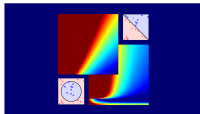


# Machine Learning Foundations

## (機器學習基石)



### Lecture 2: Learning to Answer Yes/No, Extended

Hsuan-Tien Lin (林軒田)

htlin@csie.ntu.edu.tw

Department of Computer Science  
& Information Engineering

National Taiwan University  
(國立台灣大學資訊工程系)



Handling  $\text{sign}(\cdot) = 0$ 

## Perceptron Learning Algorithm

start from some  $\mathbf{w}_0$  (say,  $\mathbf{0}$ ), and 'correct' its mistakes on  $\mathcal{D}$

When  $\mathbf{w}_0 = \mathbf{0}$ , technically  $\text{sign}(\mathbf{w}_0^T \mathbf{x}_{n(0)}) = 0$ , shall we update?

- convention -1:  $\text{sign}(0) = -1$  (update if  $y_{n(0)} = +1$ )
- convention +1:  $\text{sign}(0) = +1$  (update if  $y_{n(0)} = +1$ )
- convention 0:  $\text{sign}(0) = 0$  (always update)
- convention r:  $\text{sign}(0) = \text{random flip}$  (50% chance of update)

—usually does not matter much, **as long as  $\mathbf{w}_1$  often becomes non-zero**

$\mathbf{w}_t^T \mathbf{x}_{n(t)} = 0$  **rarely happens in practice**

Updating  $w_0$ 

## Perceptron Learning Algorithm

For  $t = 0, 1, \dots$

- ① find a **mistake** of  $\mathbf{w}_t$  called  $(\mathbf{x}_{n(t)}, y_{n(t)})$ :  $\text{sign}(\mathbf{w}_t^T \mathbf{x}_{n(t)}) \neq y_{n(t)}$
- ② (try to) correct the mistake by

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)} \mathbf{x}_{n(t)}, \text{ i.e.,}$$

$$\begin{bmatrix} w_{t+1,0} \\ w_{t+1,1} \\ \dots \\ w_{t+1,d} \end{bmatrix} = \begin{bmatrix} w_{t,0} \\ w_{t,1} \\ \dots \\ w_{t,d} \end{bmatrix} + y_{n(t)} \begin{bmatrix} x_0 (= \text{what?}) \\ x_{n(t),1} \\ \dots \\ x_{n(t),d} \end{bmatrix}$$

... until **no more mistakes**

return **last  $\mathbf{w}$**  (called  $\mathbf{w}_{\text{PLA}}$ ) as  $g$

each update changes  $w_{t,0}$  by  $y_{n(t)}$

# PLA Mistake Bound

inner product grows **fast**

$$\mathbf{w}_f^T \mathbf{w}_{t+1} \geq \mathbf{w}_f^T \mathbf{w}_t + \underbrace{\min_n y_n \mathbf{w}_f^T \mathbf{x}_n}_{\rho}$$

length<sup>2</sup> grows **slowly**

$$\|\mathbf{w}_{t+1}\|^2 \leq \|\mathbf{w}_t\|^2 + \underbrace{\max_n \|\mathbf{x}_n\|^2}_{R^2}$$

Magic Chain!

$$\mathbf{w}_f^T \mathbf{w}_1 \geq \mathbf{w}_f^T \mathbf{w}_0 + \rho$$

$$\mathbf{w}_f^T \mathbf{w}_2 \geq \mathbf{w}_f^T \mathbf{w}_1 + \rho$$

$$\mathbf{w}_f^T \mathbf{w}_3 \geq \mathbf{w}_f^T \mathbf{w}_2 + \rho$$

...

$$\mathbf{w}_f^T \mathbf{w}_T \geq \mathbf{w}_f^T \mathbf{w}_{T-1} + \rho$$

Magic Chain!

$$\|\mathbf{w}_1\|^2 \leq \|\mathbf{w}_0\|^2 + R^2$$

$$\|\mathbf{w}_2\|^2 \leq \|\mathbf{w}_1\|^2 + R^2$$

$$\|\mathbf{w}_3\|^2 \leq \|\mathbf{w}_2\|^2 + R^2$$

...

$$\|\mathbf{w}_T\|^2 \leq \|\mathbf{w}_{T-1}\|^2 + R^2$$

start from  $\mathbf{w}_0 = \mathbf{0}$ , after  $T$  mistake corrections,

$$1 \geq \frac{\mathbf{w}_f^T}{\|\mathbf{w}_f\|} \frac{\mathbf{w}_T}{\|\mathbf{w}_T\|} \geq \frac{T\rho}{1\sqrt{TR}} \implies T \leq \left(\frac{R}{\rho}\right)^2$$

# Pocket and Modern Generative AI

## Pocket

generate candidates

leverage the original PLA  $\mathbf{w}_t$ 's

keep the better

best one copied to pocket

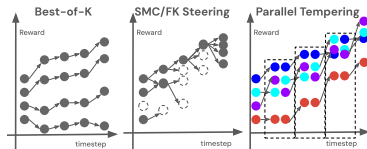
## Aligning Generative AI

generate candidates

parallism, randomness, mutation

keep the better

randomized/deterministic survival



(Lu, Lin, Wang, Workshop on Generative and Agentic AI for Biology @ ICML 2026)

generate-and-keep-better  
popular in **generative AI age**