# CIS 5200: MACHINE LEARNING

## BINARY CLASSIFICATION AND PERCEPTRON

**Surbhi Goel**

*Content here draws from material by Yingyu Liang (Princeton), Christopher De Sa and Kilian Weinberger (Cornell)*

**Spring 2023**

# LOGISTICS - UPCOMING

**Homework**:

- HW0 due on **Friday, Jan 20, 2023** end of day
- Go to OHs if you have any clarifications about HW0
- TAs will help review concepts
- For those on waitlist, email your HW0 to Keshav and Wendi (head TAs)
- HW1 will be out on Monday, Jan 23, 2023

**Recitation:**

- Sign up link will be posted on Ed this week

# LOGISTICS - RECORDING

**Recording Policy**:

- Only if you are unwell, or dealing with some extenuating circumstances and have to miss class
- Request video access via an Ed message to Keshav or Wendi
- Video lecture will be made available to you for a period of 1 week post the requested date
- Recordings will be provided as is, not intended to replace lecture

**We will run this honor-based, we will not ask any questions unless we notice excessive use**

# OUTLINE - TODAY

- Review of Supervised Learning
- Binary Classification
- Perceptron
  - History
  - Algorithm
  - Proof of convergence
  - Drawbacks
- Logistic Regression

# SUPERVISED LEARNING - REVIEW

Predict future outcomes based on past outcomes

**Inputs $x \in \mathcal{X}$**

**Labels $y \in \mathcal{Y}$**


($\mathcal{Y}$ = Breeds)
"Pug"
"Chihuahua"

**Classification**

Discrete labels


($\mathcal{Y}$ = Stock prices)
"$130.02"

**Regression**

Continuous labels

**Task:** Learn predictor $f : \mathcal{X} \rightarrow \mathcal{Y}$

# SUPERVISED LEARNING - REVIEW

**Loss function:** What is the right loss function for the task?

**Representation:** What class of functions should we use for the task?

**Optimization:** How can we efficiently solve the empirical risk minimization?

**Generalization:** Will the predictor perform well on unseen data?

# SUPERVISED LEARNING - BINARY CLASSIFICATION

**Input space:** $\mathcal{X} \subseteq \mathbb{R}^d$

**Output space:** $\mathcal{Y} = \{-1,1\}$ *we used* $\{0,1\}$ *in the last class*

**Predictor function:** $f : \mathcal{X} \rightarrow \mathcal{Y}, f \in \mathcal{F}$

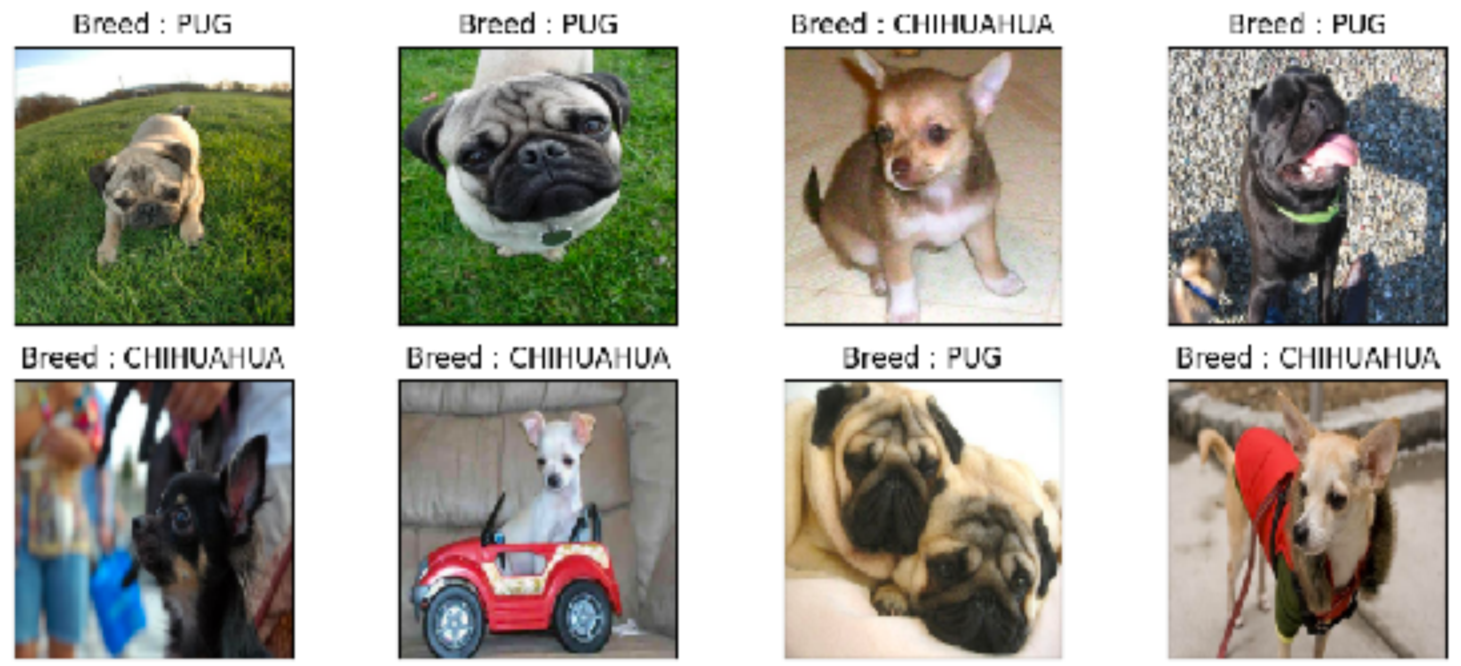
**Loss function:** $\ell(f(x), y) = \begin{cases} 0 & \text{if } f(x) = y \\ 1 & \text{otherwise.} \end{cases}$

**Data:** $\{(x_1, y_1), \ldots, (x_m, y_m)\} \subset \mathcal{X} \times \mathcal{Y}$ drawn i.i.d. from distribution $\mathcal{D}$

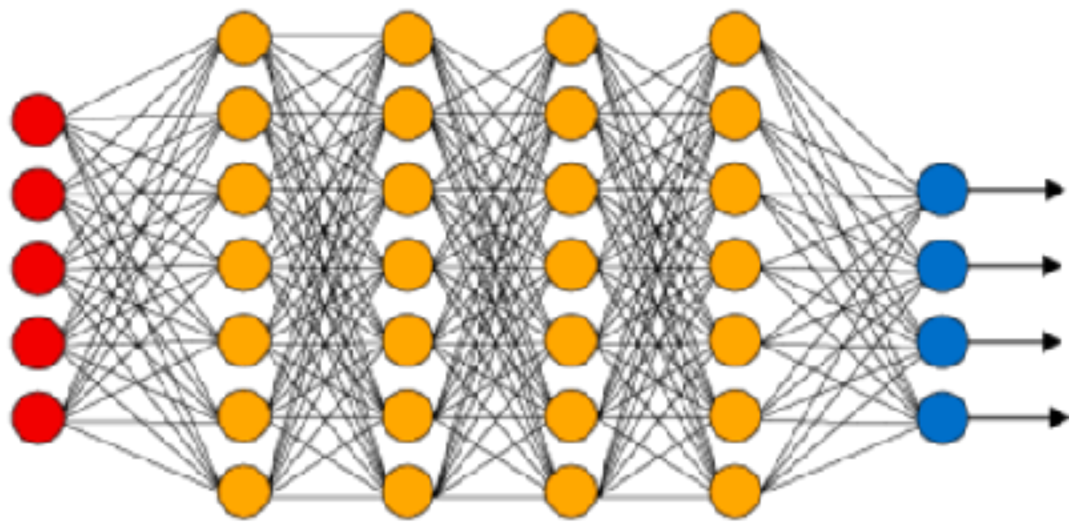
# CLASSIFICATION - PIPELINE

Training dataset

$$\mathcal{S} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$$




Function class $\mathscr{F}$

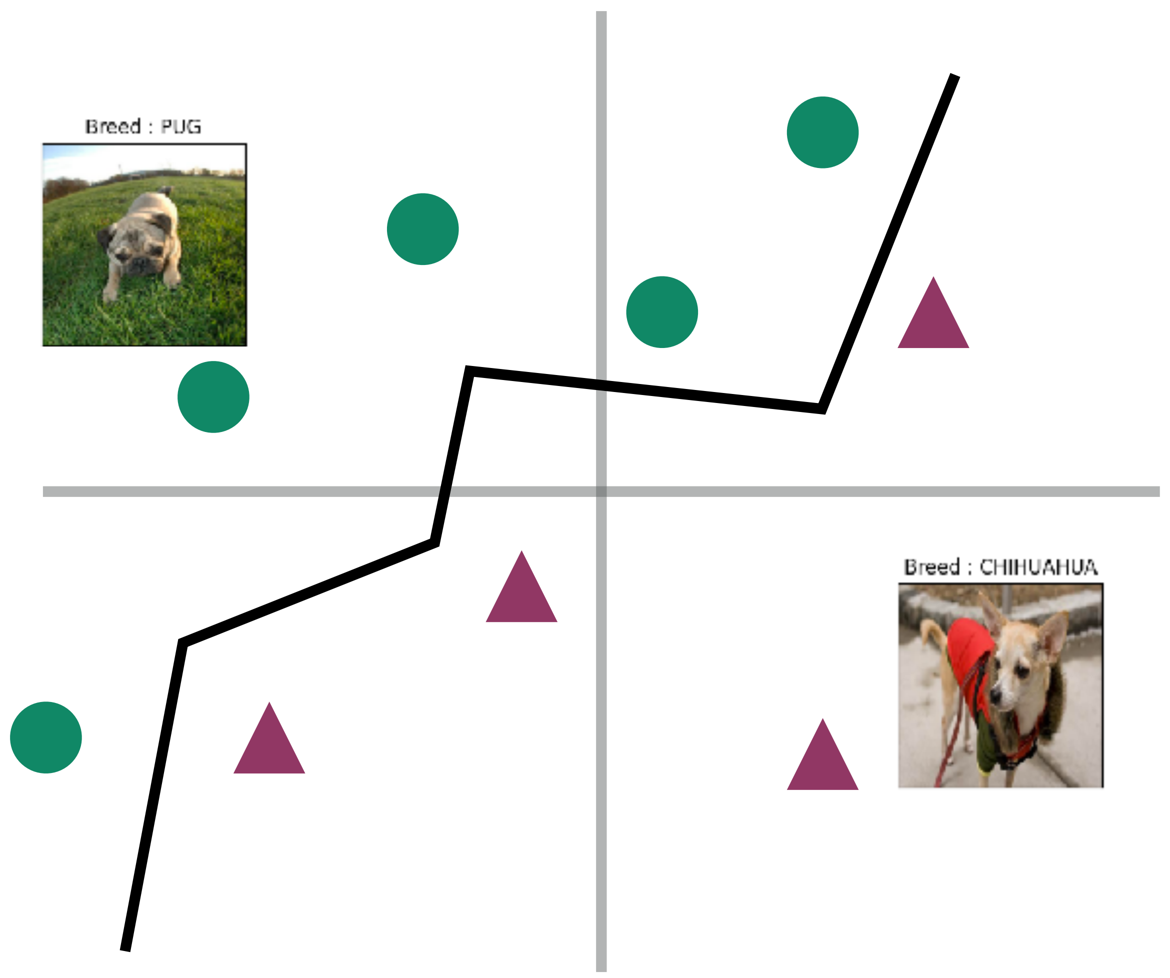


*Minimize loss on training data*

$$\min_{f \in \mathscr{F}} \frac{1}{m} \sum_{i=1}^{m} 1[f(x_i) \neq y_i]$$

*average number of mistakes*

Prediction function $\hat{f}$




*Evaluation*

$$R(\hat{f}) = \Pr_{(x,y)\sim\mathscr{D}} \left[\hat{f}(x) \neq y\right]$$

# HYPOTHESIS CLASS - LINEAR CLASSIFIER

*Halfspace* *weight* *bias*

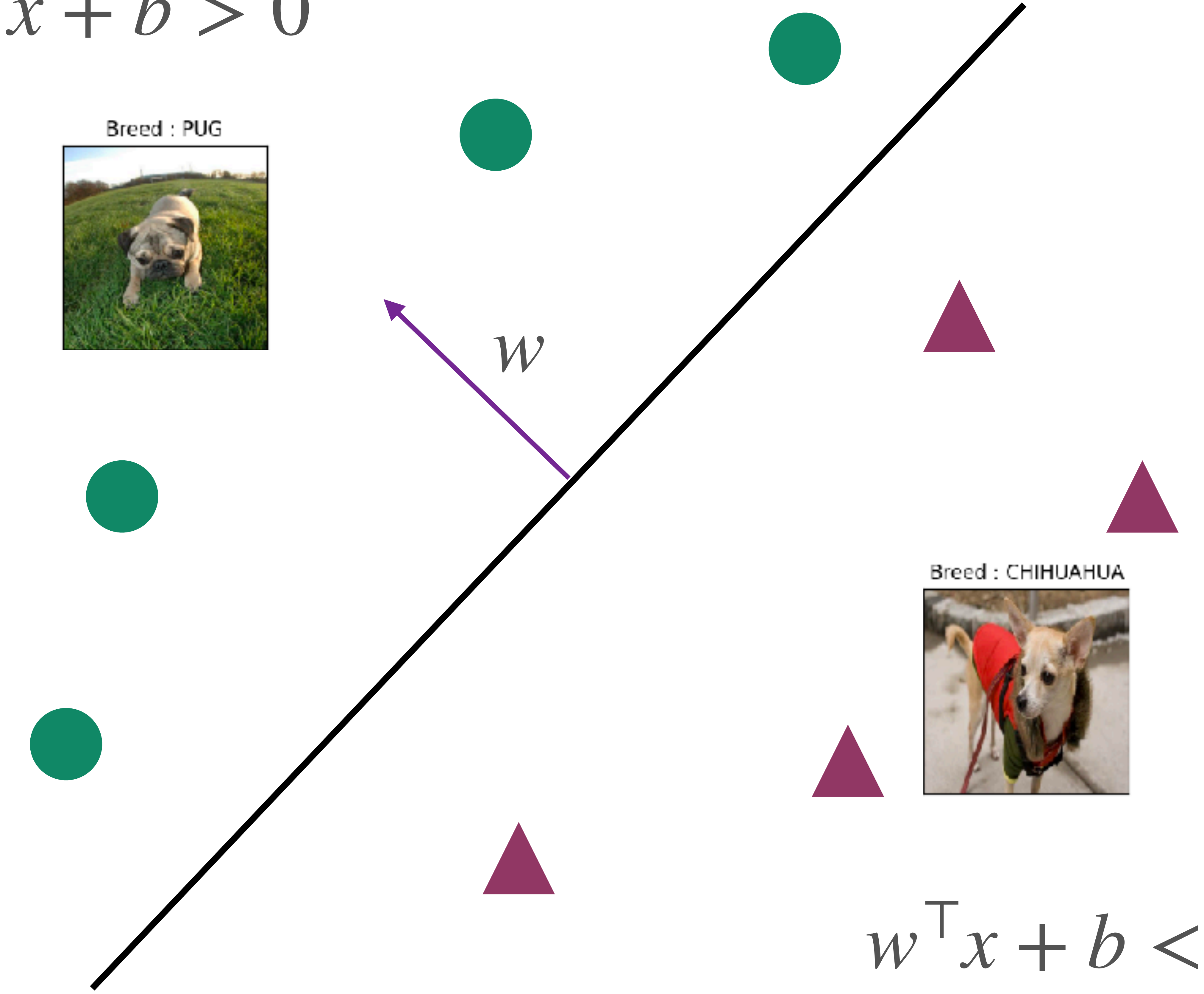
**Linear Classifier:** $\mathscr{F} := \{x \mapsto \text{sign}(w^\top x + b) \,|\, w \in \mathbb{R}^d, b \in \mathbb{R}\}$

$$\text{sign}(a) = \begin{cases} +1 & \text{if } a \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

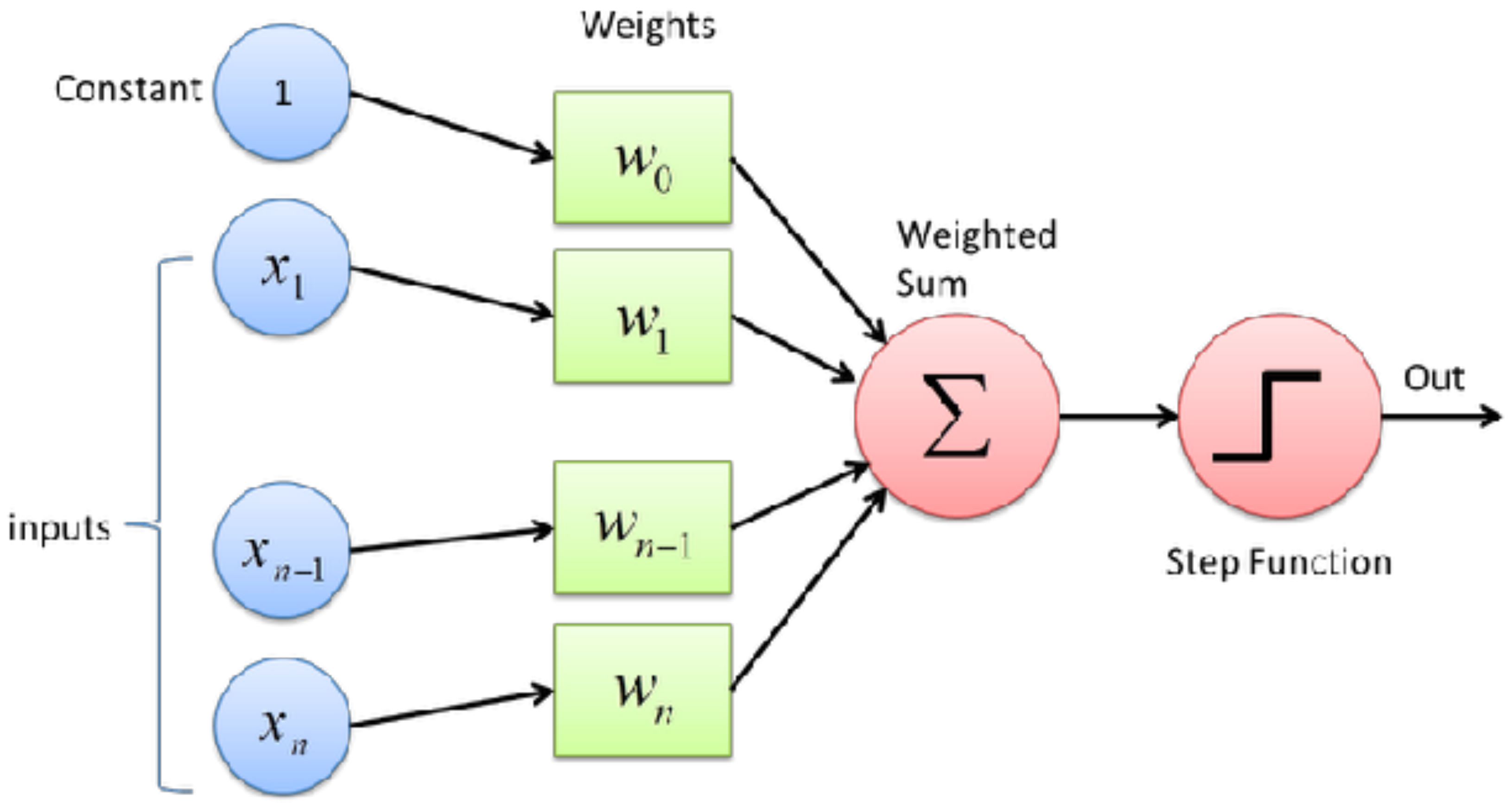
*Step function*

$w^\top x + b > 0$

$w^\top x + b = 0$




$w^\top x + b < 0$




*Perceptron*
*model of the biological neuron*

# HYPOTHESIS CLASS - LINEAR CLASSIFIER

**Linear Classifier:** $\mathscr{F} := \{x \mapsto \text{sign}(\underbrace{w^\top x}_{\text{no bias}}) \mid w \in \overbrace{\mathbb{R}^{d+1}}^{\text{extra dimension}}\}$

Map:

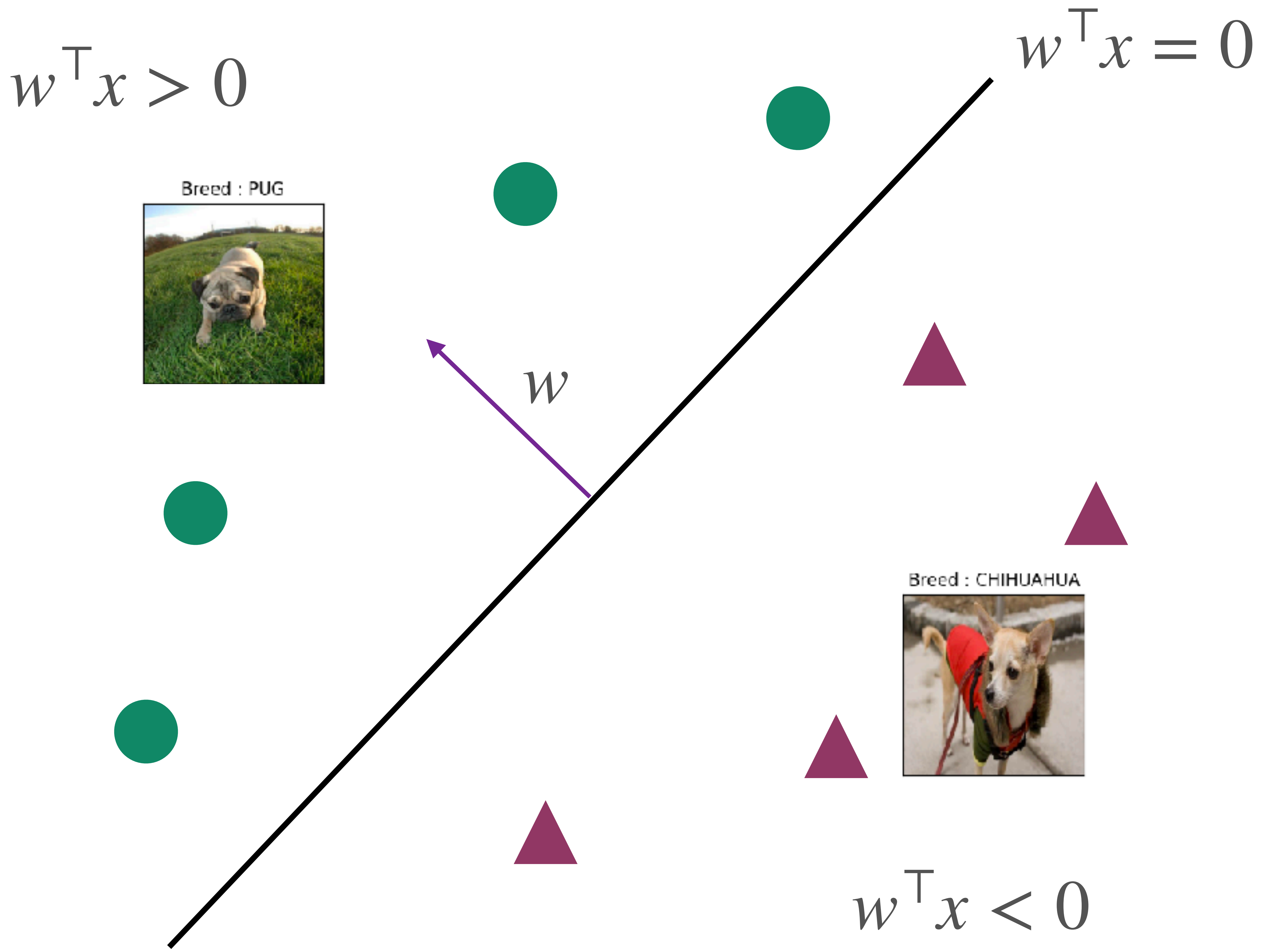
$x \mapsto \begin{bmatrix} x \\ 1 \end{bmatrix}$ and $w \mapsto \begin{bmatrix} w \\ b \end{bmatrix}$

*extra dimension*

$$\implies w^\top x + b \mapsto w^\top x$$

*no bias*

WLOG, we can assume no bias!



# LINEAR CLASSIFICATION - TRAINING

**Training Dataset:** $\mathcal{S} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$,
$x_i \in \mathbb{R}^d, y_i \in \{-1,1\}$

**Empirical Risk Minimization:** Find $\hat{w}$ that minimizes

$$\widehat{\text{err}}(w) = \frac{1}{m}\sum_{i=1}^{m} 1\left[\text{sign}(w^\top x_i) \neq y_i\right]$$

How do we solve this minimization problem?

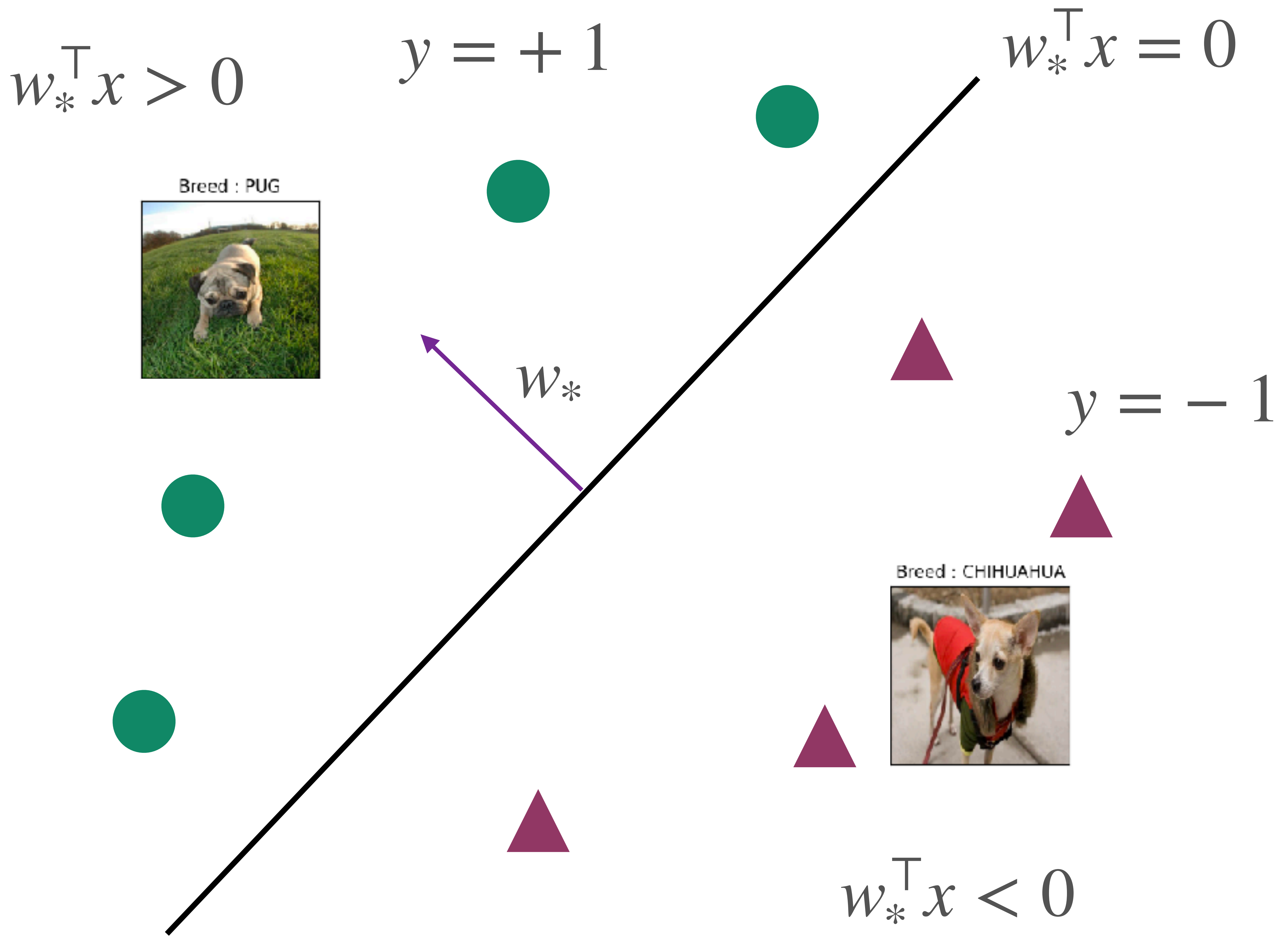
Hard in general, the problem is non-convex!

# ASSUMPTION - PERFECT CLASSIFIER

**Perfect Classifier:** $\exists w_*$ such that $y = \textbf{sign}(w_*^\top x)$ and $\|w_*\| = 1$

Data is linearly separable

$$\widehat{\text{err}}\,(w_*) = \frac{1}{m}\sum_{i=1}^{m} 1\left[\text{sign}(w_*^\top x_i) \neq y_i\right] = 0$$



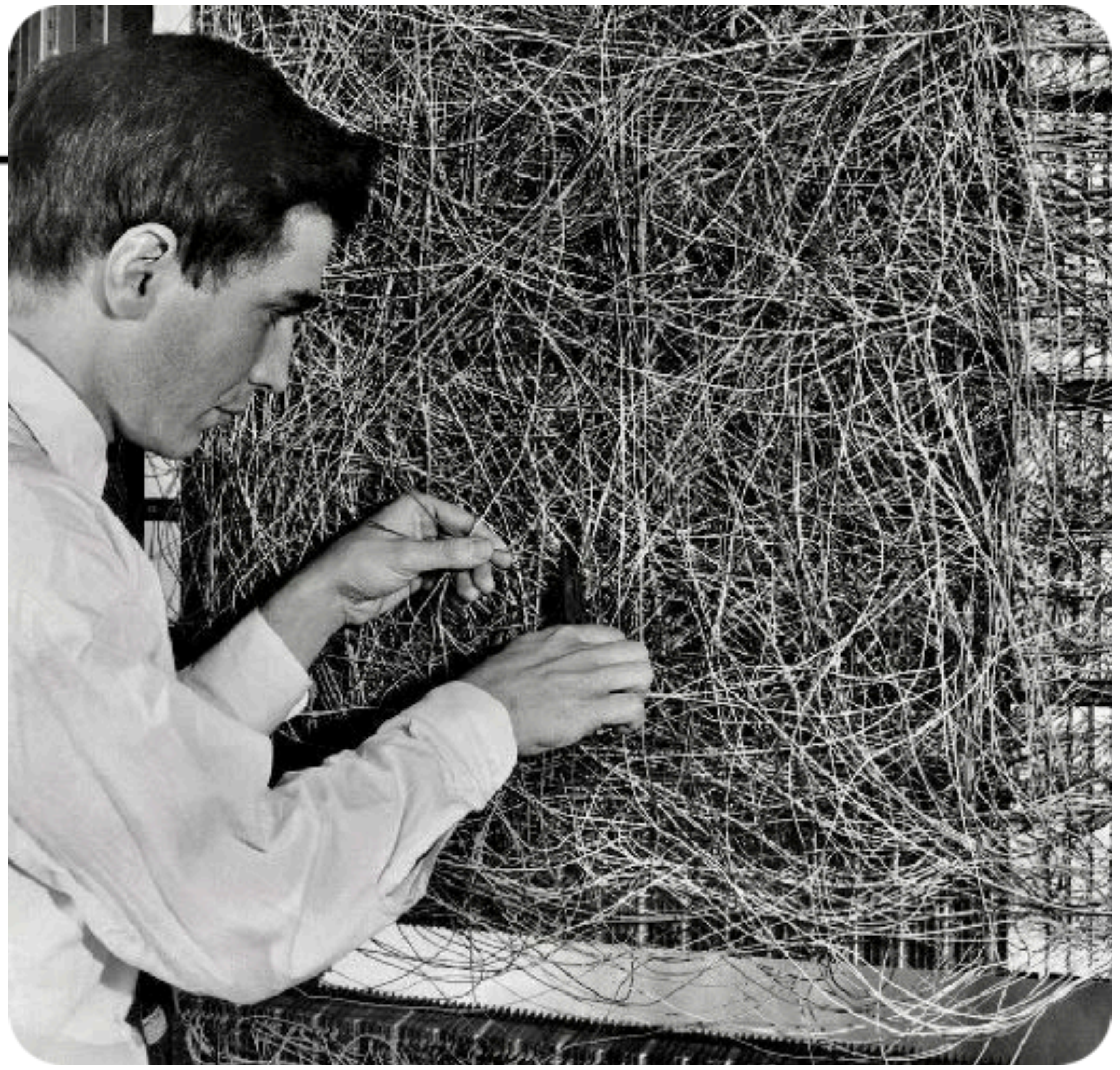
# ALGORITHM - PERCEPTRON

**Algorithm 1:** Perceptron

Initialize $w_1 = 0 \in \mathbb{R}^d$
**for** $t = 1, 2, \ldots$ **do**
  **if** $\exists i \in [m] \ s.t. \ y_i \neq \operatorname{sign}\left(w_t^\top x_i\right)$ **then** update $w_{t+1} = w_t + y_i x_i$
  **else** output $w_t$
**end**

The New York Times 1958
Electronic 'Brain' Teaches Itself

Lots of hype, expected to recognize people, and eventually gain 'consciousness'



Frank Rosenblatt with a Mark I Perceptron in 1960

# PERCEPTRON - INTUITION

**Algorithm 1:** Perceptron

Initialize $w_1 = 0 \in \mathbb{R}^d$
**for** $t = 1, 2, \ldots$ **do**
  **if** $\exists i \in [m]$ $s.t.$ $y_i \neq \text{sign}\left(w_t^\top x_i\right)$ **then** update $w_{t+1} = w_t + y_i x_i$
  **else** output $w_t$
**end**

Suppose at time $t$, example $x_i \neq 0$ is incorrectly classified

- If $y_i = 1$ then $w_{t+1}^\top x_i = w_t^\top x_i + \|x_i\|^2 > w_t^\top x_i$ Towards the positive side
- If $y_i = -1$ then $w_{t+1}^\top x_i = w_t^\top x_i - \|x_i\|^2 < w_t^\top x_i$ Towards the negative side

# PERCEPTRON - INTUITION

---

**Algorithm 1:** Perceptron

---

Initialize $w_1 = 0 \in \mathbb{R}^d$

**for** $t = 1, 2, \ldots$ **do**

  **if** $\exists i \in [m]$ $s.t.$ $y_i \neq \text{sign}\left(w_t^\top x_i\right)$ **then** update $w_{t+1} = w_t + y_i x_i$

  **else** output $w_t$

**end**

---

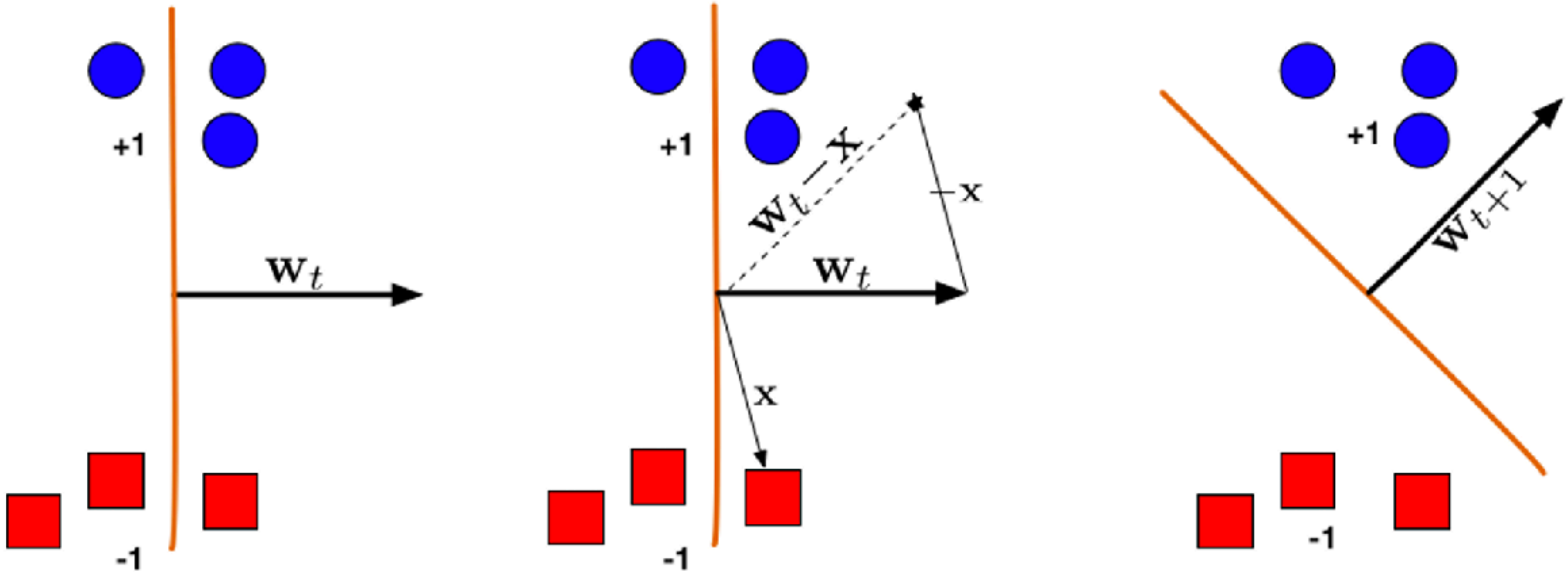
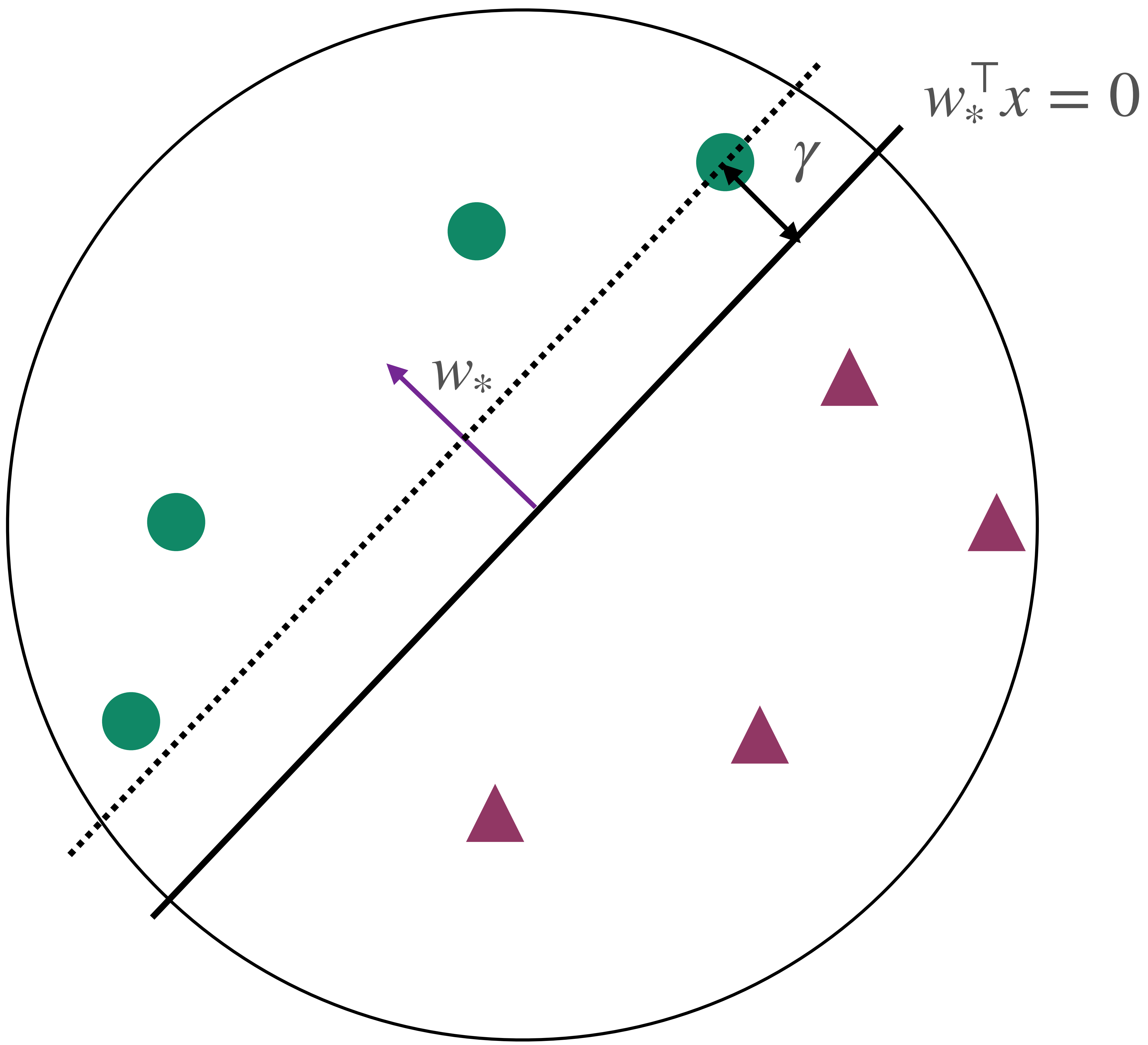



Image source: https://www.cs.cornell.edu/courses/cs4780/2022sp/notes/Notes06.pdf

# PERCEPTRON - CONVERGENCE




**Setting:**

For all $i \in [m]$, $||x_i|| \leq 1$

Margin $\gamma$ is minimum distance of any point from the hyperplane

$$\gamma = \min_{i \in [m]} |w_*^\top x_i|$$

**Theorem:**

The Perceptron algorithm stops after at most $1/\gamma^2$ rounds, and returns a hyperplane $w$ such that all examples are correctly classified.

**Algorithm 1:** Perceptron

Initialize $w_1 = 0 \in \mathbb{R}^d$
**for** $t = 1, 2, \ldots$ **do**
  **if** $\exists i \in [m]$ $s.t.$ $y_i \neq \text{sign}\left(w_t^\top x_i\right)$ **then** update $w_{t+1} = w_t + y_i x_i$
  **else** output $w_t$
**end**

## Setting:

For all $i \in [m]$, $\|x_i\| \leq 1$, $\|w_*\| = 1$

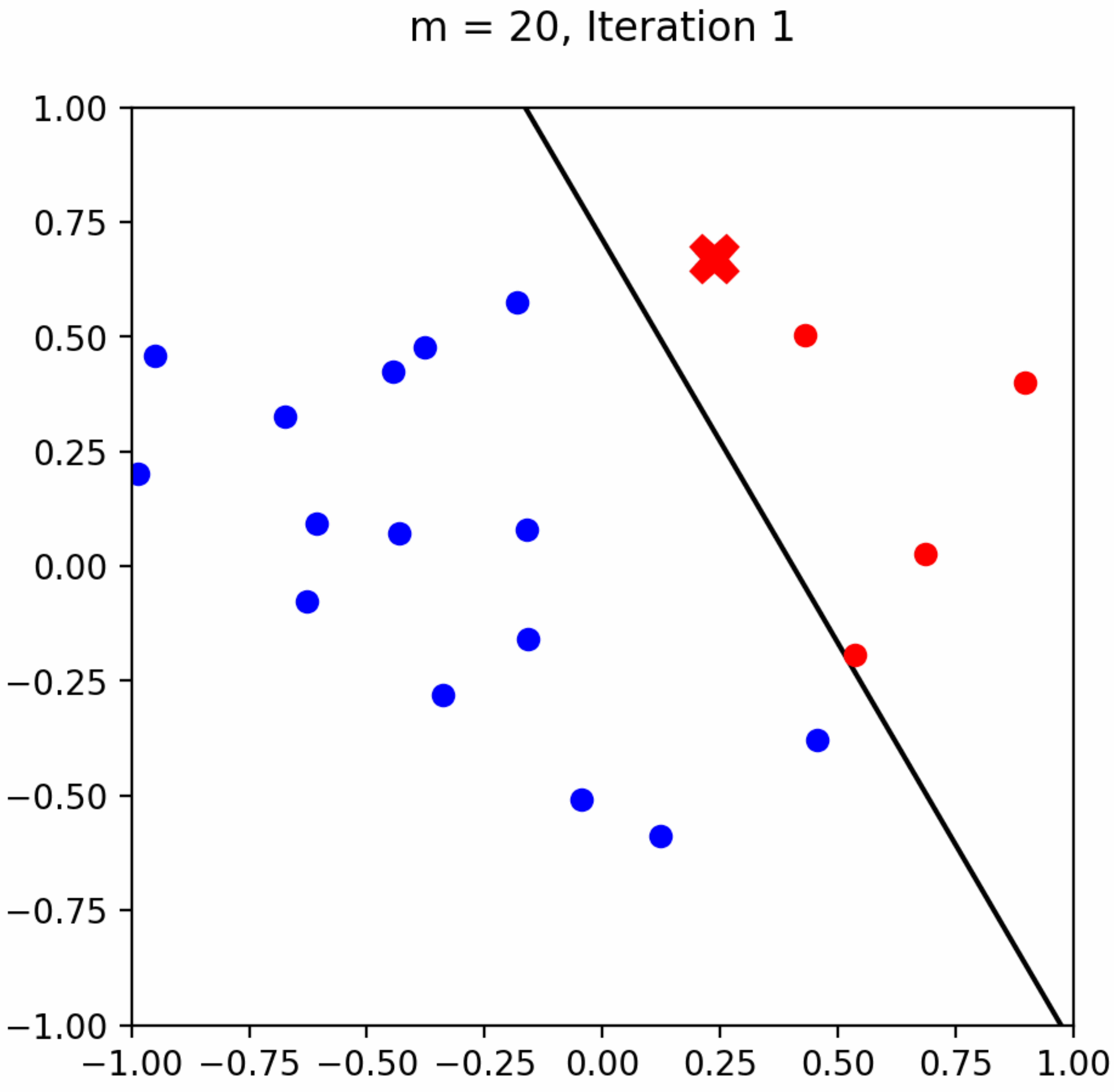
Margin $\gamma = \min_{i \in [m]} |w_*^\top x_i|$

See board/iPad

## Theorem:

The Perceptron algorithm stops after at most $1/\gamma^2$ rounds, and returns a hyperplane $w$ such that all examples are correctly classified.

# PERCEPTRON - IN ACTION



# PERCEPTRON - FAILURES

**XOR:**

Led to the AI winter till mid 1980s

Minsky and Papert in a 1969 book "Perceptrons" showed that Perceptron fails on XOR problems

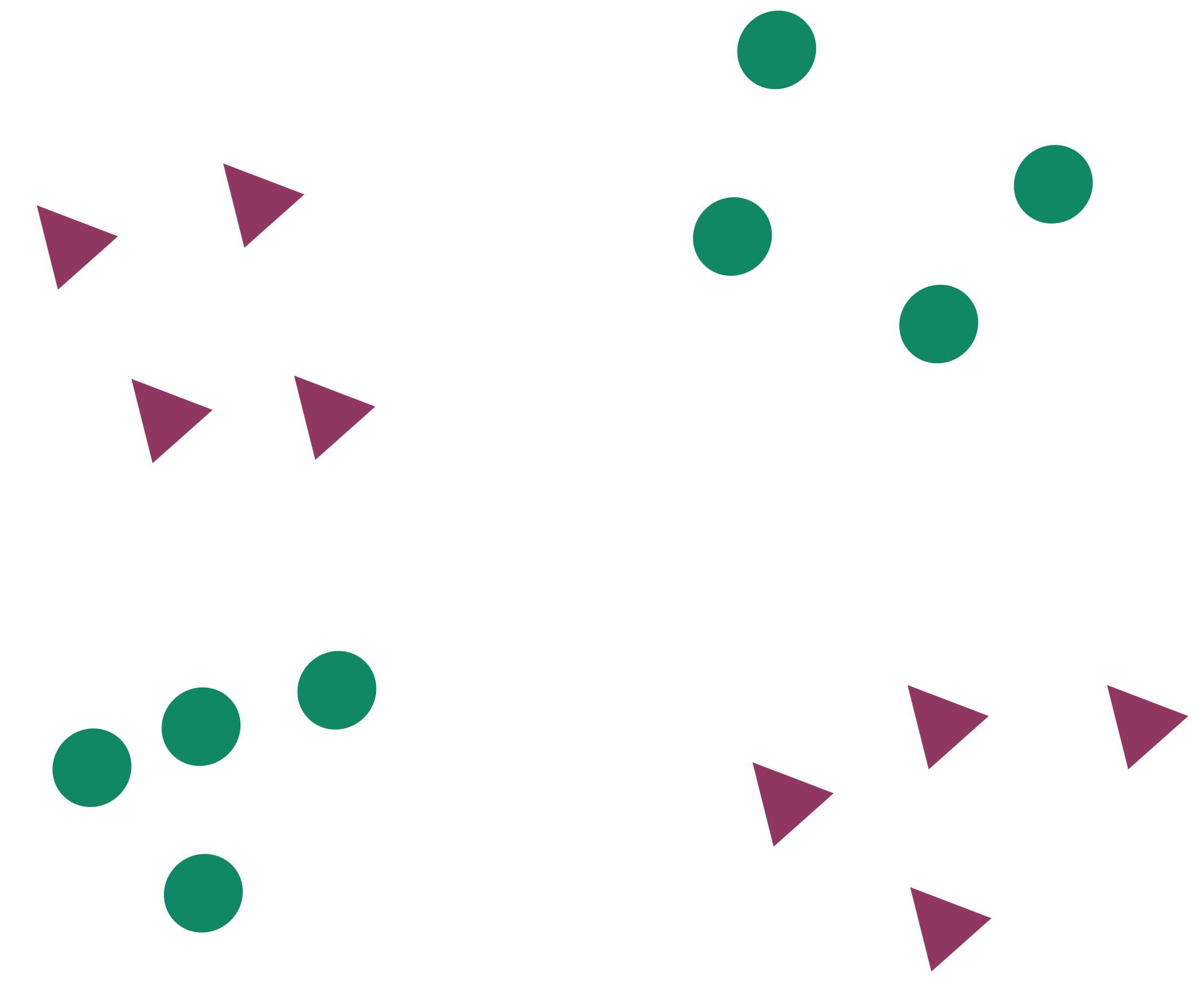
**Non-linearly separable data:**

Kernels (later in class)

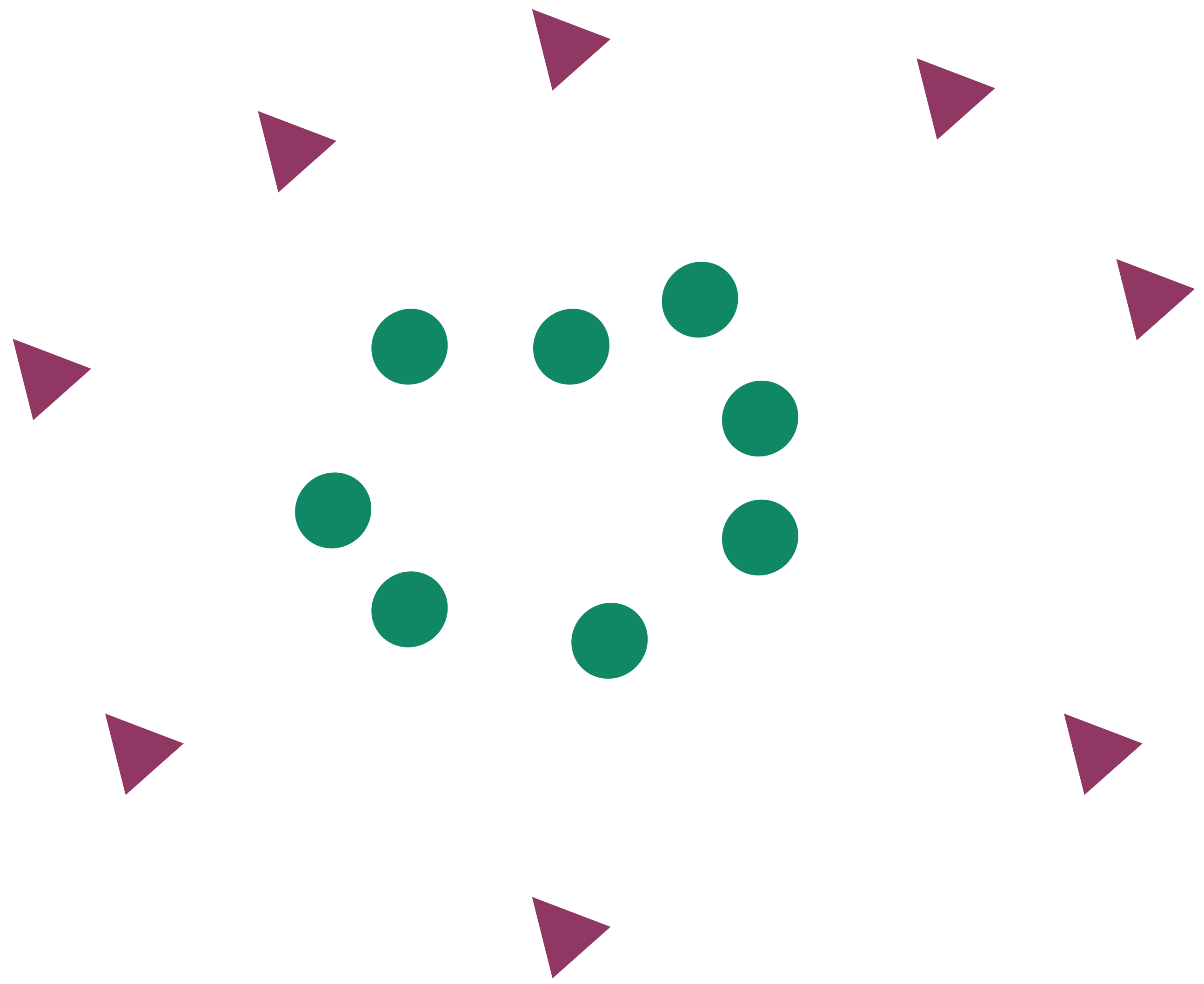
Separable in a lifted space

**Noise:**

Hard classifier, cannot model inherent noise



XOR



Non-separable Data

# PERCEPTRON - SUMMARY

**Input space:** $\mathcal{X} \subseteq \mathbb{R}^d$

**Output space:** $\mathcal{Y} = \{-1, 1\}$

**Hypothesis Class:** $\mathcal{F} := \{x \mapsto \text{sign}(w^\top x + b) \,|\, w \in \mathbb{R}^d, b \in \mathbb{R}\}$

**Loss function:** $\ell(f(x), y) = \begin{cases} 0 & \text{if } f(x) = y \\ 1 & \text{otherwise.} \end{cases}$

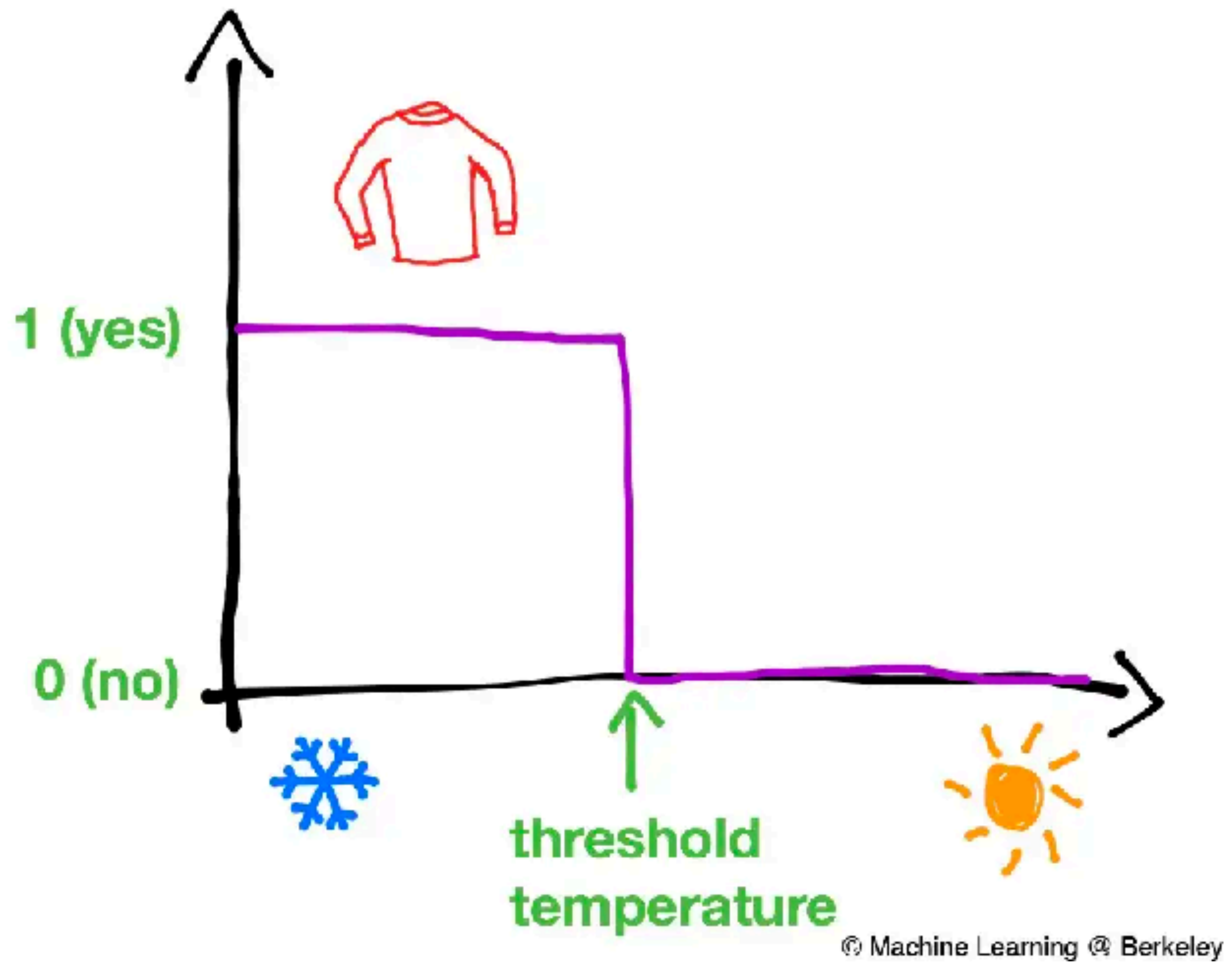
**Assumption:** Linearly separable data

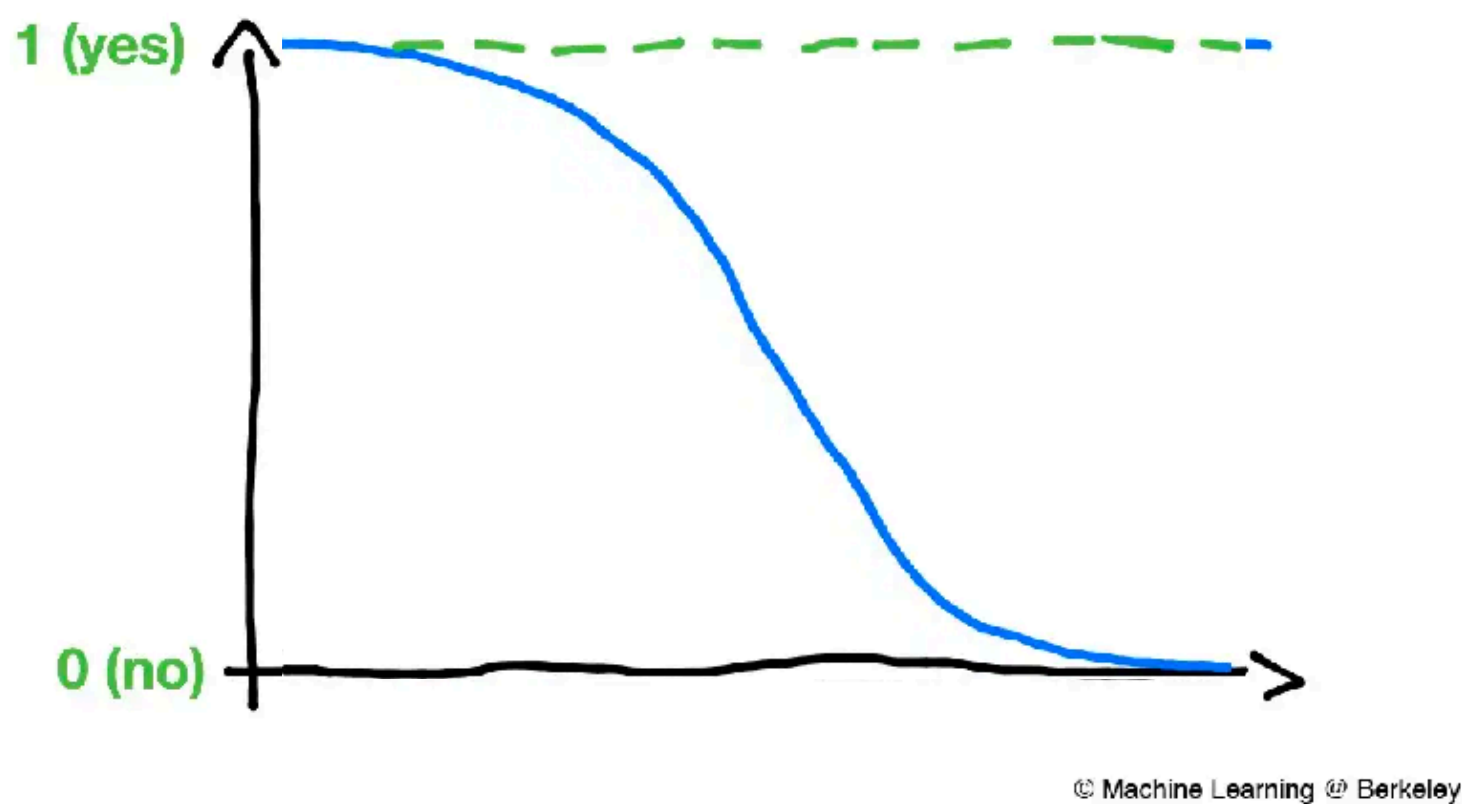
**Guarantee:** Zero-error on training data after $1/\gamma^2$ iterations for margin $\gamma$

# NON-DETERMINISTIC INPUTS

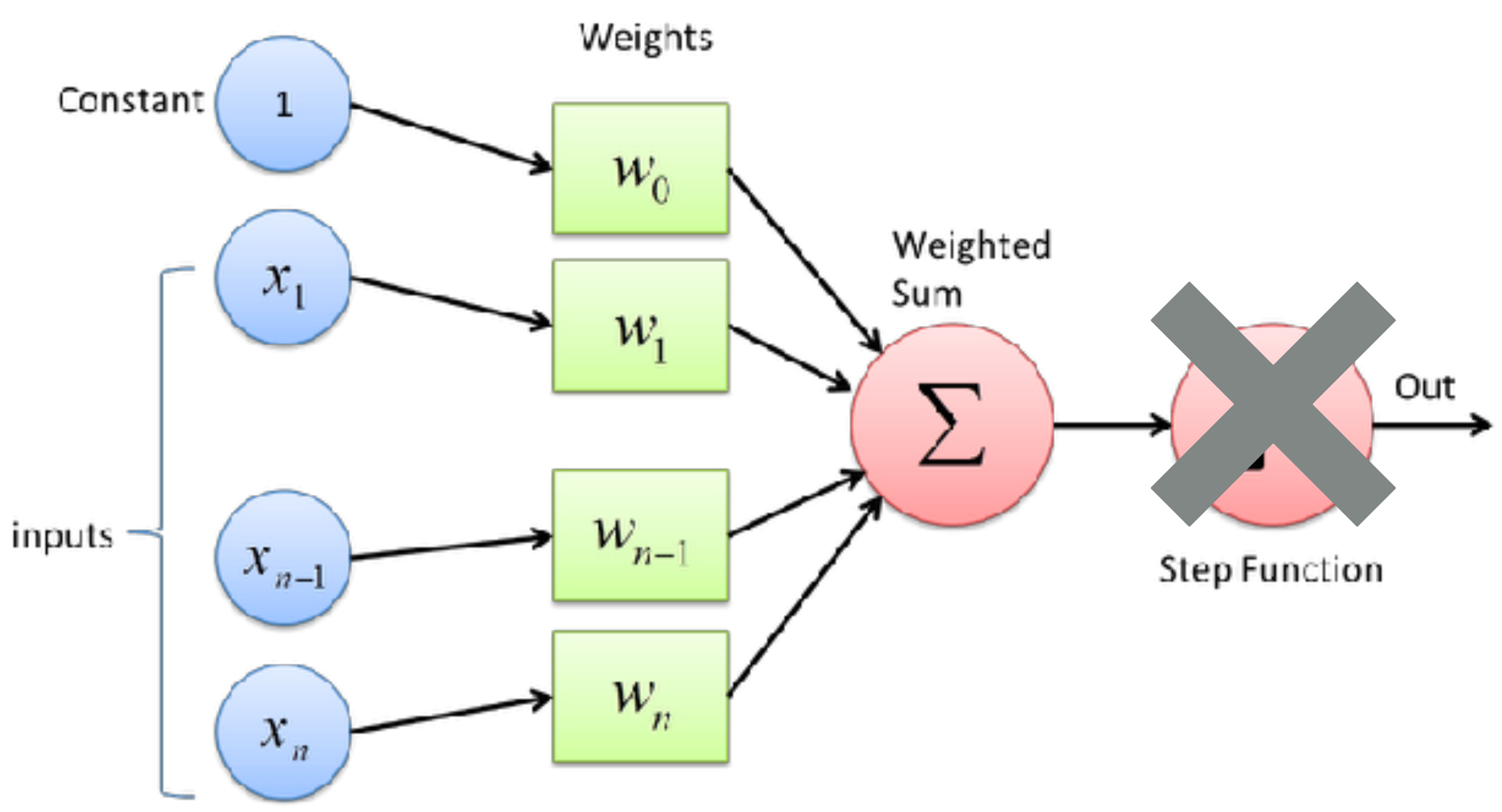
Perceptron used the **sign** function to assign deterministic labels

**But there may be inherent uncertainty in the label**



# LOGISTIC FUNCTION

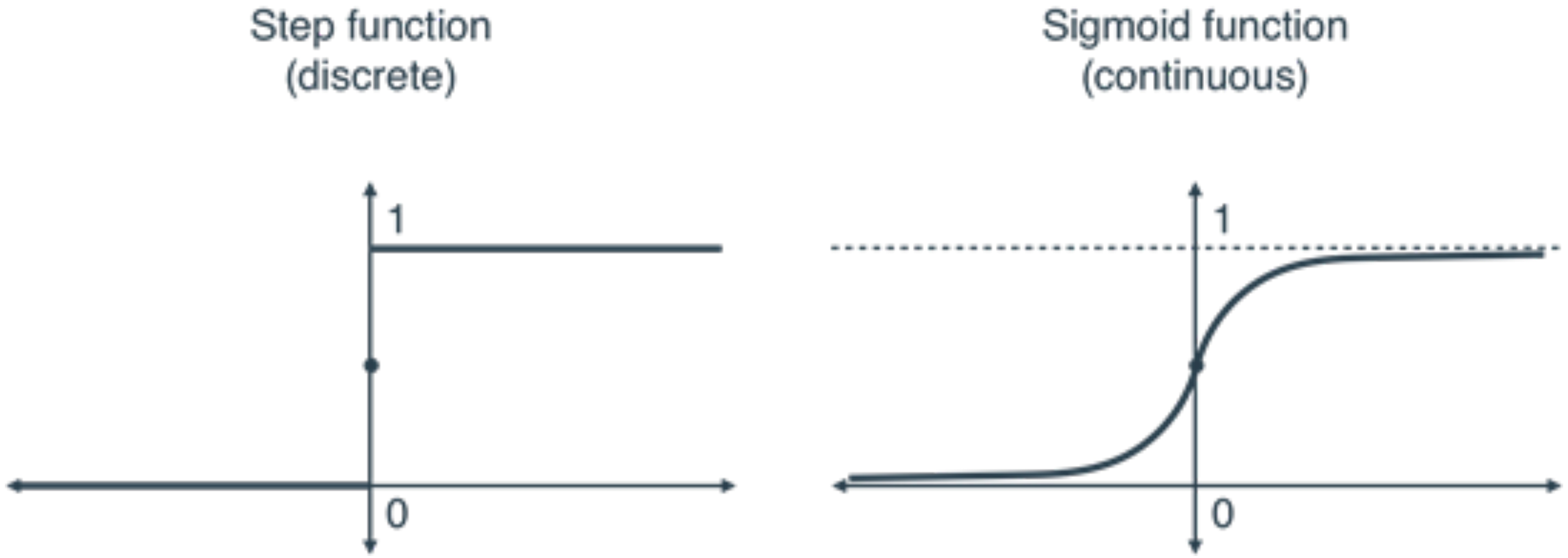



$$\text{sign}(a) = \begin{cases} +1 & \text{if } a \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

*Step function*

$$\text{sigmoid}(a) = \frac{1}{1 + \exp(-a)}$$

*Sigmoid function*




$$P(y = 1 \,|\, x) = \text{sigmoid}(w^\top x) = \frac{1}{1 + \exp(-w^\top x)}$$

$$P(y = -1 \,|\, x) = 1 - \text{sigmoid}(w^\top x) = \frac{1}{1 + \exp(w^\top x)}$$

*More unsure near the decision boundary*

*Like perceptron away from the decision boundary*

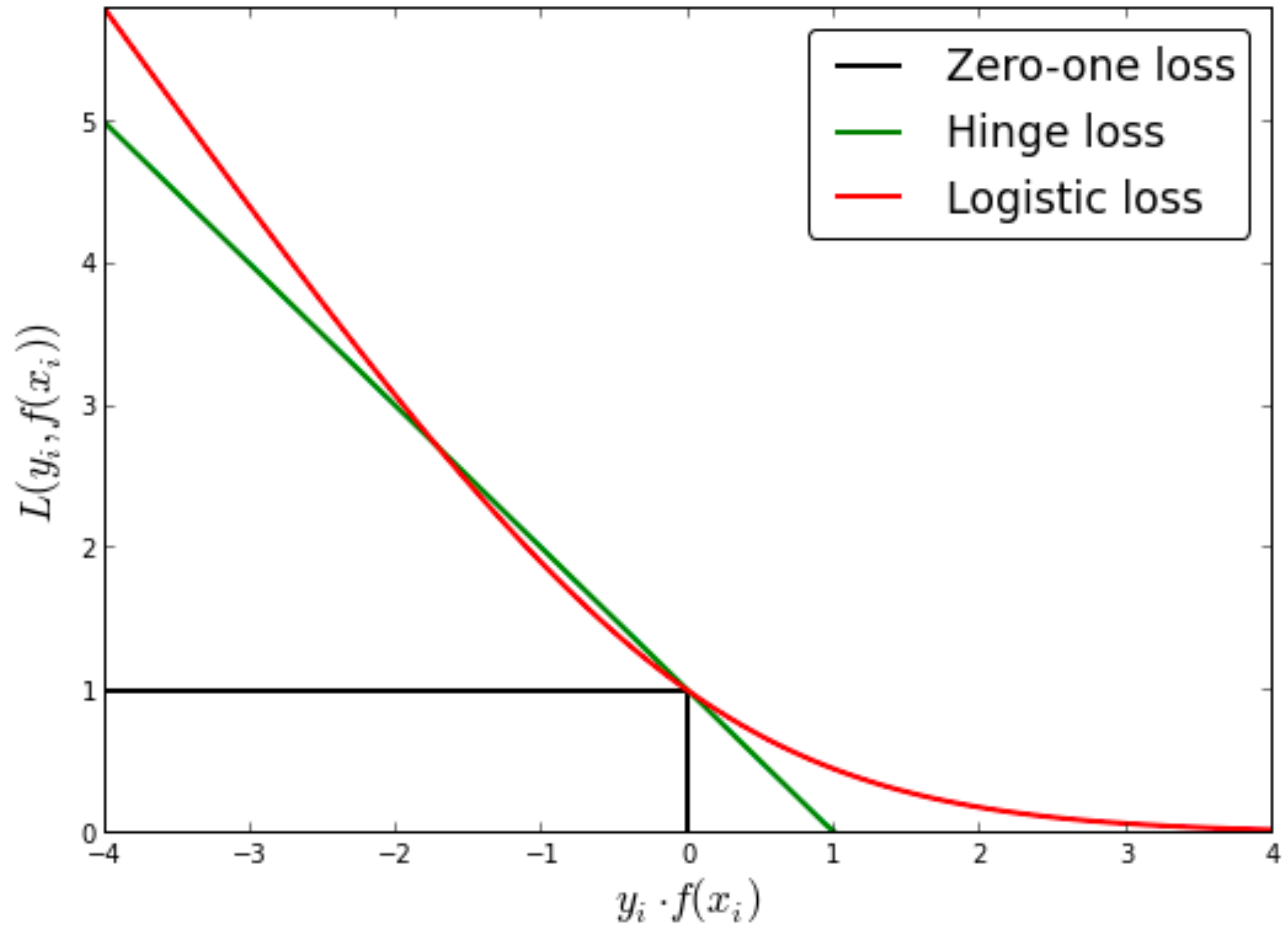
# LOGISTIC LOSS

$$P(y = 1 \,|\, x) = \text{sigmoid}(w^\top x) = \frac{1}{1 + \exp(-w^\top x)}$$

$$\text{sigmoid}(a) = \frac{1}{1 + \exp(-a)}$$

$$P(y = -1 \,|\, x) = 1 - \text{sigmoid}(w^\top x) = \frac{1}{1 + \exp(w^\top x)}$$

$$\ell(f(x), y) = \log\big(1 + \exp(-y\, f(x))\big)$$




Derivation based on probabilistic arguments, will discuss in next class

# LOGISTIC REGRESSION - SUMMARY

Predicts probability of label conditioned on input, allows uncertainty

**Input space:** $\mathscr{X} \subseteq \mathbb{R}^d$

**Output space:** $\mathscr{Y} = [0,1]$

**Hypothesis Class:** $\mathscr{F} := \{x \mapsto \text{sigmoid}(w^\top x + b) \,|\, w \in \mathbb{R}^d, b \in \mathbb{R}\}$

**Loss function:** $\ell(f(x), y) = \log\left(1 + \exp(-y\, f(x))\right)$