LDA discussions:

① As a dimensionality reduction algorithm,

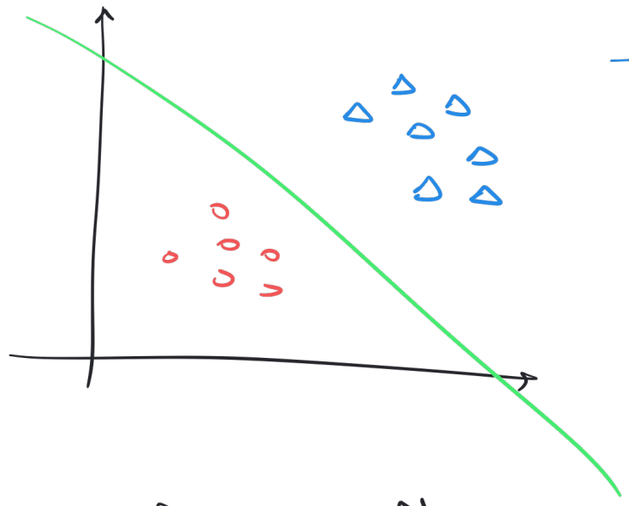
LDA projects up to

$$\min(C-1, D-1)$$

where $C \equiv$ # classes

$D \equiv$ dimensionality

② LDA assumes each class is Gaussian-distributed.



Least Squares Regression

$y = w^T x + w_0$

$y = w^T x + w_0$

DATASET : $\{(x_i, t_i)\}_{i=1}^{N}$, $x_i \in \mathbb{R}^D$, $t_i \in \mathbb{N}$

MAPPER : $y = w^T x + w_0$

OBJECTIVE FCE : $J(w, w_0) = \sum_{i=1}^{N} (t_i - y_i)^2 = \sum_{i=1}^{N} \varepsilon_i^2$

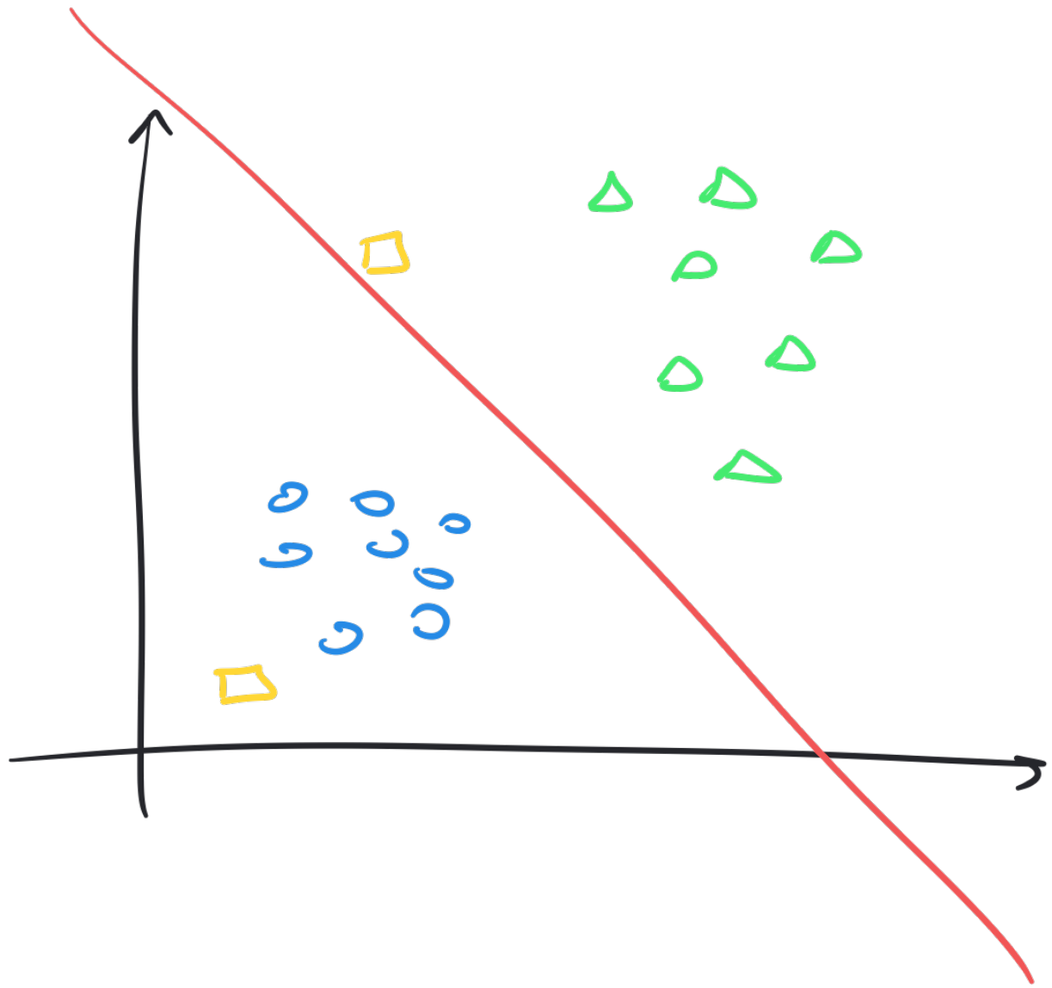
(2) the variance of estimator, $E[(y - E[y])^2]$, is dependent on its mean, $E[y]$.

∴ Avoid using Least Squares as a classifier.

Logistic Regression

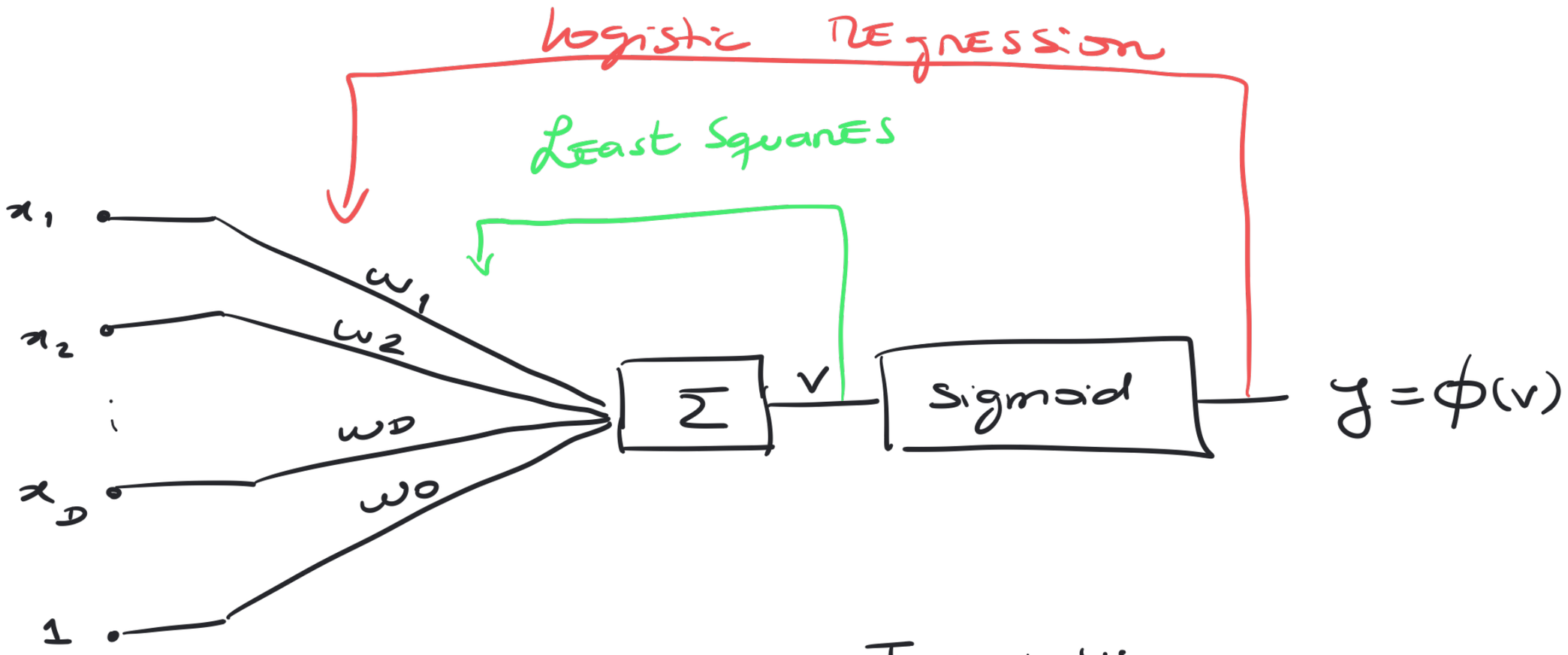
despite its name, it is a classifier.

- Linear



The idea is to find the discriminant function such that we minimize uncertainty!

$y = \phi(w^T x + w_0)$, $\phi(a) \equiv$ logistic fct.



where $v = w^T x + w_0$

and $\phi(x) = \dfrac{1}{1 + e^{-x}} \equiv$ logistic fct

$y = \phi(v) \equiv$ probabilities $\in [0, 1]$

Assume we only have 2 classes,

C_0 and $\overline{C_0}$:

$$\text{logit}\left(\underbrace{P(C_0|x)}_{\text{posterior prob. of } C_0 \text{ for sample } x}\right) = \ln\left(\frac{P(C_0|x)}{P(\overline{C_0}|x)}\right)$$

$$= \ln\left(\frac{P(C_0|x)}{1-P(C_0|x)}\right) = w^T x + b$$

$$\Rightarrow \frac{P(C_0|x)}{1-P(C_0|x)} = \exp(w^T x + b)$$

↙ Applying exp. fct

$$\Leftrightarrow P(C_0|x) = (1-P(C_0|x)) \cdot \exp(w^T x + b)$$

$$P(C_0|x) = \frac{\exp(\overbrace{w^T x + b}^{=v})}{1 + \exp(\underbrace{w^T x + b}_{=v})}$$

$$= \frac{\exp(v)}{1 + \exp(v)}$$

$$= \frac{1}{1 + \exp(-v)} \equiv \text{logistic/sigmoid function}$$

where $v = w^T x + b$

If $P(C_0|x) \gtrless P(\overline{C_0}|x)$

$x \in C_0$ if $>$, $x \in \overline{C_0}$ if $<$

OBJECTIVE function for logistic REGRESSION

TRAINING DATA : $\{(x_i, t_i)\}_{i=1}^{N}$, $t_i \in \{0, 1\}$

MAPPER : $y = \phi(w^T x + w_0) \equiv$ probabilities

OBSERVED DATA LIKELIHOOD :

$$\mathcal{L} = P(y_1, y_2, \cdots, y_N \mid x, w)$$

$$= \prod_{i=1}^{N} P(y_i \mid x, w)$$

1 — C_0

0 — $\overline{C_0}$

Example:

$$x = \{x_1, \; x_2, \; x_3\}$$

$$t = \{0, \; 1, \; 1\}$$

$$y_{w_1} = \{0.3, \; 0.8, \; 0.9\}$$

$$y_{w_2} = \{0.4, \; 0.6, \; 0.6\}$$

STEP 1

$$\mathcal{L}^{\circ} = \prod_{i=1}^{N} P(y_i / x_i, w)$$

$$= \prod_{i=1}^{N} \phi(v_i)^{t_i} \cdot (1 - \phi(v_i))^{1-t_i}$$

$$v_i = w^T \cdot x_i + w_0$$

$$\phi(v_i) = \frac{1}{1 + e^{-v_i}}$$

Example from previous page:

$$\mathcal{L}^{\circ}_{w_1} = 0.8 \times 0.9 \times (1 - 0.3)$$

STEP 2

TAKE THE OBSERVED log-likelihood:

$$\mathcal{L} = \ln(\mathcal{L}^{\circ})$$

$$= \ln\left(\prod_{i=1}^{N} \phi(v_i)^{t_i} \cdot (1-\phi(v_i))^{1-t_i}\right)$$

$$= \sum_{i=1}^{N} \left[t_i \cdot \ln(\phi(v_i)) + (1-t_i) \cdot \ln(1-\phi(v_i))\right]$$

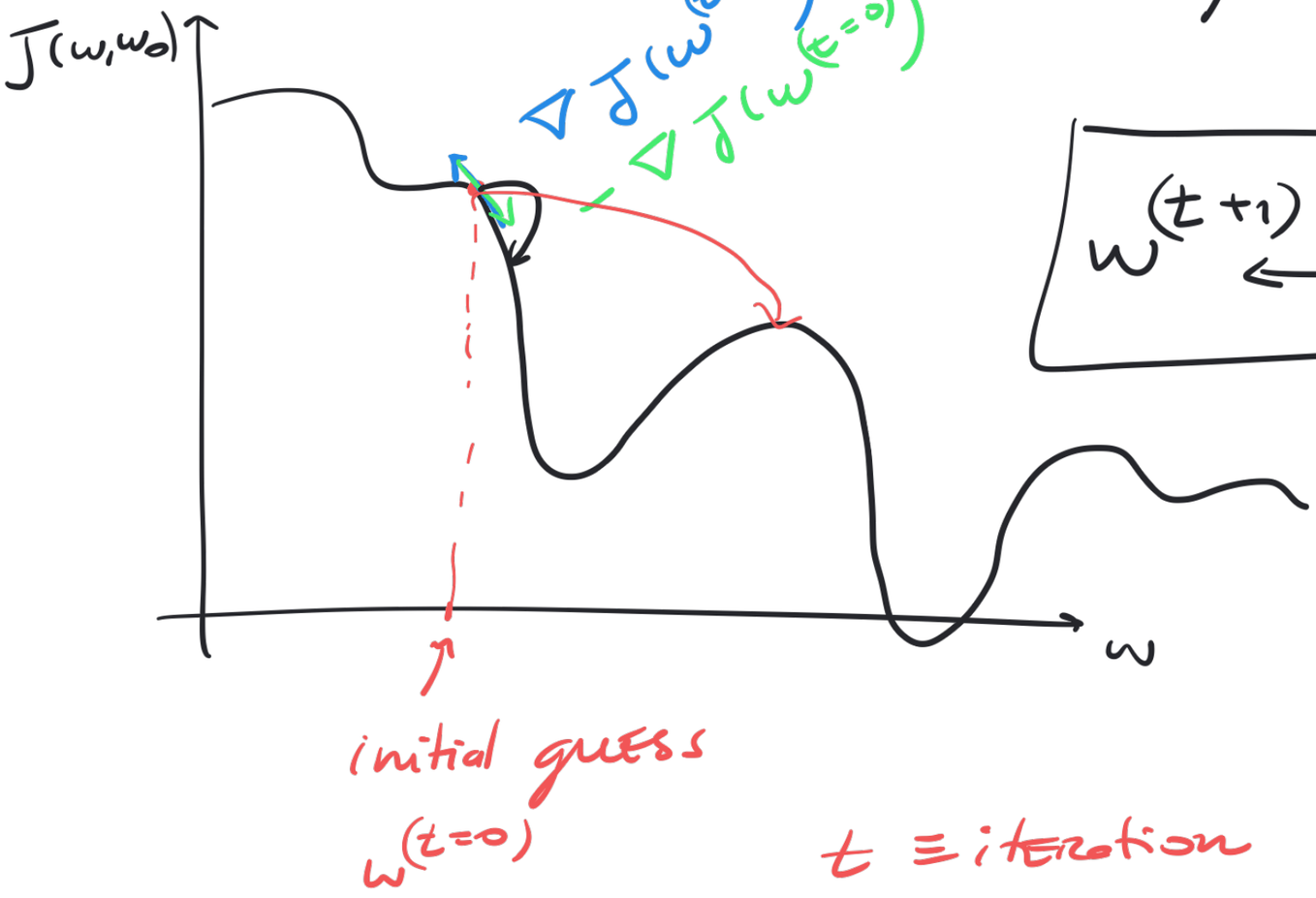
Goal: $\arg\max_{\{w, w_0\}} \mathcal{L} = \arg\min_{\{w, w_0\}} -\mathcal{L}$

As a minimization problem,

we optimize the Cross-Entropy:

$$J(w, w_0) = \sum_{i=1}^{N} \left[-t_i \cdot \ln(\phi(v_i)) - (1-t_i) \cdot \ln(1-\phi(v_i)) \right]$$

this is a non-convex fct, so we use search algorithm to find a solution ← Gradient descent



$\nabla J(w^{(t=0)})$

$\nabla J(w^{(t=0)})$

$t \equiv$ iteration

$\eta \equiv$ LEARNING RATE ("ETA")

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \nabla J(w^{(t)})$$

η is a hyperparameter

STABILITY RANGE

$0 < \eta < 1$

Solution converges to a local minima not necessarily global b/c of initialization.

$$J(w, w_0) = \sum_{i=1}^{N} \left[-t_i \cdot \ln(\phi(v_i)) - (1 - t_i) \cdot \ln(1 - \phi(v_i)) \right]$$

where $v_i = w^T x_i + w_0$

① $\dfrac{\partial J(w, w_0)}{\partial w} = 0$

$$\Leftrightarrow \sum_{i=1}^{N} \left[-t_i \cdot \underbrace{\frac{\partial \ln(\phi(v_i))}{\partial \phi(v_i)}}_{= \frac{1}{\phi(v_i)}} \cdot \underbrace{\frac{\partial \phi(v_i)}{\partial v_i}}_{= \phi'(v_i)} \cdot \underbrace{\frac{\partial v_i}{\partial w}}_{= x_i} \right.$$

$$\left. - (1 - t_i) \cdot \underbrace{\frac{\partial \ln(1 - \phi(v_i))}{\partial \phi(v_i)}}_{= \frac{-1}{1 - \phi(v_i)}} \underbrace{\frac{\partial \phi(v_i)}{\partial v_i}}_{= \phi'(v_i)} \cdot \underbrace{\frac{\partial v_i}{\partial w}}_{= x_i} \right]$$

$$\frac{\partial J}{\partial w} = \sum_{i=1}^{N} \left[\frac{t_i}{\phi(v_i)} - \frac{1 - t_i}{1 - \phi(v_i)} \right] \cdot \phi'(v_i) \cdot x_i$$

where $\phi'(x) = \phi(x) \cdot (1 - \phi(x))$

Reducing to same denominator:

$$\frac{\partial J}{\partial w} = \sum_{i=1}^{N} \left(t_i - \phi(v_i) \right) \cdot x_i$$

Similarly,

$$\frac{\partial J}{\partial w_0} = \sum_{i=1}^{N} \left(t_i - \phi(v_i) \right)$$

Pseudo-code:

Input: DATA X, target t, learning rate η.

$t = 0$ (iteration)

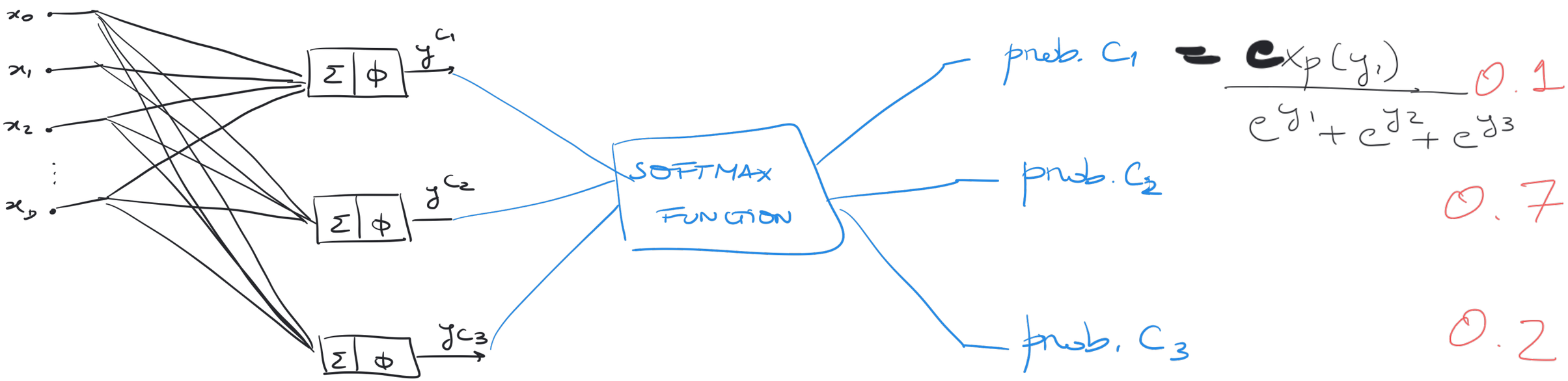
① Initialize parameters $w^{(t)}$, $w_0^{(t)}$

② Update the parameters

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \cdot \frac{\partial J(w, w_0)}{\partial w^{(t)}}$$

$$w_0^{(t+1)} \leftarrow w^{(t)} - \eta \cdot \frac{\partial J(w, w_0)}{\partial w_0^{(t)}}$$

③ $t \leftarrow t+1$. Iterate until convergence criteria is met.



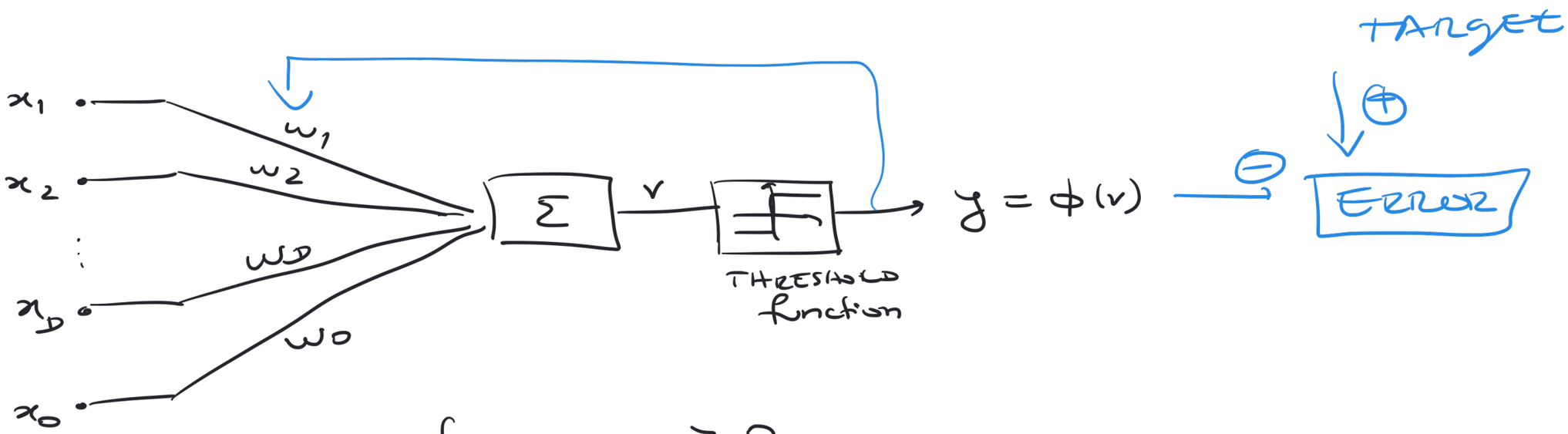
SOFTMAX function :

$$\frac{exp(y_j(x))}{\sum_j exp(y_j(x))} = p_j \qquad \sum = 1$$

The Perceptron, 1957, Rosenblatt

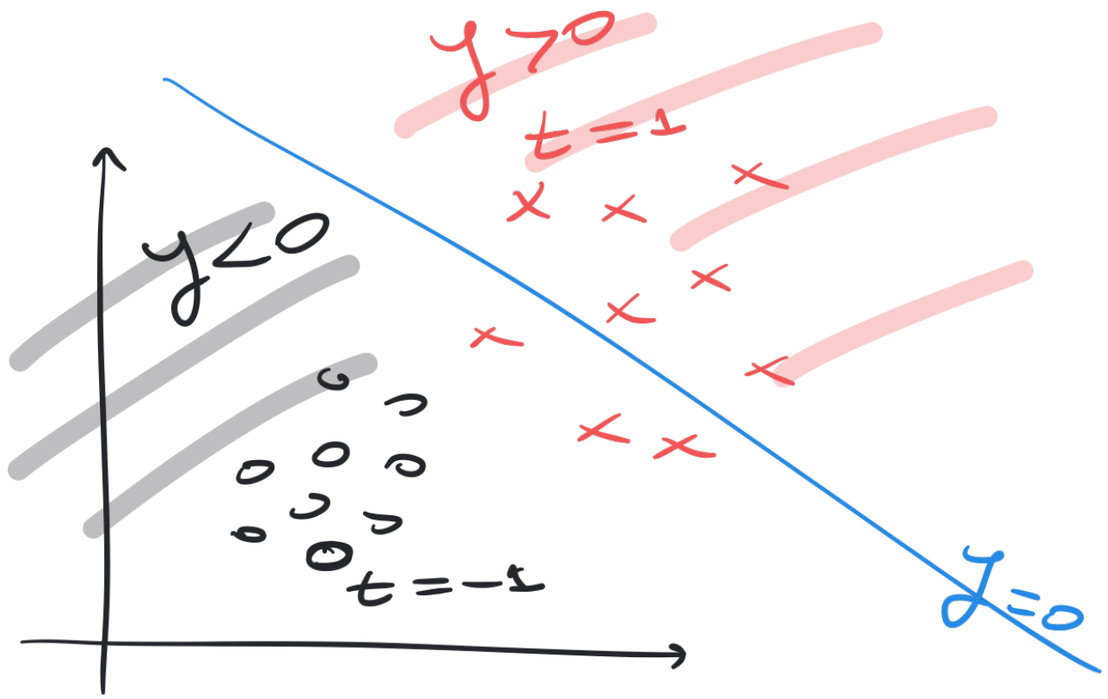
Linear binary classifier

↳ discriminative



$$\phi(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

$$y = \phi(w^T x + w_0)$$



o — class −1

x — class 1

initial discriminant function

NOTE that <u>ALL MISCLASSIFIED</u> points satisfy:

$$t_n \cdot y(x_n) < 0$$

and all <u>CORRECTLY CLASSIFIED</u> points

$$t_n \cdot y(x_n) > 0$$

OBJECTIVE function

Uses only misclassified points.

and it iterates through new values for w and w_0 until NO misclassified points exist.

$$E_p(w, w_0) = -\sum_{n \in M} t_n \cdot (w^T \cdot x_n + w_0)$$

$M \equiv$ set of Misclassified points

$$\arg\min_{\{w, w_0\}} E_p(w, w_0)$$