机器学习引论

彭玺
pengxi@scu.edu.cn
www.pengxi.me

提纲

一 . Limitations of KNN

二 . Neuron

三 . Perceptron

四 . Maximal Margin Classifier - Linear SVM

提纲

一．Limitations of KNN

二．Neuron

三．Perceptron

四．Maximal Margin Classifier - Linear SVM

Evolutional Readmap

The NN classifier:

- Prob: The training data are sufficiently distinct with each other. Insufficient robustness to noises.
- Sol: using k-nearest neighbor + max voting.

The KNN classifier:

- Prob: does not take the distance into the consideration of voting.
- Sol: Weighting by the distance!
- Prob: Choosing the value of k, i.e., model selection
- Sol: split the labeled data into training set and validation set.
- Prob: How to prove the method is good in statistics.
- Sol: Holdout method/Cross-validation
- Prob: Scaling issues
- Sol: Normalization

Two Limitations of KNN

- Prob: It do not learn knowledge from training data
- Prob: It requires that the data come from the Euclidean space so that the obtained neighbors and the data point itself come from the same subject.

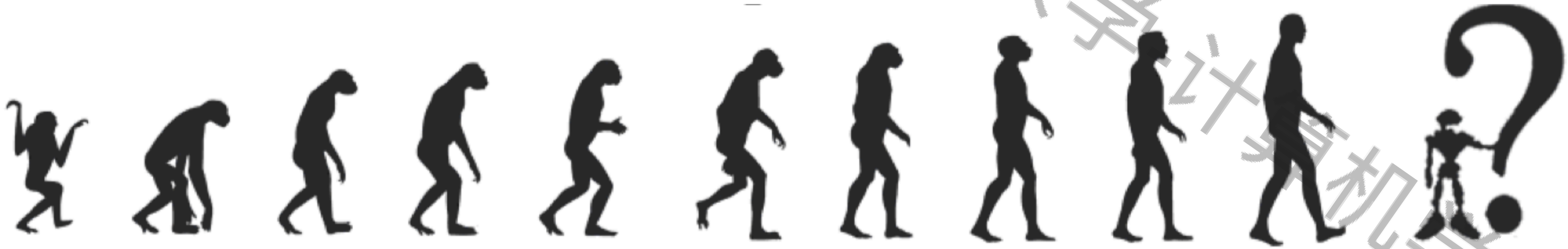
Two Limitations of KNN

- Prob: It do not learn knowledge from training data
- Sol: Perceptron -> Linear SVM
- Prob: It requires that the data come from the Euclidean space so that the obtained neighbors and the data point itself come from the same subject.
- Sol: Kernel + SVM = Nonlinear SVM

Supervised ML methods including classifier

- Given training instances (x, y)
- Learn a model/mapping f(.)
- Such that f(x) = y
- Use f(.) to predict y for new x

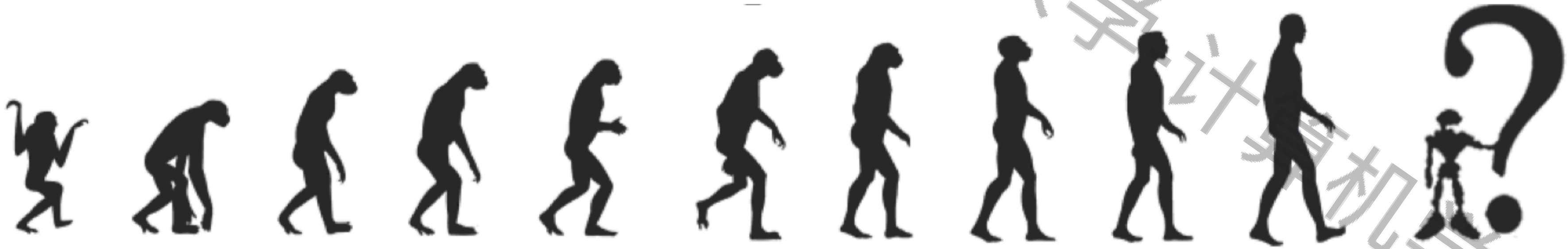
tips：人工智能发展史



- 1890：美国心理学家和哲学家William James在其著作中指出——当两个事件同时发生时，涉及到的大脑过程间的连接将会增强，这是无监督的Hebb学习规则的灵感来源。此外，James还提出了加权（weighted）、可变（modifiable）、及并行连接（parallel connections）等神经网络至今采用的基本概念。

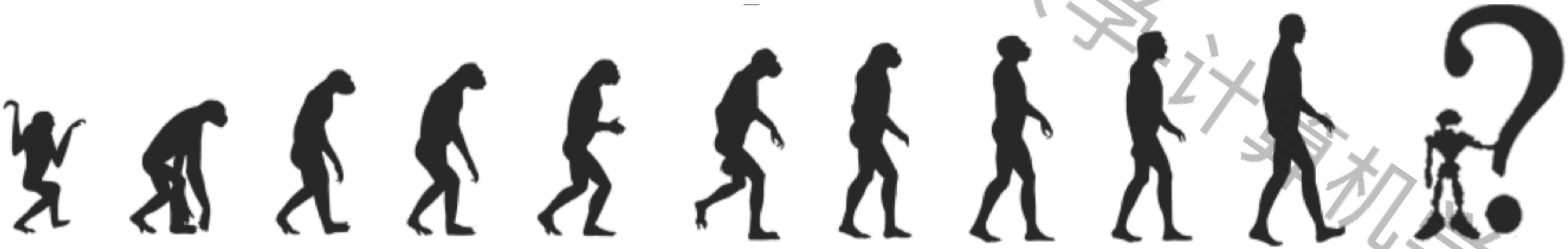
- 1906：神经科学家、诺贝尔奖得主S. R. Cajal对神经元突触的可变的连接方式的发现为神经元建模提供了基本模型。

tips：人工智能发展史



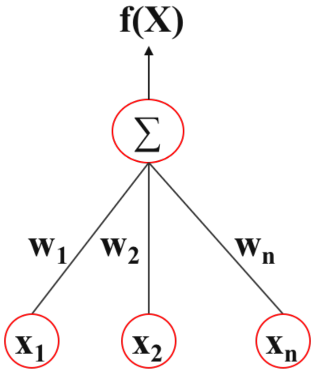
- 1932：James的学生Edward Lee Thorndike被认为是第一位真正的连接主义者。其专著《The Fundamentals of Learning》中提出学习是刺激和响应之间建立关联的结果。y?=f(x)
- 1943：C. L. Hull则提出神经元的输出是其输入与其他多个传入脉冲叠加的非线性函数，学习是对神经元连接的修改。y=f(**W**x)
  - 这一理论是神经元连接模型、激活状态和学习算法的生理学依据。此外，Hull还提出了几个经验性的公式，其本质上就是Bernard Widrow和Marcian Hoff提出的Delta rule。

tips：人工智能发展史



- 1943：Warren McCulloch和Walter Pitts提出第一个人工神经元模型——M-P模型，其通过阈值加权和定义了输入与输出的关系，奠定了人工神经元的数学模型基础。此外，McCulloch和Pitts于1947年的工作是模式识别领域的开端；$y=f(\mathbf{W}x+b)$

- 1957：基于上述发现和研究，Frank Rosenblatt发明了Perceptron，是第一个2层的ANN，同时也是第一个基于学习的能处理线性可分数据的分类器。



提纲

一. Limitations of KNN

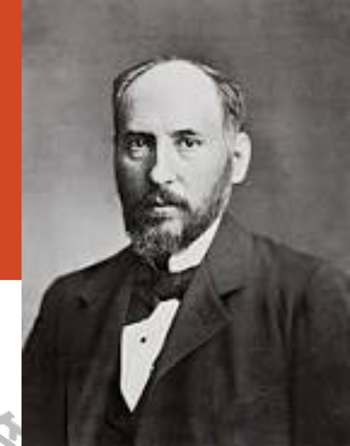
二. Neuron

三. Perceptron

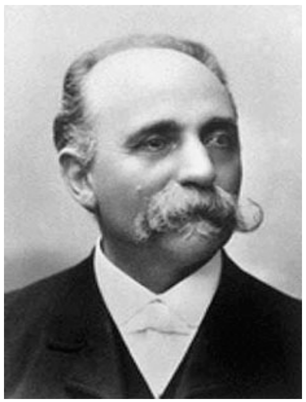
四. Maximal Margin Classifier - Linear SVM

Neuron – see through brains

Neuron – 突触可变

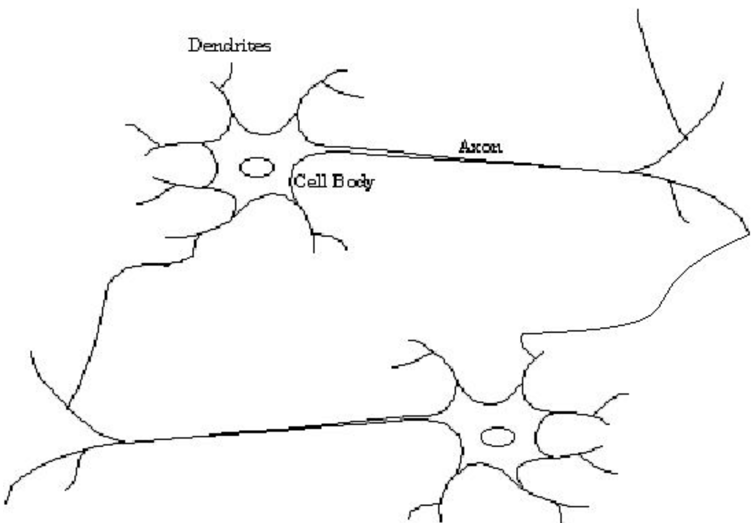


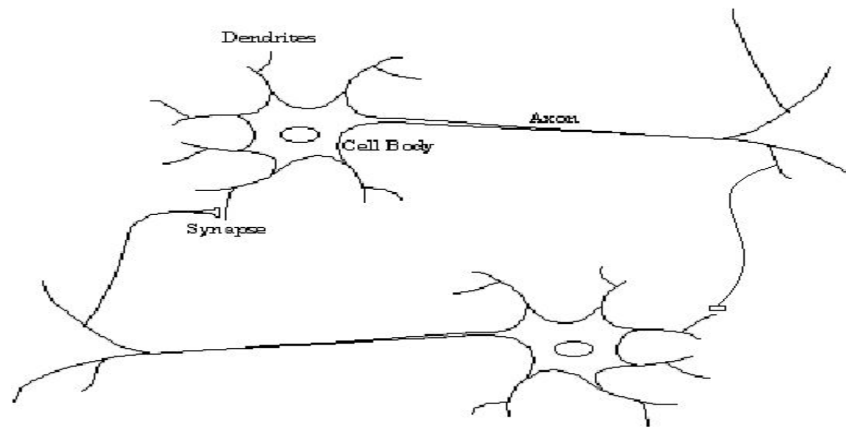
- 1906：神经科学家、诺贝尔奖得主S. R. Cajal对神经元突触的可变的连接方式的发现为神经元建模提供了基本模型。



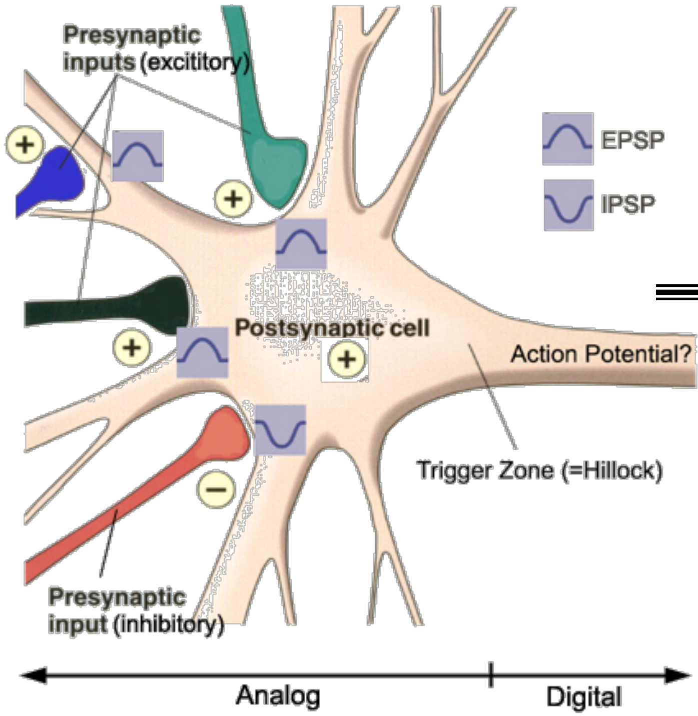
C. Golgi

S. R. Cajal





Neuron

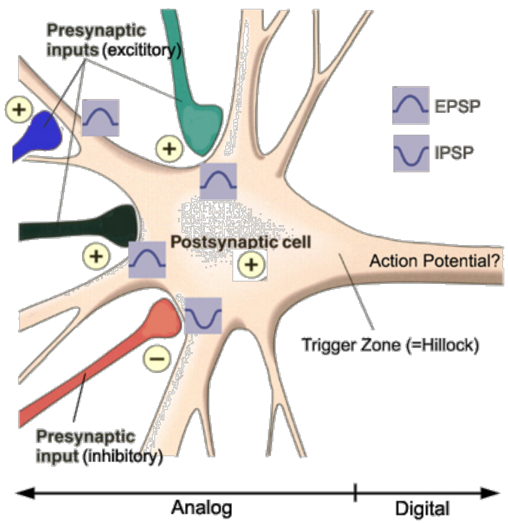


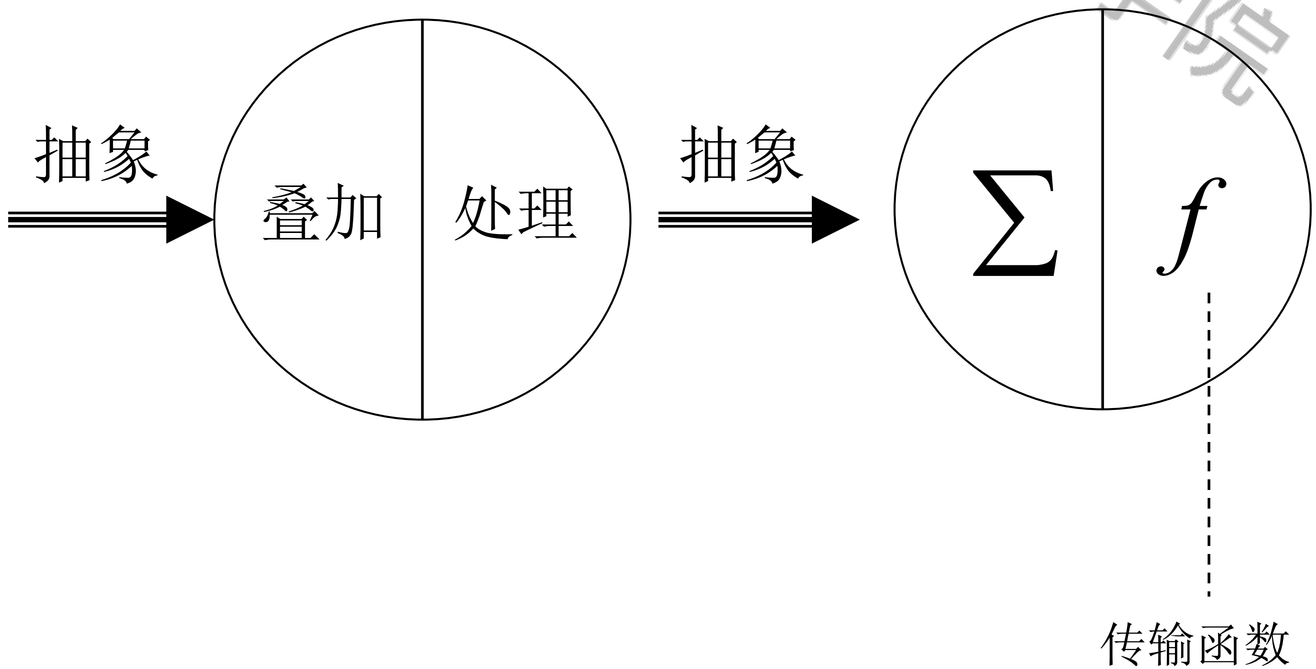
抽象？

数学模型

Neuron

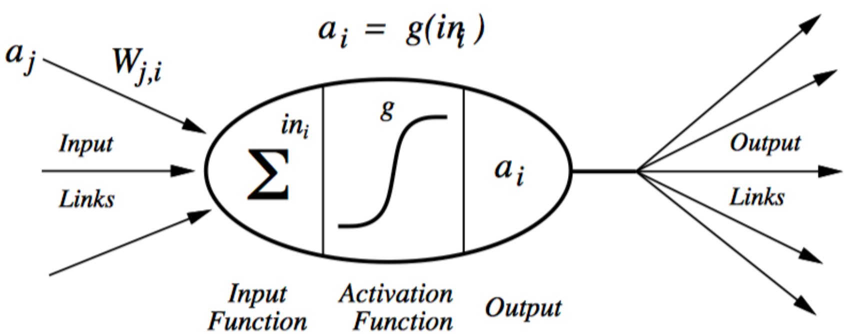
- 1943：C. L. Hull则提出神经元的输出是其输入与其他多个传入脉冲叠加的非线性函数，学习是对神经元连接的修改。y=f(**W**x)



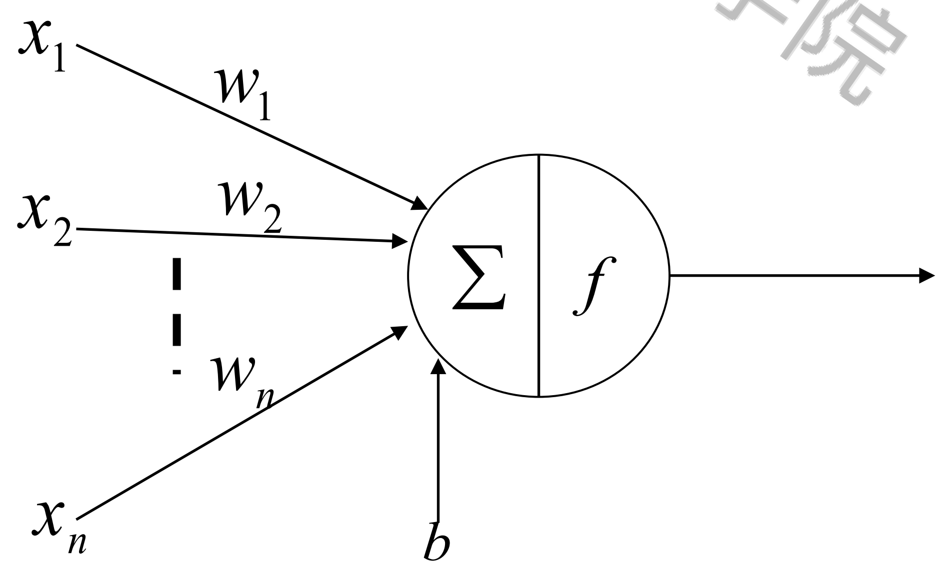


Neuron

- 1943：Warren McCulloch和Walter Pitts提出第一个人工神经元模型——M-P模型，其通过阈值加权和定义了输入与输出的关系，奠定了人工神经元的数学模型基础。此外，McCulloch和Pitts于1947年的工作是模式识别领域的开端；y=f(**W**x+b)



$$a_i = g\left(\sum_j W_{j,i} a_j\right)$$



$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

提纲

一．Limitations of KNN

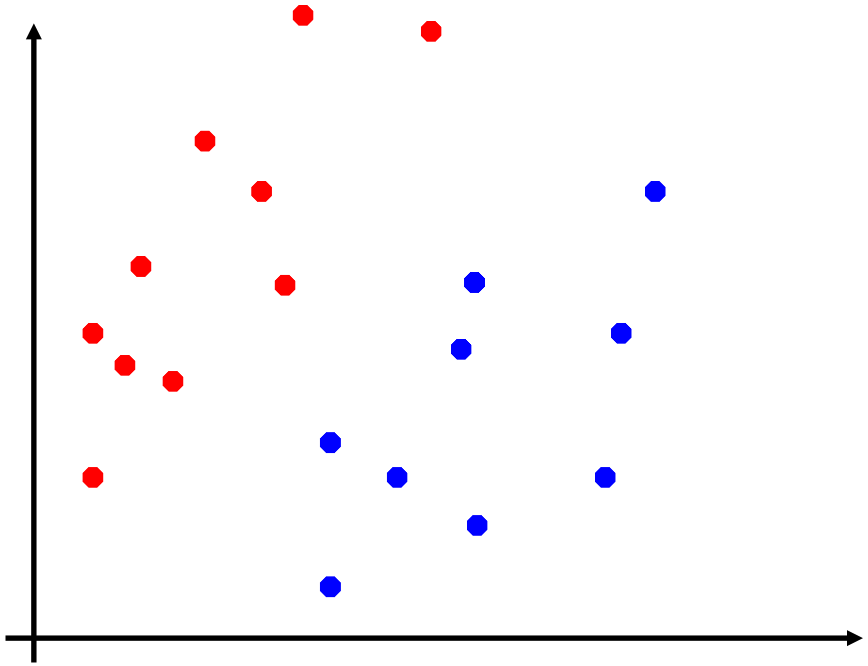
二．Neuron

三．Perceptron

四．Maximal Margin Classifier - Linear SVM

Perceptron

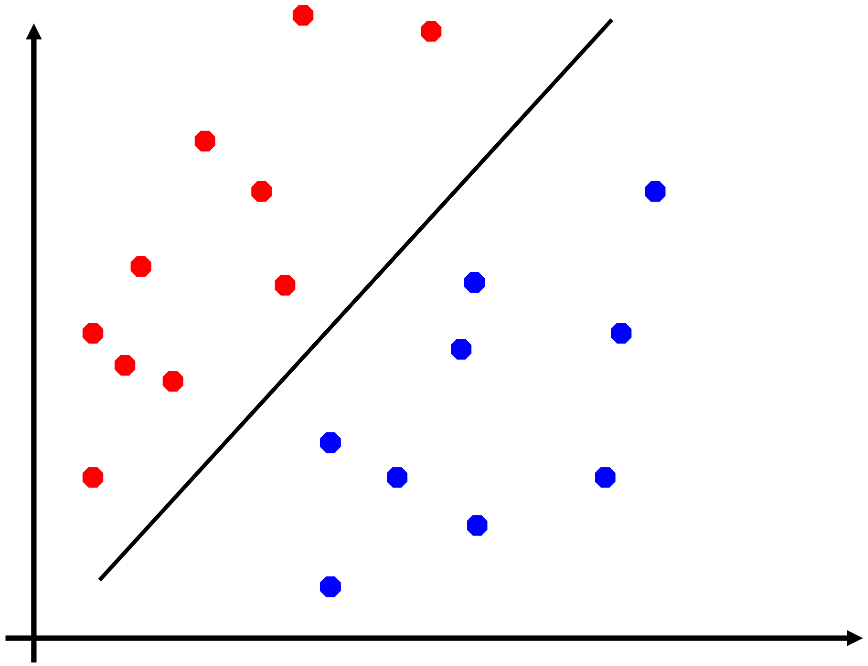
- Binary classification can be viewed as the task of separating classes in feature space:



What **knowledge** should be learned?

Perceptron

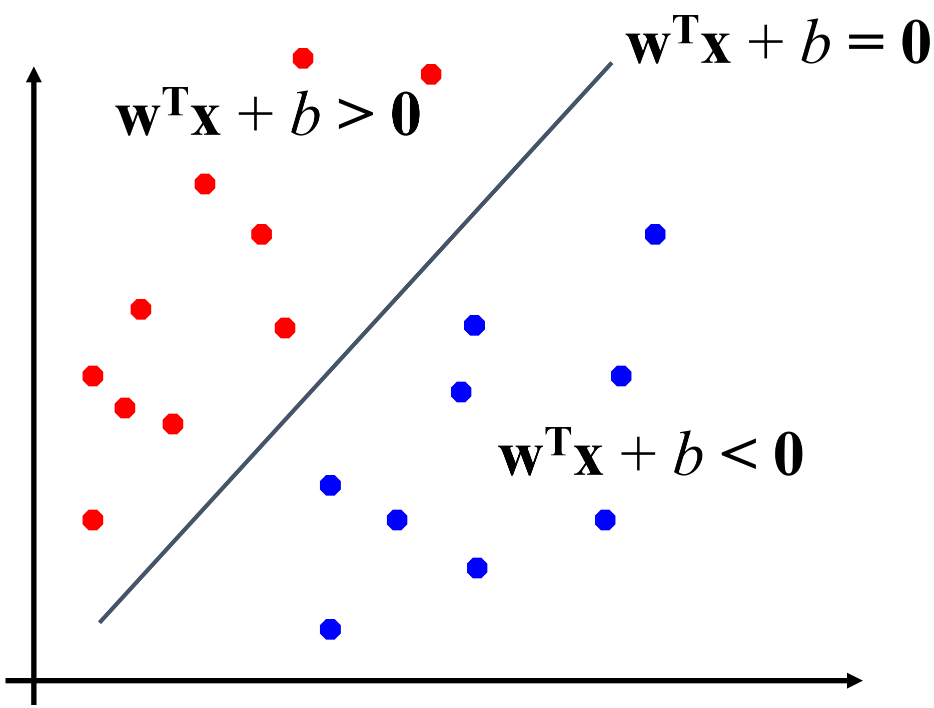
- Binary classification can be viewed as the task of separating classes in feature space:



Decision boundary/Hyperplane !

Perceptron

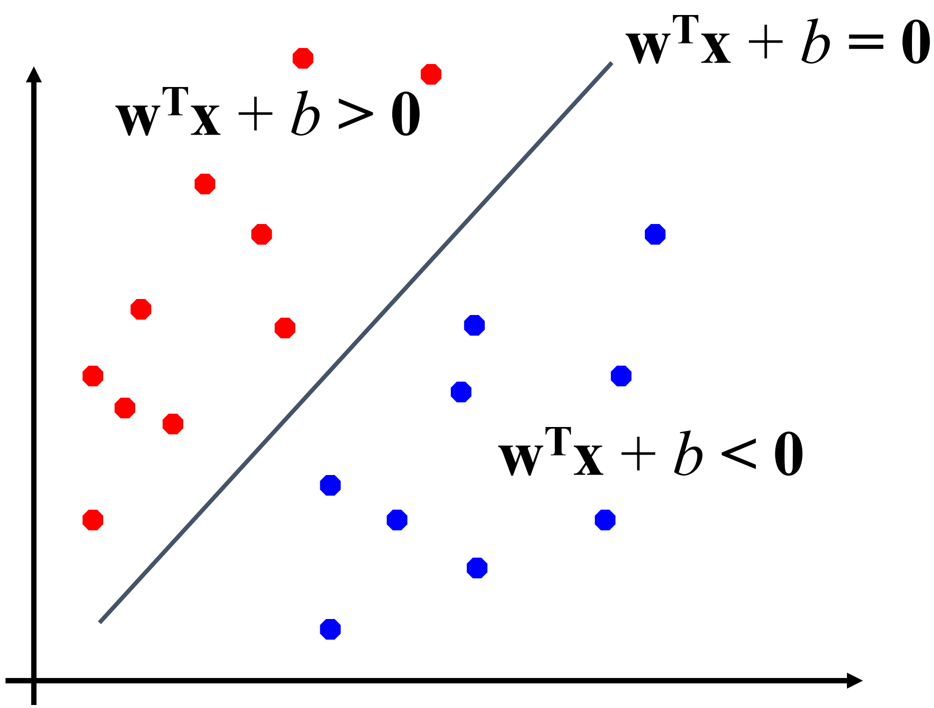
- Binary classification can be viewed as the task of separating classes in feature space:

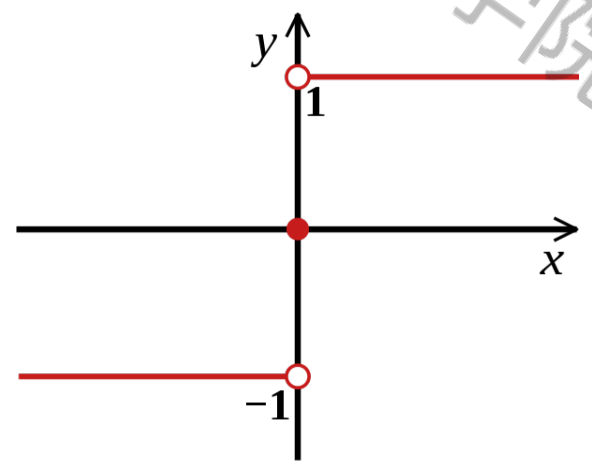


$$\mathbf{w}^{\mathrm{T}}\mathbf{x} = [\mathbf{w}^{\mathrm{T}} \quad \mathbf{b}] \times \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

Perceptron

- Binary classification can be viewed as the task of separating classes in feature space:



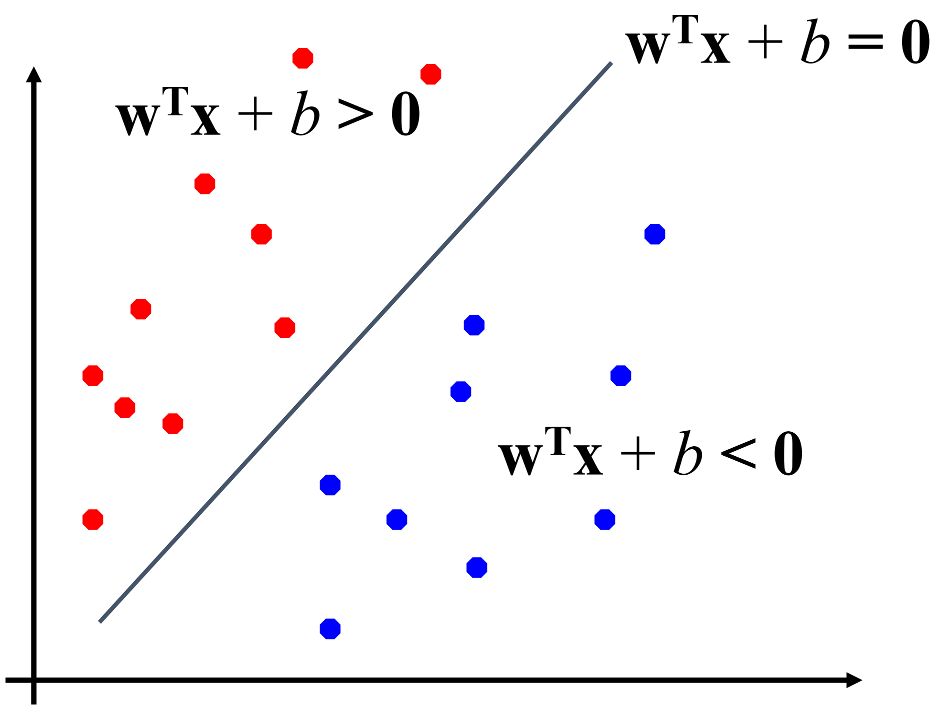


$$\operatorname{sgn}(x) := \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases}$$

Activate function

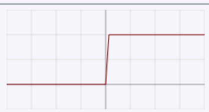
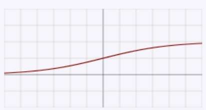
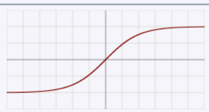
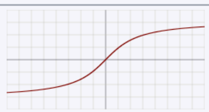
Perceptron

- Binary classification can be viewed as the task of separating classes in feature space:



$$y = \mathrm{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b)$$

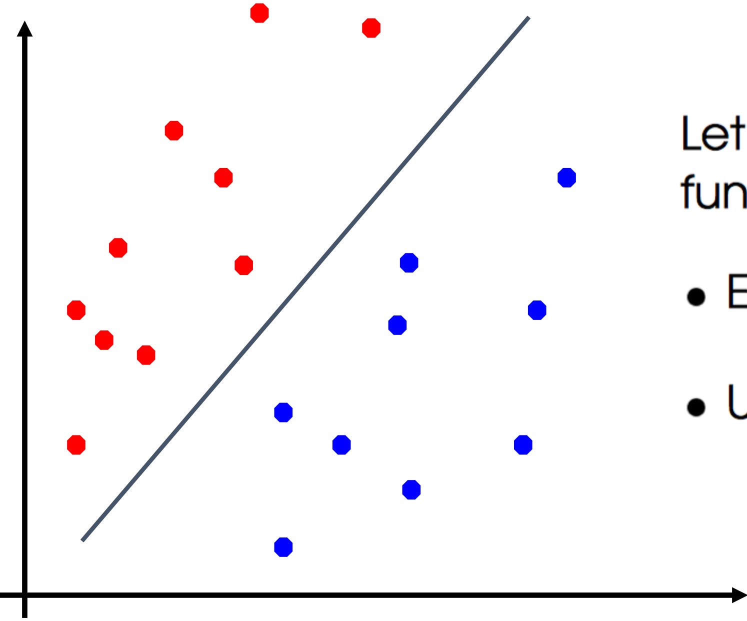
Tips: Activation Function

Name	Plot	Equation	Derivative (with respect to x)	Range	Order of continuity
Identity		$f(x) = x$	$f'(x) = 1$	$(-\infty, \infty)$	C^∞
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$	$\{0, 1\}$	C^{-1}
Logistic (a.k.a. Sigmoid or Soft step)		$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$ [1]	$f'(x) = f(x)(1 - f(x))$	$(0, 1)$	C^∞
TanH		$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$	$f'(x) = 1 - f(x)^2$	$(-1, 1)$	C^∞
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$	$\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$	C^∞
Softsign [7][8]		$f(x) = \frac{x}{1 + \|x\|}$	$f'(x) = \frac{1}{(1 + \|x\|)^2}$	$(-1, 1)$	C^1

See here for more choices:
https://en.wikipedia.org/wiki/Activation_function

Perceptron - Learning

- 1890：美国心理学家和哲学家William James在其著作中指出——当两个事件同时发生时，涉及到的大脑过程间的连接将会增强，这是无监督的Hebb学习规则的灵感来源。此外，James还提出了加权（weighted）、可变（modifiable）、及并行连接（parallel connections）等神经网络至今采用的基本概念。

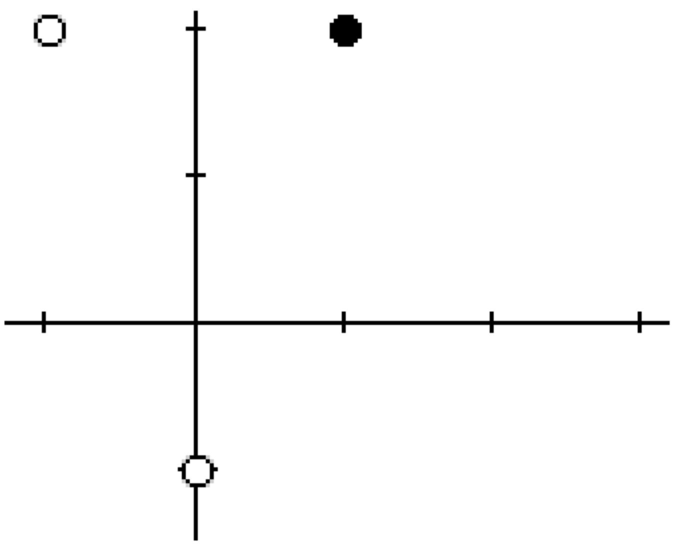


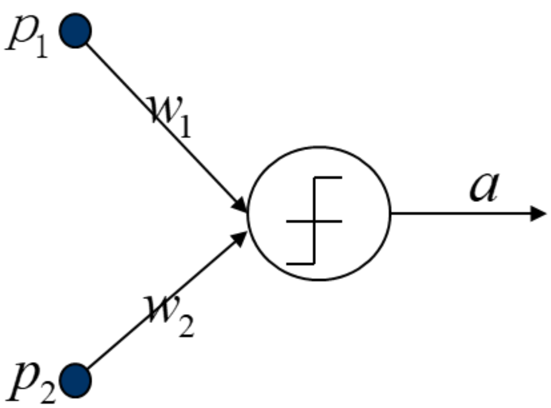
Let y be the correct output, and $f(x)$ the output function of the network.

- Error: E $= y - f(x)$
- Update weights: $W_j \leftarrow W_j + \alpha x_j E$

Perceptron - example

$$\left\{\mathbf{p}_1 = \begin{bmatrix}1\\2\end{bmatrix}, t_1 = 1\right\} \quad \left\{\mathbf{p}_2 = \begin{bmatrix}-1\\2\end{bmatrix}, t_2 = 0\right\} \quad \left\{\mathbf{p}_3 = \begin{bmatrix}0\\-1\end{bmatrix}, t_3 = 0\right\}$$

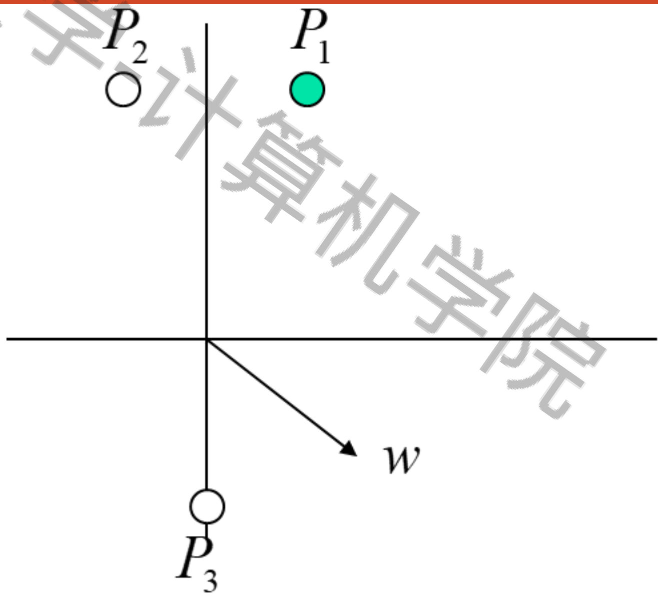




$$a = f\left(w^T p\right) = f\left(w_1 p_1 + w_2 p_2\right)$$

Perceptron – example – step 0

Random initial weight: $w = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix}$

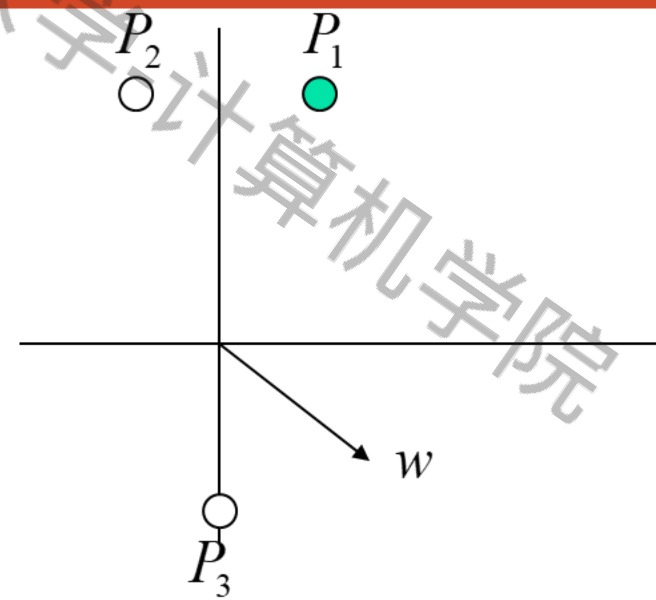


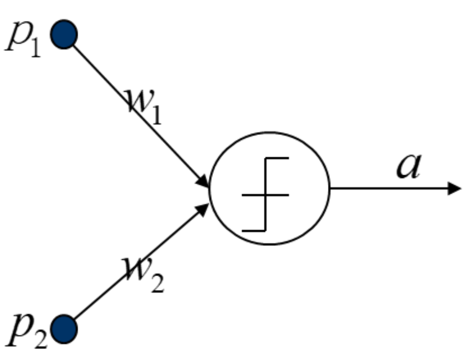
Perceptron – example – step 1

Random initial weight: $w = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix}$

pass $\mathbf{p}_1$ through the network: $\left\{ \mathbf{p}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, t_1 = 1 \right\}$

$$a = f\left(w^T P_1\right) = f\left(\begin{bmatrix} 1.0 & -0.8 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right) = f(-0.6) = 0$$



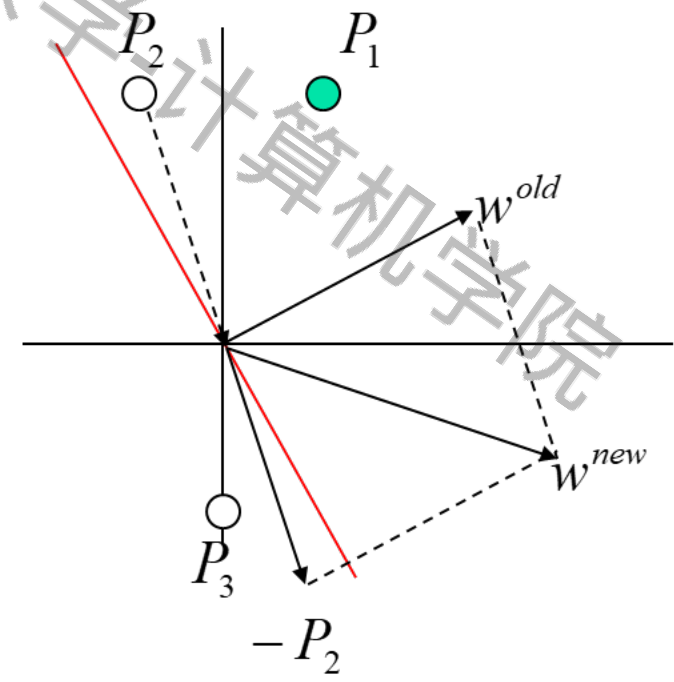


Perceptron – example – step 1

Random initial weight: $w = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix}$

pass $\mathbf{p}_1$ through the network: $\left\{ \mathbf{p}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, t_1 = 1 \right\}$

$$a = f\left(w^T P_1\right) = f\left(\begin{bmatrix} 1.0 & -0.8 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right) = f(-0.6) = 0$$



Let y be the correct output, and $f(x)$ the output function of the network.

- Error: $E = y - f(x)$
- Update weights: $W_j \leftarrow W_j + \alpha x_j E$

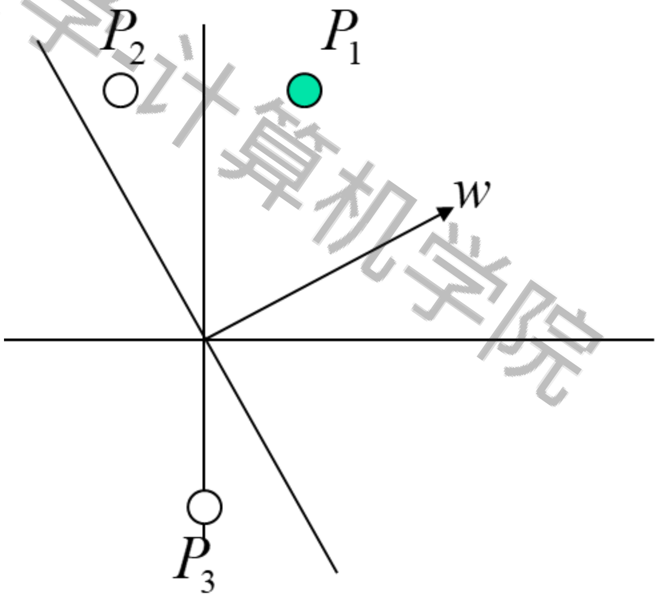
$$w^{new} = w^{old} + p_1 = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix}$$

Perceptron – example – step 2

$$w = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix}$$

pass $\mathbf{p}_2$ through the network: $\left\{ \mathbf{p}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, t_2 = 0 \right\}$

$$a = f\left(w^T P_2\right) = f\left(\begin{bmatrix} 2.0 & 1.2 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \right) = f(0.4) = 1$$

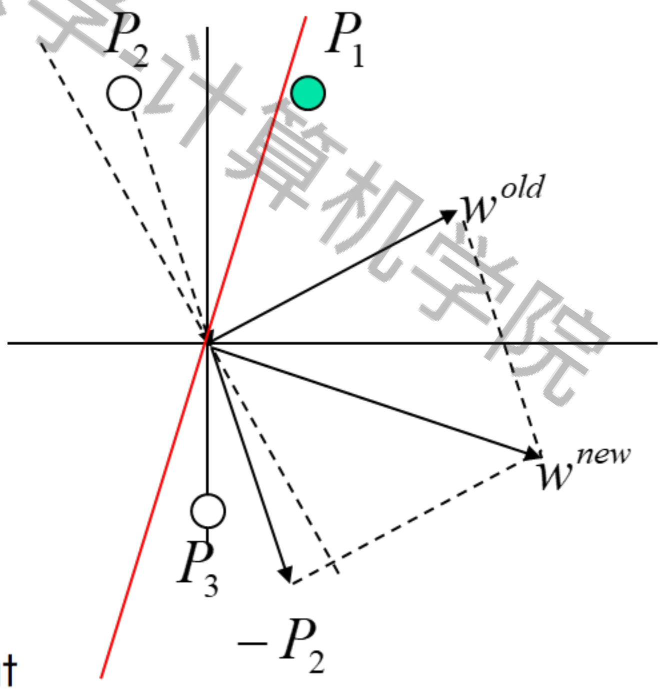


Perceptron – example – step 2

$$w = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix}$$

pass $\mathbf{p}_2$ through the network: $\left\{ \mathbf{p}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, t_2 = 0 \right\}$

$$a = f\left(w^T P_2\right) = f\left(\begin{bmatrix} 2.0 & 1.2 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \right) = f(0.4) = 1$$



Let y be the correct output, and $f(x)$ the output function of the network.

- Error: E $= y - f(x)$
- Update weights: $W_j \leftarrow W_j + \alpha x_j E$

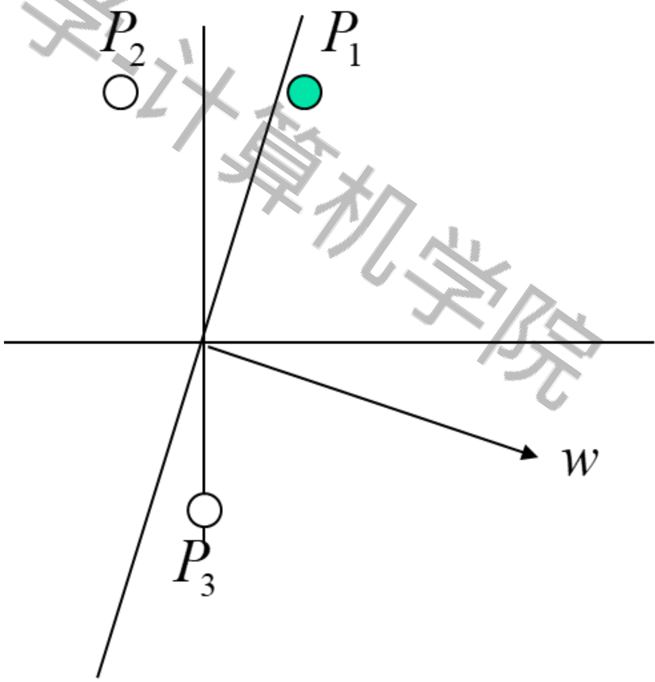
$$w^{new} = w^{old} - p_2 = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix} - \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3.0 \\ -0.8 \end{bmatrix}$$

Perceptron – example – step 3

$$w = \begin{bmatrix} 3.0 \\ -0.8 \end{bmatrix}$$

pass $\mathbf{p}_3$ through the network: $\left\{ \mathbf{p}_3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, t_3 = 0 \right\}$

$$a = f\left(w^T P_3\right) = f\left(\begin{bmatrix} 3.0 & -0.8 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right) = f(0.8) = 1$$

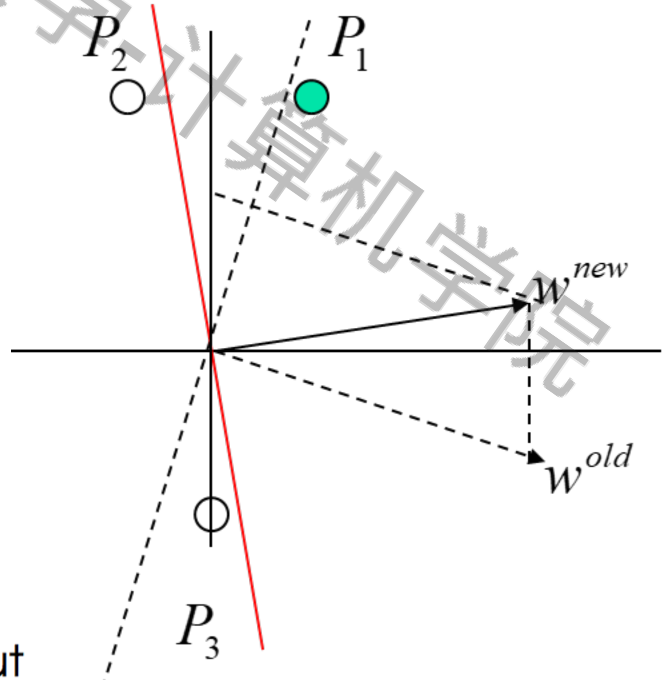


Perceptron – example – step 3

$$w = \begin{bmatrix} 3.0 \\ -0.8 \end{bmatrix}$$

pass $\mathbf{p_3}$ through the network: $\left\{ \mathbf{p}_3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, t_3 = 0 \right\}$

$$a = f\left(w^T P_3\right) = f\left(\begin{bmatrix} 3.0 & -0.8 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right) = f(0.8) = 1$$



Let y be the correct output, and $f(x)$ the output function of the network.

- Error: $\mathrm{E} = y - f(x)$
- Update weights: $W_j \leftarrow W_j + \alpha x_j E$

$$w^{new} = w^{old} - p_3 = \begin{bmatrix} 3.0 \\ -0.8 \end{bmatrix} - \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 3.0 \\ 0.2 \end{bmatrix}$$

提纲

一. Limitations of KNN

二. Neuron

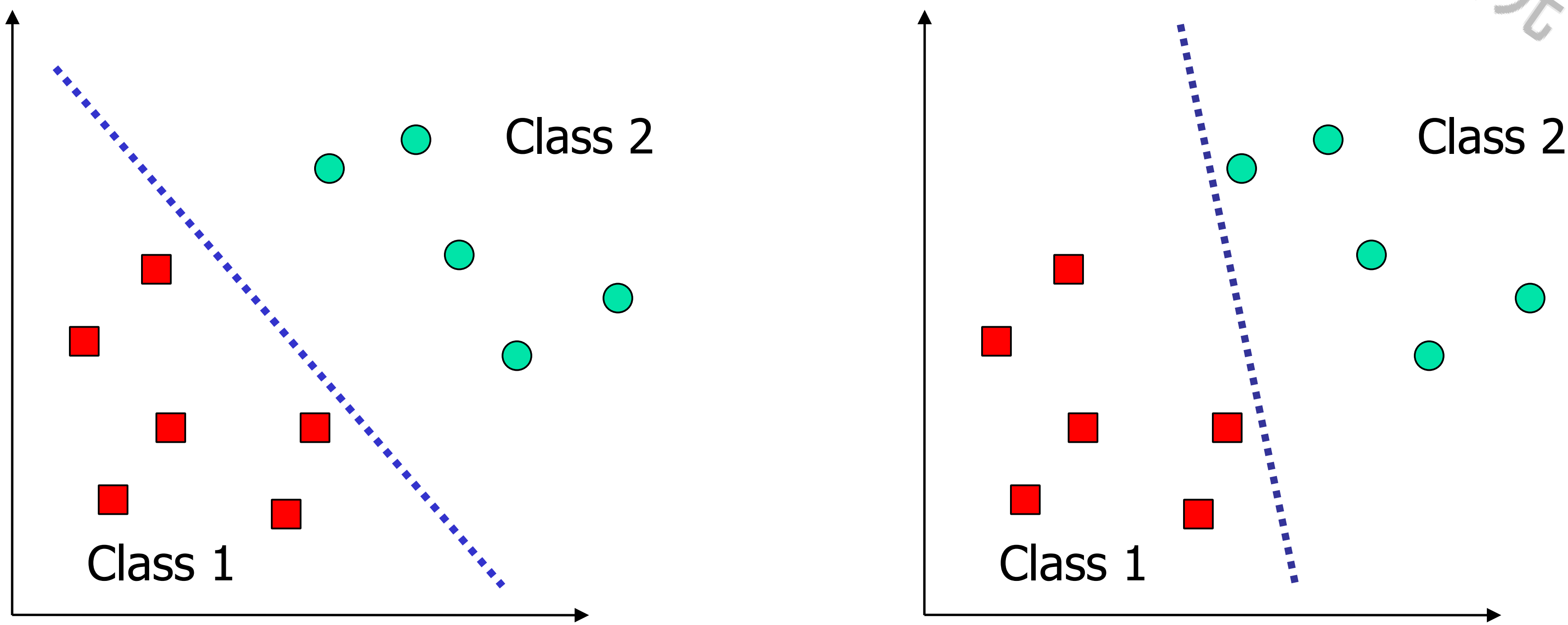
三. Perceptron

四. Maximal Margin Classifier - Linear SVM

Maximal Margin Classifier

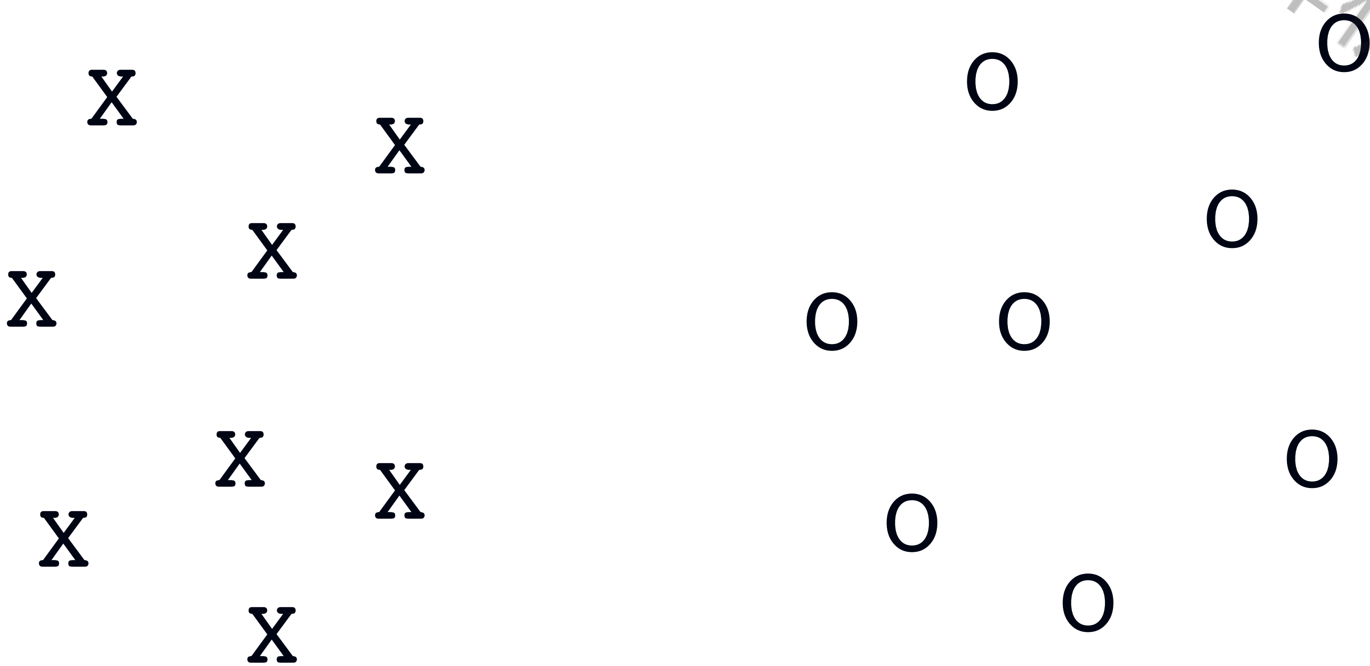
Limitation of Perceptron:

Many decision boundaries can separate these two classes
Which one should we choose?



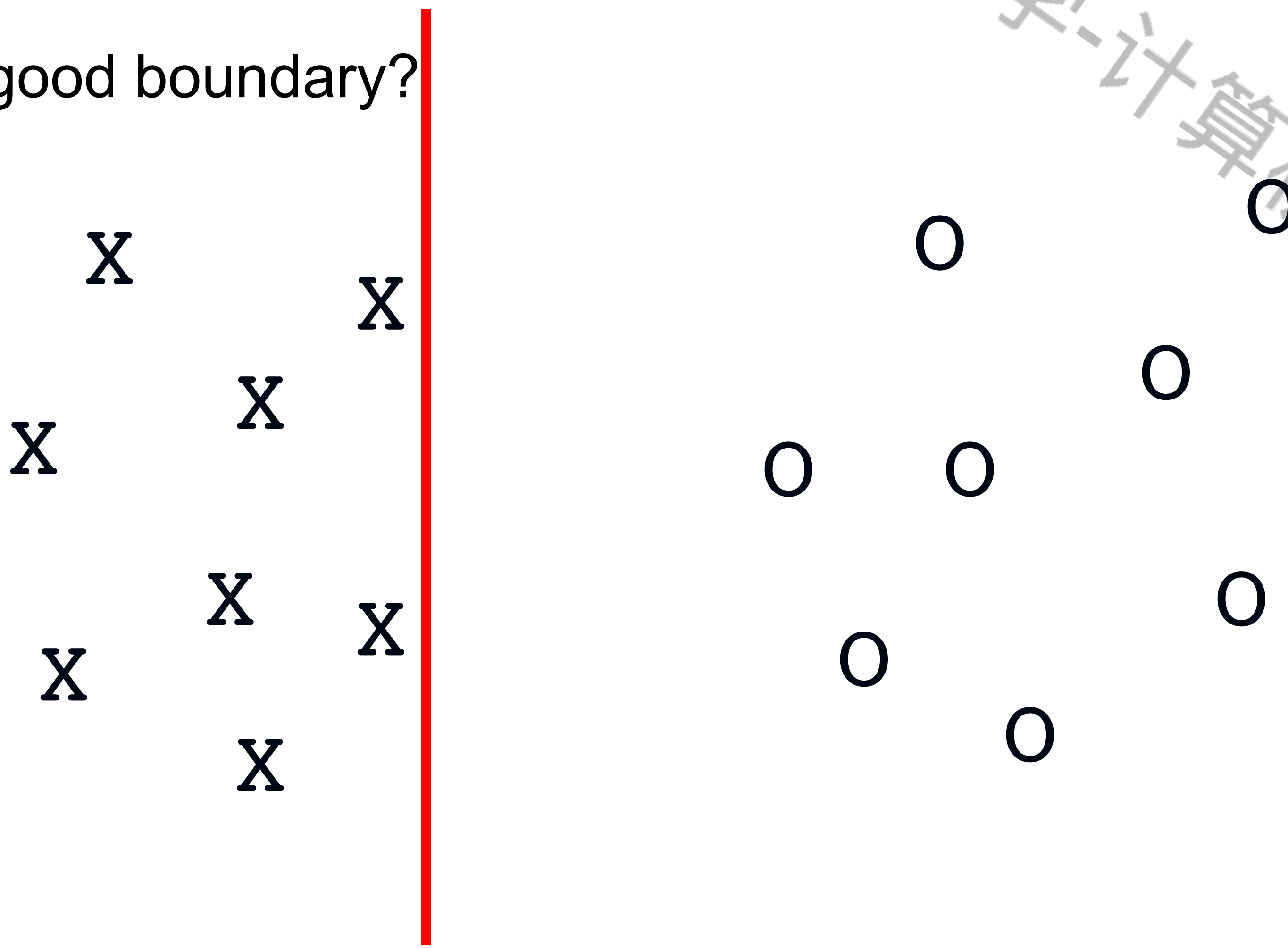
Maximal Margin Classifier

A good boundary?



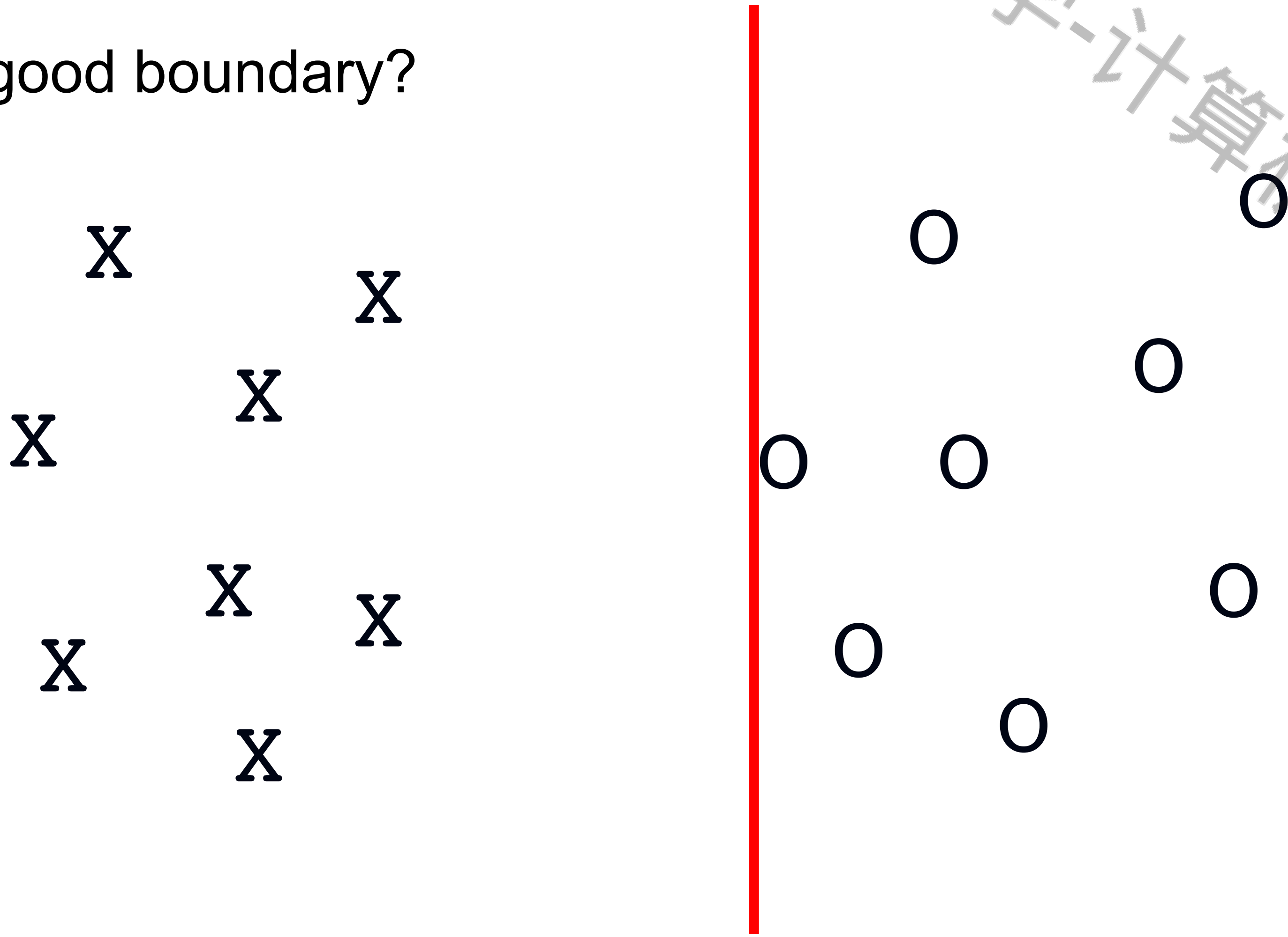
Maximal Margin Classifier

A good boundary?



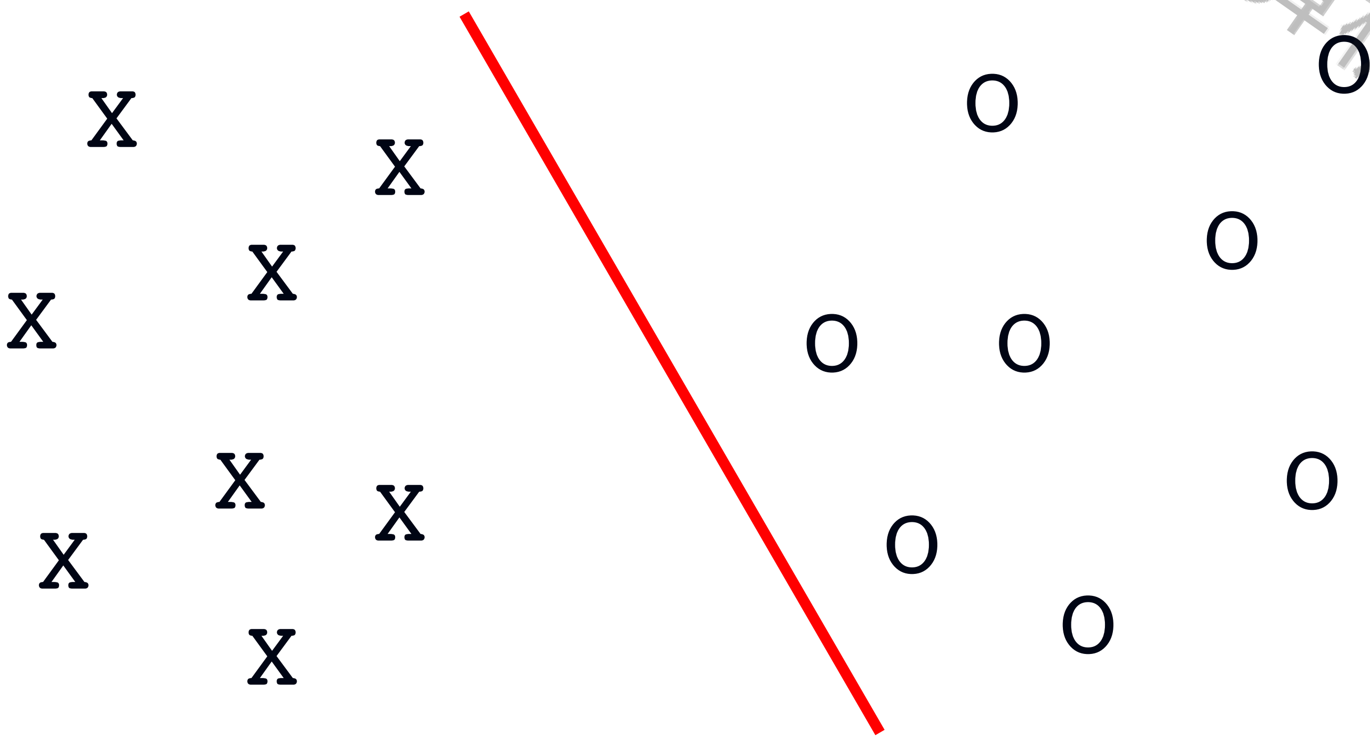
Maximal Margin Classifier

A good boundary?



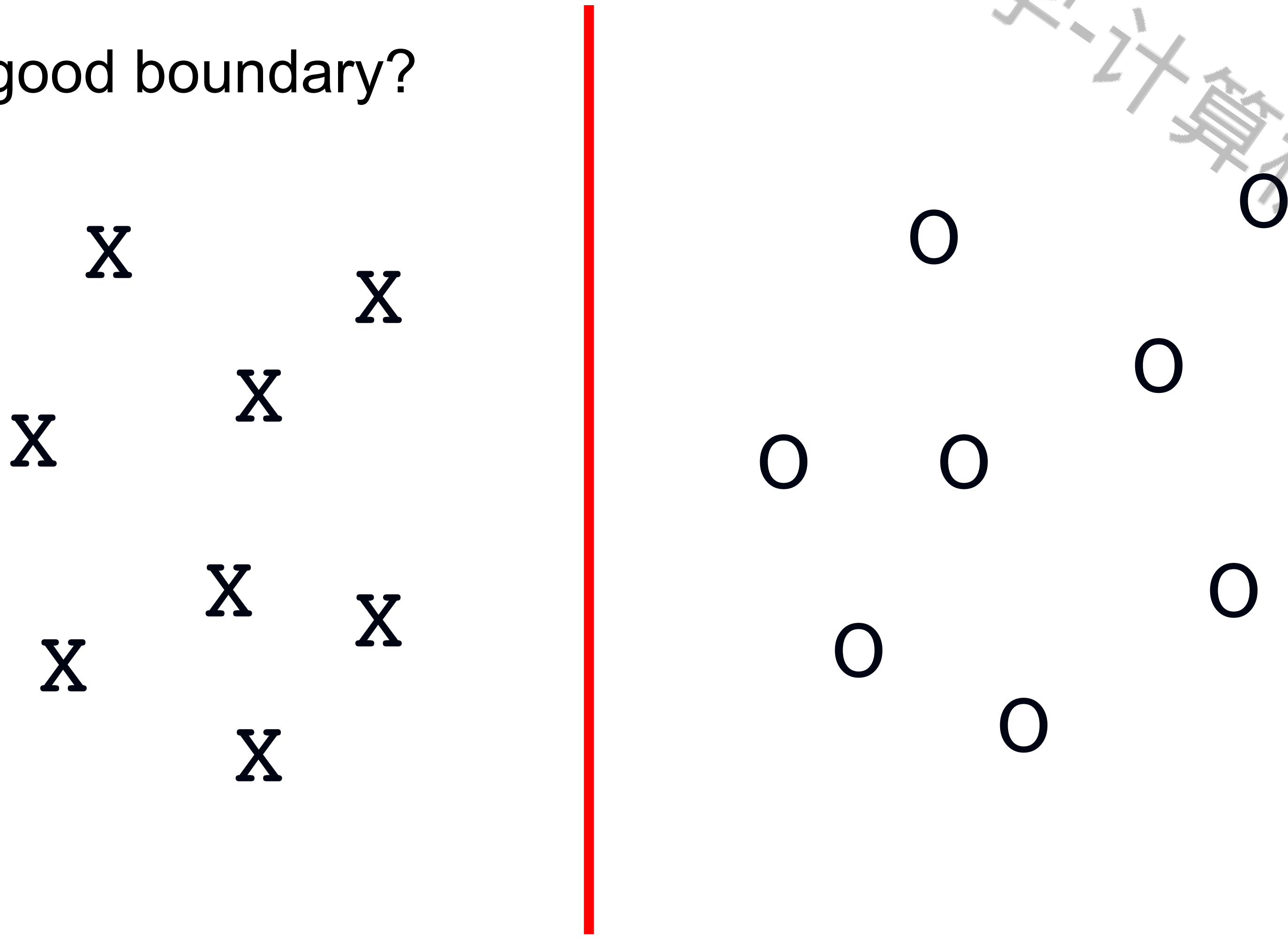
Maximal Margin Classifier

A good boundary?

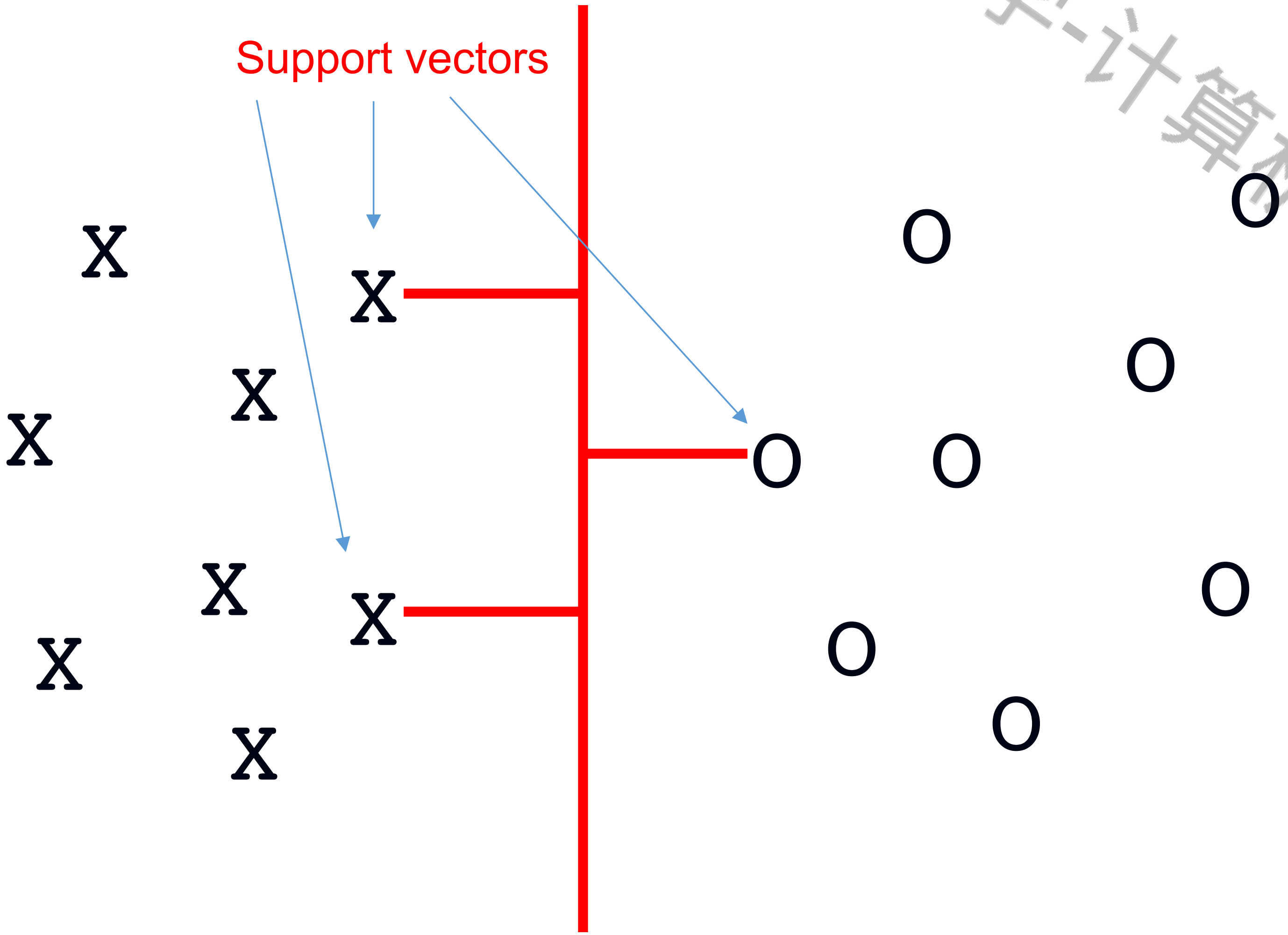


Maximal Margin Classifier

A good boundary?



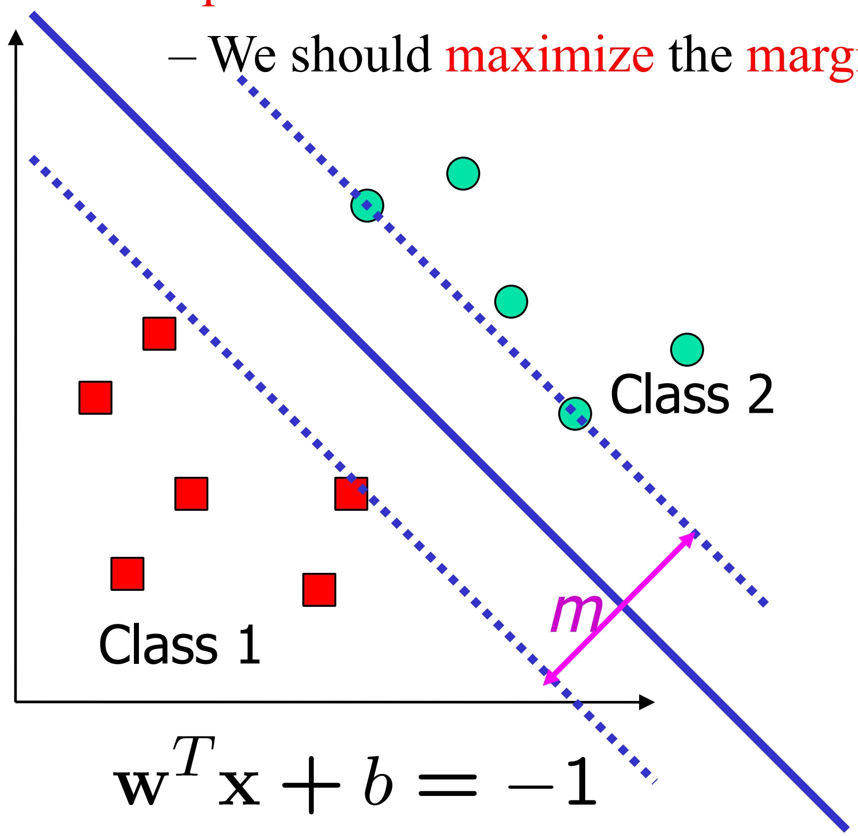
Maximal Margin Classifier



Maximal Margin Classifier

The decision boundary should be as far away from the data of both classes as possible

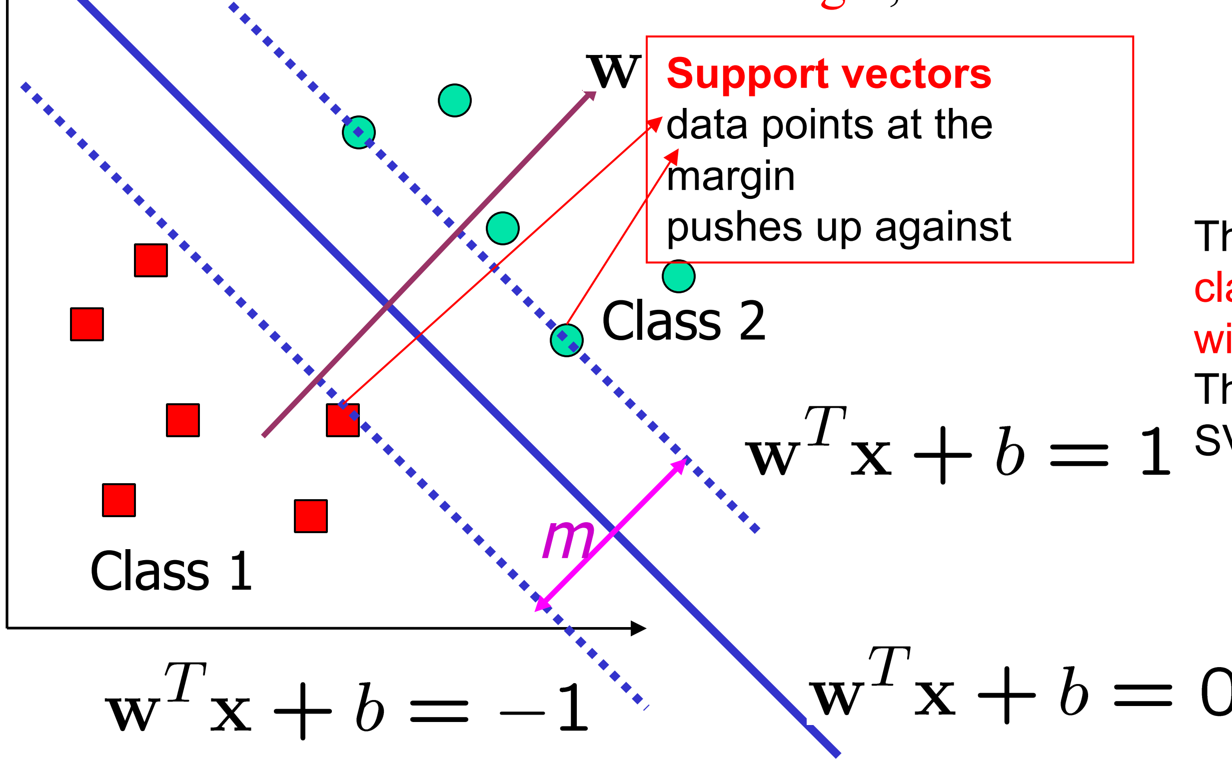
– We should maximize the margin, *m*



Maximal Margin Classifier

The decision boundary should be as far away from the data of both classes as possible

– We should maximize the margin, m

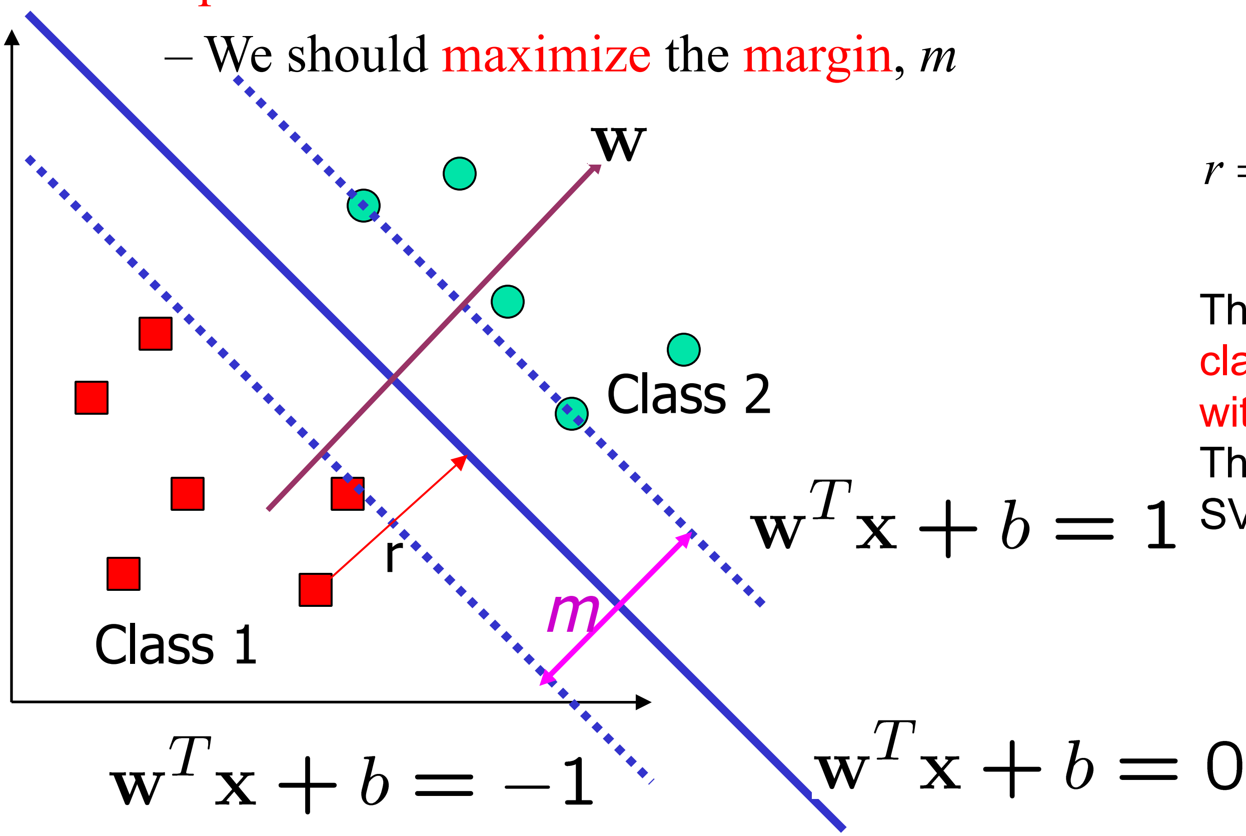


The maximum margin linear classifier is the linear classifier with the maximum margin. This is the simplest kind of SVM (Called an Linear SVM)

Maximal Margin Classifier

The decision boundary should be as far away from the data of both classes as possible

– We should maximize the margin, m

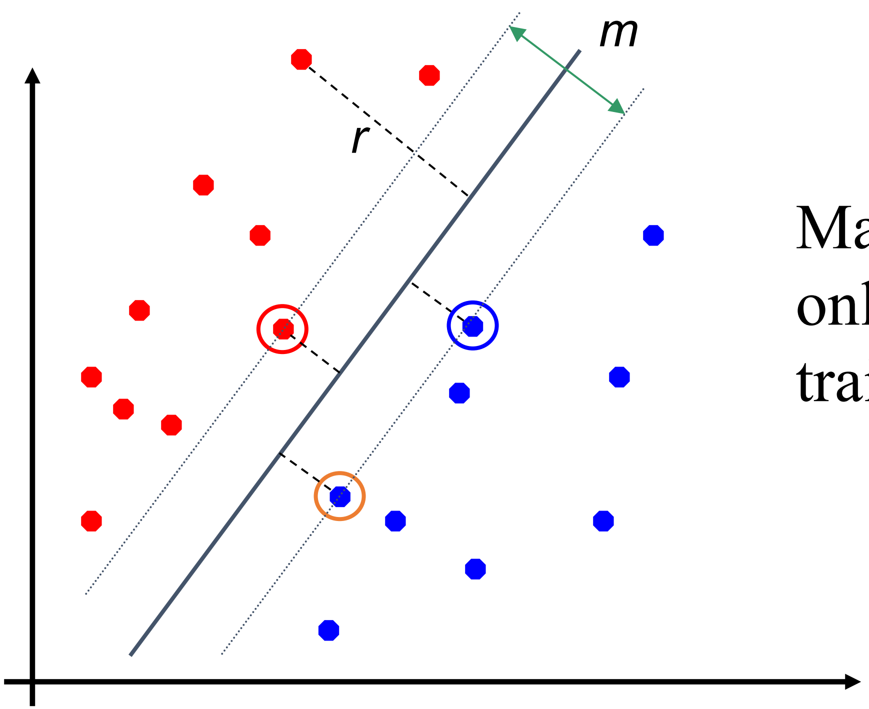


$$r = \frac{\mathbf{w}^T\mathbf{x}_i + b}{\|\mathbf{w}\|}$$

The maximum margin linear classifier is the linear classifier with the maximum margin. This is the simplest kind of SVM (Called an Linear SVM)

Maximal Margin Classifier

- Distance from example $\mathbf{x}_i$ to the separator is $r = \frac{\mathbf{w}^T\mathbf{x}_i + b}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane are ***support vectors***.
- ***Margin*** *m* of the separator is the distance between support vectors.



Maximizing margin implies that only support vectors matter; other training examples are ignorable.

Maximal Margin Classifier

- Let training set $\{(\mathbf{x}_i, y_i)\}_{i=1..n}$, $\mathbf{x}_i \in R^d$, $y_i \in \{-1, 1\}$ be separated by a hyperplane with margin m. Then for each training example $(\mathbf{x}_i, y_i)$:

$$\begin{array}{ll} \mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b \le -m/2 & \text{if } y_i = -1 \\ \mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b \ge m/2 & \text{if } y_i = 1 \end{array} \quad \Leftrightarrow \quad y_i(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b) \ge m/2$$

- For every support vector $\mathbf{x}_s$, the above inequality is an equality. After rescaling $\mathbf{w}$ and b by $m/2$ in the equality, we obtain that distance between each $\mathbf{x}_s$ and the hyperplane is $r = \frac{y_s(\mathbf{w}^T\mathbf{x}_s + b)}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$

- Then the margin can be expressed through (rescaled) $\mathbf{w}$ and b as:

$$m = 2r = \frac{2}{\|\mathbf{w}\|}$$

Maximal Margin Classifier

- Then we can formulate the *quadratic optimization problem:*

Find $\mathbf{w}$ and b such that

$$\rho = \frac{2}{\|\mathbf{w}\|} \text{ is maximized}$$

and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $\quad y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1$

Which can be reformulated as:

Find $\mathbf{w}$ and b such that

$$\mathbf{\Phi}(\mathbf{w}) = ||\mathbf{w}||^2 = \mathbf{w}^{\mathrm{T}}\mathbf{w} \text{ is minimized}$$

and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $\quad y_i\,(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1$

Maximal Margin Classifier

The objective function:

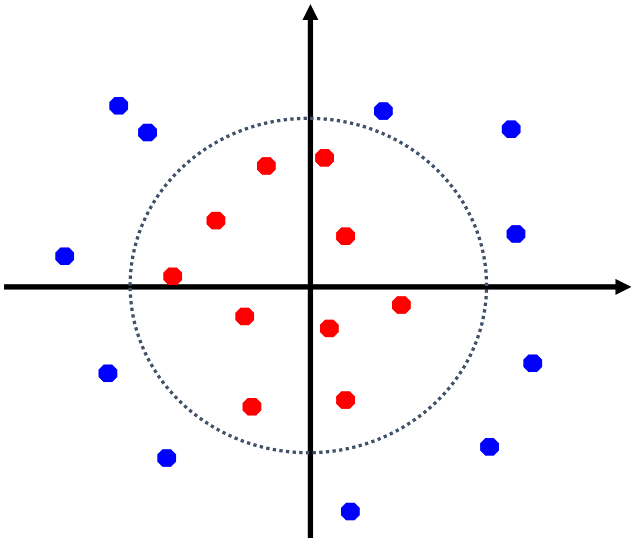
Minimize $\frac{1}{2}||\mathbf{w}||^2$

subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$

Test Questions:

- The evolution of neuron from biology to mathematics?
- The key concepts of Perceptron and its limitations.
- Who is Vladimir N. Vapnik.
- Maximum Margin Principle and why support vector is important?
- How to compute the distance between a given data point to the boundary?
- What limitations the linear SVM suffered from?

others



Q&A

THANKS!