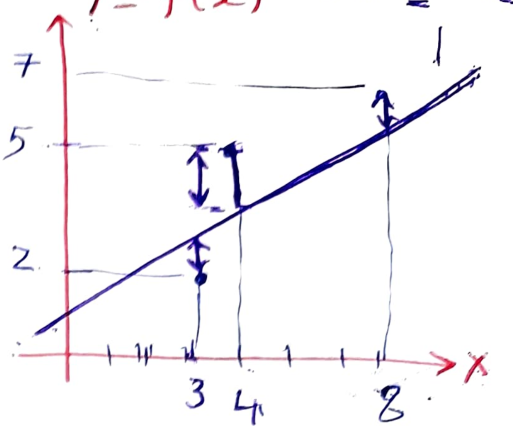
- Outcomes (Numerical optimization);
  - understanding "Convexity"
  - " optimization
  - " Gradient Descent algorithm
    variants of
    ? ?
    $\boxed{\nabla}$
  - momentum-based Algorithms.
  - Coding & implementation of such algorithms.

why numerical optimatization?

→ example of Analytic solution of "least-squares"

$$\min_{(a,b)} \Big(\sum_i (y_i - \tilde{y}_i)^2 \Big)$$

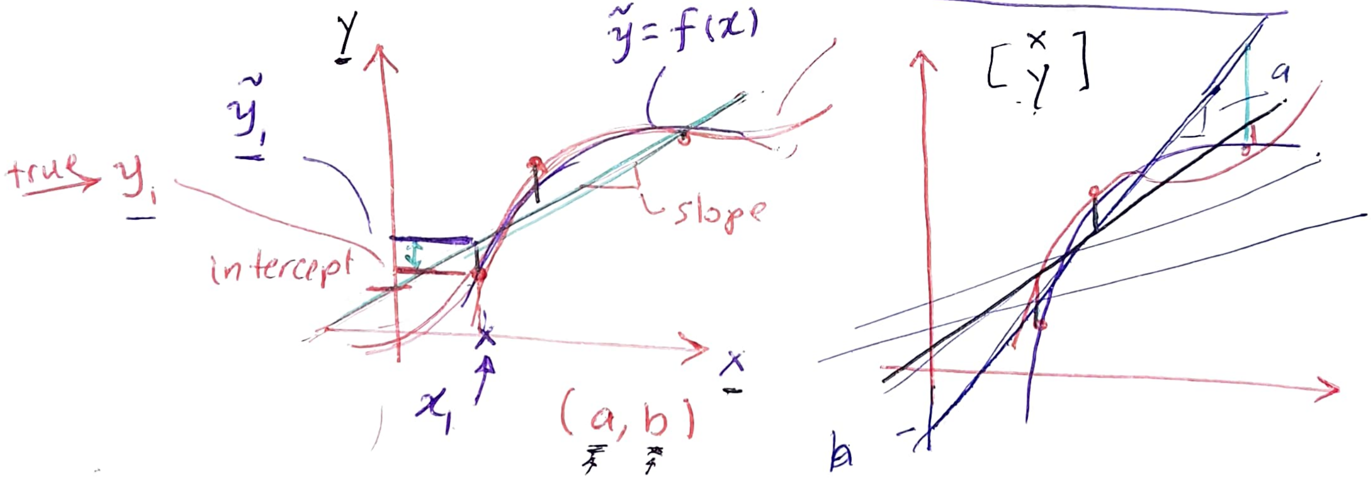
real   predicted



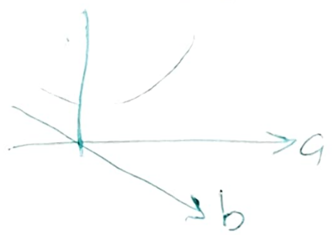
→ for this problem ~~a~~ analytic solution exist.

"pseudo-inverse" $\vec{\theta} = X^T (X X^T)^{-1} y$

a, b



→ RSS : sum of squared Residuals.



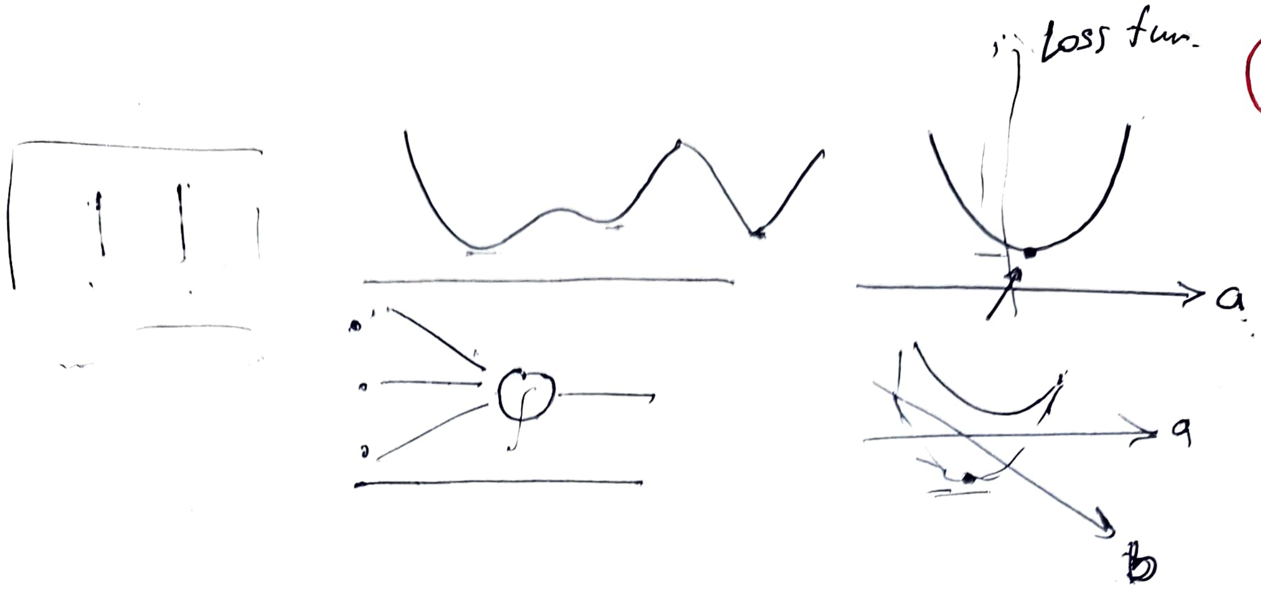
$$\sum_i (y_i - \tilde{y}_i)^2$$

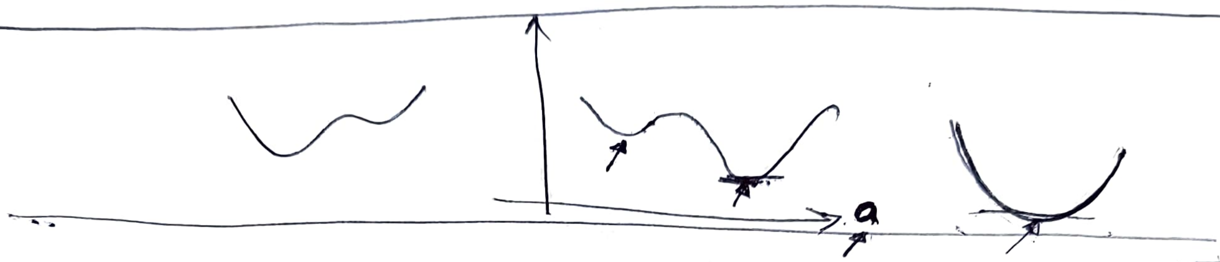
$f(x) = a x_i + b$

$$RSS = \sum_i (y_i - (a x_i + b))^2$$

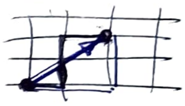
MSE

supervised learning (Regression)





Cost function (loss function)

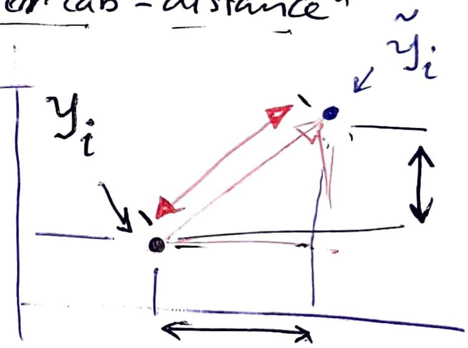


L_1 Norm $\Rightarrow$ MAE $\rightarrow$ Manhatten distance
"taxi ~~or~~ cab - distance"

L_2 Norm $\Rightarrow$ MSE

$\Downarrow$

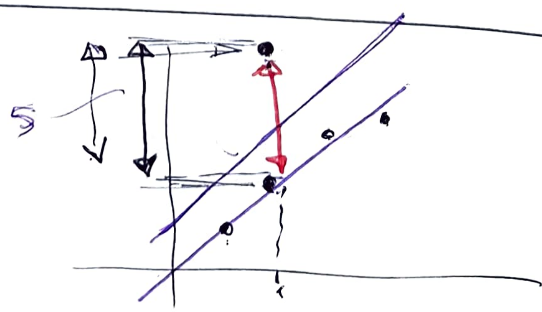
Eculidian distance $\sqrt{\| \;-\; \|^2}$



$$MAE = |\vec{y}_i - \vec{y}|_1$$

MAE : robust to outliers.

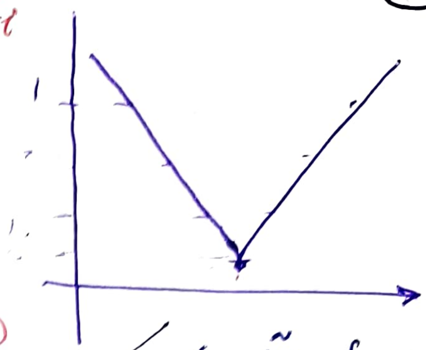
MSE.

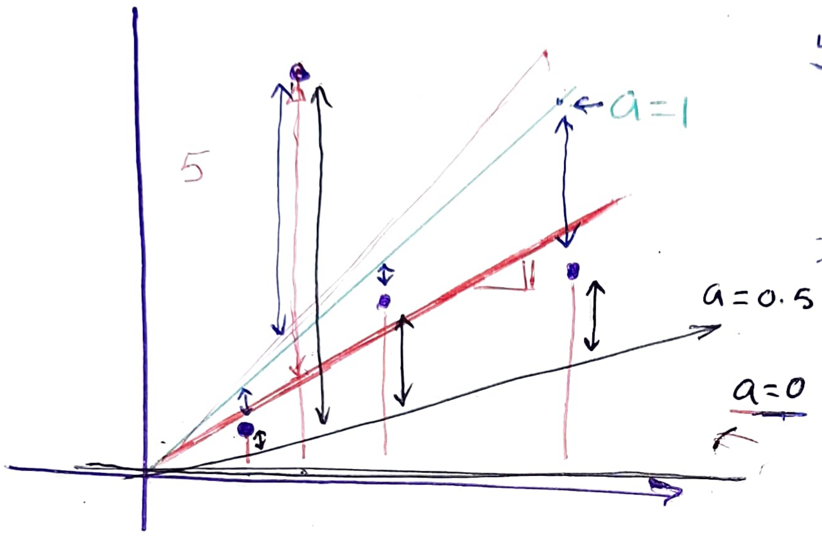


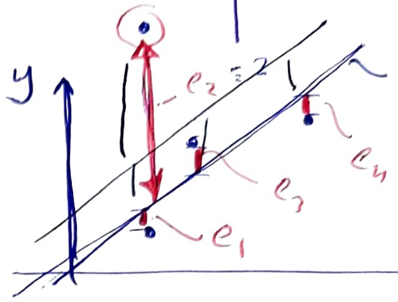
$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} \left| \vec{y}_i - \underbrace{\tilde{\vec{y}}_i}_{f(a)} \right| \qquad \tilde{y}_i \to a\,x_i$$

n ← data points

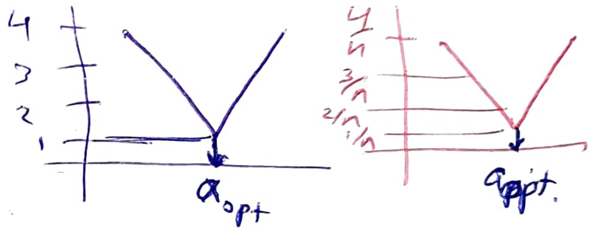
$\tilde{y} = a\,x$

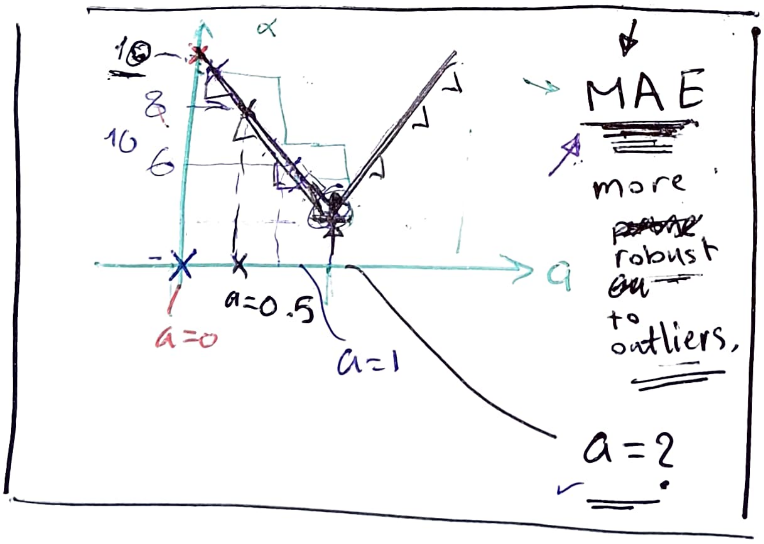


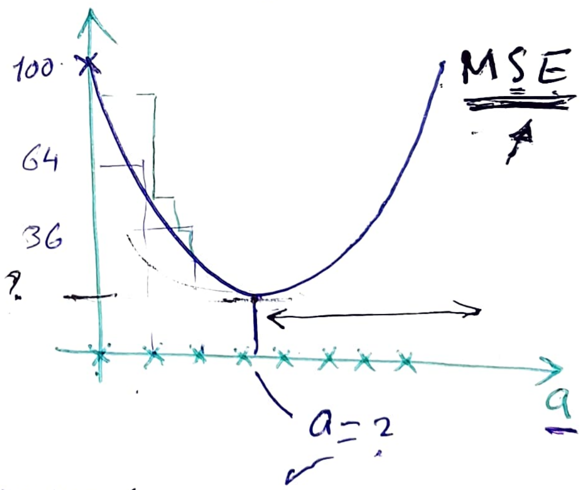




$\tilde{y} = f(x) = a\,x$





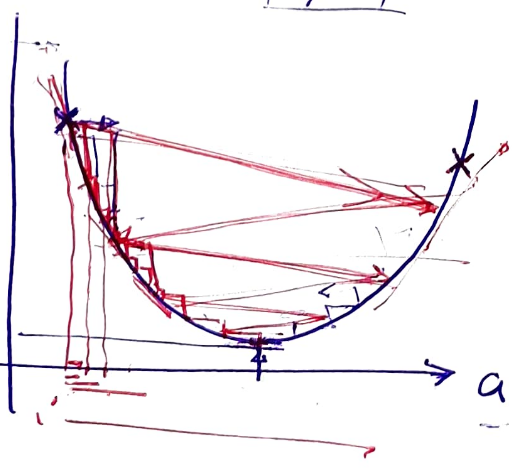


~~MSE =~~

$$\mathrm{RSS} = \mathrm{SSE} = \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

learning rate ~ α

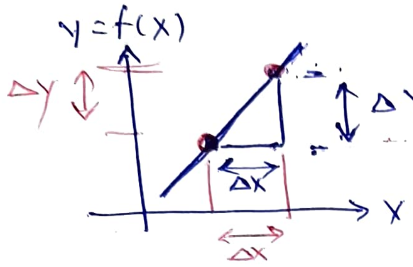


derivative of a function

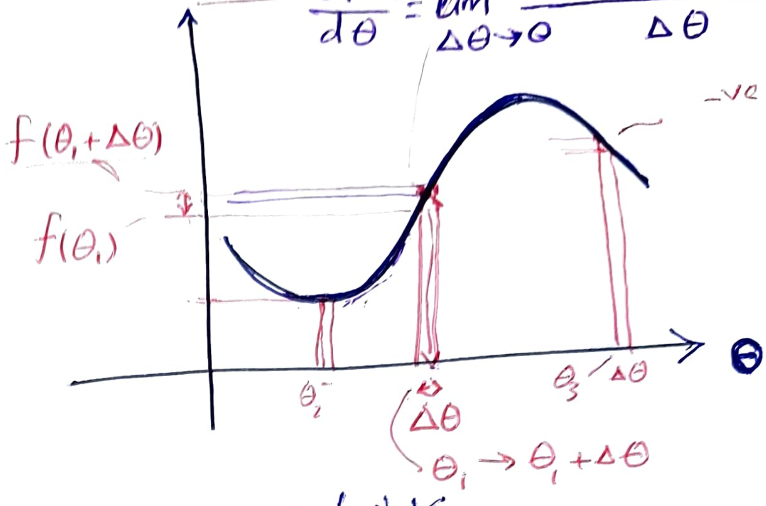
$$\frac{d\,f(\theta)}{d\theta} = \underline{\qquad\qquad}$$

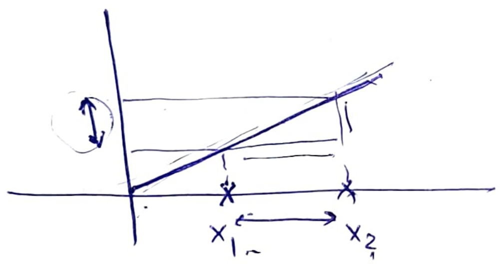
$$\frac{d f(\theta)}{d\theta} = \lim_{\Delta\theta \to 0} \frac{f(\theta+\Delta\theta)-f(\theta)}{\Delta\theta}$$

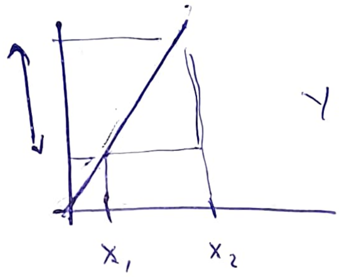
$$\frac{d\,f(\theta)}{d\,\theta}$$

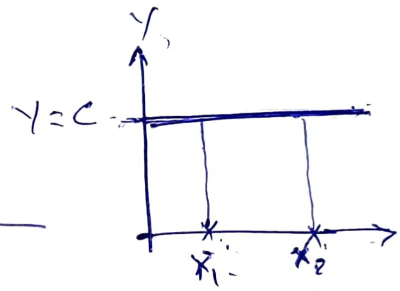


$$\text{slope} = \frac{\Delta Y}{\Delta X} = \frac{dY}{dX}$$







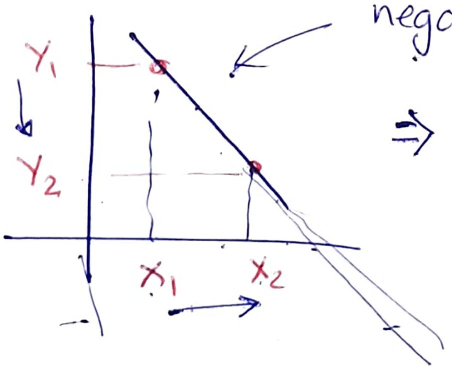


rate of change ↓ | rate of change ↑ | rate of change

$$\frac{\Delta Y}{\Delta X} = \downarrow \qquad \frac{\Delta Y}{\Delta X} = \uparrow \qquad \frac{\Delta Y}{\Delta X} = 0$$

---

negative slope

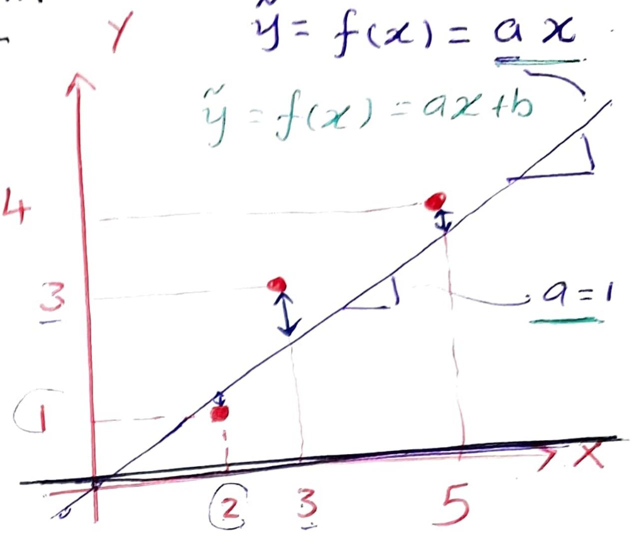


$$\Rightarrow \frac{\Delta Y}{\Delta X} = \frac{Y_2 - Y_1}{X_2 - X_1} = \frac{-ve}{+ve}$$

Gradient Descent Algorithm

$$\tilde{y} = f(x) = ax$$

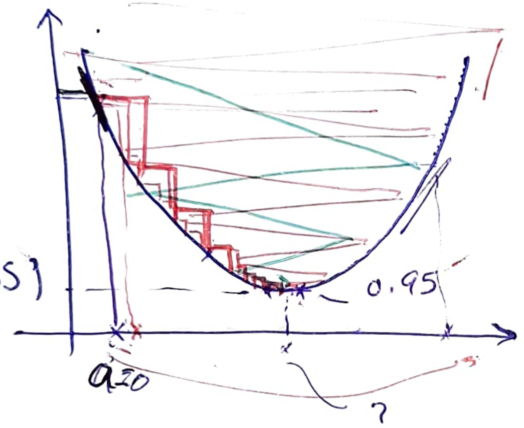
$$\tilde{y} = f(x) = ax + b$$



$$RSS = \sum_i (y_i - \tilde{y}_i)^2$$

cost fun.

$$= \frac{1}{2} \sum_i (y_i - a x_i)^2$$



gradient.

$$\frac{d}{a}(\text{cost function})$$

$$= 2 \sum_{i=1}^{3} (y_i - a x_i)(-x_i)$$

$$\text{gradient} = 2\big((1-2a)(-2) + (3-3a)(-3) + (4-5a)(-5)\big)$$

$$-2 \qquad -9 \qquad -20$$

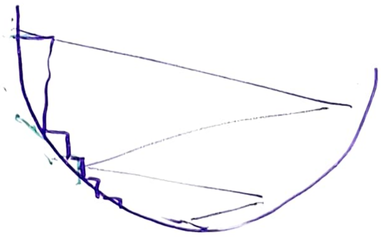
randomly let "a = 0" ⇒ gradient = −62

update estimate of "a"

$$a_{new} = a_{old} - \boxed{\alpha \times \text{gradient}}$$

learning rate

$$= 0 - 0.1 \times -62$$



Iteration 2 $a_{new} = 6.2$

gradient =

.9
0.95
a.96

$$\alpha = 0.001$$

$$a_{new} = 0 - \left(-\frac{62}{1000}\right)$$

$$= 0 + 0.062 = 0.062$$

$$\alpha = 1$$

$$a_{new} = 0 - (-62 \times 1)$$

$$= 62$$