Deep Learning - Theory and Practice

IE 643
Lecture 3

August 6, 2024.

Perceptron

Perceptron

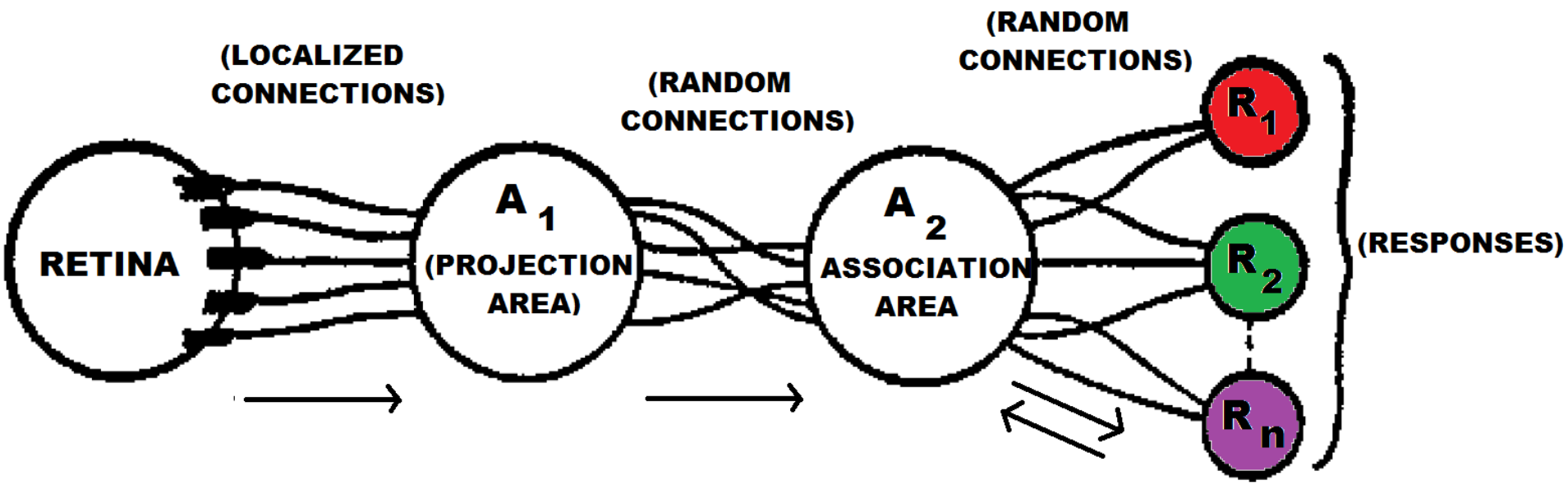
Psychological Review
Vol. 65, No. 6, 1958

THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN

F. ROSENBLATT

Cornell Aeronautical Laboratory

Perceptron

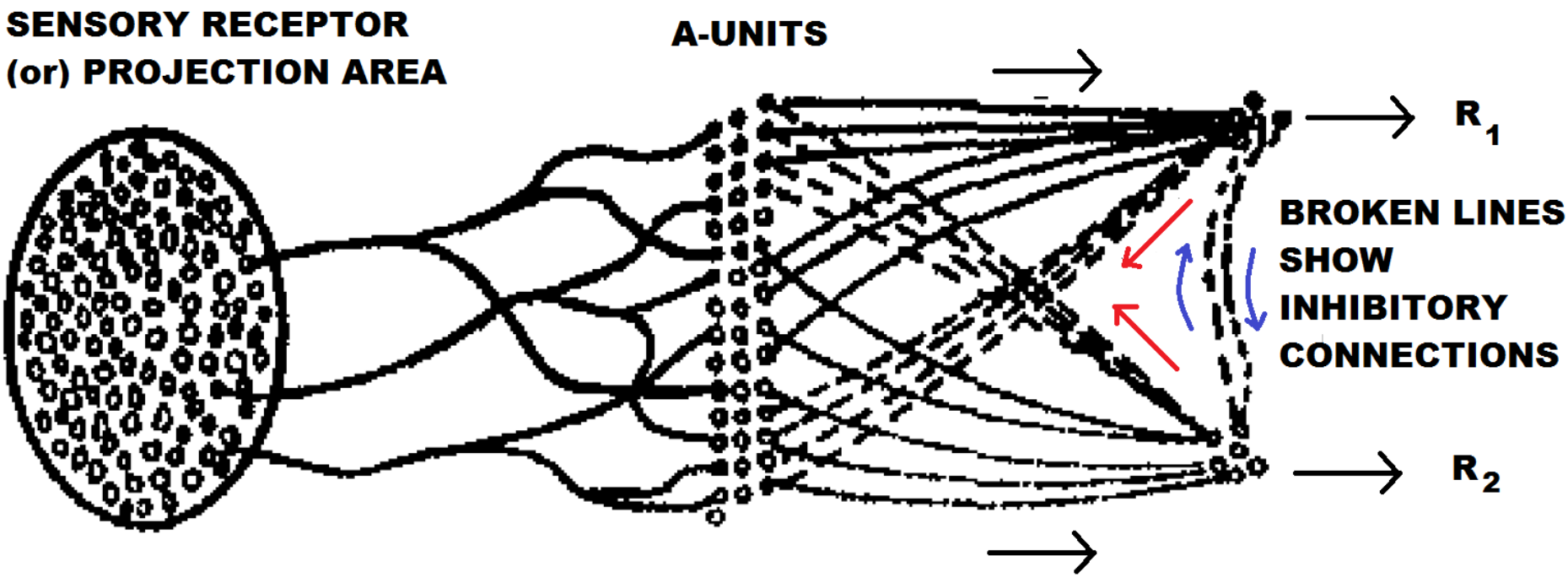

(LOCALIZED
CONNECTIONS)
(RANDOM
CONNECTIONS)
(RANDOM
CONNECTIONS)
RETINA
A 1
(PROJECTION
AREA)
A 2
ASSOCIATION
AREA
R 1
R 2
R n
(RESPONSES)

Perceptron

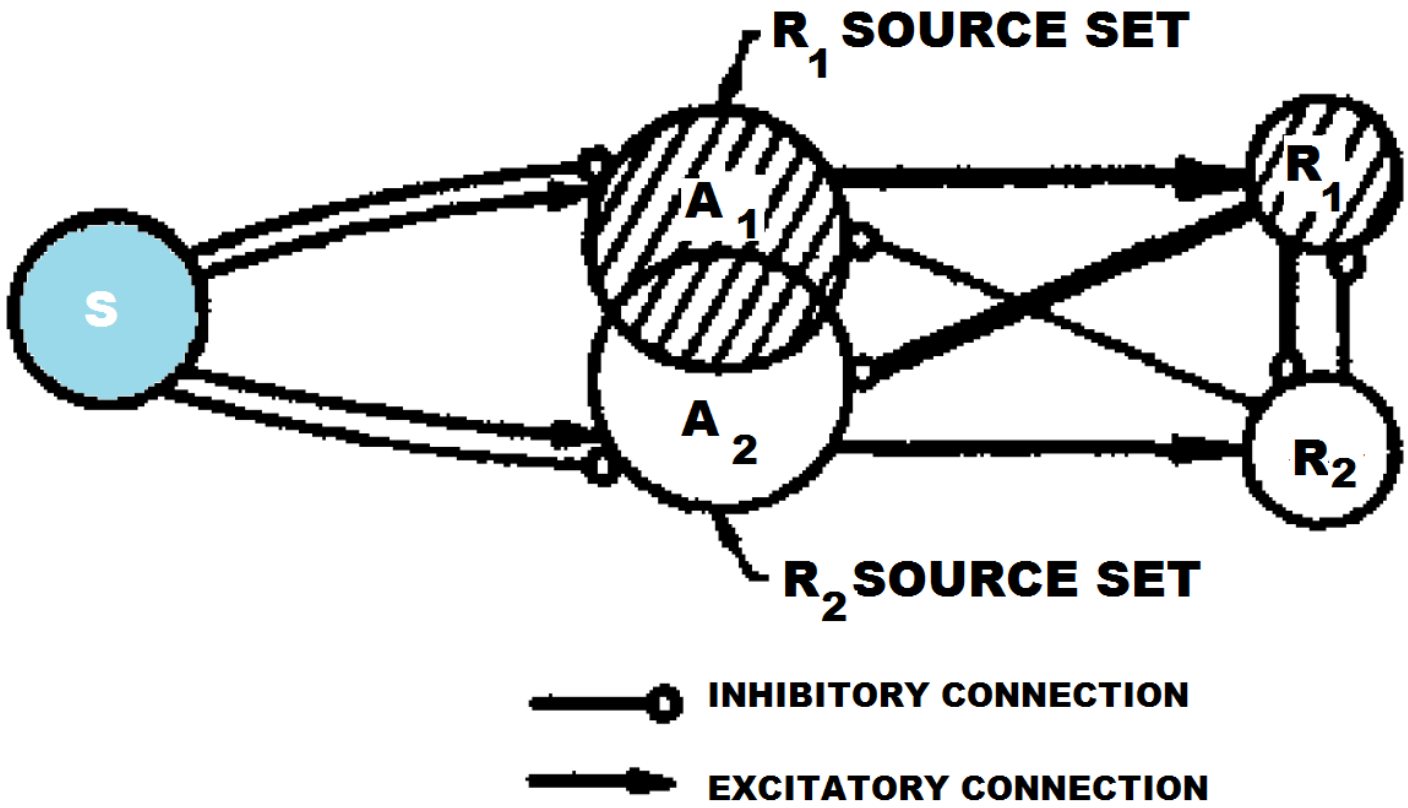
Key Assumptions

- Stimuli which are similar will tend to form pathways to same sets of response cells.
- Stimuli which are dissimilar will tend to form pathways to different sets of response cells.
- Application of positive or negative reinforcements may facilitate or hinder the formation of connections.
- **Similarity** of stimuli is a dynamically evolving attribute.

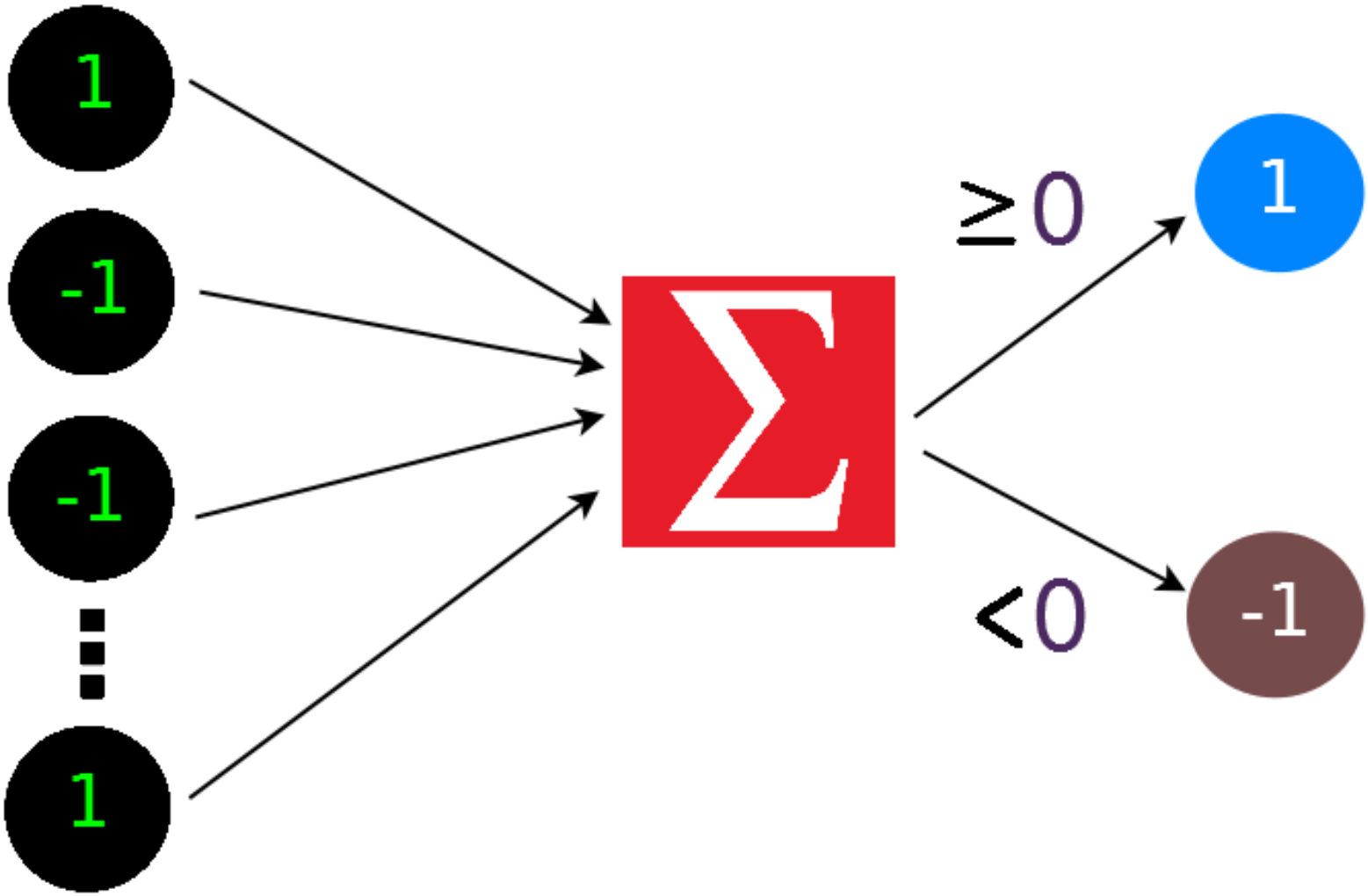
Perceptron


SENSORY RECEPTOR
(or) PROJECTION AREA
A-UNITS
R1
BROKEN LINES
SHOW
INHIBITORY
CONNECTIONS
R2

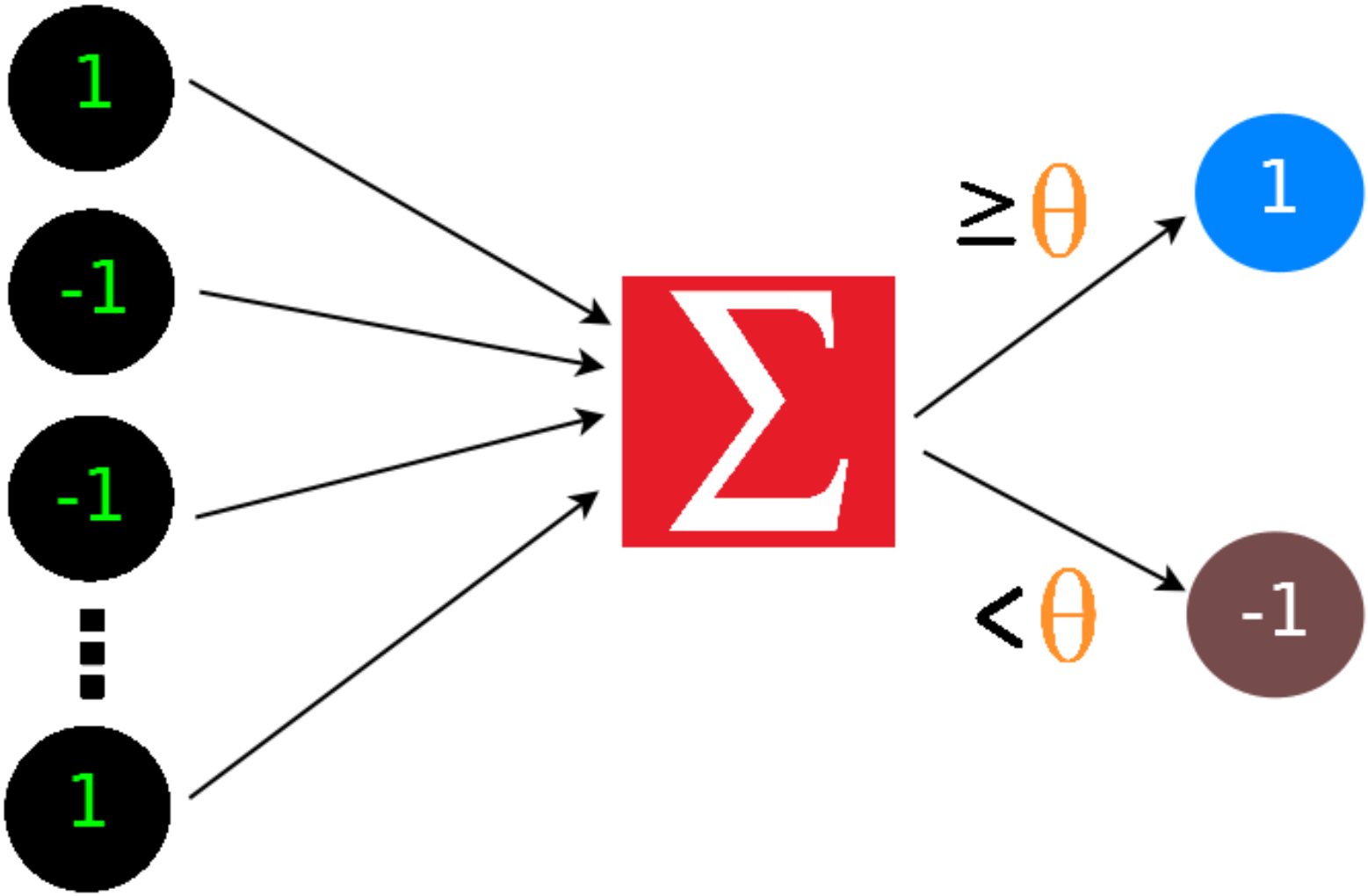
Perceptron


R1 SOURCE SET
S
A1
A2
R1
R2
R2 SOURCE SET
INHIBITORY CONNECTION
EXCITATORY CONNECTION

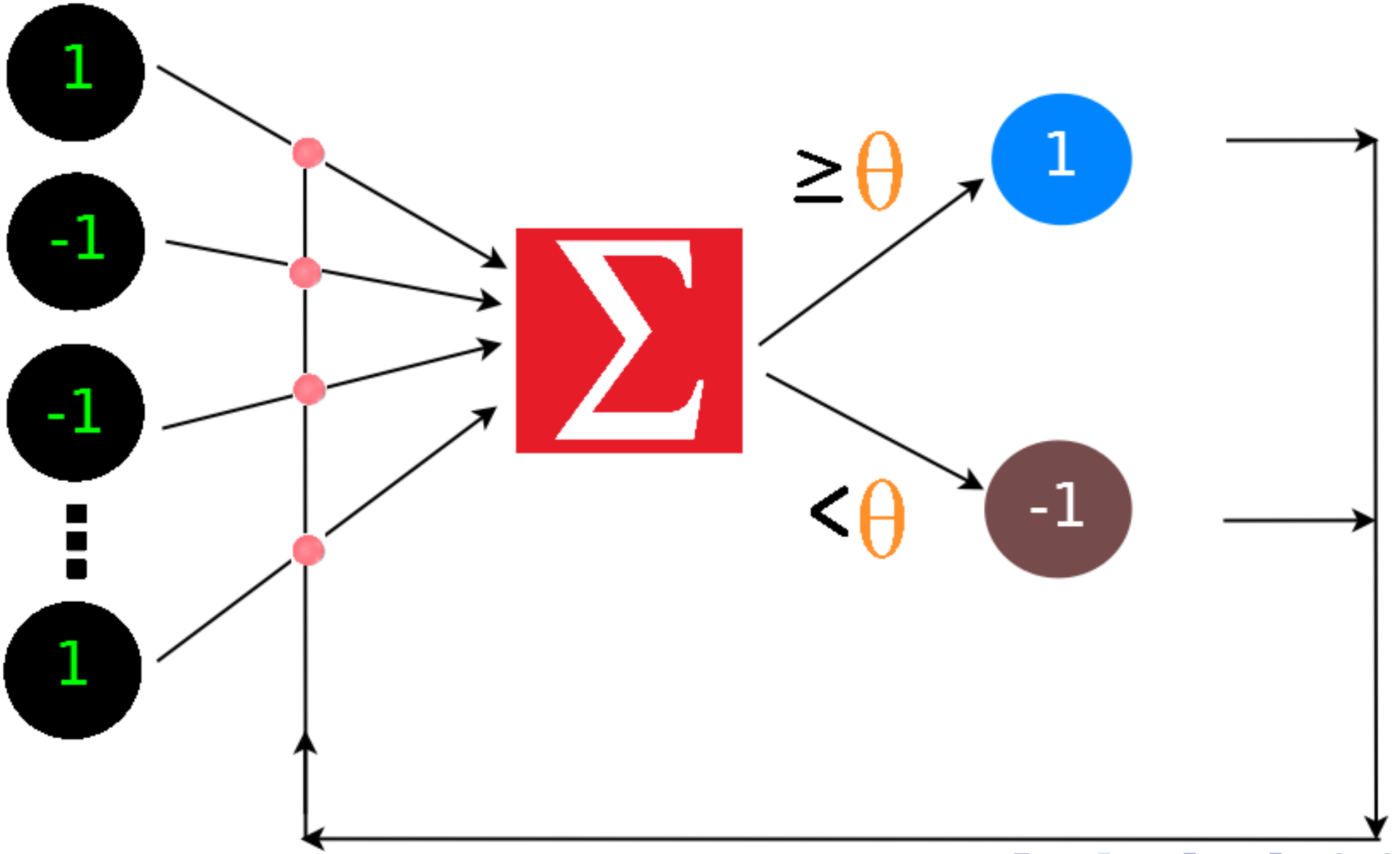
Perceptron



Perceptron



Perceptron



Nature of Machine Learning Tasks

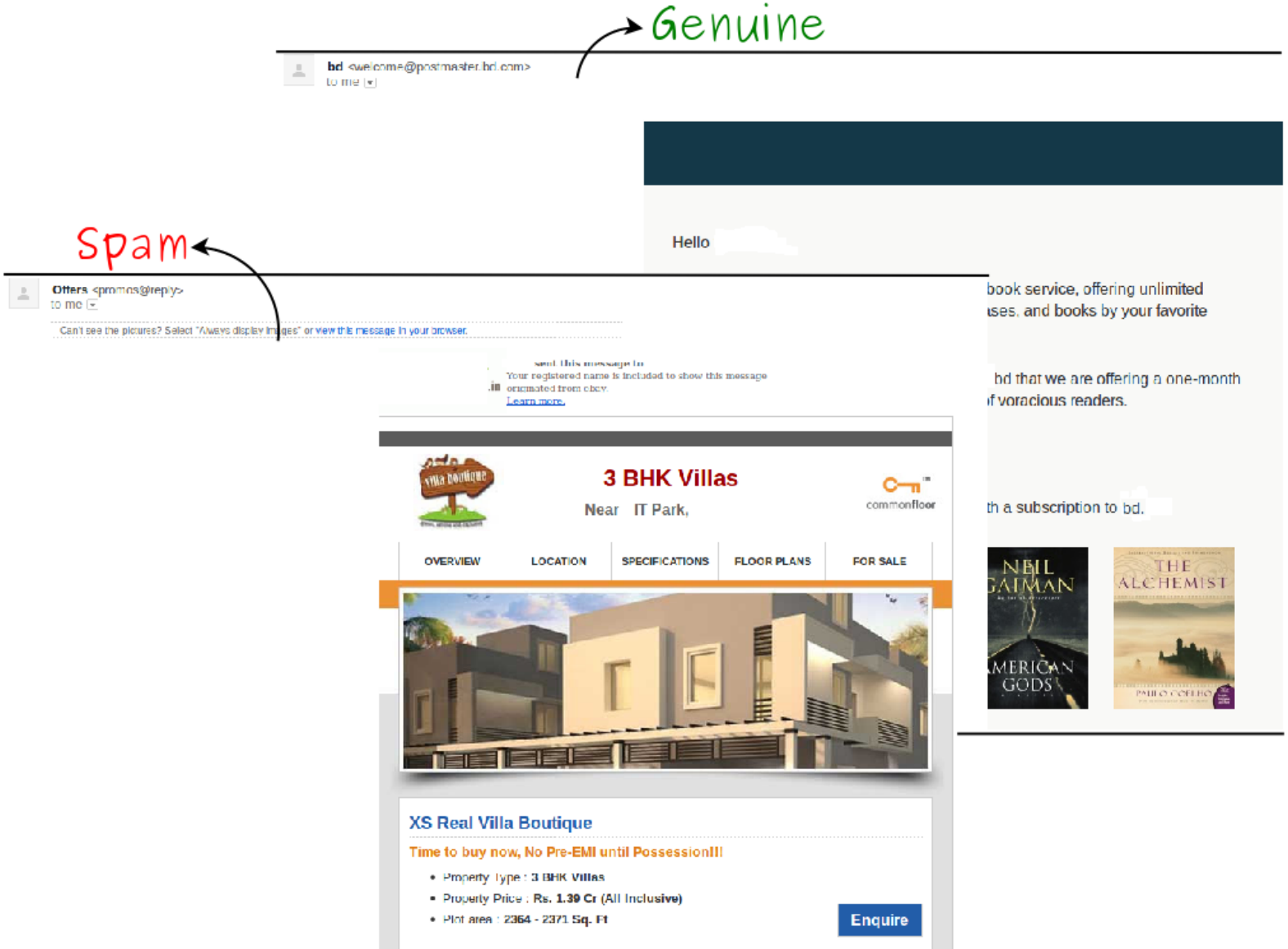
Nature of Machine Learning Tasks

- Supervised Learning
  - Inputs and corresponding outputs are known during learning
  - *e.g.* Classification (Binary, Multi-class, Multi-label)
- Unsupervised Learning
  - Input objects are not generally labeled
  - *e.g.* Clustering, Principal-component Analysis
- Semi-supervised Learning
  - learning from a few labeled data
- Transfer Learning
  - transferring a learned model from task T_1 to T_2
  - *e.g.* transfer from image captioning to video captioning

Supervised Machine Learning

Binary Classification

Recall: e-mail Spam Classification



Binary Classification

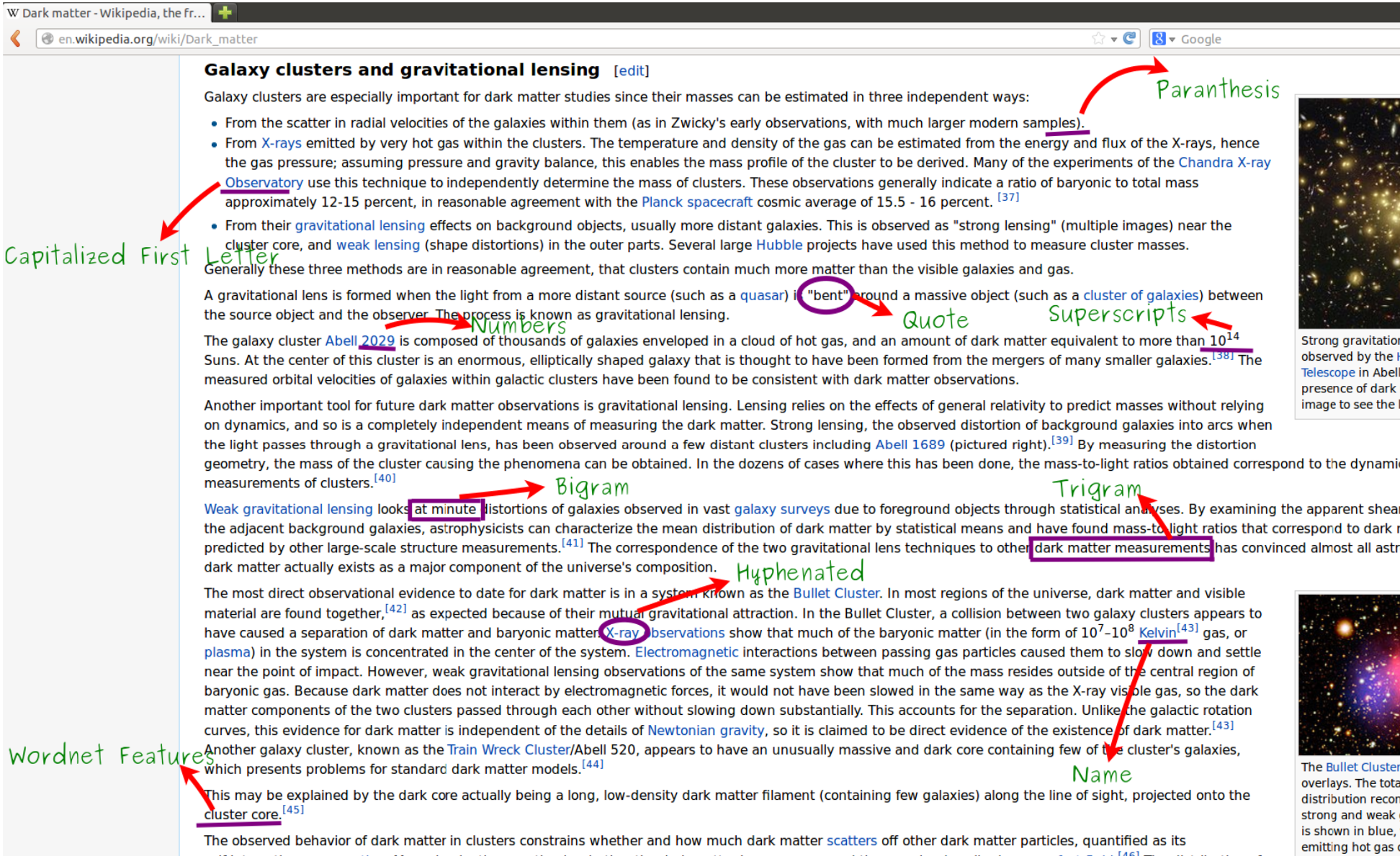
- **Input:** e-mail messages
- **Output:** Spam/Not spam

Binary Classification

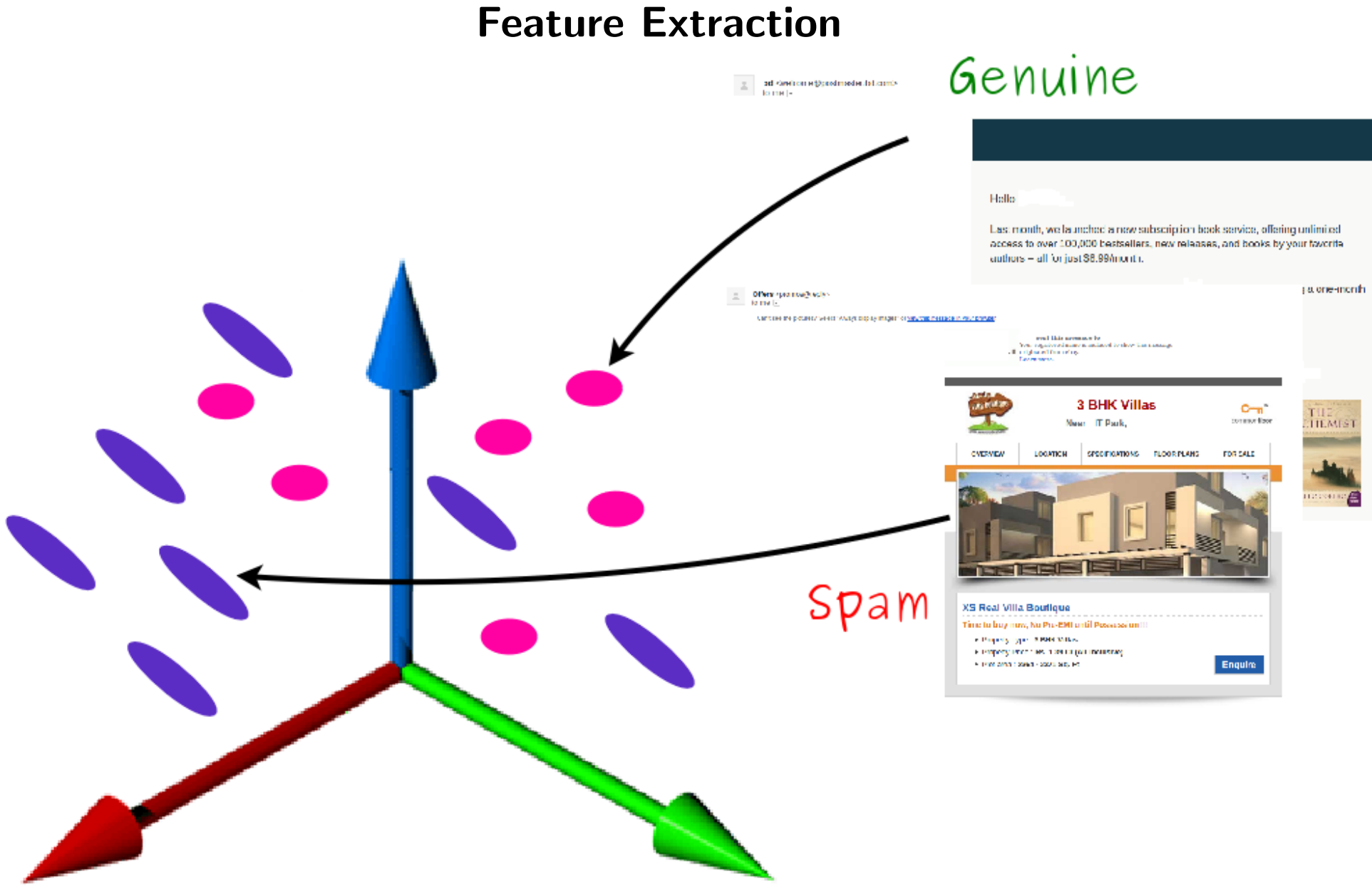
- **Input:** e-mail messages $\Longrightarrow$ some feature space
- **Output:** Spam/Not spam $\Longrightarrow$ $\{+1, -1\}$
- Generally many input/output pairs are given for learning the machine learning model.

Binary Classification

Feature Extraction



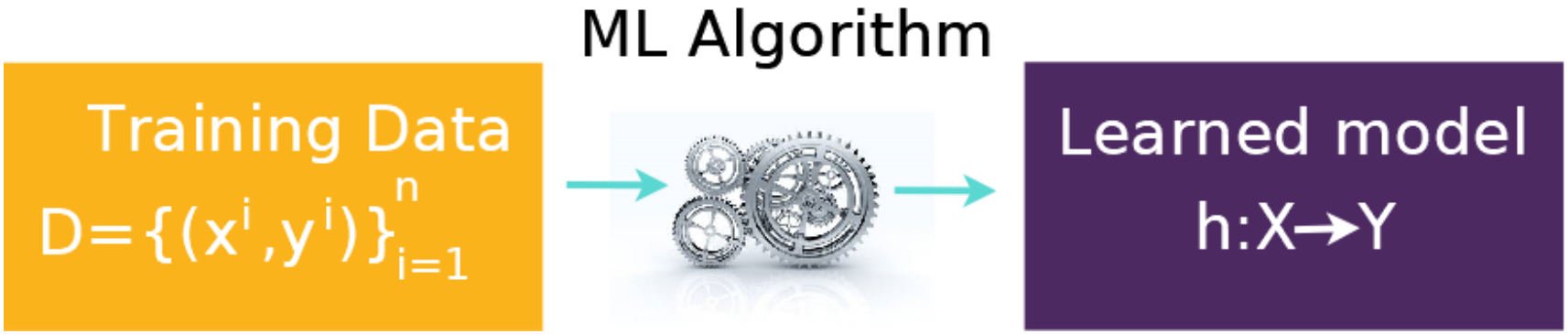
Binary Classification



Binary Classification

- **Input:** e-mail messages $\implies$ some feature space $\subseteq \mathbb{R}^d$.
  - $x \in \mathcal{X} \subseteq \mathbb{R}^d$
- **Output:** Spam/Not spam $\implies \{+1, -1\}$
  - $y \in \mathcal{Y} = \{+1, -1\}$
- Generally n input/output pairs $\{(x^i, y^i)\}_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$ are given for learning the machine learning model.
- $D = \{(x^i, y^i)\}_{i=1}^n$ called the training data.

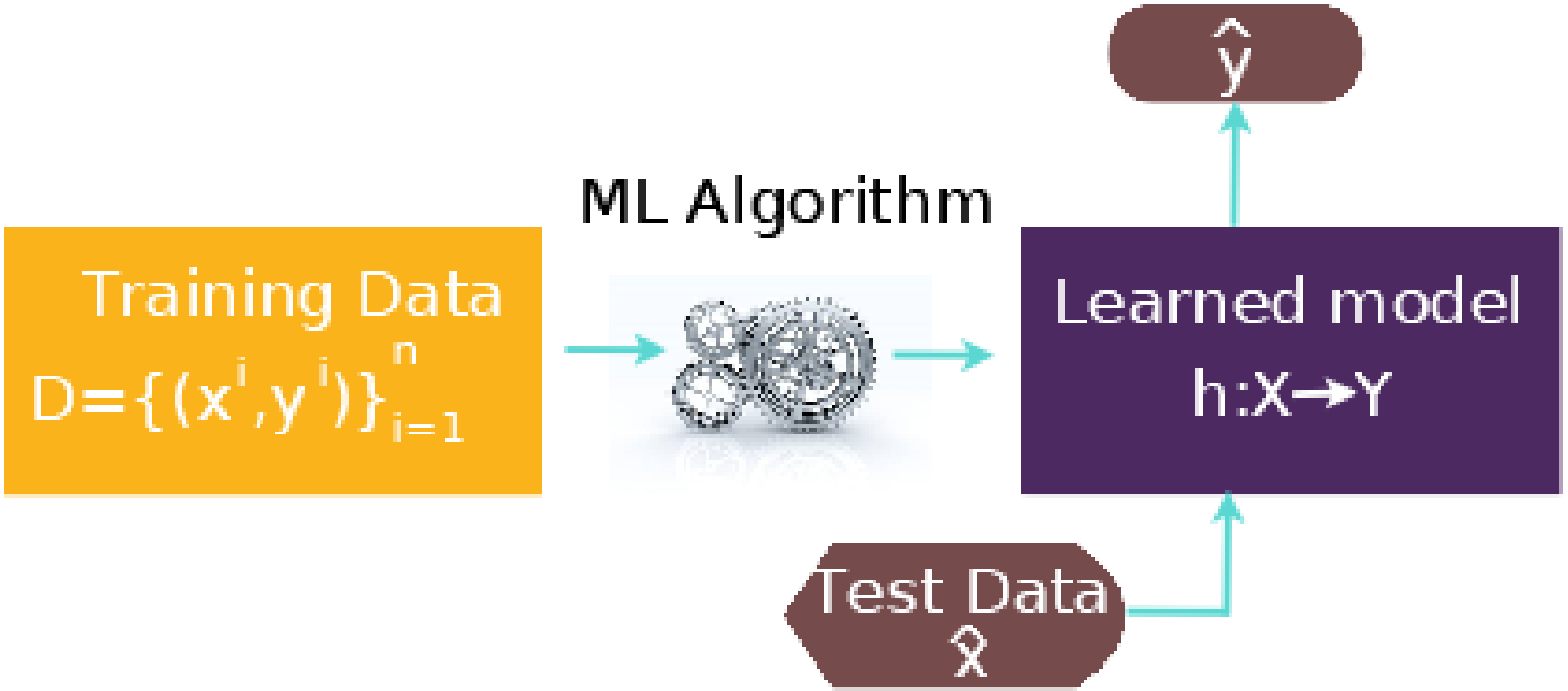
General Nature of a Supervised Machine Learning Task



Training

- **Input:** Training data $D = \{(x^i, y^i)\}_{i=1}^{n}$
- **Aim:** Learn a model $h : \mathcal{X} \rightarrow \mathcal{Y}$

General Nature of a Supervised Machine Learning Task



Training

- **Input:** Training data $D = \{(x^i, y^i)\}_{i=1}^{n}$
- **Aim:** Learn a model $h : \mathcal{X} \rightarrow \mathcal{Y}$

Testing

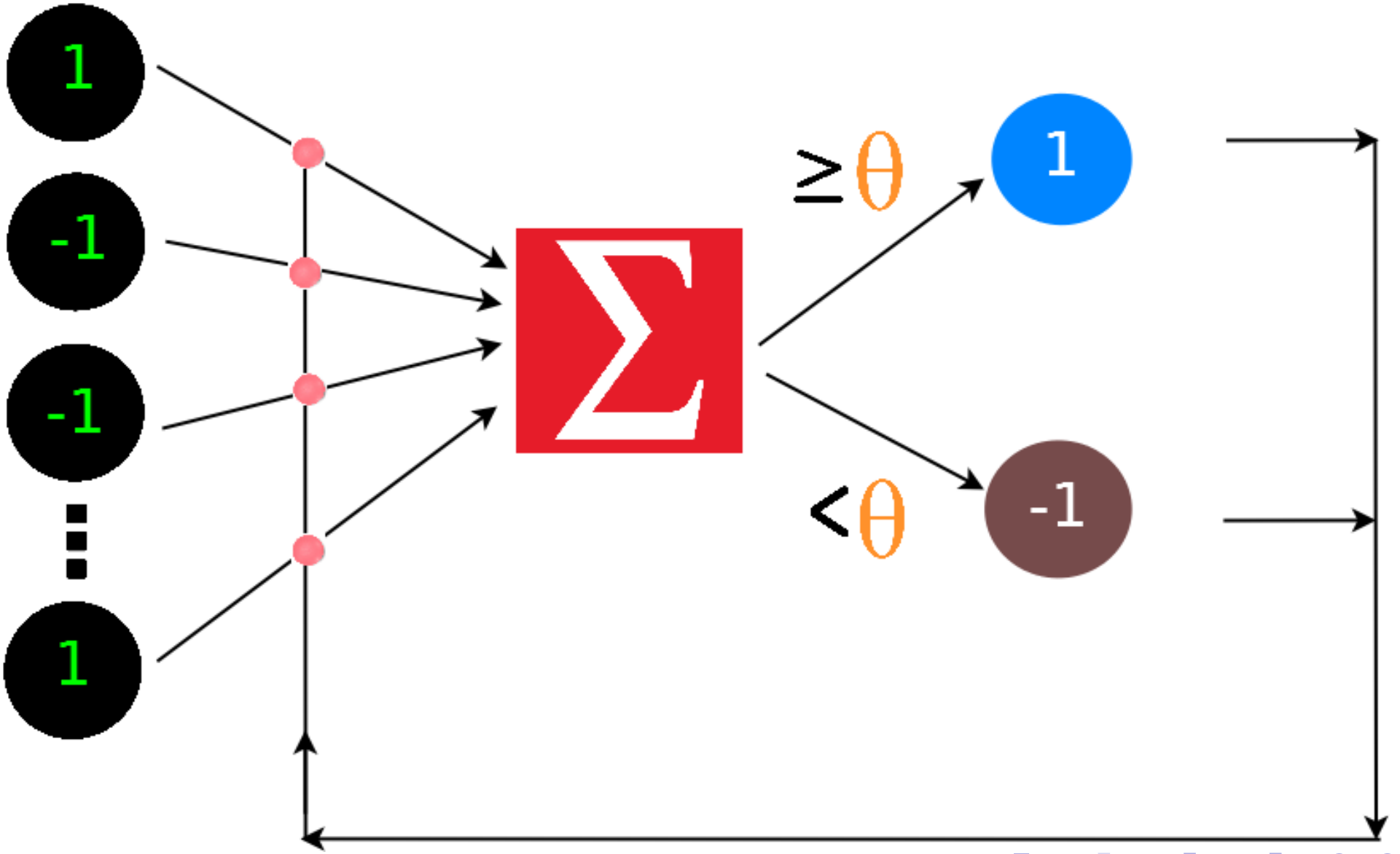
- Given $\hat{x}$, predict $\hat{y} = h(\hat{x})$

A sample data set for email classification

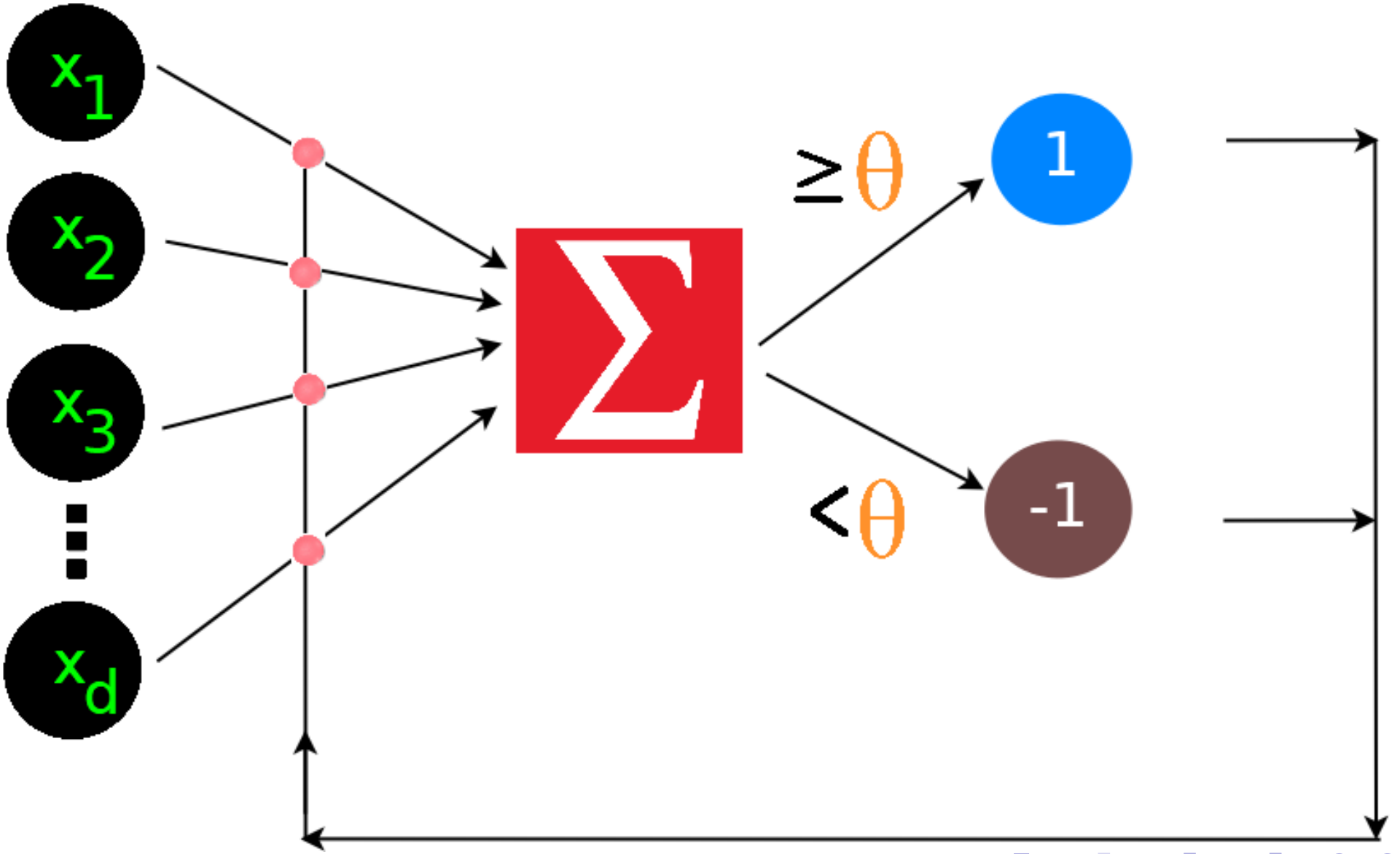
spam	to_multiple	from	cc	sent_email	image	attach	dollar	winner	inherit	viagra	password	num_char
0	0	1	0	0	0	0	0	no	0	0	0	11.37
0	0	1	0	0	0	0	0	no	0	0	0	10.504
0	0	1	0	0	0	0	0	no	0	0	0	13.256
0	0	1	0	0	0	0	0	no	0	0	2	1.231
1	0	1	0	0	0	0	0	no	0	0	0	5.108
1	0	1	4	0	0	2	0	no	0	0	0	0.631
1	0	1	0	0	0	0	1	yes	0	0	0	0.963
1	0	1	0	0	0	0	2	no	0	0	0	2.182

Perceptron and Learning

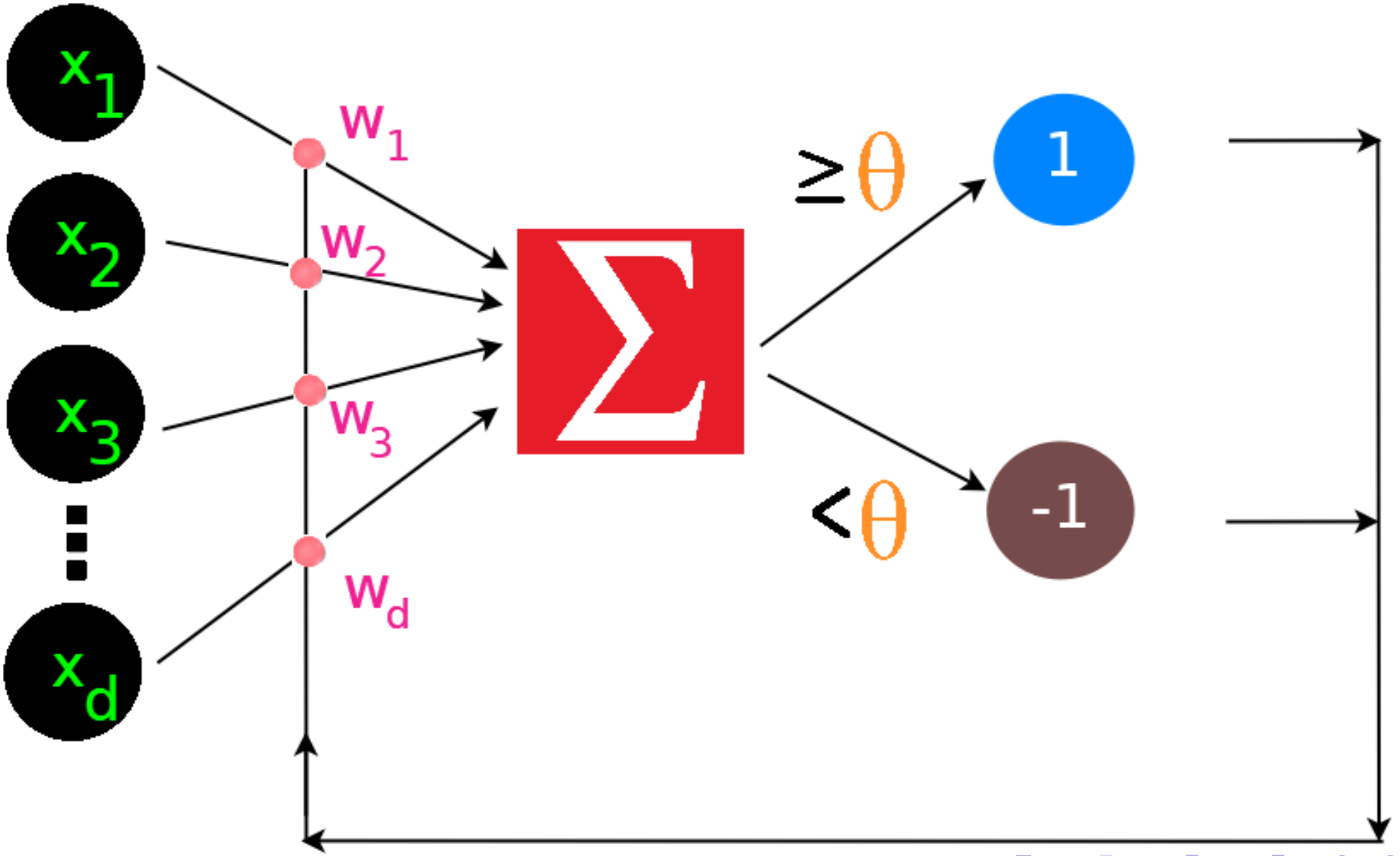
Perceptron



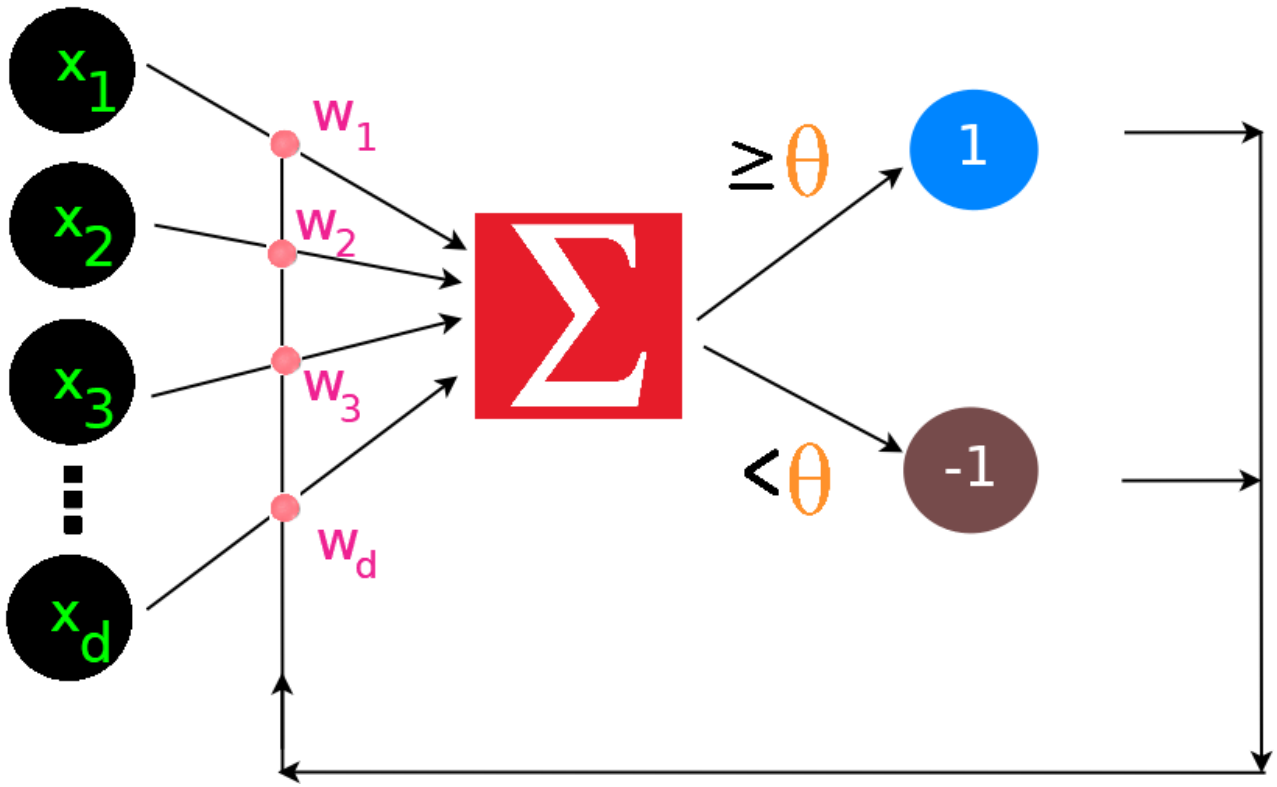
Perceptron



Perceptron

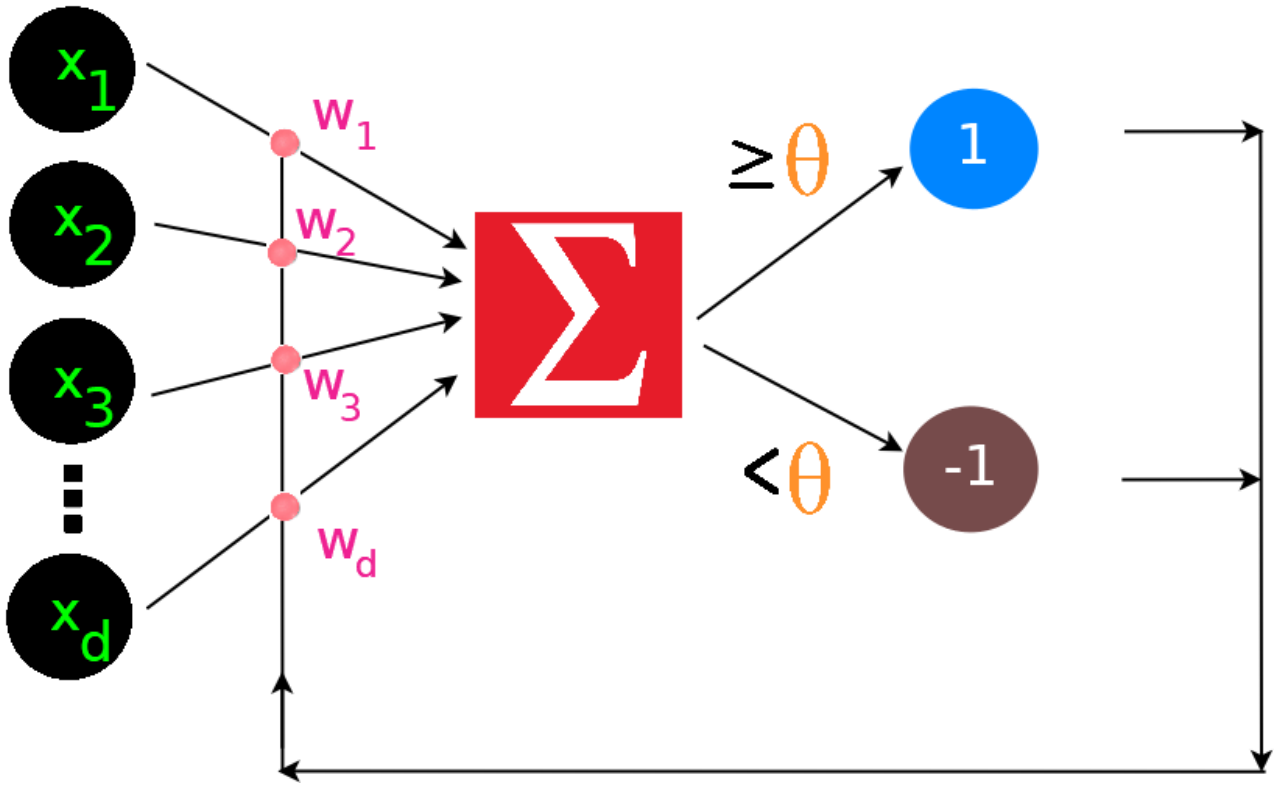


Perceptron



- Input is of the form $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$.
- We associate weights $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$ to the connections.
- **Prediction Rule:**
  - $\sum_{i=1}^{d} w_i x_i \geq \theta \implies$ predict 1.
  - $\sum_{i=1}^{d} w_i x_i < \theta \implies$ predict -1.

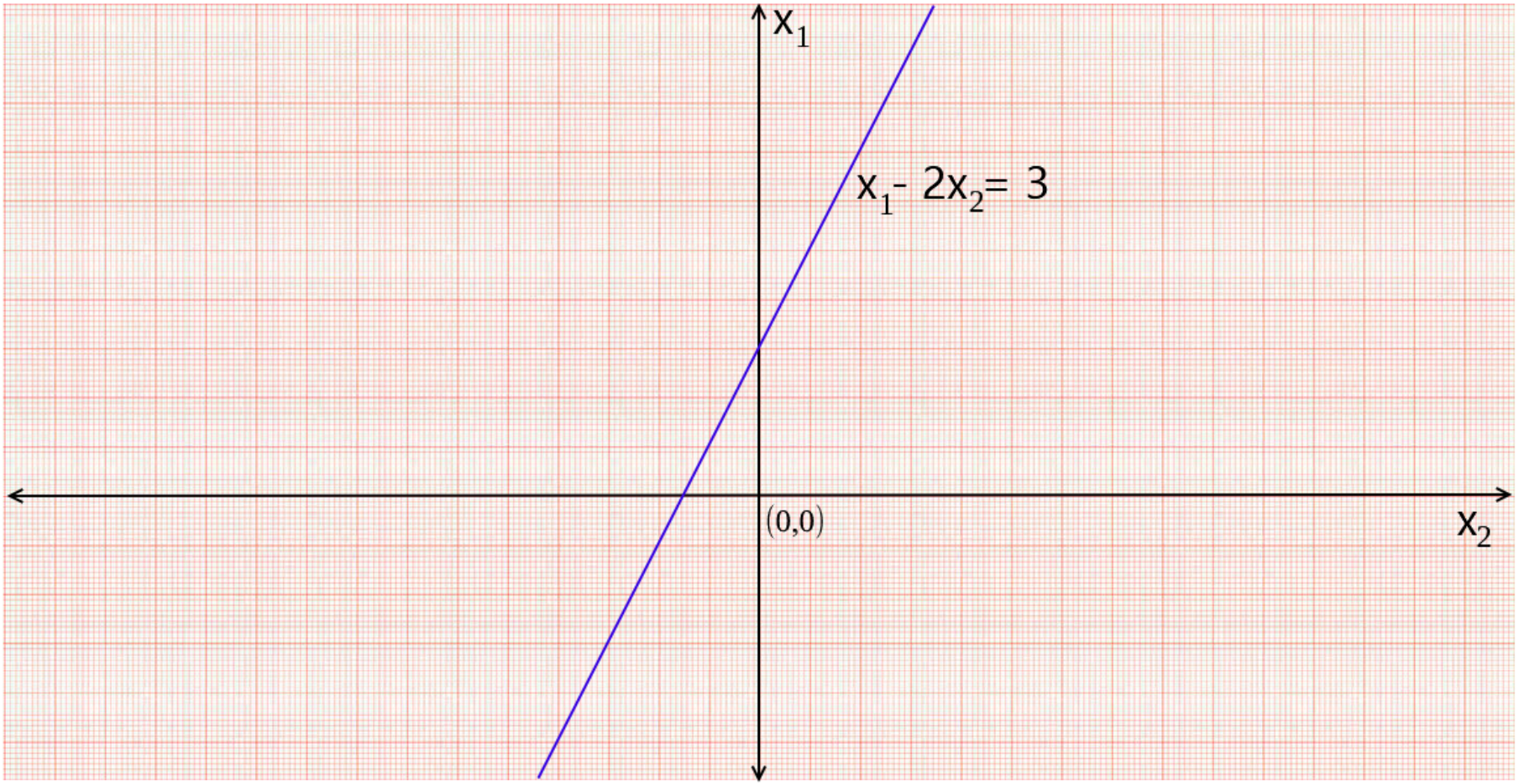
Perceptron



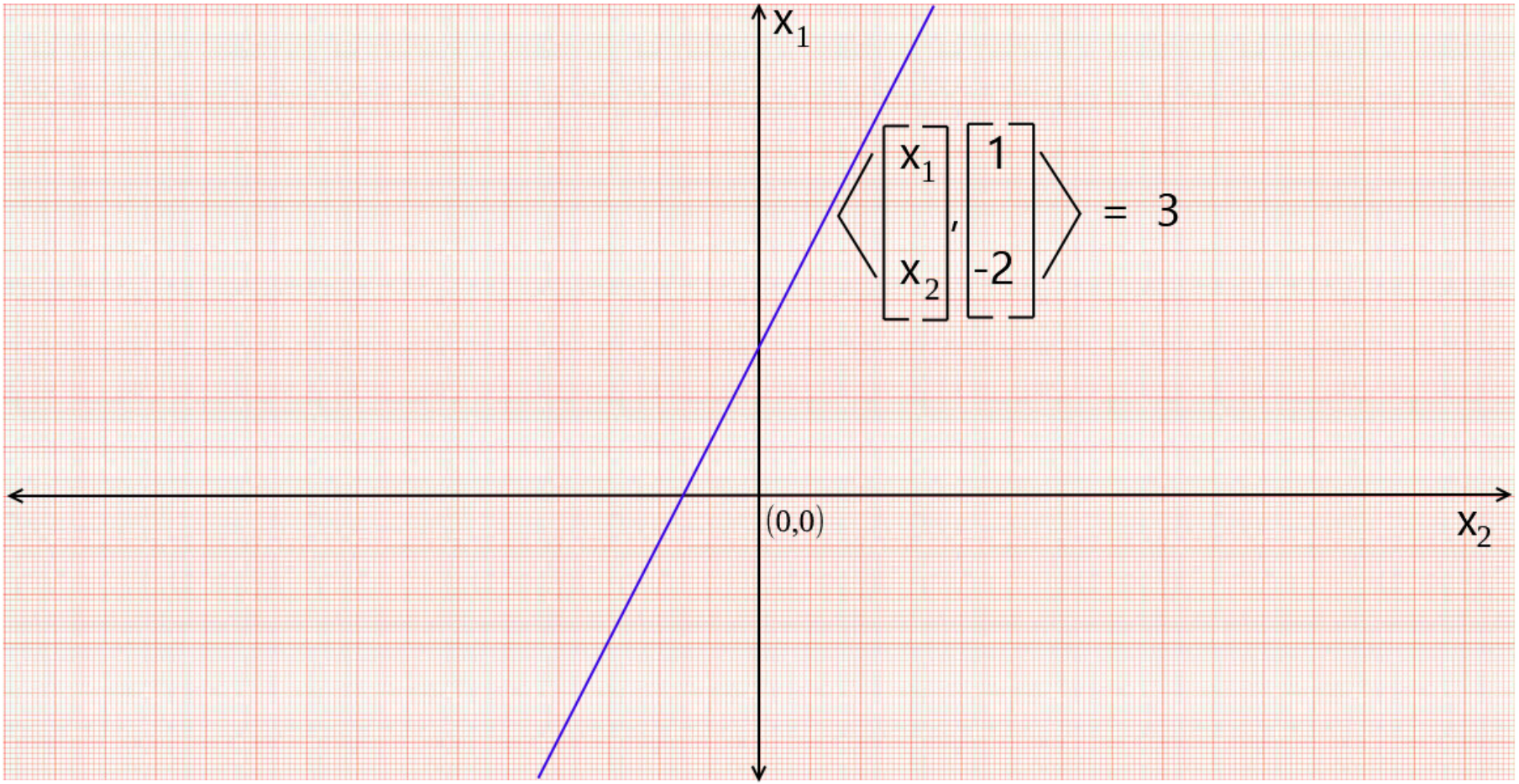
- Input is of the form $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$.
- We associate weights $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$ to the connections.
- **Prediction Rule:**
  - $\langle w, x \rangle \geq \theta \implies$ predict 1.
  - $\langle w, x \rangle < \theta \implies$ predict -1.

Note: $\langle w, x \rangle = \sum_{i=1}^{d} w_i x_i$ denotes inner product between w and x.

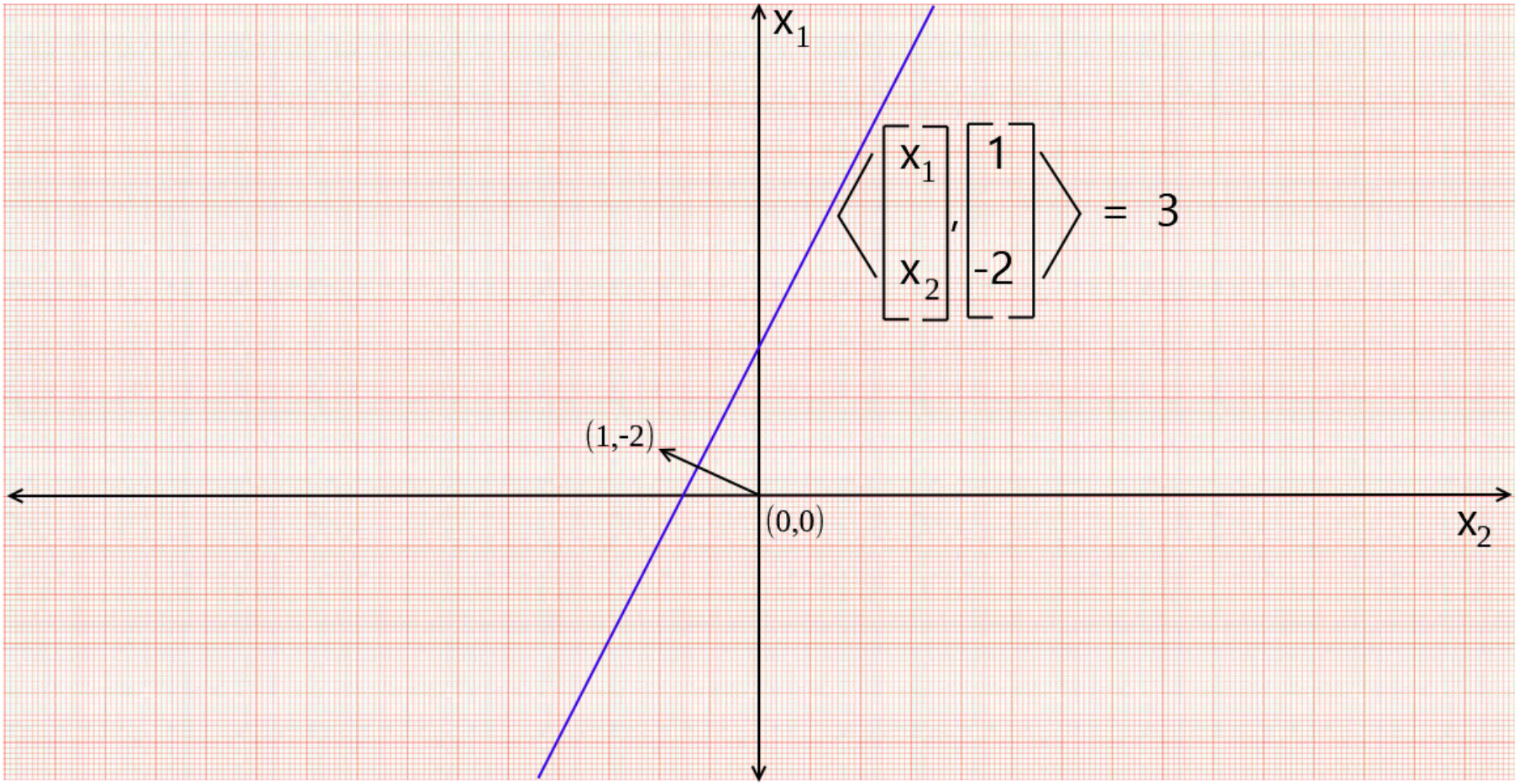
Perceptron - Geometry



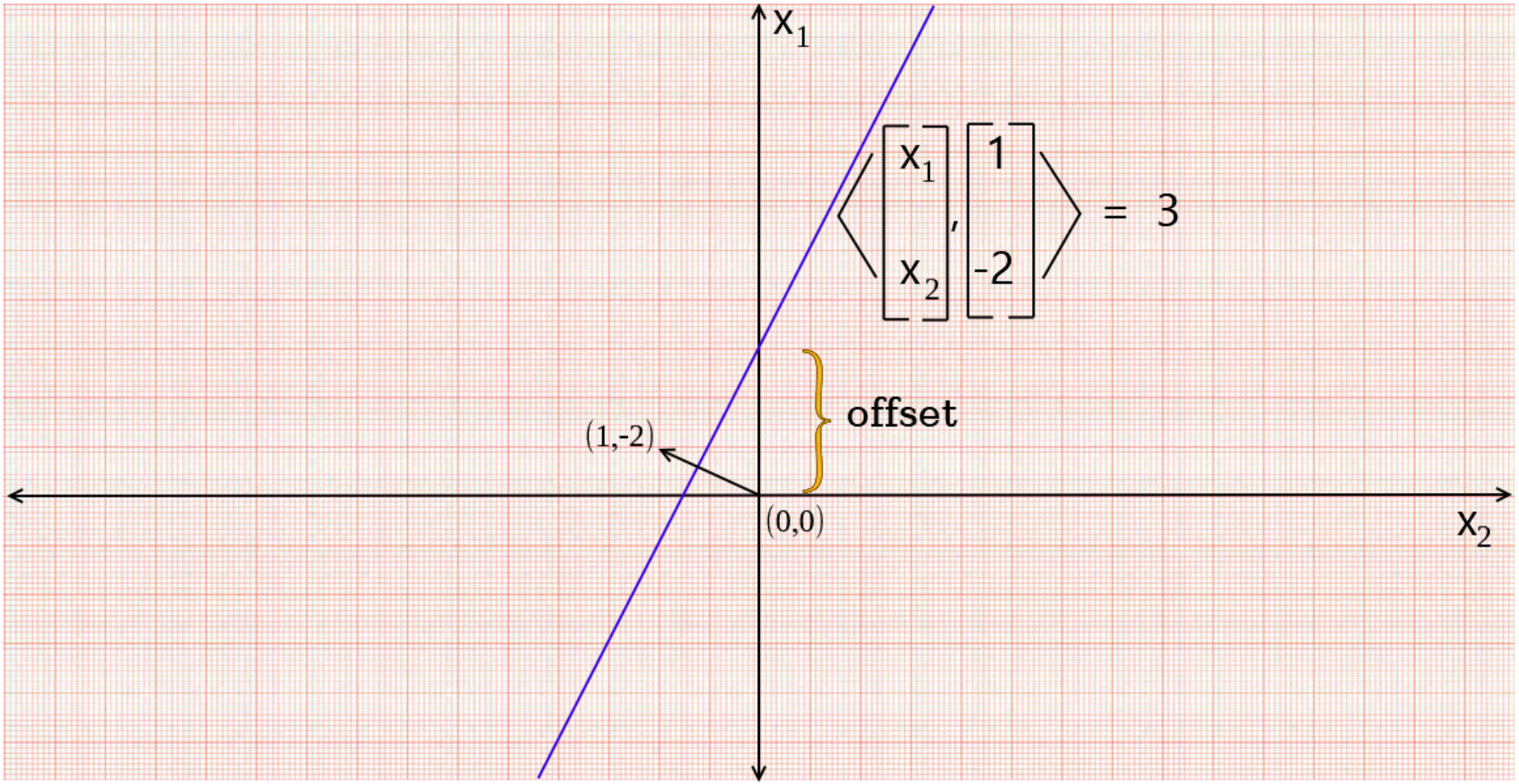
Perceptron - Geometry



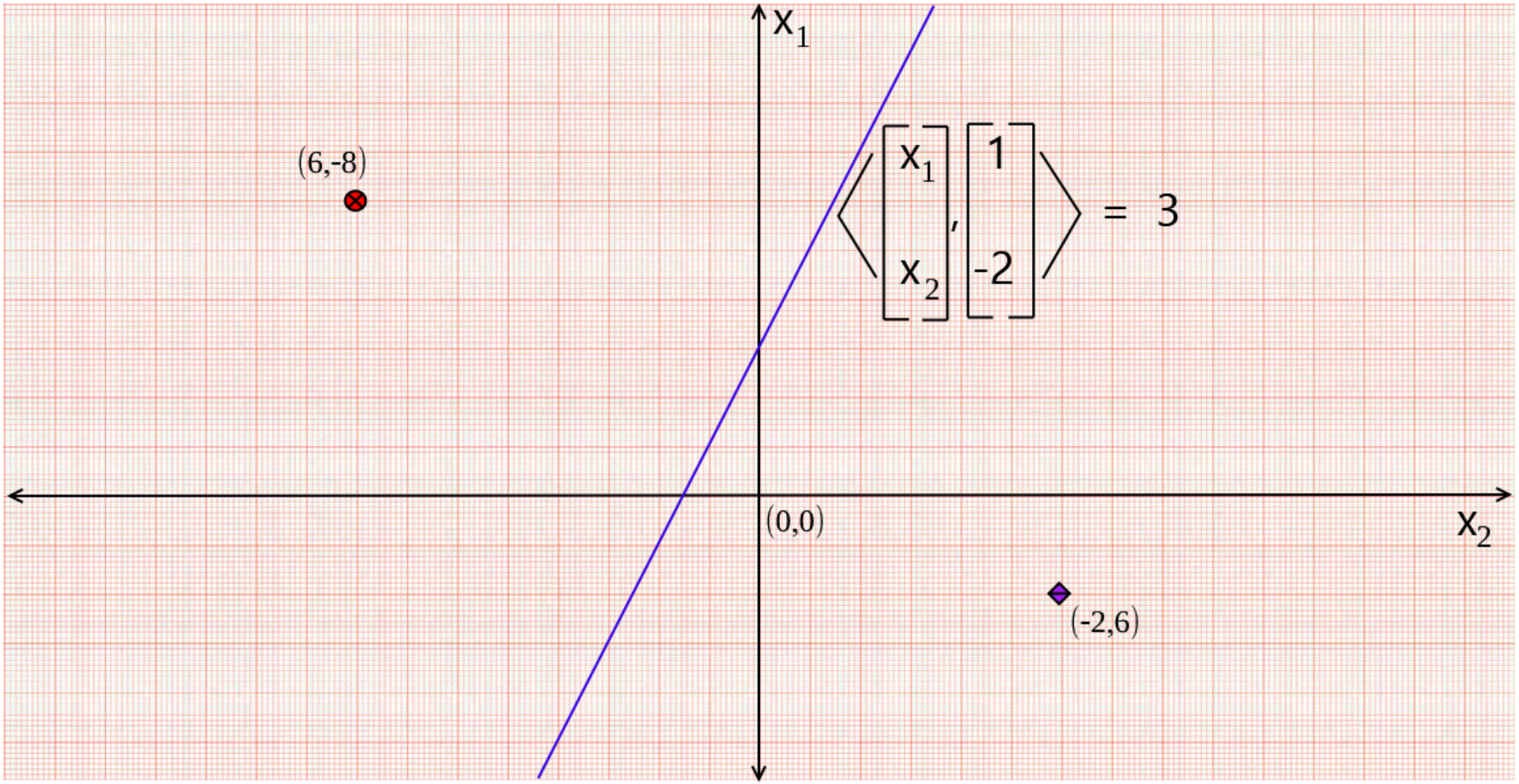
Perceptron - Geometry



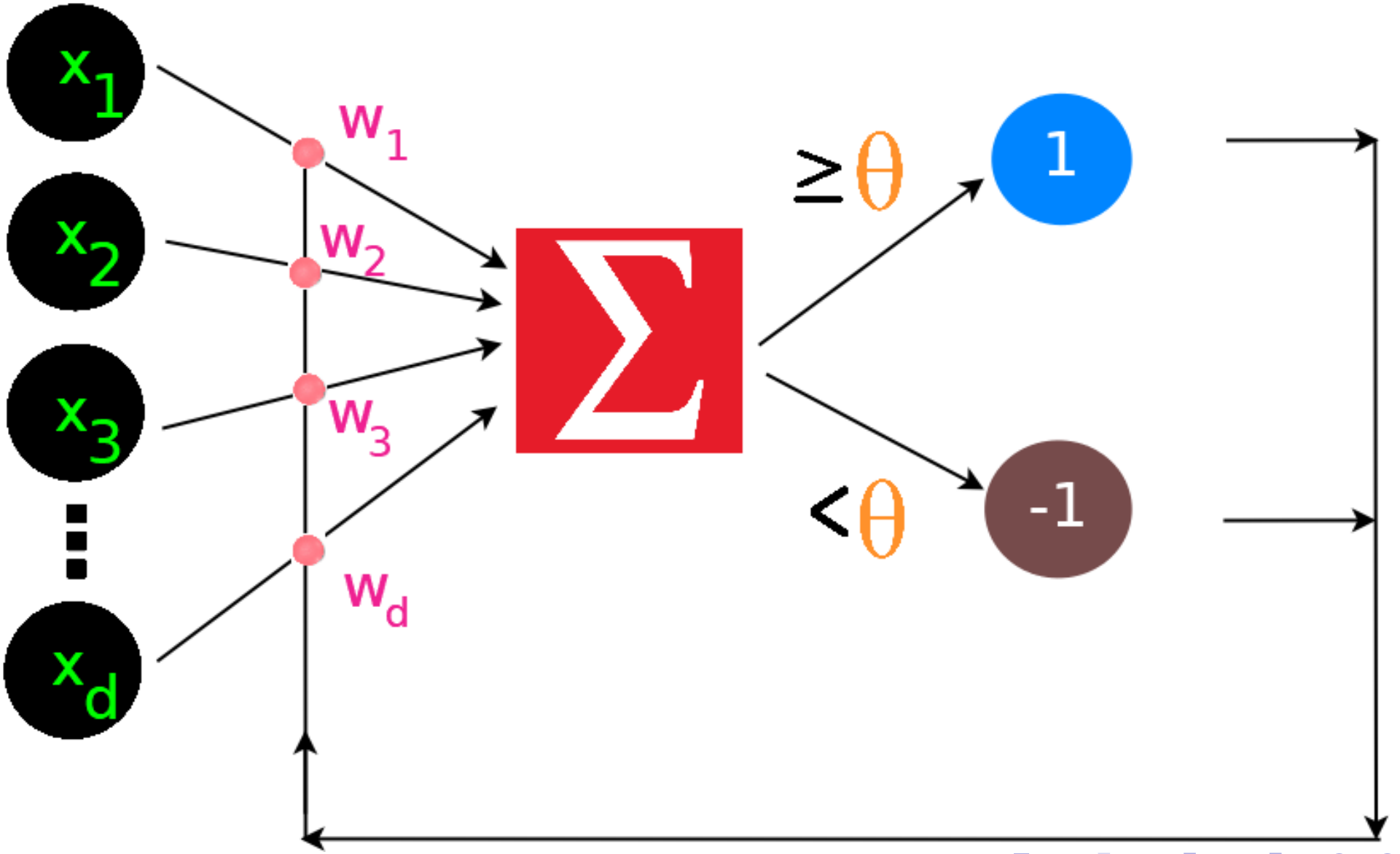
Perceptron - Geometry



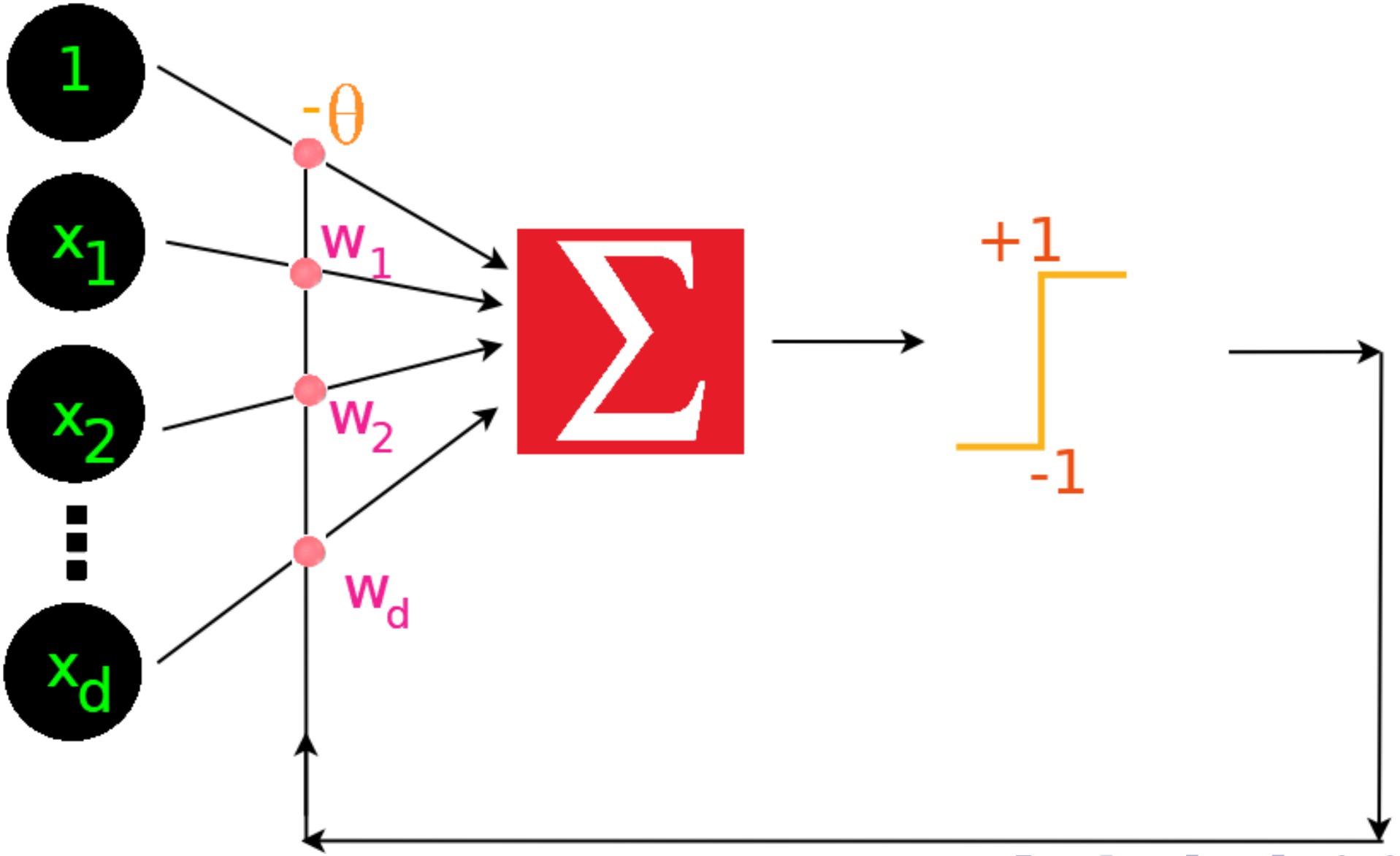
Perceptron - Geometry



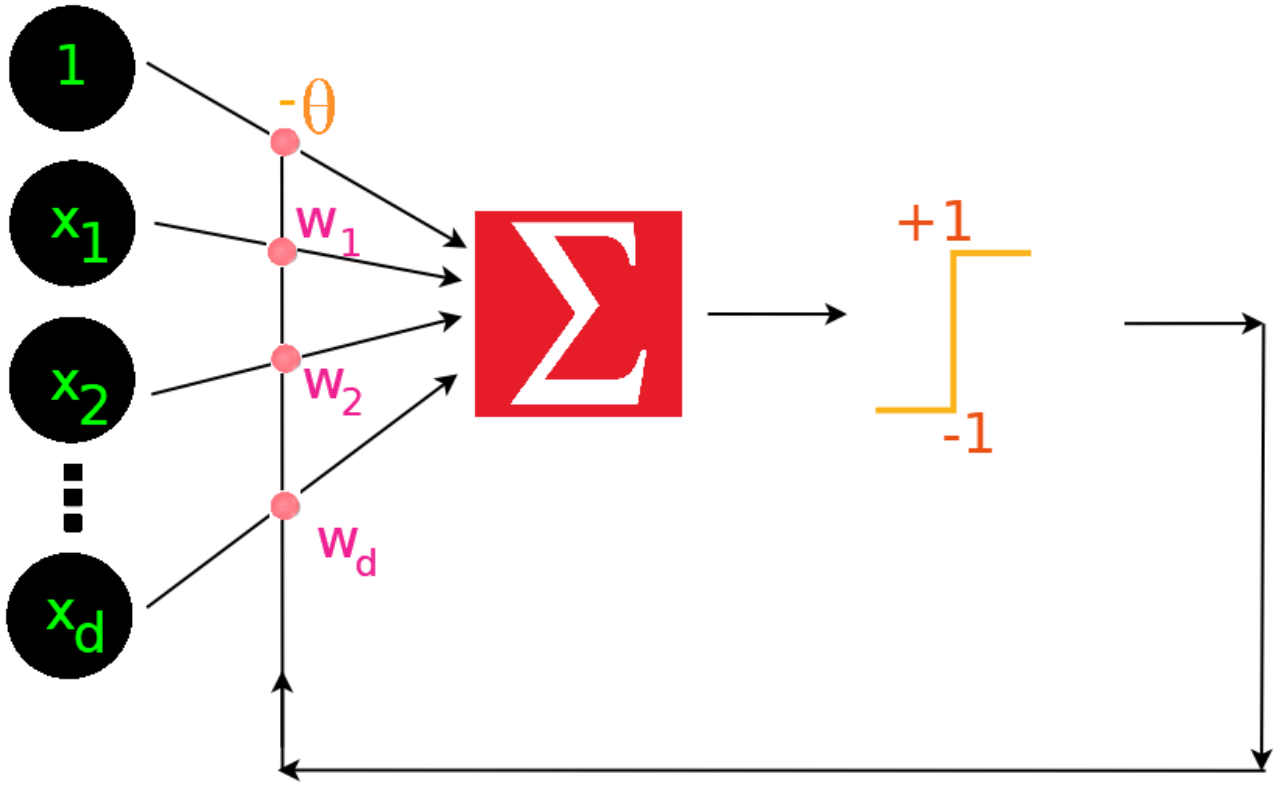
Perceptron



Perceptron

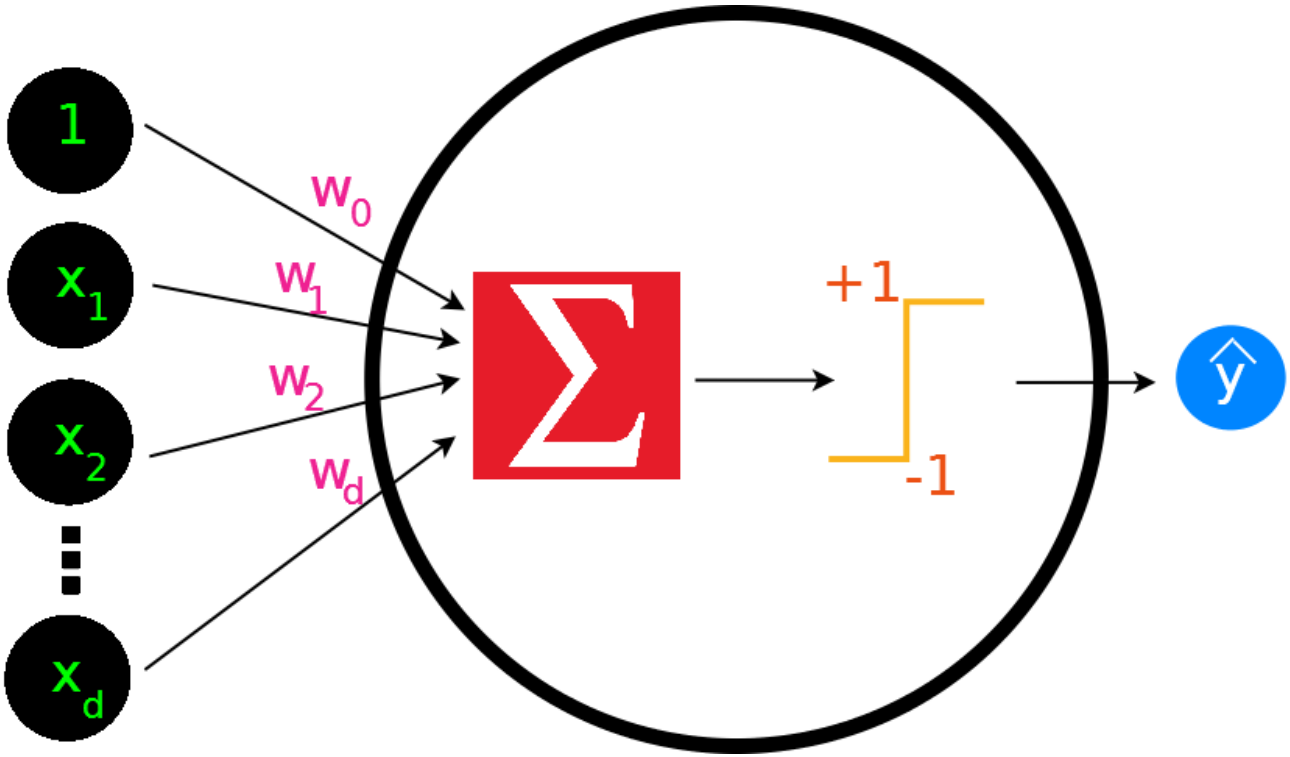


Perceptron



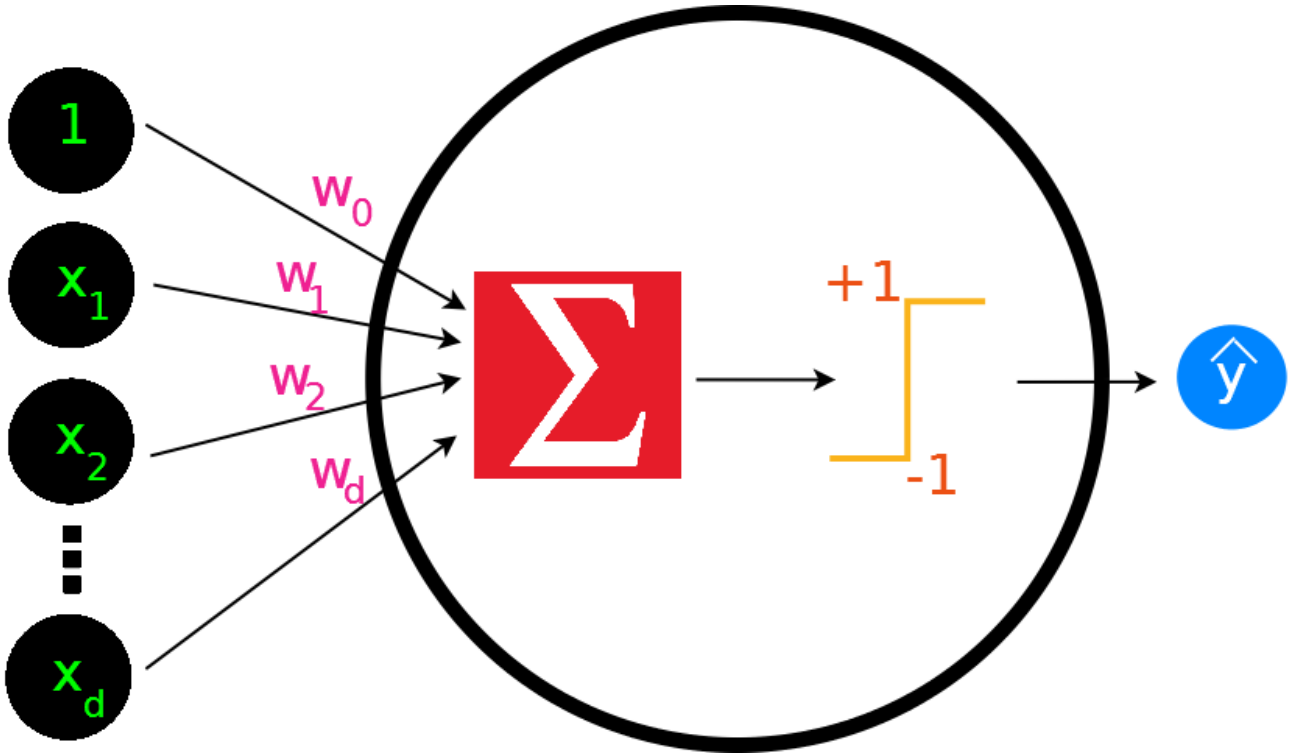
- Input is of the form $\tilde{x} = (1, x) = (1, x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d+1}$.
- We associate weights $\tilde{w} = (-\theta, w) = (-\theta, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}$ to the connections.
- **Prediction Rule:**
  - $\langle w, x \rangle - \theta \geq 0 \implies$ predict 1.
  - $\langle w, x \rangle - \theta < 0 \implies$ predict -1.

Perceptron - Data Perspective



- **Actual Input:** $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$
- **Transformed Input:** $\tilde{x} = (1, x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d+1}$
- **Actual Weights:** $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d, \theta \in \mathbb{R}$
- **Transformed Weights:** $\tilde{w} = (w_0, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}$

Perceptron - Data Perspective



Equivalently we might use:

- **Transformed Input:** $\tilde{x} = (x_1, x_2, \ldots, x_d, 1) \in \mathbb{R}^{d+1}$
- **Transformed Weights:** $\tilde{w} = (w_1, w_2, \ldots, w_d, w_0) \in \mathbb{R}^{d+1}$

Perceptron - Data Perspective

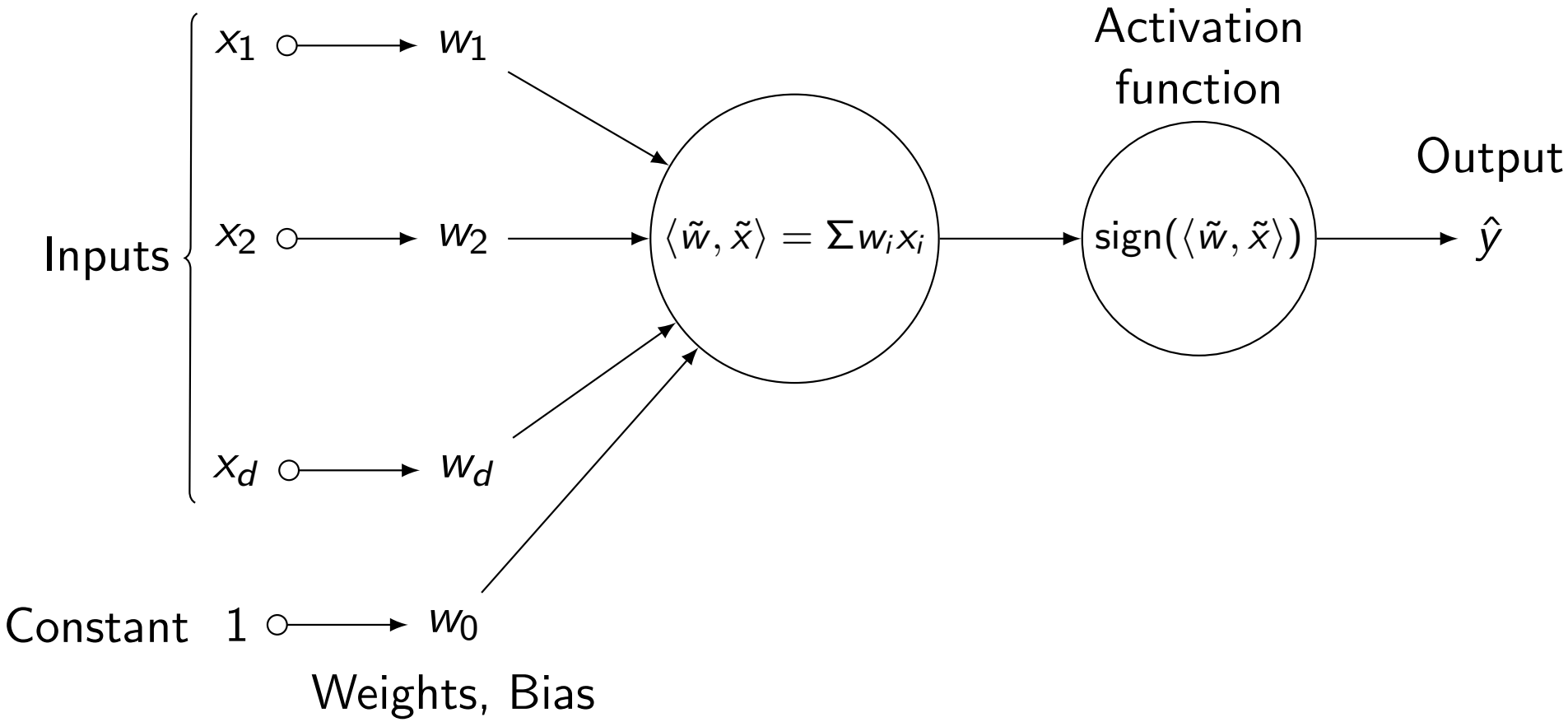
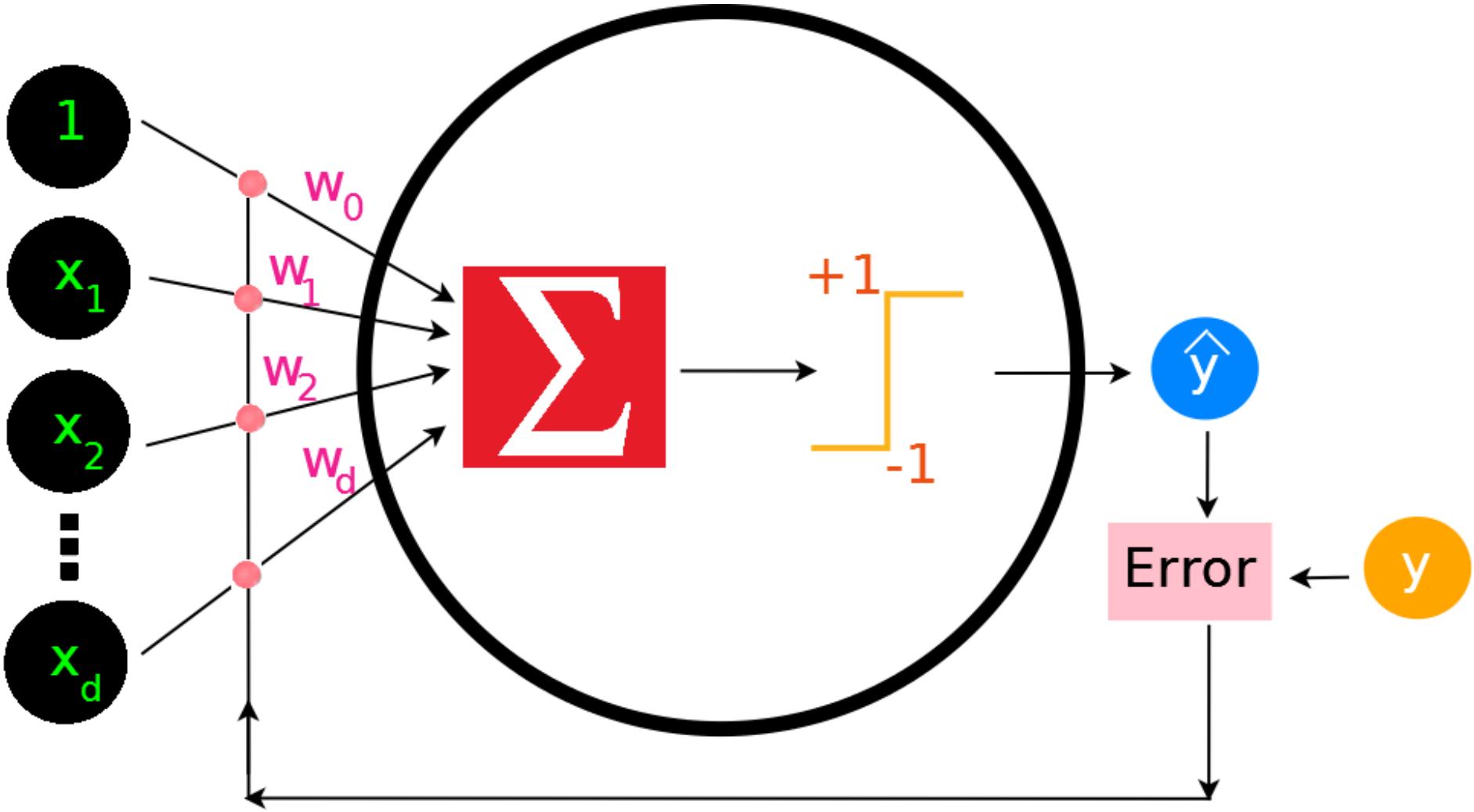


Figure: Perceptron unit

Perceptron - Data Perspective



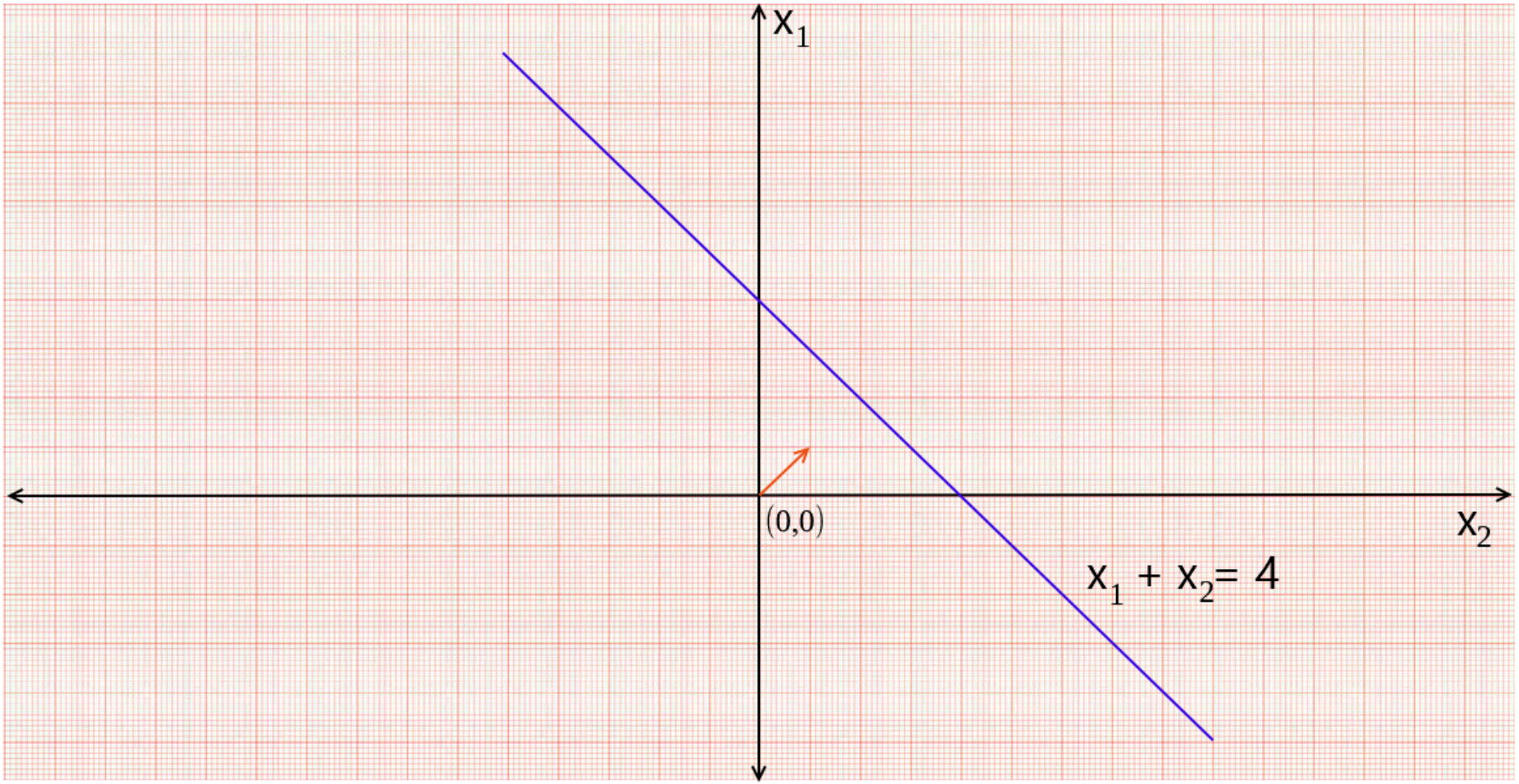
- **Input: data point** $x = (x_1, x_2, \ldots, x_d)$, **label** $y \in \{+1, -1\}$.

Perceptron - Training

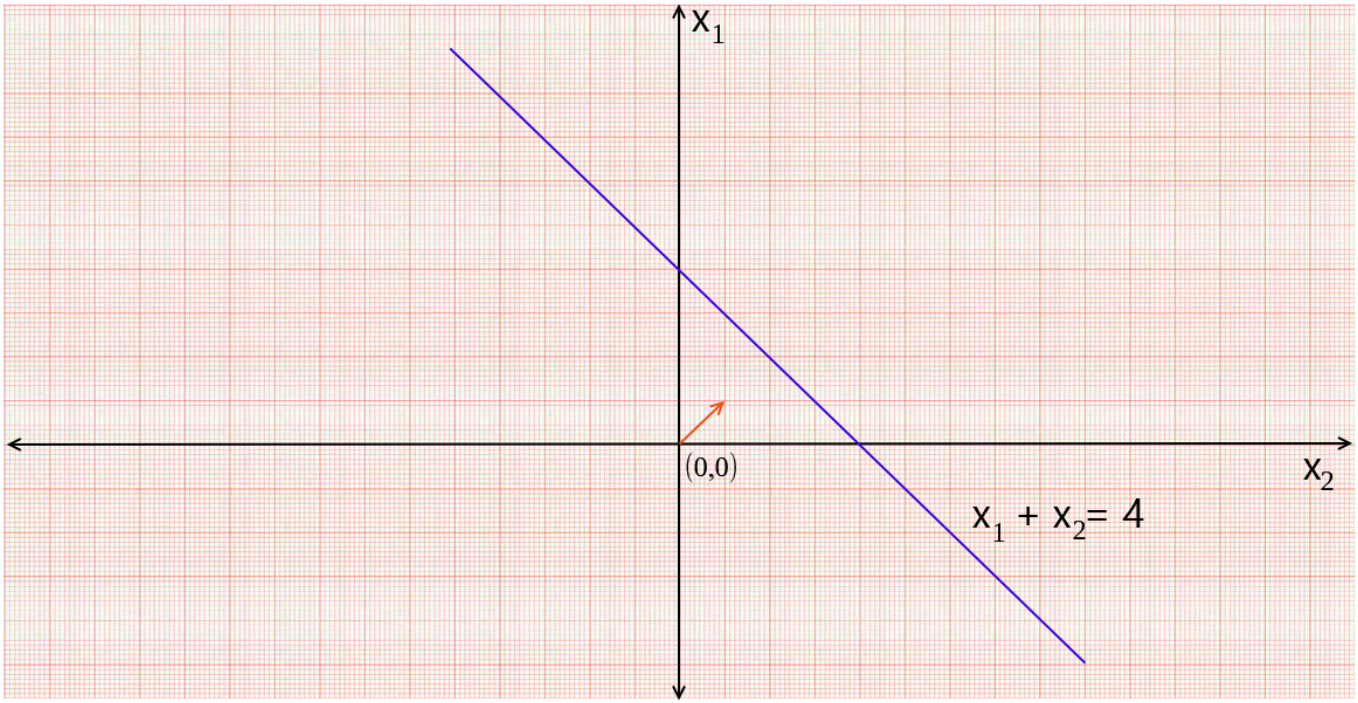
Perceptron Training Procedure

1: $\tilde{w}^1 = 0$
2: **for** $t \leftarrow 1, 2, 3, \ldots$ **do**
3: receive (x^t, y^t), $x^t \in \mathbb{R}^d$, $y^t \in \{+1, -1\}$.
4: Transform x^t into $\tilde{x}^t = (x^t, 1) \in \mathbb{R}^{d+1}$.
5: $\widehat{y} = Perceptron(\tilde{x}^t; \tilde{w}^t)$
6: **if** $\widehat{y} \neq y^t$ **then**
7: $\tilde{w}^{t+1} = \tilde{w}^t + y^t \tilde{x}^t$
8: **else**
9: $\tilde{w}^{t+1} = \tilde{w}^t$

Perceptron Update Rule - Geometric Idea

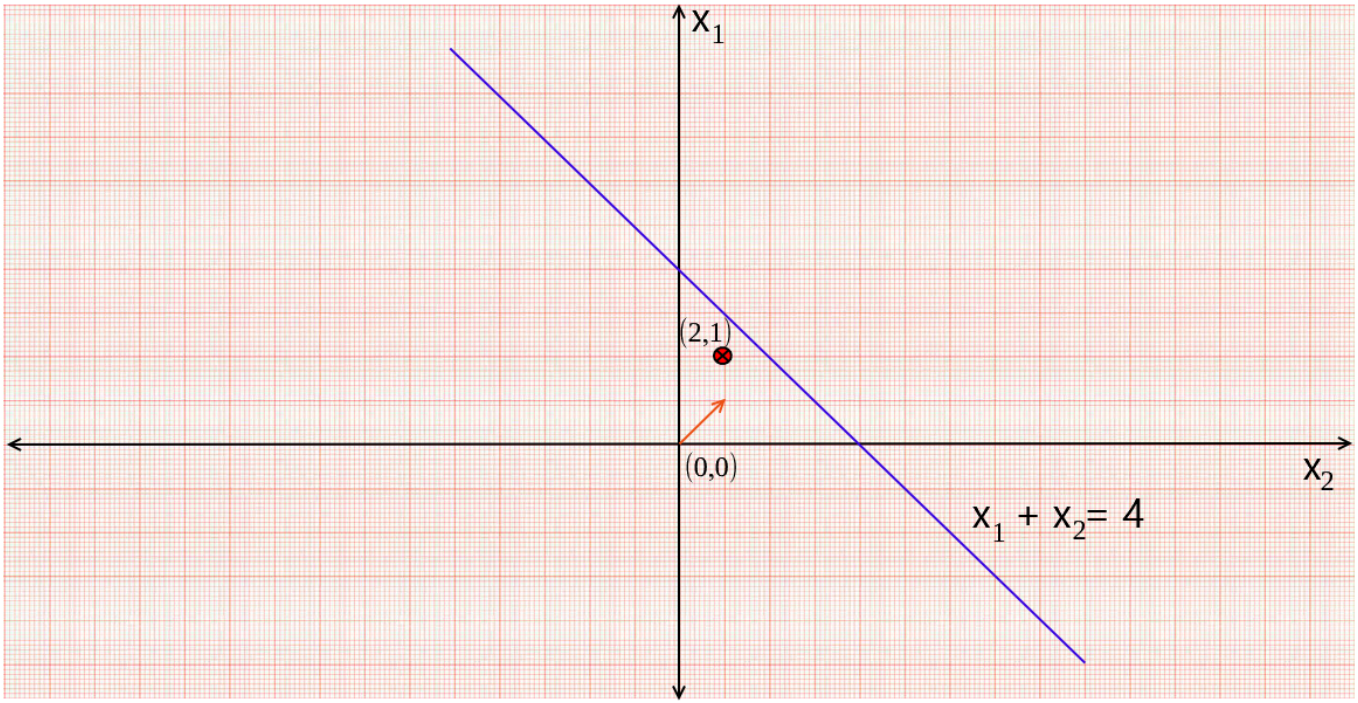


Perceptron Update Rule - Geometric Idea



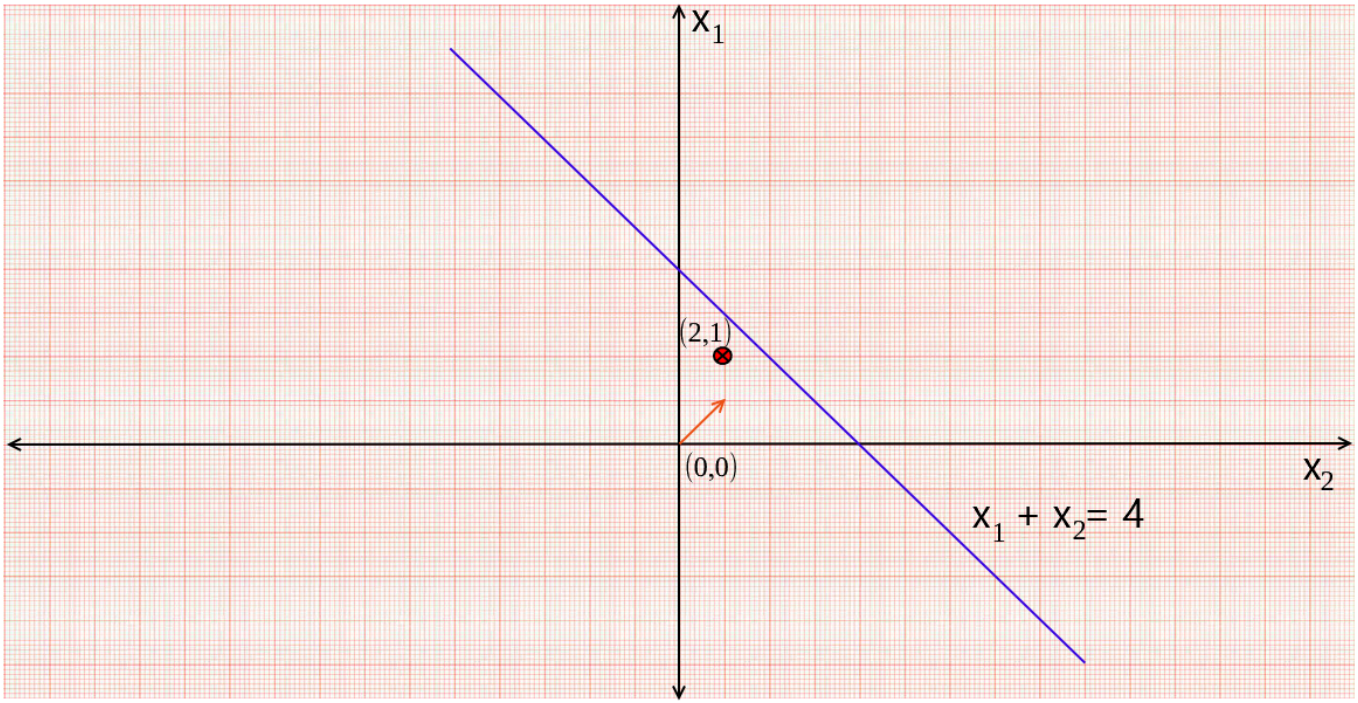
- The current weights are $\tilde{w}^t = (1, 1, -4)$.

Perceptron Update Rule - Geometric Idea



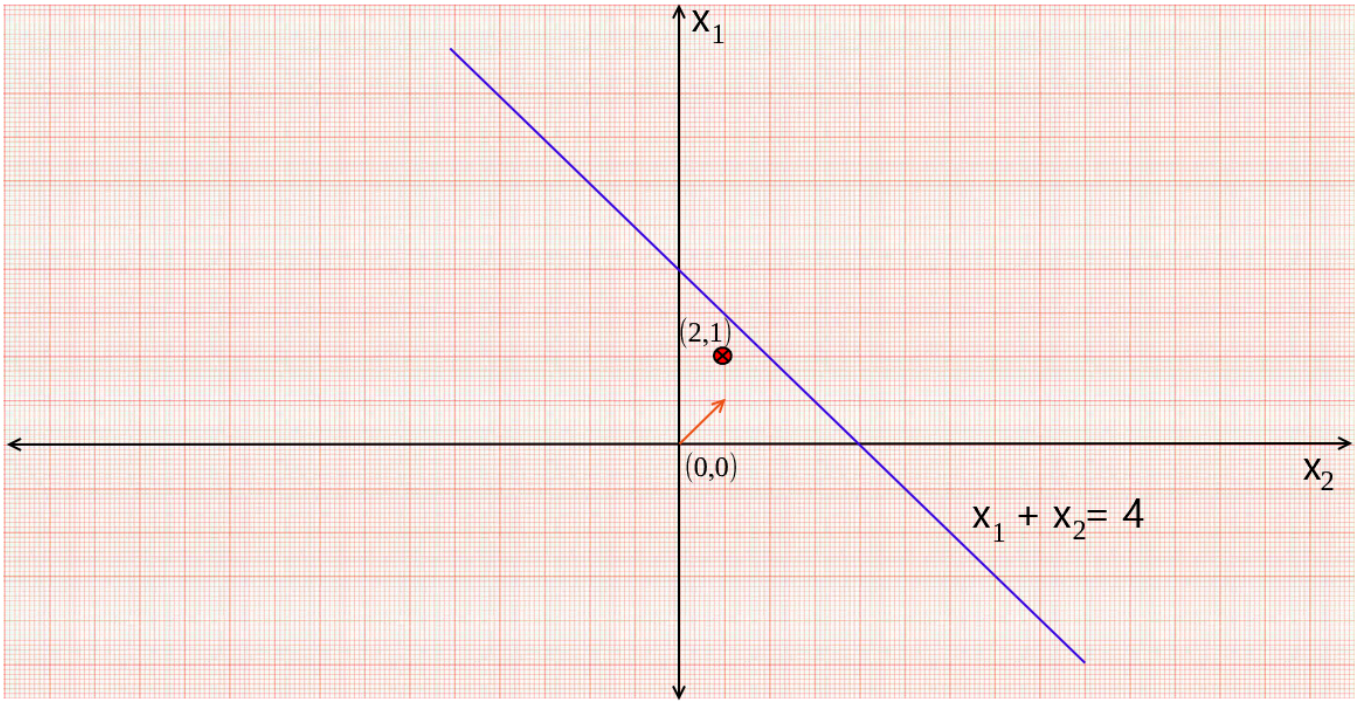
- Suppose a new point $x^t = (2, 1)$, $y^t = 1$ arrives.

Perceptron Update Rule - Geometric Idea



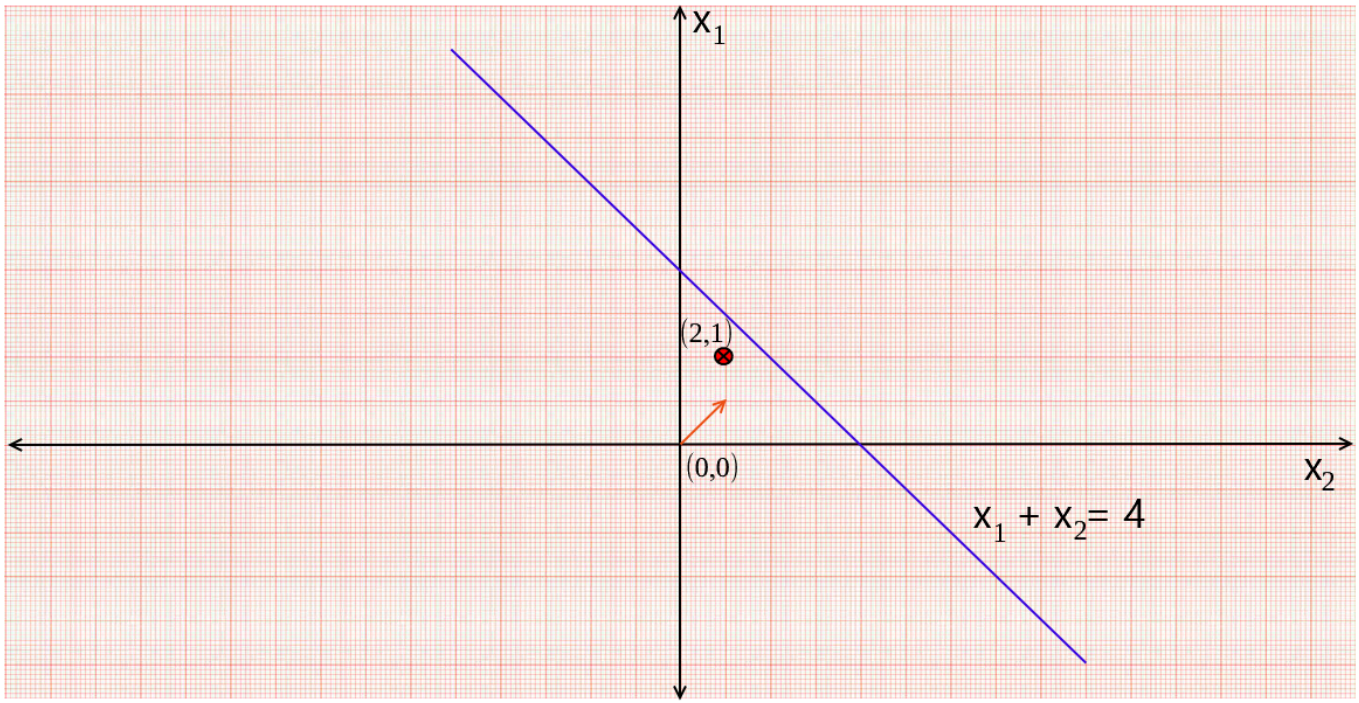
- Suppose a new point $x^t = (2, 1)$, $y^t = 1$ arrives.
- Transform x^t into $\tilde{x}^t = (2, 1, 1)$.

Perceptron Update Rule - Geometric Idea



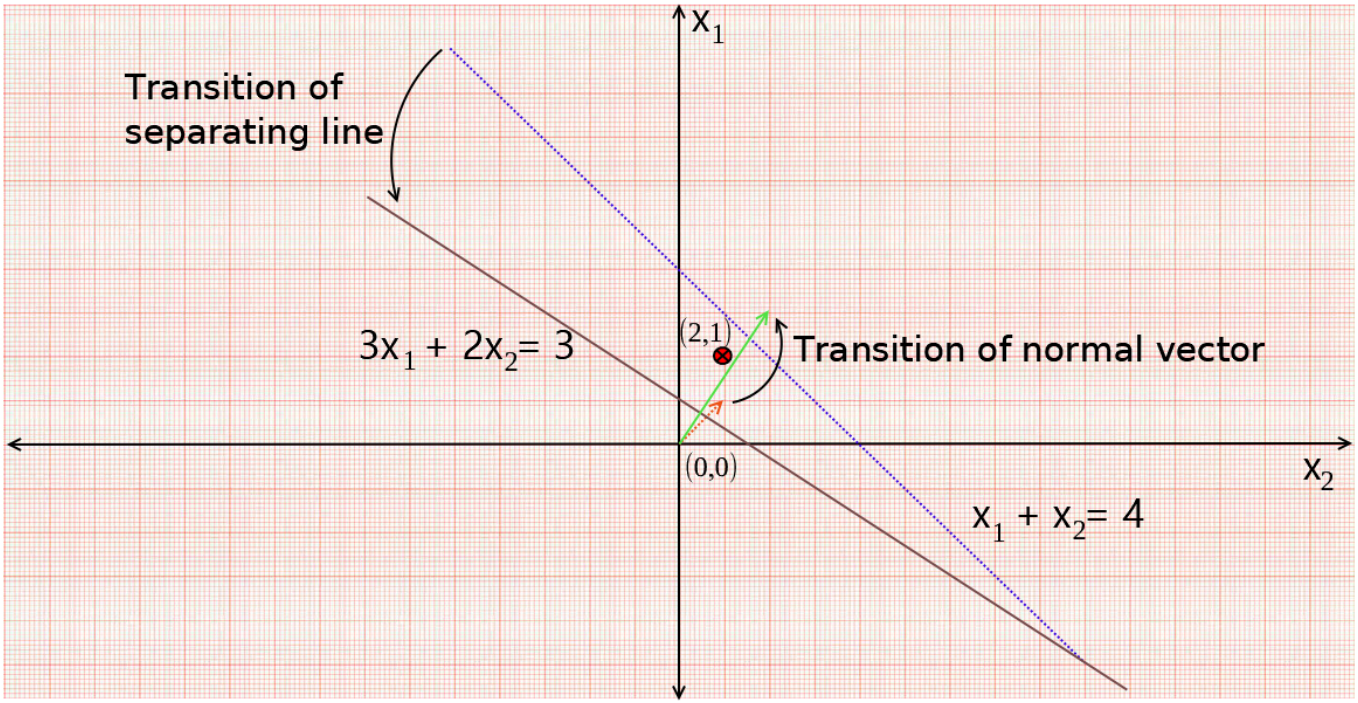
- Suppose a new point $x^t = (2, 1)$, $y^t = 1$ arrives.
- However with the current weights, perceptron outputs $\hat{y}^t = -1$. (why?)

Perceptron Update Rule - Geometric Idea



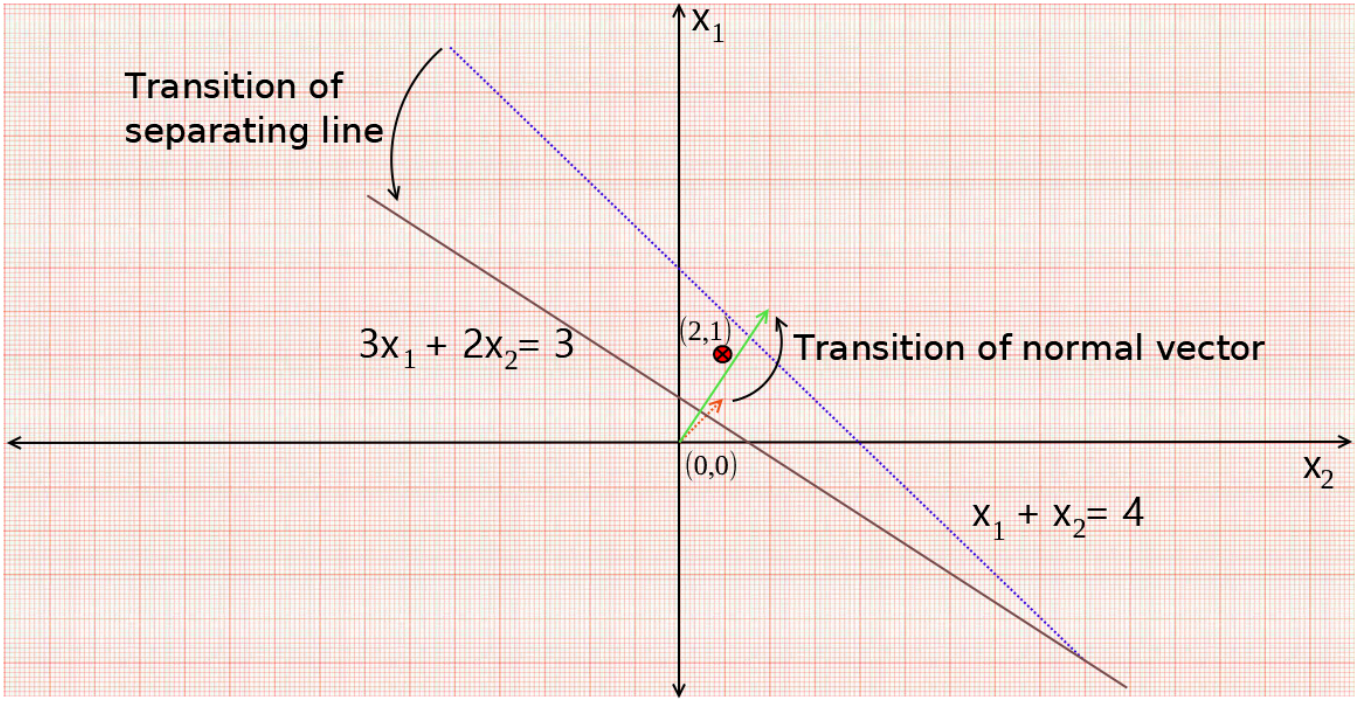
- Suppose a new point $x^t = (2, 1)$, $y^t = 1$ arrives.
- However with the current weights, perceptron outputs $\hat{y}^t = -1$.
- Hence an error occurs and $\tilde{w}^t$ gets updated to $\tilde{w}^{t+1} = \tilde{w}^t + y^t \tilde{x}^t$, .

Perceptron Update Rule - Geometric Idea



- Suppose a new point $x^t = (2, 1)$, $y^t = 1$ arrives.
- However with the current weights, perceptron outputs $\hat{y}^t = -1$.
- After update, the new weights become $\tilde{w}^{t+1} = (3, 2, -3)$.

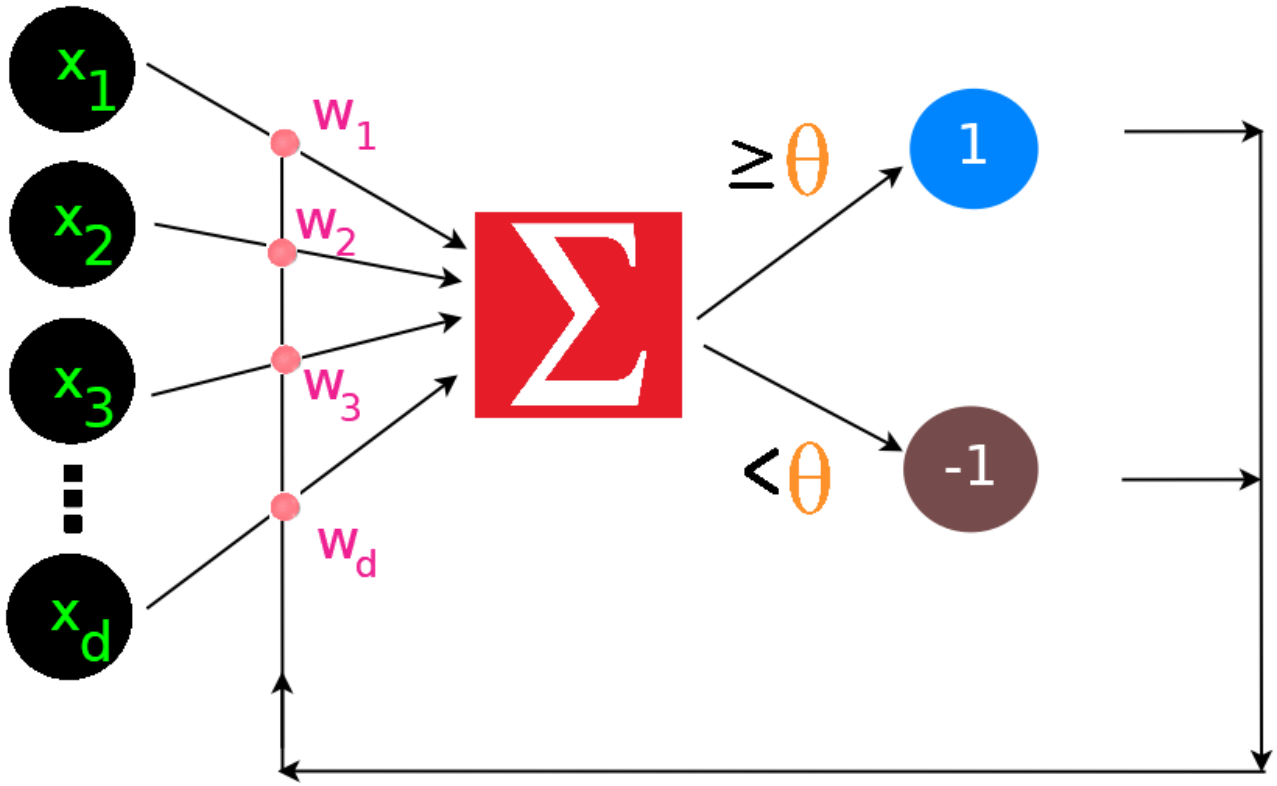
Perceptron Update Rule - Geometric Idea



Homework:

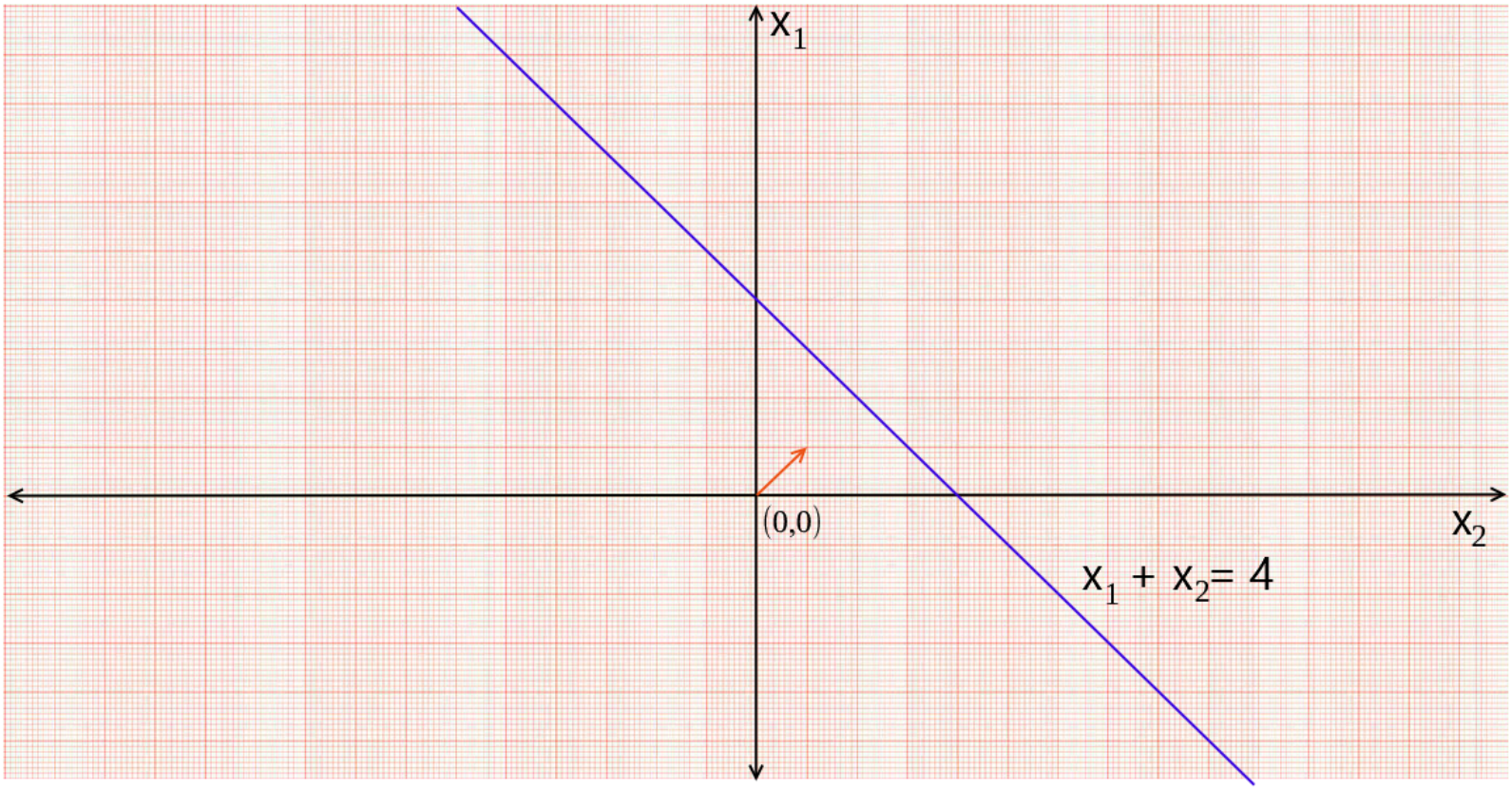
- Suppose now a new point $x^t = (-1, -1)$ with label -1 comes up. How will the weights change?
- Suppose a different new point $x^t = (-2, 3)$ with label $+1$ comes up. How will the weights change?

Perceptron - Geometric Idea



- Input is of the form $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$.
- We associate weights $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$ to the connections.
- **Prediction Rule:**
  - $\langle w, x \rangle \geq \theta \implies$ predict 1.
  - $\langle w, x \rangle < \theta \implies$ predict -1.

Perceptron - Geometric Idea

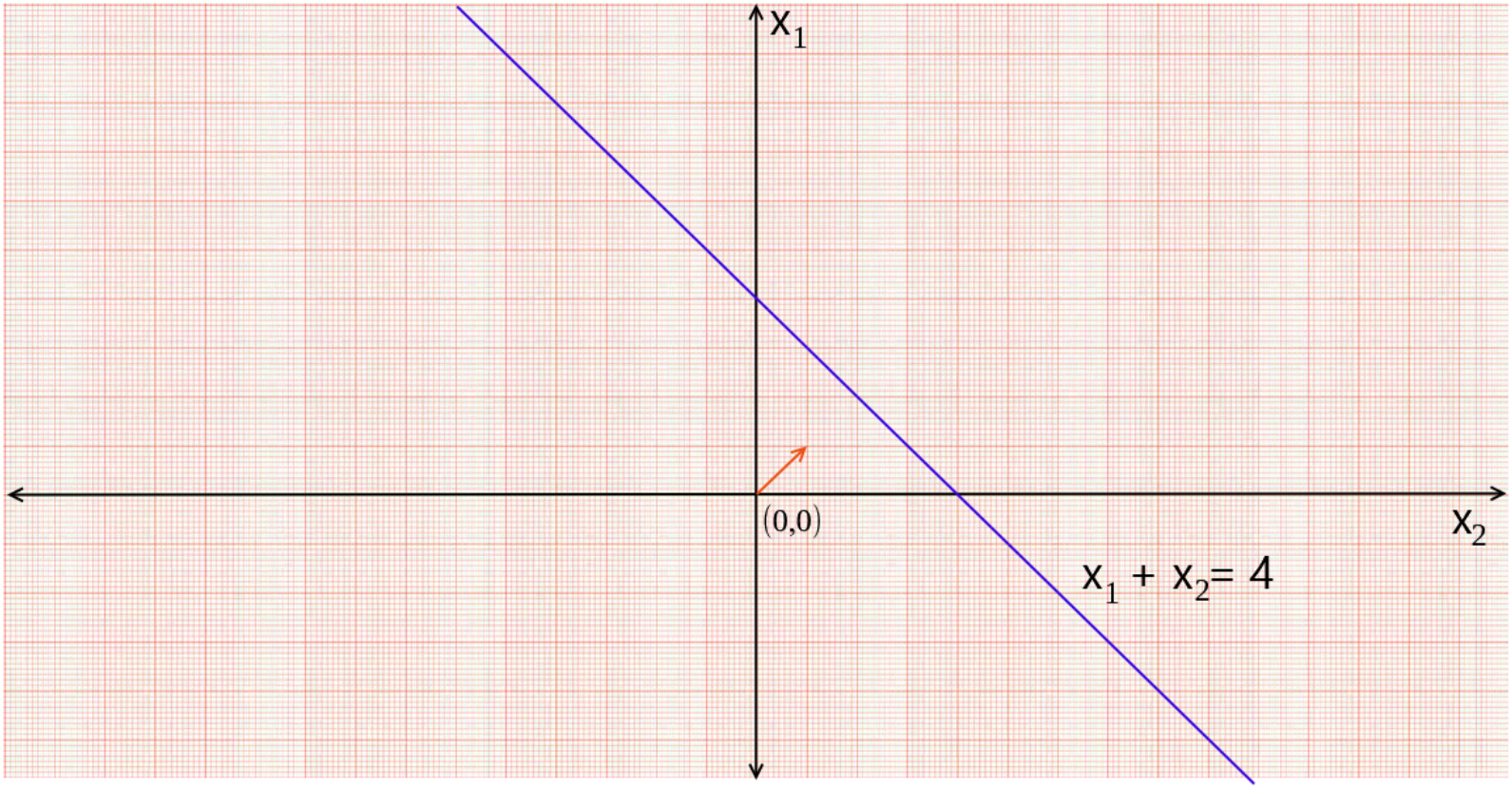


Prediction Rule

- $\langle w, x \rangle \geq \theta \implies$ predict 1.
- $\langle w, x \rangle < \theta \implies$ predict -1.

Geometric Idea: To find a separating hyperplane.

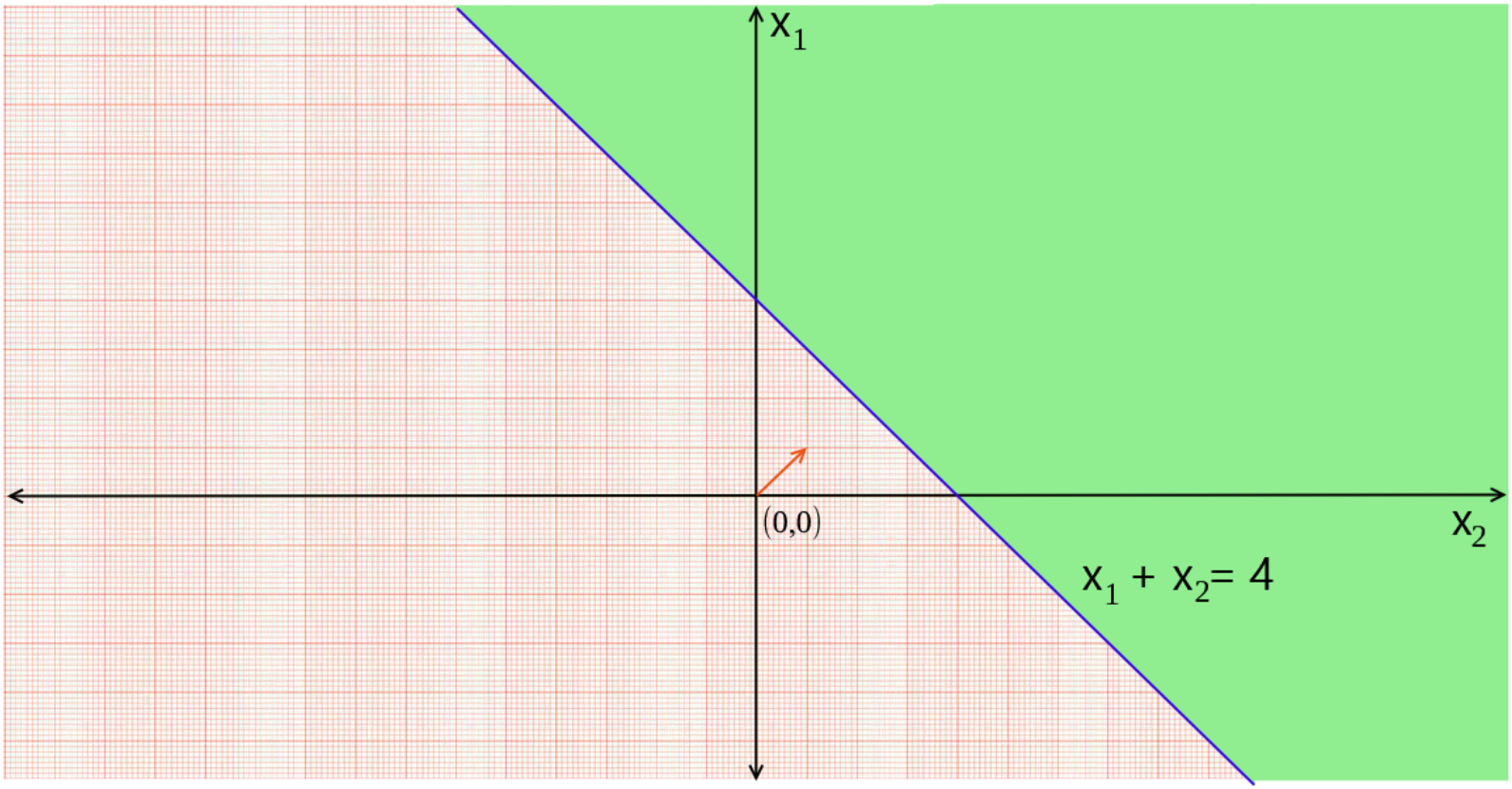
Perceptron - Geometric Idea



Hyperplane in $\mathbb{R}^d$

$\mathcal{H} = \{x \in \mathbb{R}^d : \exists w \neq 0 \text{ and } b \in \mathbb{R} \text{ s.t. } \langle w, x \rangle = b\}$.

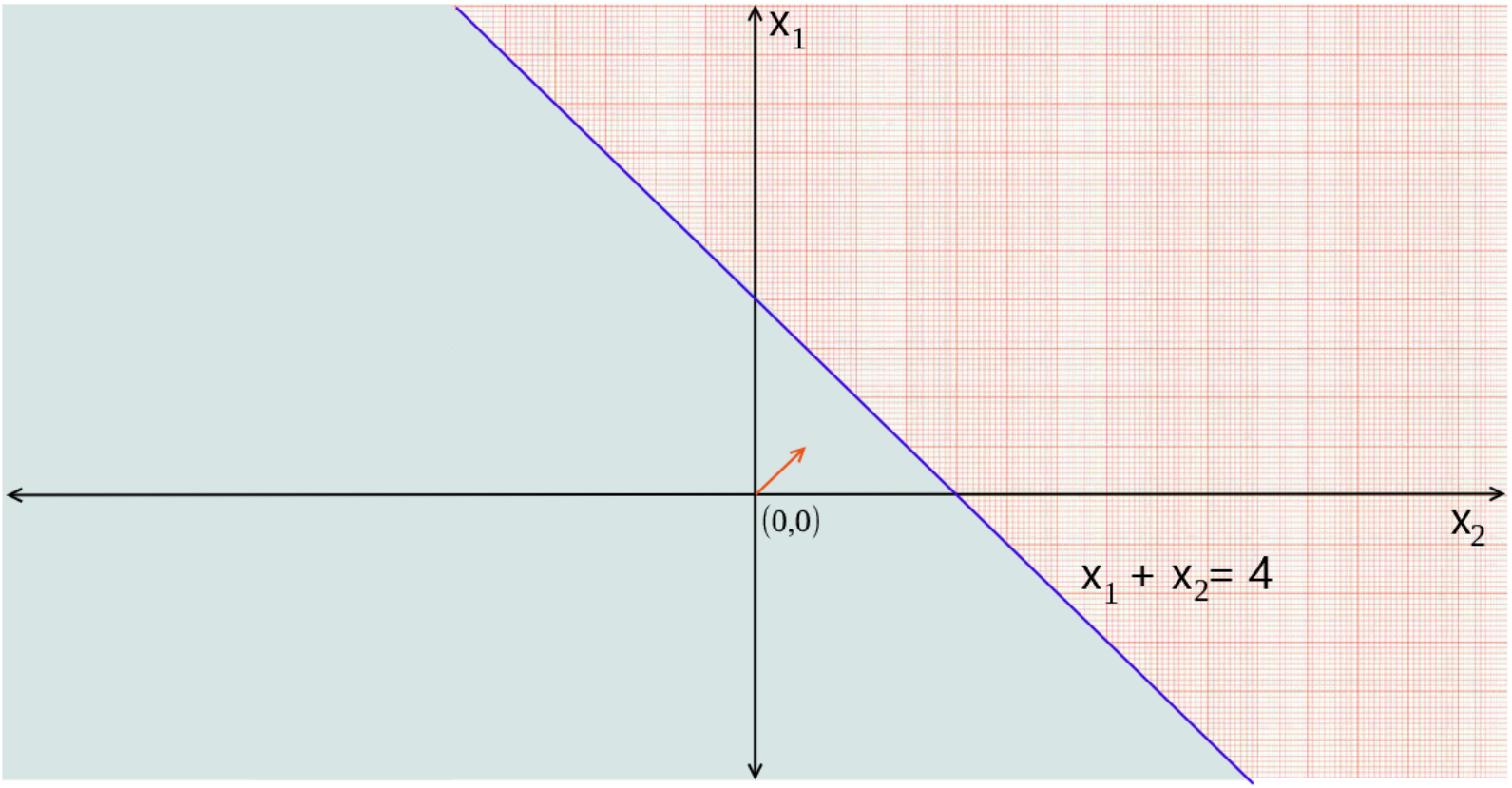
Perceptron - Geometric Idea



Open Halfspaces associated with a hyperplane $\langle w, x \rangle = b$

$\mathcal{Q}_1 = \{x \in \mathbb{R}^d : \langle w, x \rangle > b\}$ and $\mathcal{Q}_2 = \{x \in \mathbb{R}^d : \langle w, x \rangle < b\}$.

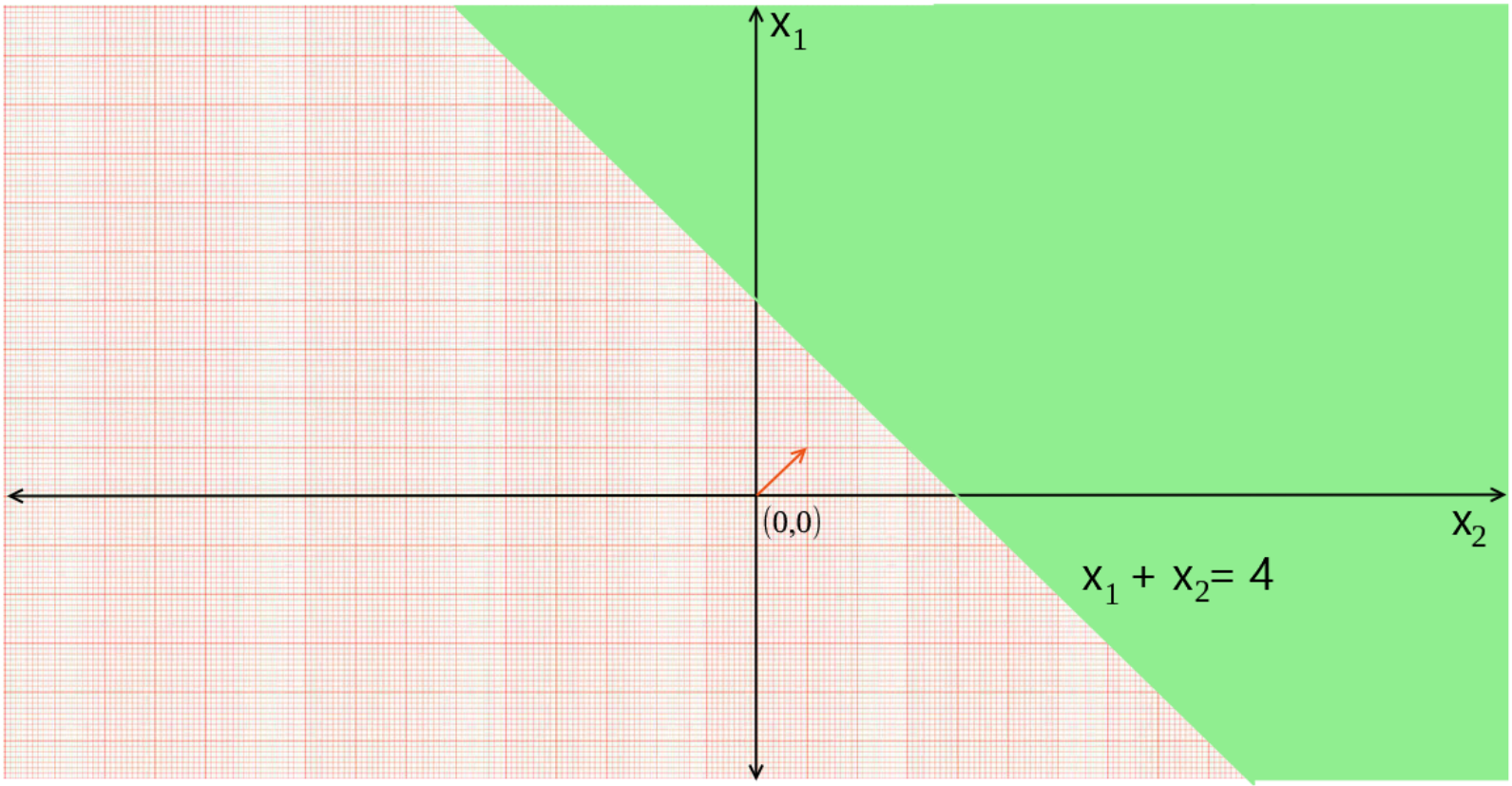
Perceptron - Geometric Idea



Open Halfspaces associated with a hyperplane $\langle w, x \rangle = b$

$\mathcal{Q}_1 = \{x \in \mathbb{R}^d : \langle w, x \rangle > b\}$ and $\mathcal{Q}_2 = \{x \in \mathbb{R}^d : \langle w, x \rangle < b\}$.

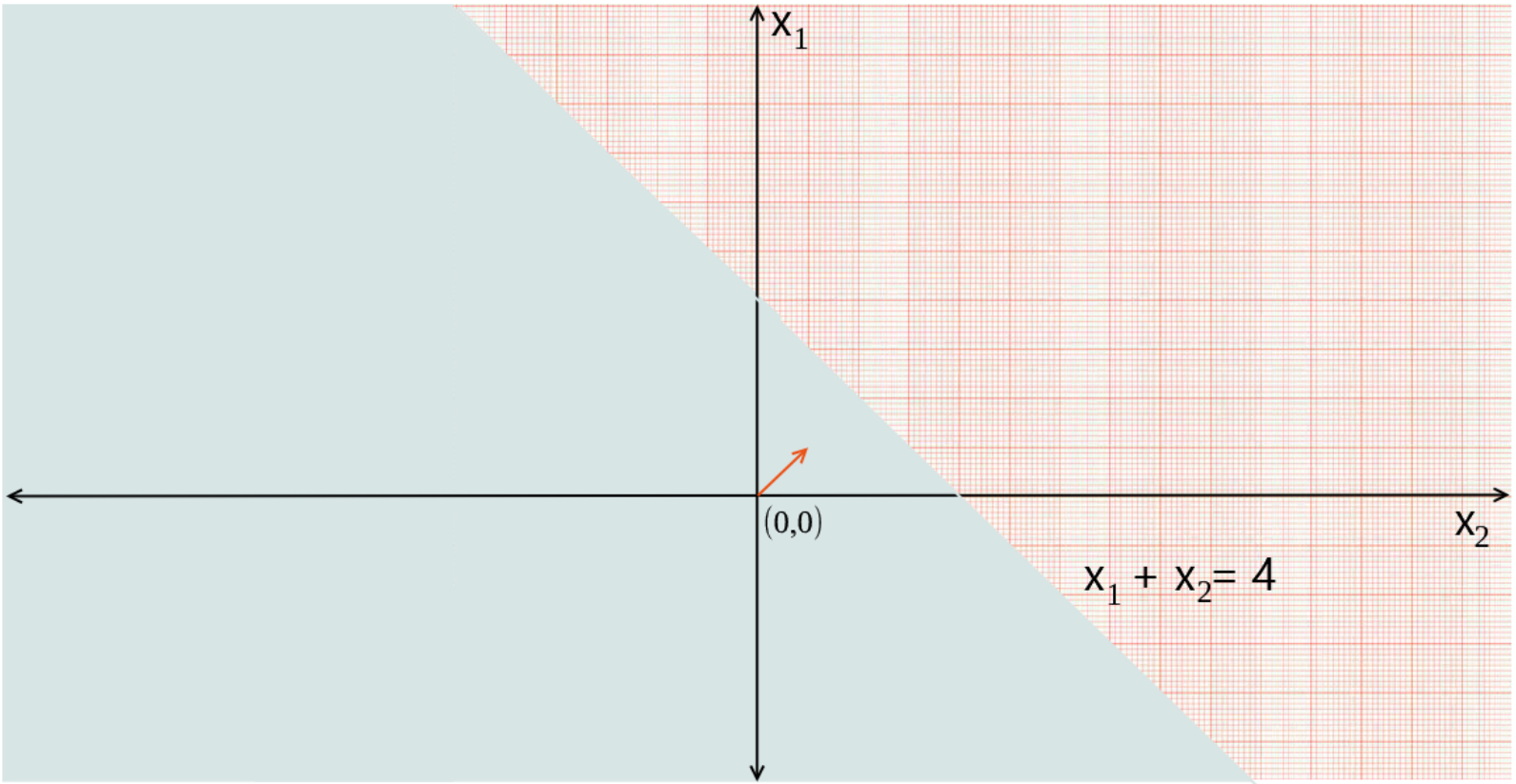
Perceptron - Geometric Idea



Closed Halfspaces associated with a hyperplane $\langle w, x \rangle = b$

$\mathcal{S}_1 = \{x \in \mathbb{R}^d : \langle w, x \rangle \geq b\}$ and $\mathcal{S}_2 = \{x \in \mathbb{R}^d : \langle w, x \rangle \leq b\}$.

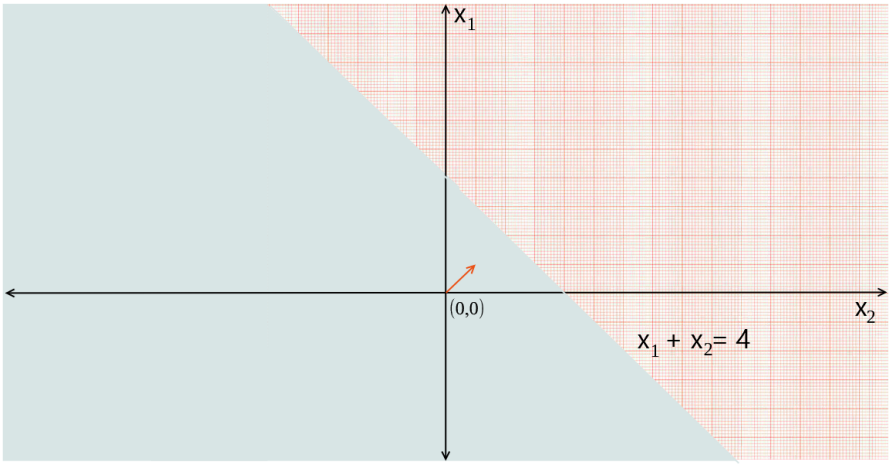
Perceptron - Geometric Idea

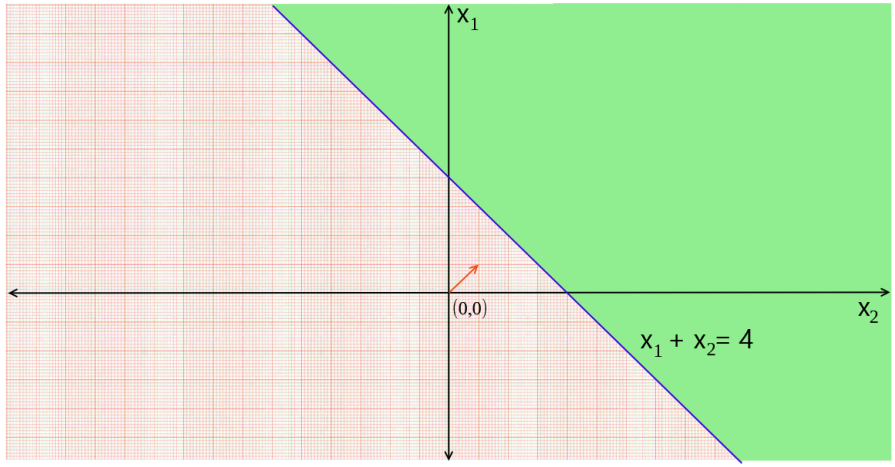


Closed Halfspaces associated with a hyperplane $\langle w, x \rangle = b$

$\mathcal{S}_1 = \{x \in \mathbb{R}^d : \langle w, x \rangle \geq b\}$ and $\mathcal{S}_2 = \{x \in \mathbb{R}^d : \langle w, x \rangle \leq b\}$.

Perceptron - Geometric Idea



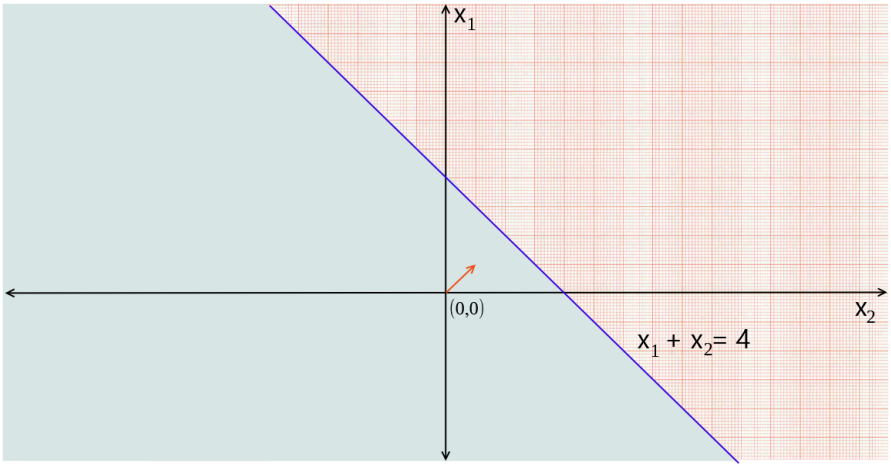


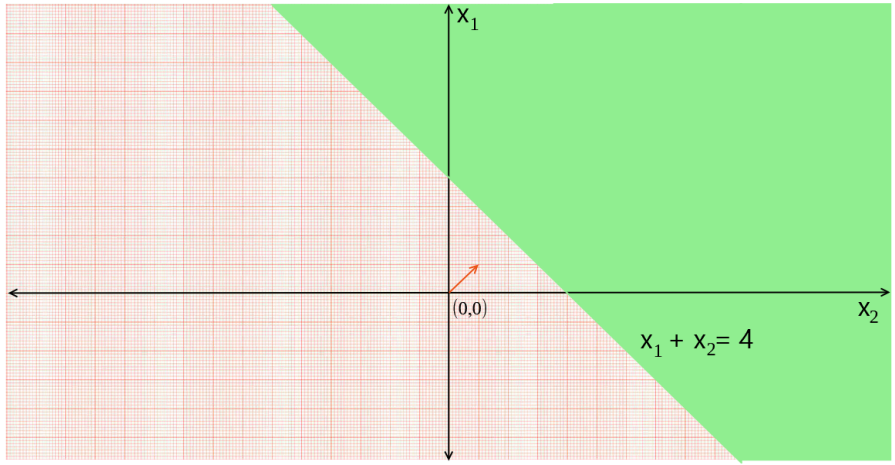
Prediction Rule

- $\langle w, x \rangle \geq \theta \implies$ predict 1.
- $\langle w, x \rangle < \theta \implies$ predict -1.

Geometric Idea: To find a separating hyperplane (w, θ) such that samples with class labels 1 and -1 lie on alternate sides of the hyperplane.

Perceptron - Geometric Idea





Prediction Rule

- $\langle w, x \rangle \geq \theta \implies$ predict 1.
- $\langle w, x \rangle < \theta \implies$ predict -1.

Geometric Idea: To find a separating hyperplane (w, θ) such that samples with class labels 1 and -1 lie on alternate sides of the hyperplane.

Perceptron - Convergence

Convergence of Perceptron Training Procedure

Under a suitable **separation assumption** of the positive and negative labeled data, the training procedure for Perceptron converges in finite time.