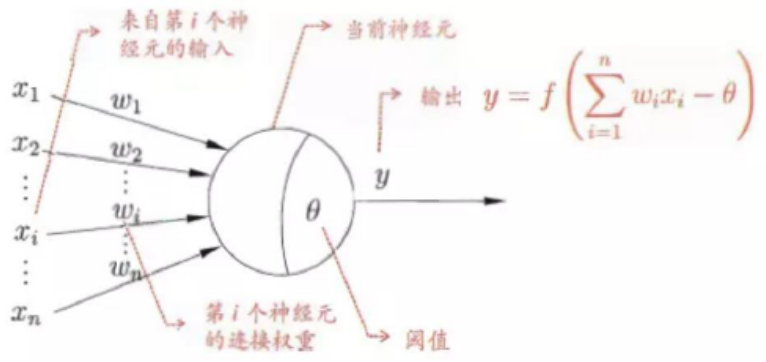
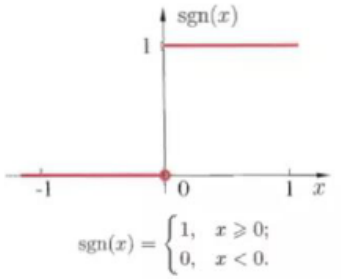
模式识别 U2 感知机

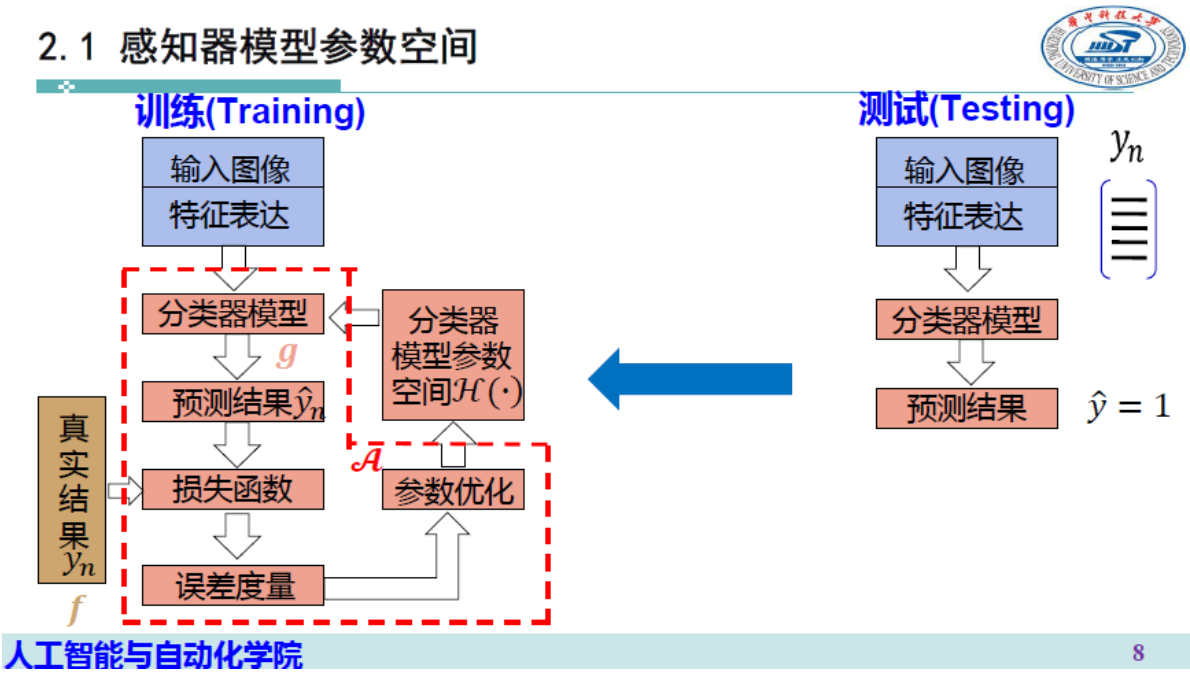
课堂内容

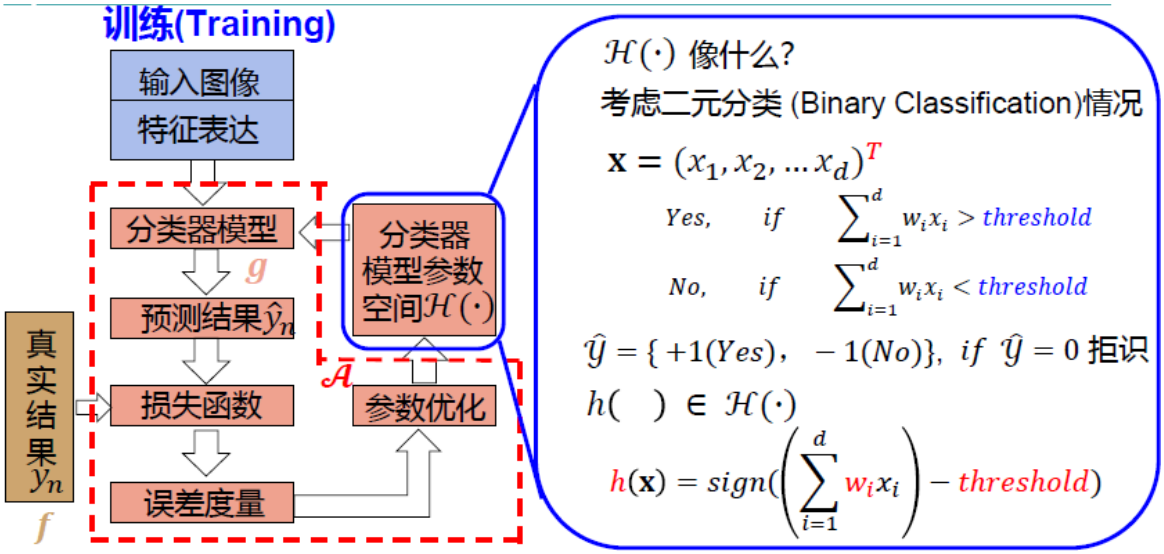
感知器模型参数空间

感知器 Perceptron









用向量形式（Vector Form）来表示感知器模型

$$
\begin{aligned}
h(x) &= sign((\Sigma_{i=1}^{d} w_i x_i) - threshold) \\
&= sign((\Sigma_{i=1}^{d} w_i x_i) + (-threshold) \cdot (+1)) \\
&= sign(\Sigma_{i=0}^{d} w_i x_i) \\
&= sign(\overrightarrow{w^T} \cdot \overrightarrow{x})
\end{aligned}
$$

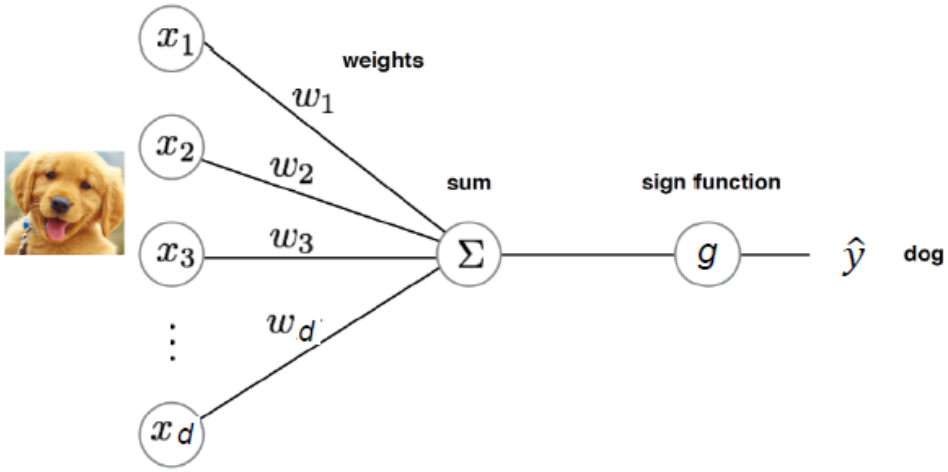
由上式我们可知，我们将阈值threshold扩展进了原来的w权重向量中，使其作为常数偏置存在；在进行这一操作时也在X中扩展出了一维全为1的增广X

我们称新的w，X为增广化后的$\mathbf{w}^T$、$\mathbf{X}$

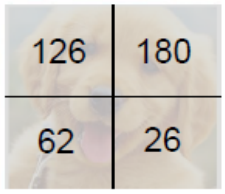
$$
\begin{aligned}
\mathbf{W} &= [w_1, w_2, w_3, \ldots, w_d, w_{d+1}]_{1\times(d+1)} \\
\overrightarrow{x_i} &= [x_1, x_2, x_3, \ldots, x_d, 1]_{1\times(d+1)} \\
\mathbf{X} &= \begin{bmatrix} \overrightarrow{x_1} \\ \overrightarrow{x_2} \\ \overrightarrow{x_3} \\ \ldots \\ \overrightarrow{x_n} \end{bmatrix}_{n\times(d+1)} \\
\text{则}\mathbf{X}\cdot\mathbf{W}^T &= \begin{bmatrix} \overrightarrow{x_1}\cdot\mathbf{W}^T \\ \overrightarrow{x_2}\cdot\mathbf{W}^T \\ \overrightarrow{x_3}\cdot\mathbf{W}^T \\ \cdots \\ \overrightarrow{x_n}\cdot\mathbf{W}^T \end{bmatrix}_{n\times 1} \\
&= \begin{bmatrix} x_1^{(1)}w_1 + x_2^{(1)}w_2 + \ldots + x_d^{(1)}w_d + 1^{(1)}\cdot(w_{d+1}) \\ x_1^{(2)}w_1 + x_2^{(2)}w_2 + \ldots + x_d^{(2)}w_d + 1^{(2)}\cdot(w_{d+1}) \\ x_1^{(3)}w_1 + x_2^{(3)}w_2 + \ldots + x_d^{(3)}w_d + 1^{(3)}\cdot(w_{d+1}) \\ \cdots \\ x_1^{(n)}w_1 + x_2^{(n)}w_2 + \ldots + x_d^{(n)}w_d + 1^{(n)}\cdot(w_{d+1}) \end{bmatrix}_{n\times 1}
\end{aligned}
$$

$\mathbf{W}$是$1\times(d+1)$维，$\mathbf{X}$是$n\times(d+1)$维

将感知器算法用于图像分类(*image classification*)示例：



将感知器算法用于图像分类(*image classification*)示例：



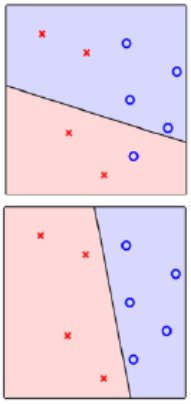
$$\mathbf{x} = \begin{pmatrix} 126 \\ 180 \\ 62 \\ 26 \end{pmatrix} \rightarrow \mathbf{x} = \begin{pmatrix} 1 \\ 126 \\ 180 \\ 62 \\ 26 \end{pmatrix}$$

$$if: \mathbf{w}^T = (3.2 \quad 1.5 \quad 1.3 \quad 2.1 \quad 0.8)$$

$$\hat{y} = h(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x}) = sign((3.2 \quad 0.5 \quad 1.3 \quad 2.1 \quad 0.8)\begin{pmatrix} 1 \\ 126 \\ 180 \\ 62 \\ 26 \end{pmatrix})$$
$$= sign(451.2) = 1$$

在二维空间观察感知器的分类面(*候选的*) $\mathcal{H}(\cdot)$

$$h(\mathbf{x}) = sign(w_0 + w_1x_1 + w_2x_2)$$



样本 $\mathbf{x}$ ➡ 用平面(或者 $\mathbf{R}^d$ 超平面)上的点表示

标签 y ➡ ○ (+1), × (-1)

$h(\mathbf{x})$ ➡ 平面上的线(或者 $\mathbf{R}^d$ 上的超平面)

平面被分成两个区域，分别表示+1类和-1类所在的区域

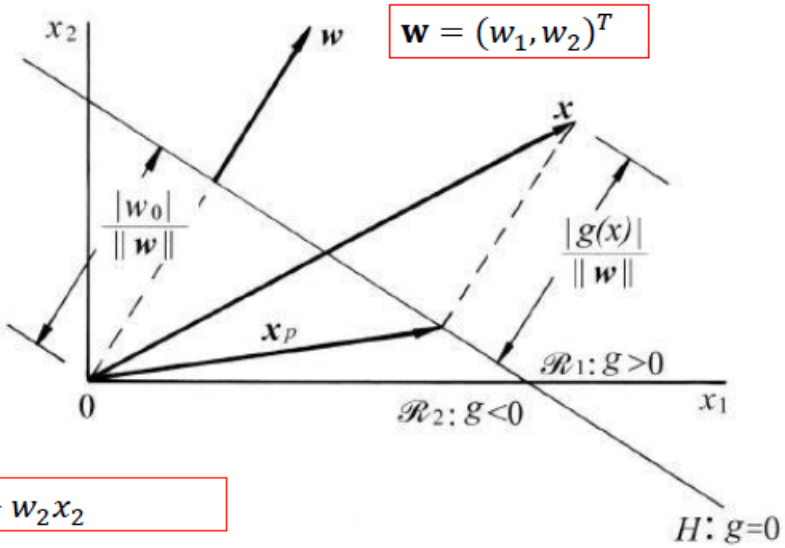
不同的线有可能将同一样本分到不同的类别中去

感知器也被称作二元线性分类器(*binary linear classifiers*)

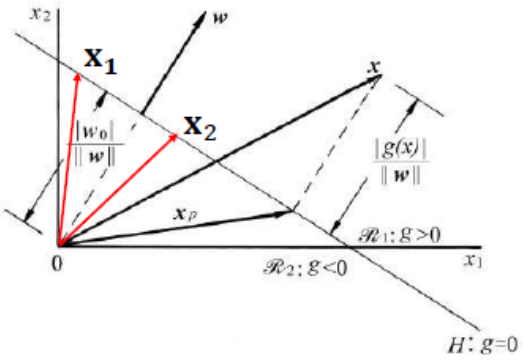
在高维空间中感知器的分类面

$$h(x) = sign(w_0 + w_1x_1 + w_2x_2 + \ldots + w_dx_d)$$
$$= sign(\sum_{i=0}^{d} w_ix_i)$$
$$= sign(W \cdot \mathbf{x})$$

用向量几何知识来分析感知器模型



$g(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2$



$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$ $\quad$ $\mathbf{w} = (w_1, w_2)^T$

假定 $\mathbf{x_1}$ 和 $\mathbf{x_2}$ 在分类面上：

$$g(\mathbf{x_1}) = \mathbf{w}^T\mathbf{x_1} + w_0 = 0$$
$$g(\mathbf{x_2}) = \mathbf{w}^T\mathbf{x_2} + w_0 = 0$$

$$\mathbf{w}^T(\mathbf{x_1} - \mathbf{x_2}) = 0$$

$$\mathbf{w} \perp (\mathbf{x_1} - \mathbf{x_2})$$

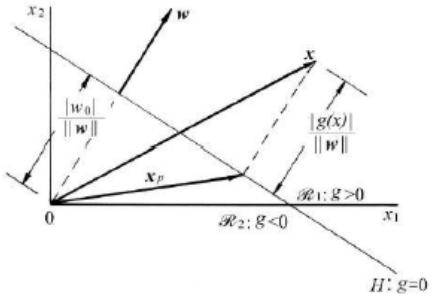
$g(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2$

权向量w垂直于分类面

用向量几何知识来分析感知器模型

如果 $\mathbf{x} \in D$ 在特征平面上，$\mathbf{x}_p$ 在分类面上

r 表示 $\mathbf{x}$ 与分类面 $(g(\mathbf{x}_p) = 0)$ 之间的距离



$$\mathbf{x} = \mathbf{x}_p + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$$

$$\begin{aligned} g(\mathbf{x}) &= \boldsymbol{w}^T\mathbf{x} + w_0 \\ &= \boldsymbol{w}^T\left(\mathbf{x}_p + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right) + w_0 \\ &= \boldsymbol{w}^T\,\mathbf{x}_p + w_0 + r\frac{\boldsymbol{w}^T\boldsymbol{w}}{\|\boldsymbol{w}\|} \\ &= r\|\boldsymbol{w}\| \end{aligned}$$

$$g(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2$$

几何知识：二维中 点到直线的距离

$$l: ax_1 + bx_2 + c = 0$$

则距离

$$d = \frac{|ax_{p1} + bx_{p2} + c|}{\sqrt{a^2 + b^2}}$$

扩展到如今向量几何当中

$$d = r\frac{\mathbf{w}}{||\mathbf{w}||}$$

其中 $\frac{\mathbf{w}}{||\mathbf{w}||}$ 表示单位法向量，r则可以用标量指示距离

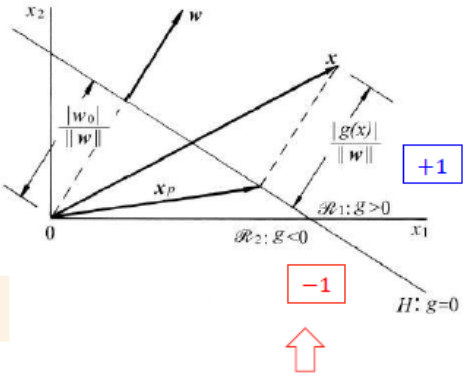
$P.\ ||\mathbf{w}||$指向量的模，也可理解为向量$\mathbf{w}$的L_2范数

用向量几何知识来分析感知器模型

$$r = \frac{g(\mathbf{x})}{\|w\|}$$

当 $\mathbf{x} = \mathbf{0}$，$g(\mathbf{x}) = w_0 \Rightarrow r = \frac{w_0}{\|\mathbf{w}\|}$

即：原点到分类面的距离为 $r = \frac{w_0}{\|w\|}$



- 当 $w_0 > 0$，原点处于分类面的正区域
- 当 $w_0 < 0$，原点处于分类面的负区域
- 当 $w_0 = 0$，分类面穿过原点

$$h(\mathbf{x}) = sign(w_0 + w_1x_1 + w_2x_2)$$
$$= sign(g(\mathbf{x}))$$

有上述推导我们可得：

$$r = \frac{g(\mathbf{x})}{||\mathbf{w}||}$$

其中

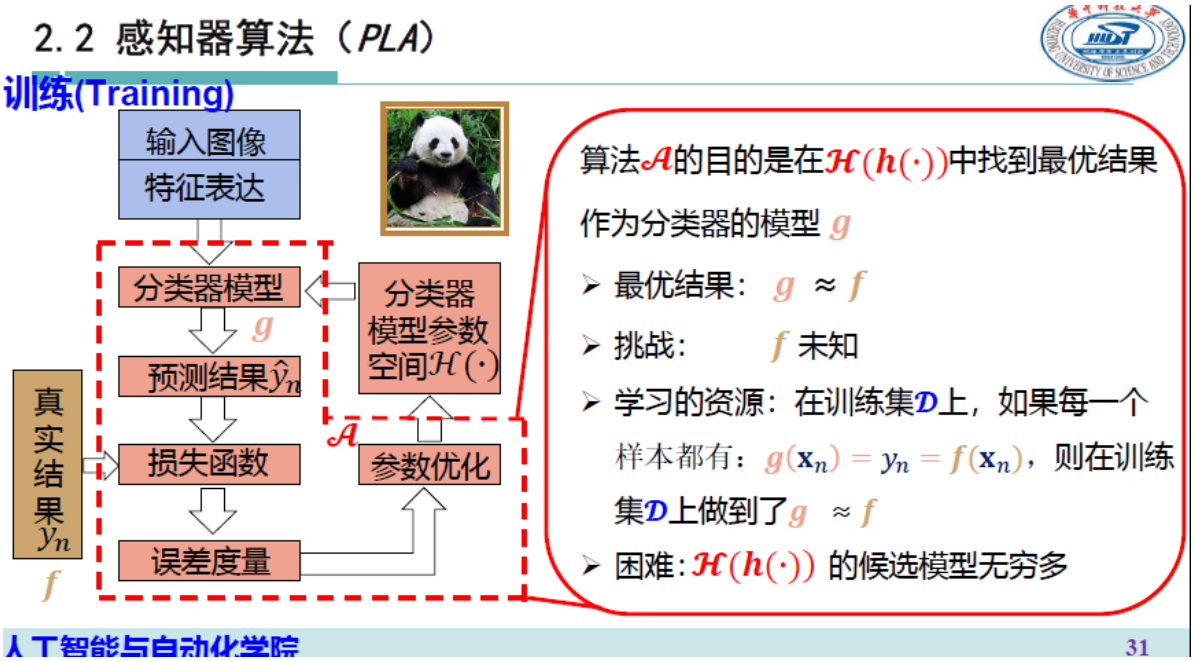
$\mathbf{w}$是训练得到的感知器模型，其本质是可学习迭代的参数集合；$g(\mathbf{x})$则是将该数据点代入模型中取得的结果

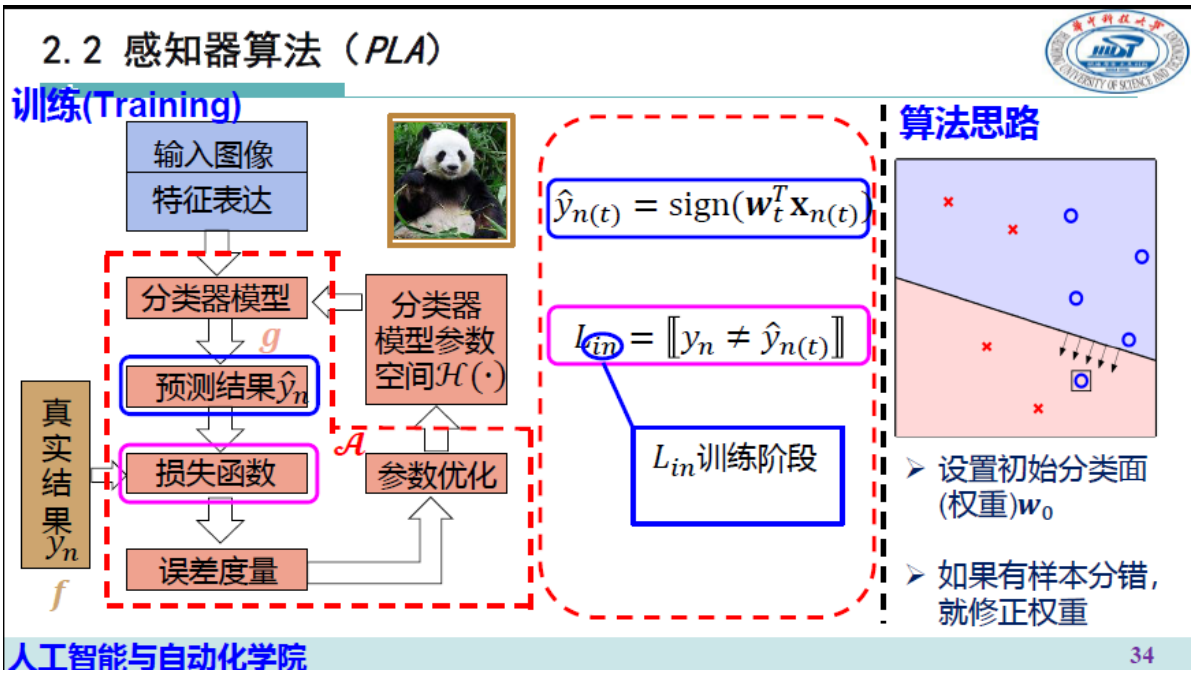
$$W^T \cdot X = ||W|| * ||X|| cos\theta$$

感知器算法 PLA

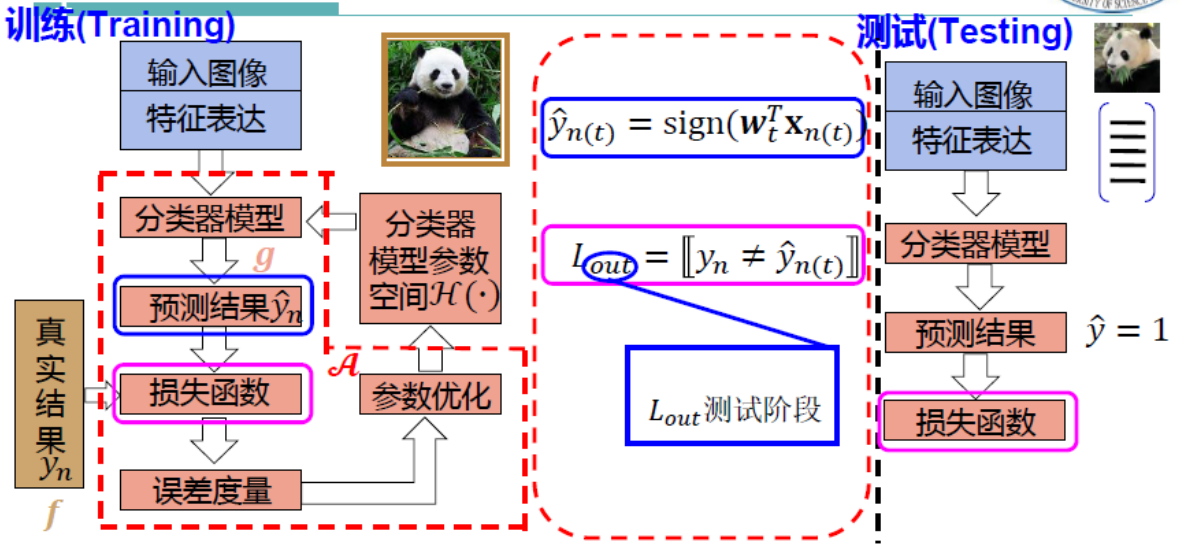
PLA（Perceptron Learning Algorithm）

算法思路

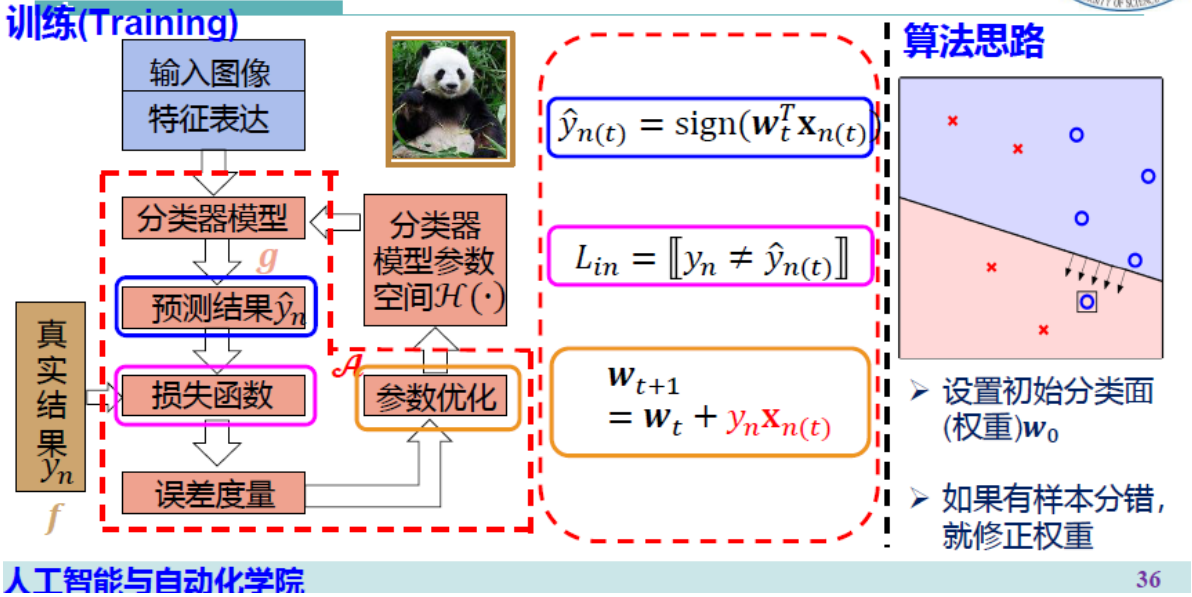

2.2 感知器算法（PLA）
训练(Training)
输入图像
特征表达
分类器模型
g
预测结果$\hat{y}_n$
损失函数
误差度量
真实结果y_n
f
分类器模型参数空间$\mathcal{H}(\cdot)$
参数优化
$\mathcal{A}$
算法$\mathcal{A}$的目的是在$\mathcal{H}(h(\cdot))$中找到最优结果
作为分类器的模型 g
最优结果： $g \approx f$
挑战： f 未知
学习的资源：在训练集$\mathcal{D}$上，如果每一个样本都有：$g(\mathbf{x}_n) = y_n = f(\mathbf{x}_n)$，则在训练集$\mathcal{D}$上做到了$g \approx f$
困难：$\mathcal{H}(h(\cdot))$ 的候选模型无穷多
人工智能与自动化学院
31


2.2 感知器算法（PLA）
训练(Training)
输入图像
特征表达
分类器模型
g
预测结果$\hat{y}_n$
损失函数
误差度量
真实结果y_n
f
分类器模型参数空间$\mathcal{H}(\cdot)$
参数优化
$\mathcal{A}$
$\hat{y}_{n(t)} = \text{sign}(\boldsymbol{w}_t^T \mathbf{x}_{n(t)})$
$L_{in} = [\![y_n \neq \hat{y}_{n(t)}]\!]$
L_{in}训练阶段
算法思路
设置初始分类面(权重)$\boldsymbol{w}_0$
如果有样本分错，就修正权重
人工智能与自动化学院
34

2.2 感知器算法（PLA）



2.2 感知器算法（PLA）



算法流程

- 对样本的特征向量x和权向量$\boldsymbol{w}$ 增广化
- 初始化权向量$\boldsymbol{w}_0$ (例如： $\boldsymbol{w}_0 = \mathbf{0}$)
- ***for*** $t = 0,1,2,\ldots$ (t 代表迭代次数)

① 进行到第t 次迭代时权向量为$\boldsymbol{w}_t$，它对样本$(\mathbf{x}_{n(t)}, y_{n(t)})$错分

$$\text{sign}(\boldsymbol{w}_t^T \mathbf{x}_{n(t)}) \neq y_n$$

② 通过下式对权向量$\boldsymbol{w}_t$进行更新： $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_n \mathbf{x}_{n(t)}$

…直到所有样本均能被正确分类，此时的$\boldsymbol{w}_{t+1}$作为学到的g

回顾感知器算法：

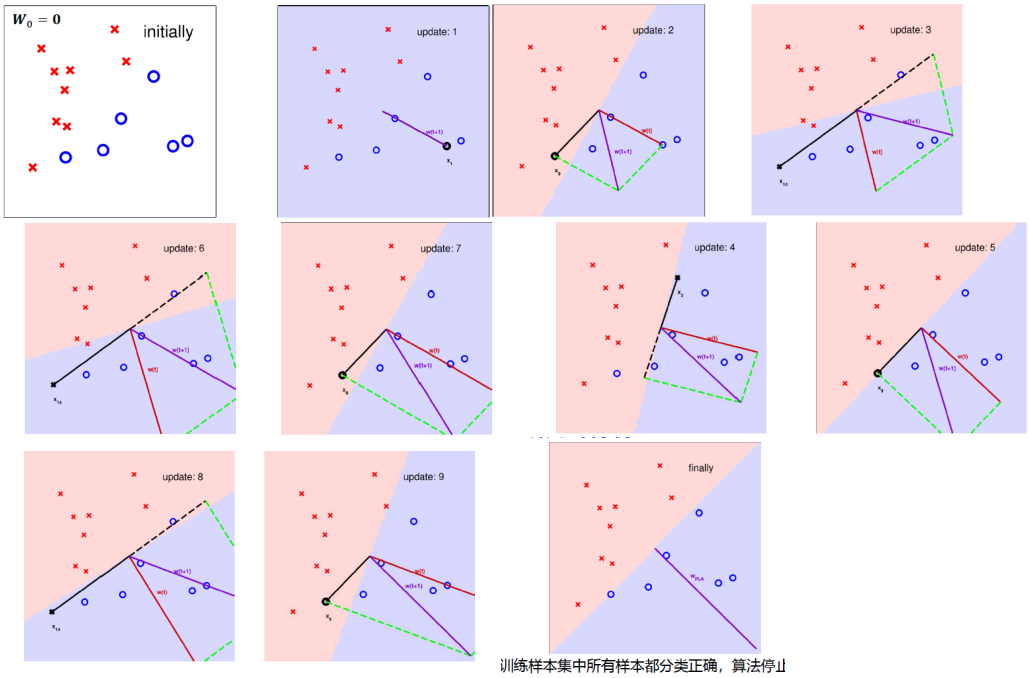
- 对样本的特征向量x和权向量$\boldsymbol{w}$ 增广化
- 初始化权向量$\boldsymbol{w}_0$ (例如： $\boldsymbol{w}_0 = \mathbf{0}$)
- ***for*** $t = 0,1,2,\ldots$ (t 代表迭代次数)

① 对某些样本n，通过下式对权向量$\boldsymbol{w}_t$进行更新：

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + 1 \cdot (\llbracket \text{sign}(\boldsymbol{w}_t^T \mathbf{x}_{n(t)}) \neq y_n \rrbracket \mathbf{x}_{n(t)})$$

…直到满足停止条件，此时的$\boldsymbol{w}_{t+1}$作为学到的g

算法迭代示例



训练样本集中所有样本都分类正确，算法停止

算法问题

问题1：算法会收敛吗？

结果与输入样本顺序是否有关？

问题2：能学到 $g \approx f$ 吗？

- *在$\mathcal{D}$上，如果$L_{in} = 0$，$g \approx f$？*
- *不在$\mathcal{D}$上时，$L_{out} = 0$？$g \approx f$？*
- *算法不能收敛时，$g \approx f$？*

感知器算法的收敛性

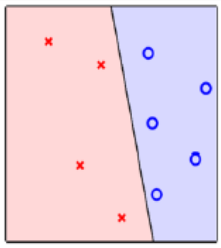
PLA收敛条件：数据集中所有样本线性可分

线性可分性 (*linear separable*)

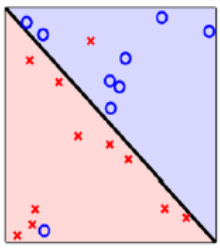
PLA算法收敛 ⟺ 算法停止 ⟺ $\boldsymbol{w}$ 对 $\mathcal{D}$ 上所有样本正确分类

$\sqrt{}$

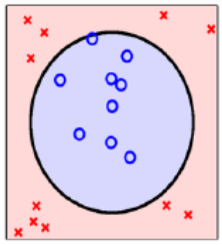
$\mathcal{D}$ 上所有样本是线性可分的



线性可分



线性不可分



线性不可分

所有样本线性可分是否意味着PLA一定收敛?

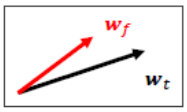
假设 $\boldsymbol{w}_f$ 是理想的分类面:

$\mathcal{D}$ 是线性可分的 ⟺ $y_n = sign(\boldsymbol{w}_f^T \mathbf{x}_n)$

第t次迭代时，任意一个样本$\mathbf{x}_{n(t)}$满足

$$y_{n(t)} \boldsymbol{w}_f^T \mathbf{x}_{n(t)} \geq \min_n y_n \boldsymbol{w}_f^T \mathbf{x}_n > 0$$

迭代次数增加



$$\boldsymbol{w}_f^T \boldsymbol{w}_{t+1} = \boldsymbol{w}_f^T (\boldsymbol{w}_t + y_{n(t)} \mathbf{x}_{n(t)}) \geq \boldsymbol{w}_f^T \boldsymbol{w}_t + \min_n y_n \boldsymbol{w}_f^T \mathbf{x}_n > \boldsymbol{w}_f^T \boldsymbol{w}_t + 0$$

结论：随着迭代次数增加， $\boldsymbol{w}_f^T \boldsymbol{w}_t$随之增加，意味着$\boldsymbol{w}_t$与 $\boldsymbol{w}_f$越来越接近

假设 $\boldsymbol{w}_t$ 是第 t 次迭代得到的分类面:

当 $\mathbf{x}_n$被分类错误时，才更新 $\boldsymbol{w}_t$ ⟺ $sign(\boldsymbol{w}_t^T \mathbf{x}_n) \neq y_n$ ⟺ $y_{n(t)} \boldsymbol{w}_t^T \mathbf{x}_{n(t)} \leq 0$

假设 $\mathbf{x}_n$在样本集中模值最大，随着迭代次数增加，$\|\boldsymbol{w}_t\|$?

$$\begin{aligned} \|\boldsymbol{w}_{t+1}\|^2 &= \|\boldsymbol{w}_t + y_{n(t)} \mathbf{x}_{n(t)}\|^2 \\ &= \|\boldsymbol{w}_t\|^2 + 2 y_{n(t)} \boldsymbol{w}_t^T \mathbf{x}_{n(t)} + \|y_{n(t)} \mathbf{x}_{n(t)}\|^2 \\ &\leq \|\boldsymbol{w}_t\|^2 + 0 + \|y_{n(t)} \mathbf{x}_{n(t)}\|^2 \leq \|\boldsymbol{w}_t\|^2 + \max_n \|y_n \mathbf{x}_n\|^2 \end{aligned}$$

结论：随着迭代次数增加， $\boldsymbol{w}_t$模值增长不会太快， 意味着$\boldsymbol{w}_t$与 $\boldsymbol{w}_f$的接近是方向上在靠近，而非模值的贡献

课后证明（作业）：

(1) 针对线性可分训练样本集，*PLA*算法中，当$\boldsymbol{w}_0 = \mathbf{0}$，在对分错样本进行了*T*次修正后，下式成立：$\frac{\mathbf{w}_f^T}{\|\mathbf{w}_f\|}\frac{\mathbf{w}_T}{\|\mathbf{w}_T\|} \geq \sqrt{T} \cdot constant$

(2) 针对线性可分训练样本集，*PLA*算法中，假设对分错样本进行了*T*次修正后，得到的分类面不再出现错分状况，定义：$R^2 = \max_n \|\mathbf{x}_n\|^2$，$\rho = \min_n y_n \frac{\mathbf{w}_f^T}{\|\mathbf{w}_f\|}\mathbf{x}_n$，证明：$\mathrm{T} \leq \frac{R^2}{\rho^2}$

线性不可分情况

线性不可分分析

算法收敛性保证：

$\mathcal{D}$ 是线性可分的

$\boldsymbol{w}_f^T \boldsymbol{w}_t$随$t$ 增加而增加

$\mathbf{x}_n$被分类错误时才更新 $\boldsymbol{w}_t$

$\boldsymbol{w}_t$模值增长不会太快

算法学习的分类面与 $\boldsymbol{w}_f$越来越接近

算法收敛

算法的长处：

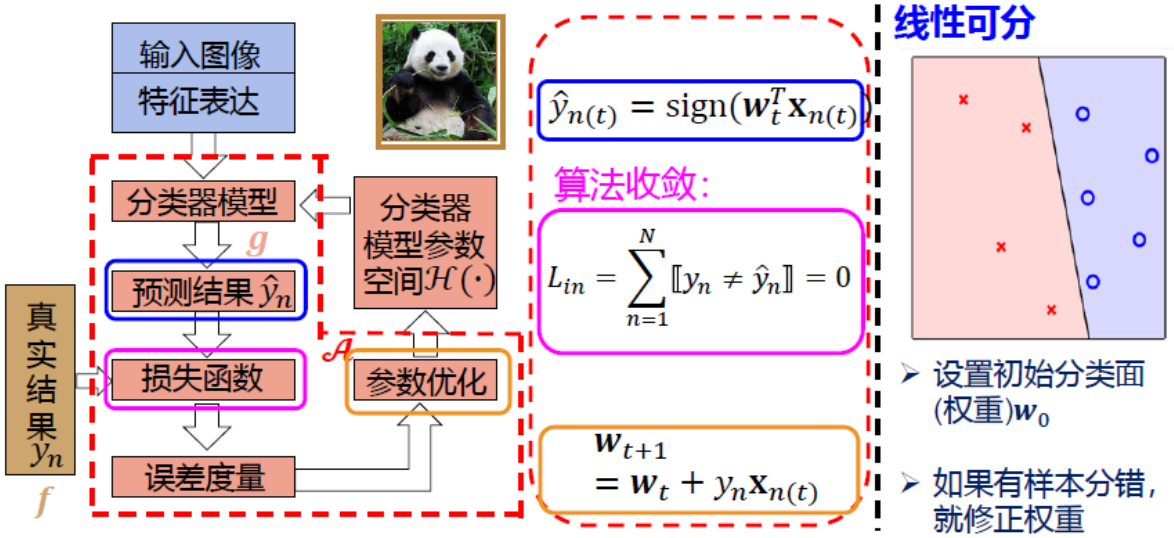
- 实现简单
- 运行快速
- 适用于任意维数d

算法的不足：

- $\mathcal{D}$ 线性可分，算法收敛
  --是否线性可分，事先未知
- $\mathrm{T} \leq \frac{\max_n \|\mathbf{x}_n\|^2}{(\min_n y_n \frac{\mathbf{w}_f^T}{\|\mathbf{w}_f\|}\mathbf{x}_n)^2}$，T有界
  --$\mathbf{W}_f$未知，实际无法得到T

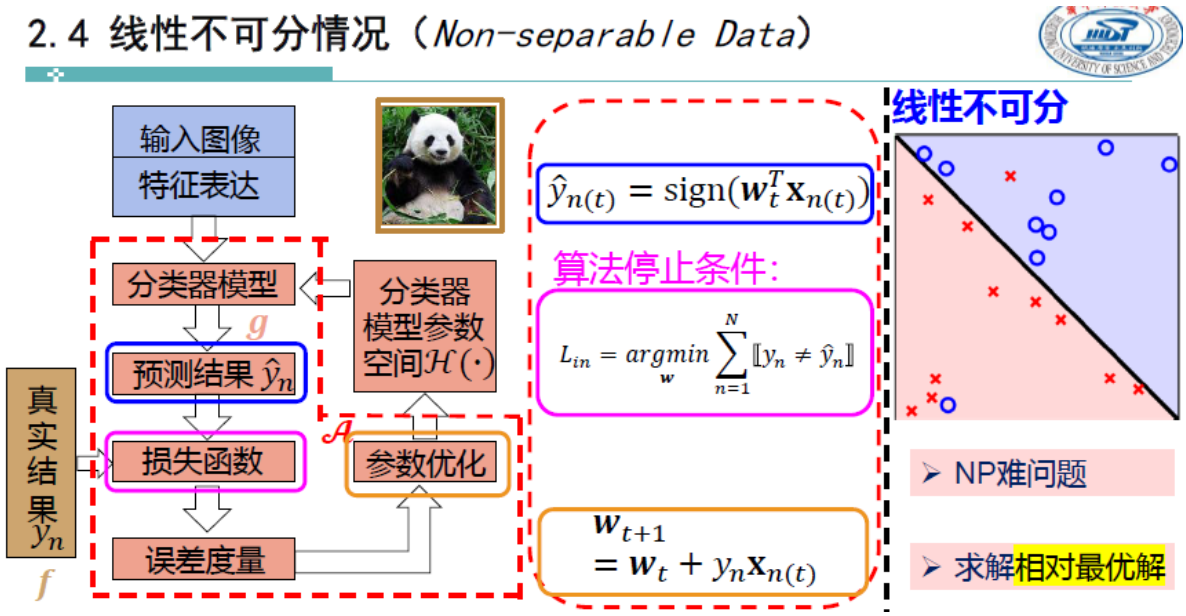
如果 $\mathcal{D}$ 是线性不可分的，算法不收敛，如何处理?

对于**线性可分**情况



模型的最终目的是实现收敛，即全部样本完全正确划分

对于**线性不可分**情况



调整模型算法停止条件为:损失函数最小

Pocket算法

为处理线性不可分情况而对PLA算法的修正

***Pocket* 算法 –** *为处理线性不可分情况而对PLA算法的修正*

- 对样本的特征向量$\mathbf{x}$和权向量$\boldsymbol{w}$ 增广化
- 初始化权向量$\boldsymbol{w}_0$ (例如： $\boldsymbol{w}_0 = \mathbf{0}$)，并任意选一个“Pocket”向量$\hat{\boldsymbol{w}}$
- ***for*** $t = 0,1,2,\ldots$ (t 代表迭代次数)
  ① 进行到第t 次迭代时权向量为$\boldsymbol{w}_t$，它对样本$(\mathbf{x}_{n(t)}, y_{n(t)})$错分

$$\text{sign}(\boldsymbol{w}_t^T \mathbf{x}_{n(t)}) \neq y_n$$

  ② 通过下式对权向量$\boldsymbol{w}_t$进行更新： $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_n \mathbf{x}_{n(t)}$

  ③ 如果 $\boldsymbol{w}_{t+1}$在所有样本集上错分的样本少于$\hat{\boldsymbol{w}}$，则用$\boldsymbol{w}_{t+1}$代替$\hat{\boldsymbol{w}}$，并在错分样本中随机选一个对权向量进行更新

…达到指定的迭代次数

返回此时的 “Pocket”向量$\hat{\boldsymbol{w}}$ 作为算法学到的g

Pocket算法

- 贪心思想对PLA的优化：每次更新权值前，先遍历所有点，如果错误点数量下降，则更新权值，否则不更新
  - 在选择修正点时存在随机性，迭代次数可能比PLA小也可能比PLA大，每次找到的直线也不相同
  - 执行时间比PLA长，这是因为在每次调整参数后需要去比对修正后与当前最优参数的结果好坏
  - Pocket算法可以保存最优参数，当得到相应迭代次数或者完全正确分类的时候停止
  - 在足够的迭代次数的情况下，便可以找到全局最优解，对于线性不可分数据集，也可以在迭代到相应次数时，停止算法，输出最优结果

小结

2.1 感知器模型参数空间

在R^d空间的超平面的线性分类面

2.2 感知器算法（*PLA*）

通过迭代的方式对错分样本的分类面进行修正

2.3 感知器算法的收敛性

如果训练样本集是线性可分的，算法能对所有的样本正确分类并停止

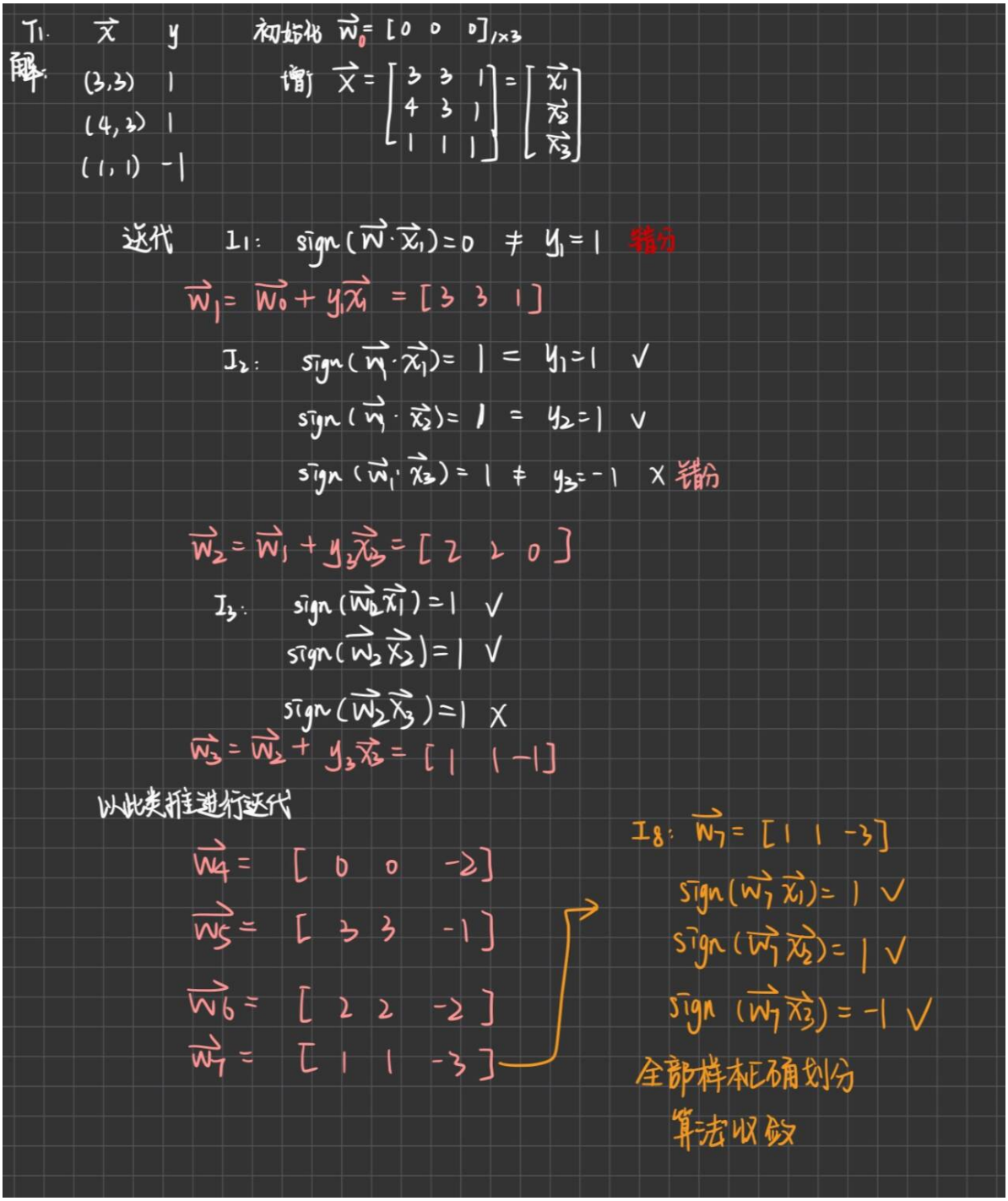
2.4 线性不可分情况

通过“Pocket”算法在设置的迭代次数下寻找相对最佳的分类面

作业部分

手写作业

1，假设训练样本集为 $D=\{(\mathbf{x}_1,y_1)=((3,3)^T,1),(\mathbf{x}_2,y_2)=((4,3)^T,1),(\mathbf{x}_3,y_3)=((1,1)^T,-1)\}$，使用感知器算法设计分类面，并判断测试样本$\mathbf{x}=(0,1)^T$属于哪个类别。↵



2，对于感知器算法（PLA），假设第 t 次迭代时，选择的是第 n 个样本：$\text{sign}(\boldsymbol{w}^T\mathbf{x}_n)\neq y_n$，$\boldsymbol{w}_{t+1}\leftarrow\boldsymbol{w}_t+y_n\mathbf{x}_n$，下述那个式子正确？

（a）$\boldsymbol{w}_{t+1}^{\boldsymbol{T}}\mathbf{x}_n=y_n$

（b）$\text{sign}\left(\boldsymbol{w}_{t+1}^{\boldsymbol{T}}\mathbf{x}_n\right)=y_n$

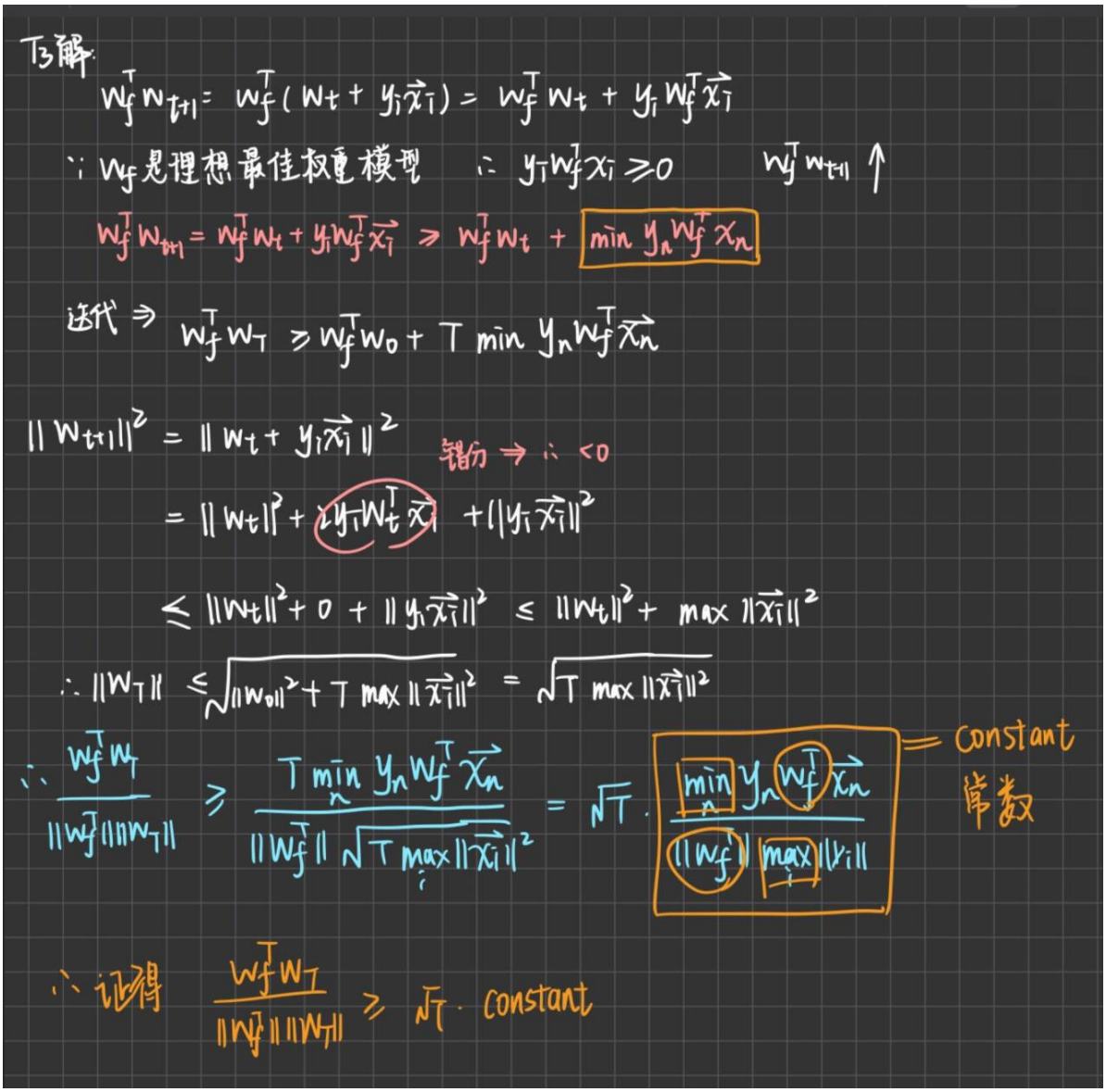
（c）$y_n\boldsymbol{w}_{t+1}^{\boldsymbol{T}}\mathbf{x}_n\geq y_n\boldsymbol{w}_t^{\boldsymbol{T}}\mathbf{x}_n$

（d）$y_n\boldsymbol{w}_{t+1}^{\boldsymbol{T}}\mathbf{x}_n<y_n\boldsymbol{w}_t^{\boldsymbol{T}}\mathbf{x}_n$

T2. 解. $\vec{w}_{t+1} = \vec{w}_t + y_n \vec{x}_n$

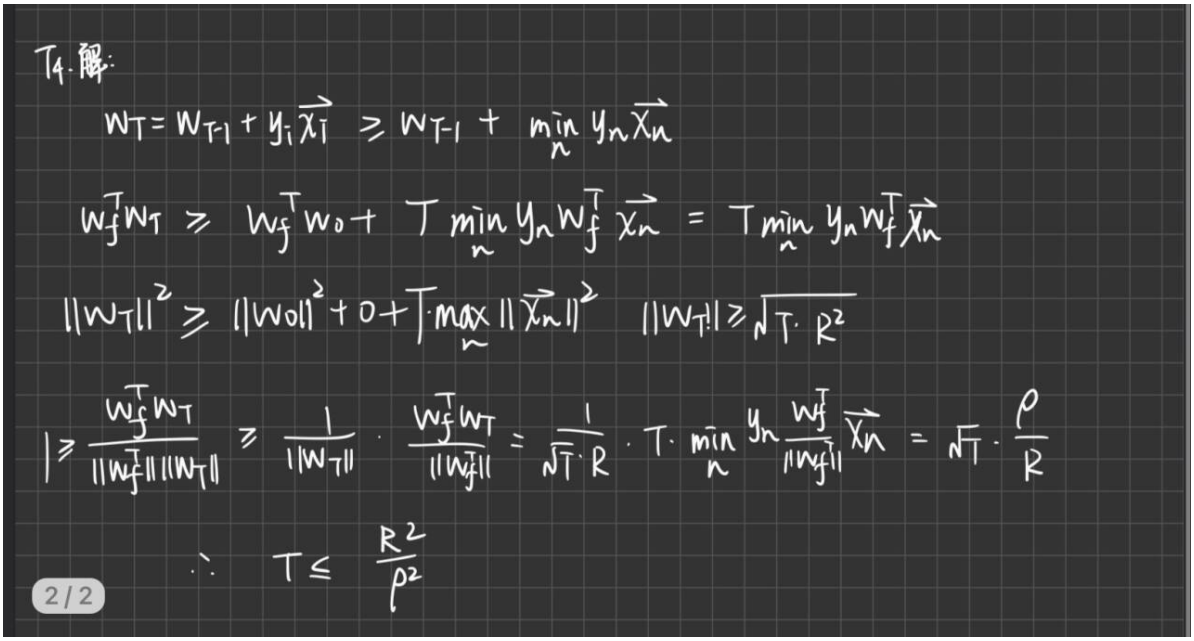
$y_n \vec{w}_{t+1} \vec{x}_n = y_n(\vec{w}_t + y_n \vec{x}_n)\ \vec{x}_n = y_n \vec{w}_t \vec{x}_n + \vec{y_n}^2 |\vec{x}_n|^2 \geq y_n \vec{w}_t \vec{x}_n$

3，证明：针对线性可分训练样本集，PLA 算法中，当$\boldsymbol{w}_0 = \mathbf{0}$，在对分错样本进行了 T 次纠正后，下式成立：$\frac{\mathbf{w}_f^T}{\|\mathbf{w}_f\|}\frac{\mathbf{w}_T}{\|\mathbf{w}_T\|} \geq \sqrt{T} \cdot constant$



4，针对线性可分训练样本集，PLA 算法中，假设对分错样本进行了 T 次纠正后得到的分类面不再出现错分状况，定义：$\boldsymbol{R^2 = \max_n \|x_n\|^2}$，

$\rho = \min_n y_n \frac{\mathbf{w}_f^T}{\|\mathbf{w}_f\|} \mathbf{x}_n$，试证明：$T \leq \frac{R^2}{\rho^2}$


T4. 解:
$w_T = w_{T-1} + y_i \vec{x_i} \geq w_{T-1} + \min_n y_n \vec{x_n}$
$w_f^T w_T \geq w_f^T w_0 + T \min_n y_n w_f^T \vec{x_n} = T \min_n y_n w_f^T \vec{x_n}$
$\|w_T\|^2 \geq \|w_0\|^2 + 0 + T \cdot \max_n \|\vec{x_n}\|^2 \quad \|w_T\| \geq \sqrt{T \cdot R^2}$
$1 \geqslant \frac{w_f^T w_T}{\|w_f^T\| \|w_T\|} \geqslant \frac{1}{\|w_T\|} \cdot \frac{w_f^T w_T}{\|w_f^T\|} = \frac{1}{\sqrt{T} \cdot R} \cdot T \cdot \min_n y_n \frac{w_f^T}{\|w_f^T\|} \vec{x_n} = \sqrt{T} \cdot \frac{\rho}{R}$
$\therefore \quad T \leq \frac{R^2}{\rho^2}$
2 / 2

编程作业