Lecture 6. GD.

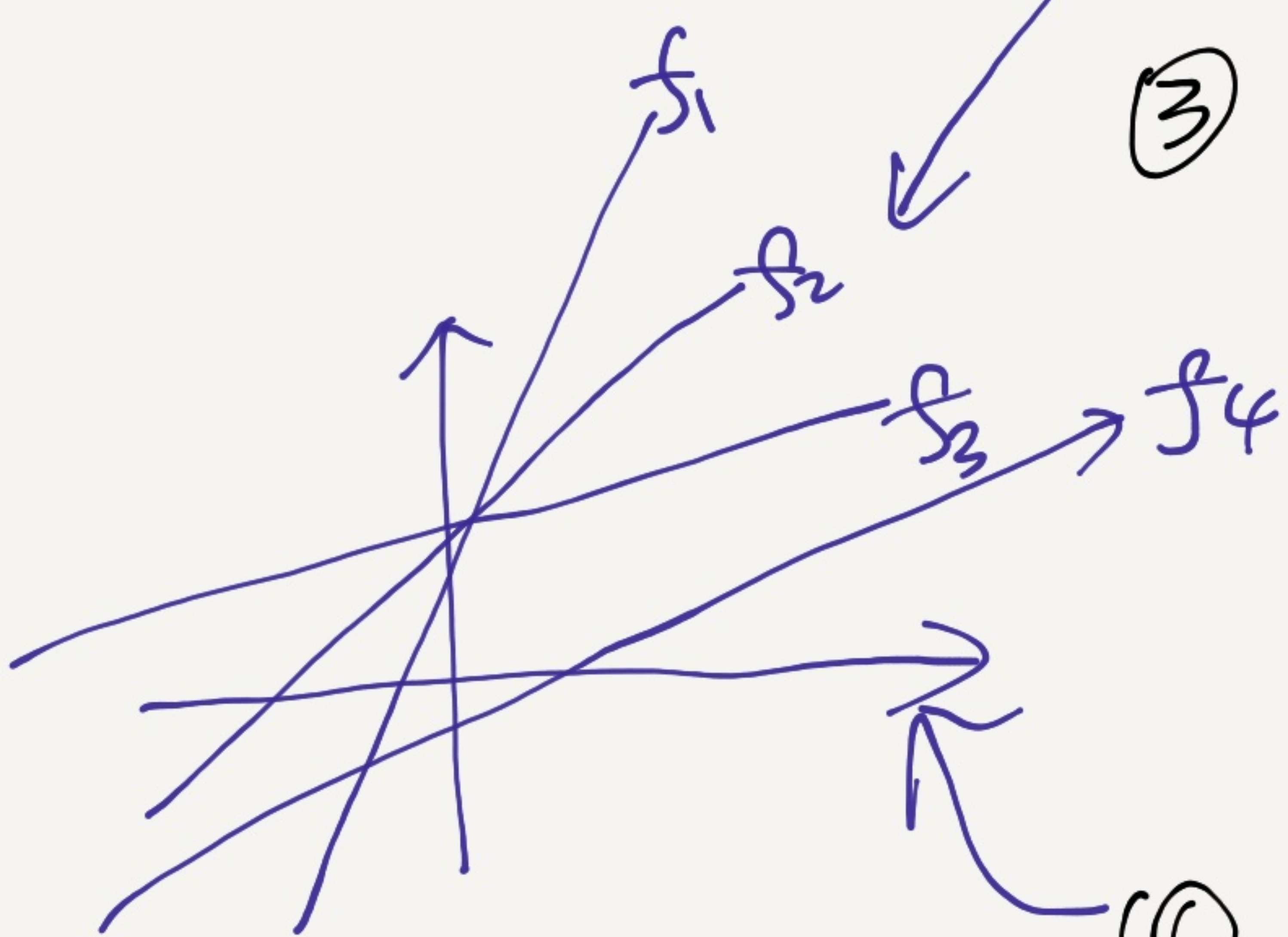
1. Linear Regression

Four component: ① $\{(x_i, y_i)\}_{i=1}^{n}$

generalized feature vector $x_i = \begin{pmatrix} x_{i0}=1 \\ x_{i1} \\ \vdots \\ x_{im} \end{pmatrix}_{(m+1)\times 1}$ target value. $y_i \in R$

② model: $f(x_i) = w^T \cdot x_i$ model parameters $w = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{pmatrix}_{(m+1)\times 1}$

$= w_0 \cdot x_{i0} + w_1 \cdot x_{i1} + \cdots + w_m \cdot x_{im}$



③ loss: $L(w) = \frac{1}{2}\sum_{i=1}^{n}(\underbrace{\hat{y}_i}_{\text{prediction } f(x)} - y_i)^2$

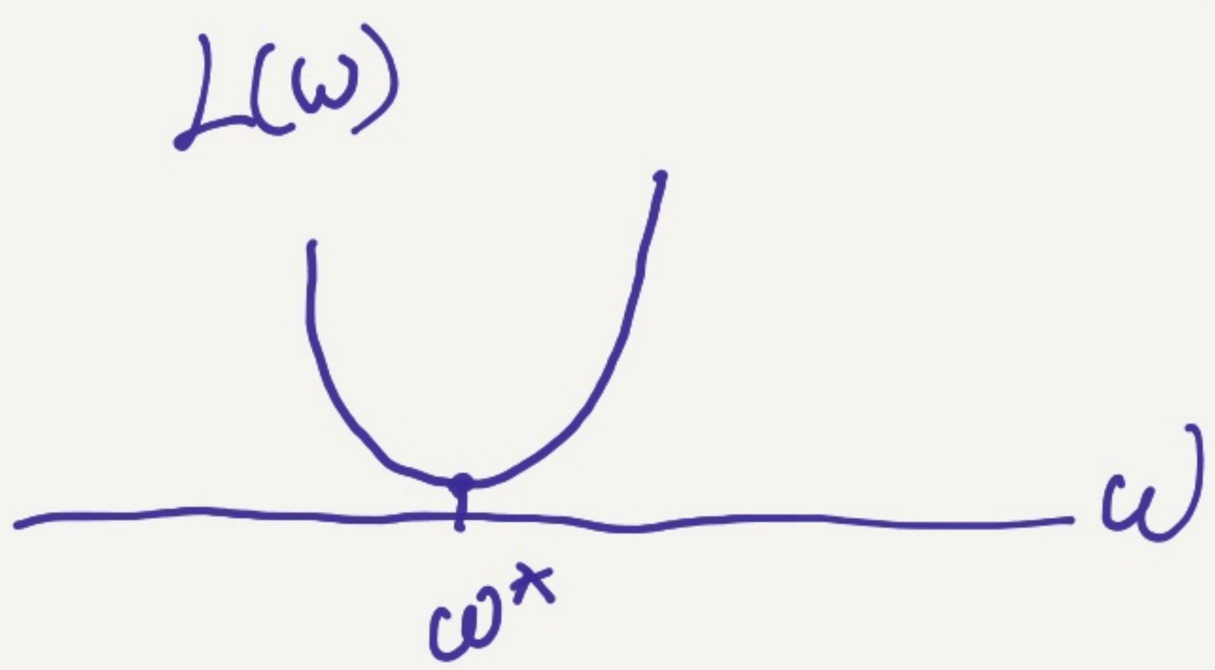
$= \frac{1}{2}\sum_{i=1}^{n}(w^T x_i - y_i)^2$

④ Optimization: Find w^* (best model parameters)

$$w^* = \arg\underset{w}{\text{minimize}}\, L(w) = \frac{1}{2}\sum_{i=1}^{n}(w^T x_i - y_i)^2$$

$\sum_{i=1}^{n} z_i^2 = z^T \cdot z \quad z = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}$

$= \frac{1}{2}\begin{pmatrix} w^T x_1 - y_1 \\ w^T x_2 - y_2 \\ \vdots \\ w^T x_n - y_n \end{pmatrix}^T (w^T x_1 - y_1, \cdots, w^T x_n - y_n)$



$\nabla L(w)|_{w=w^*} = 0$ (m+1, equations)

$b = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}_{n\times 1}$ $A = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}_{n\times m}$

$= \frac{1}{2}(Aw - b)^T(Aw - b)$

$$\nabla_w L(w) = \frac{1}{2} \cdot 2 \cdot A^T(Aw - b) = 0$$

$$L(w) = \frac{1}{2}(Aw-b)^T(Aw-b)$$

$$A^TAw = A^Tb$$

inverse

$$w^* = \underline{(A^TA)^{-1}A^Tb}$$ (Analytic solution)

challenges:

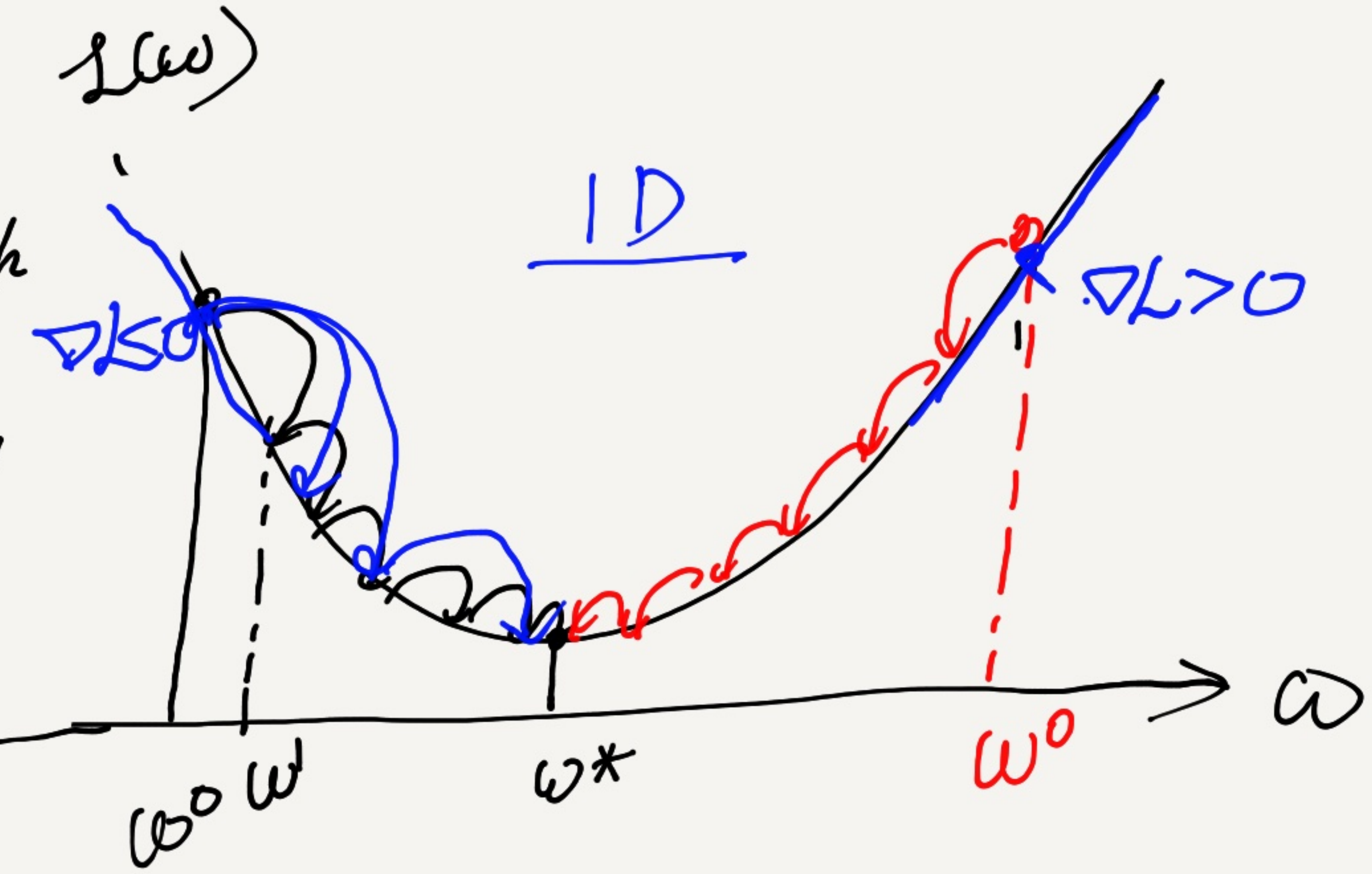
1) A could be very large. we may do not have enough resources to load it and perform the calculation.

$$A^T \cdot A \ . \quad (A^T \cdot A)^{-1}$$

2) $(A^TA)^{-1}$ may not exist.

2. Gradient-based approaches.

1). main idea: start from a random location, e.g., w^0, update w^0 to approach w^*. Generate a sequence of model parameters.



$$w^0 \to w^1 \to w^2 \to \cdots \to w^k = w^*$$

$$\mathcal{L}(w^0) > \mathcal{L}(w^1) > \mathcal{L}(w^2) \cdots > \mathcal{L}(w^k)$$

To design this optimization, we need to determine

① moving direction.

② step size

③ stopping condition. (k)

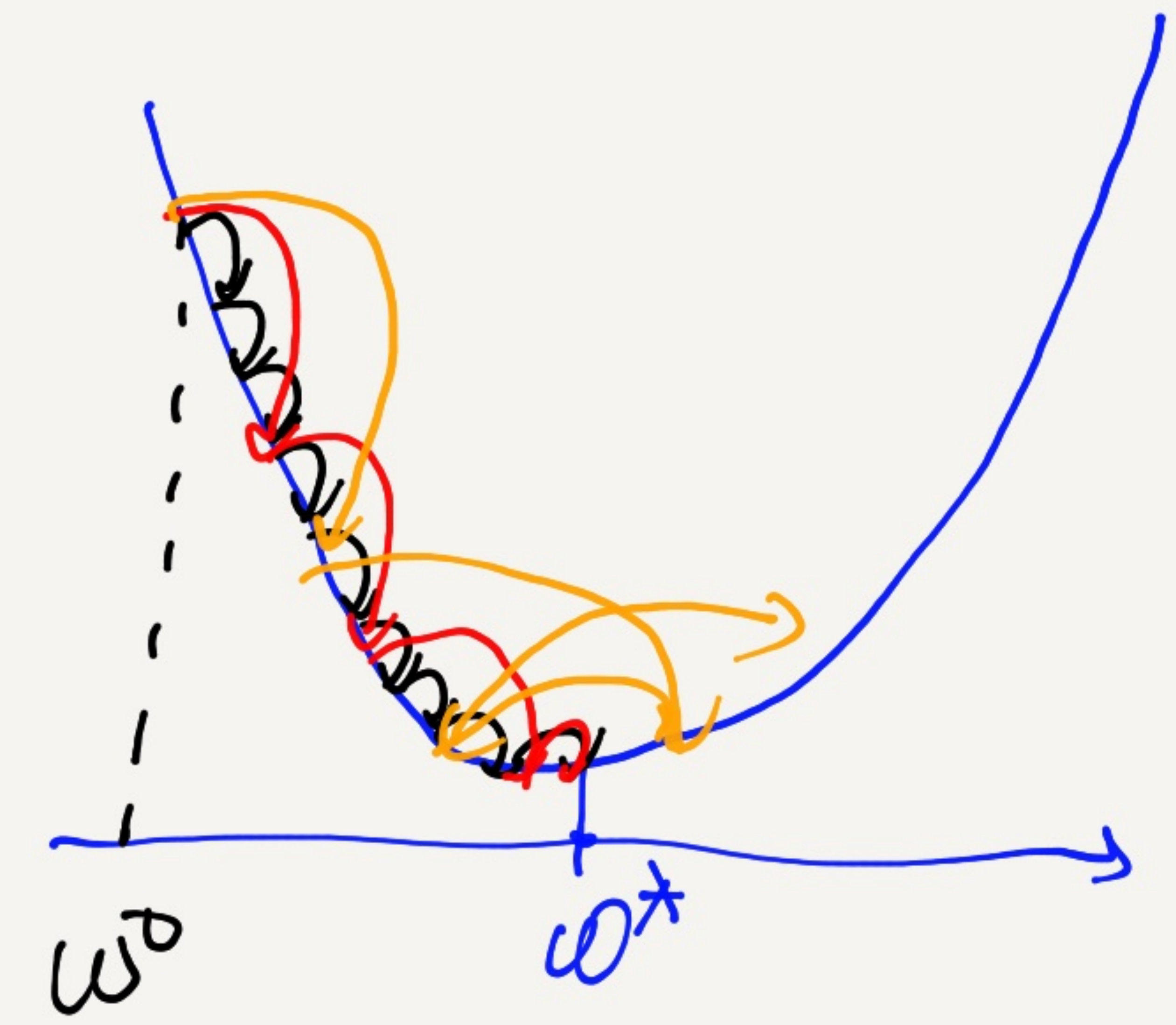
2) Moving direction (use the sign of the gradient to determine the trend of the $\mathcal{L}(w)$)

	direction
∇L is positive >0,	$-\nabla L$ $(-)$
∇L is <0,	$-\nabla L$ $(+)$

N.D.: $w = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{pmatrix}$ $\nabla L = \begin{pmatrix} \frac{\partial L}{\partial w_0} \\ \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_m} \end{pmatrix} \longrightarrow$ moving direction

$-\nabla L$ (opposite direction)

3) Step size. ε / learning rate



if step size is small, it takes longer time to reach w^*.

Too large step size may not converge to w^*.

① define ε as a small positive value.

10^{-3}, 10^{-4}, 10^{-5}

② line search. $\varepsilon^* = \arg\min_{\varepsilon} L(\overbrace{w + \varepsilon \cdot (-\nabla L)}^{\text{new } w})$

③ Dynamic ε: $\varepsilon_1 > \varepsilon_2 > \varepsilon_3 \cdots > \varepsilon_k$

4) stopping conditions

① $\|\nabla L(\omega)\|_2 < \delta$ (small positive value, e.g. 10^{-5}) $\|\nabla L(\omega)\|_2 = 0$

② define the max number of iterations. $k = 1000$.

5) GD Algorithm.

Input: $\{(x_i, y_i)\}_{i=1}^{n}$, ε, $\delta = 10^{-5}$, max_iter.

Output: ω^*

1. Generate ω^0 randomly: $\omega = \omega^0$, $i = 0$

2. while $\|\nabla L(\omega)\|_2 > \delta$ and $i <$ max_iter

   ① calculate the gradient. $\hat{g} = \nabla L(\omega) = A^T(A\omega - b)$

   ② update ω: $\omega = \omega + \varepsilon \cdot (-\hat{g})$

   $i = i + 1$

3. $\omega^* = \omega$. return ω^*