中国科学院南京分院
Nanjing Branch of Chinese Academy of Sciences

人工智能原理与算法

4. 感知机

夏睿

2023.3.1

目录

- （二分类）感知机
- 多分类感知机
- 线性模型回顾

感知机

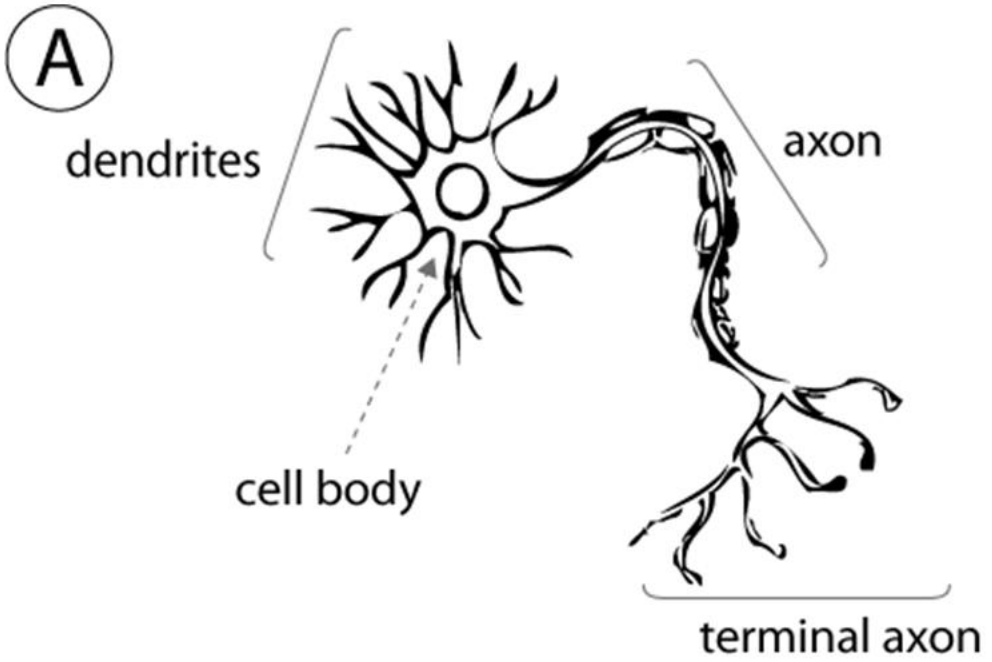
感知机（Perceptron）

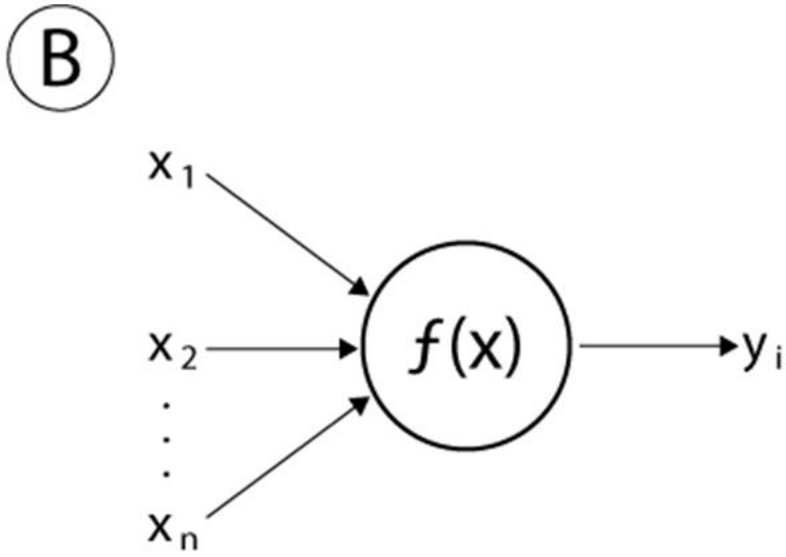
- 感知机算法是Frank Rosenblatt于1957年就职于康奈尔航空实验室时所发明的。
- 感知机是一种监督分类算法。
- 它是一种线性分类算法。
- 它是人工神经网络（ANN）的早期代表性工作，奠定了ANN的基础。

从神经网络到人工神经网络

神经元

人工神经元

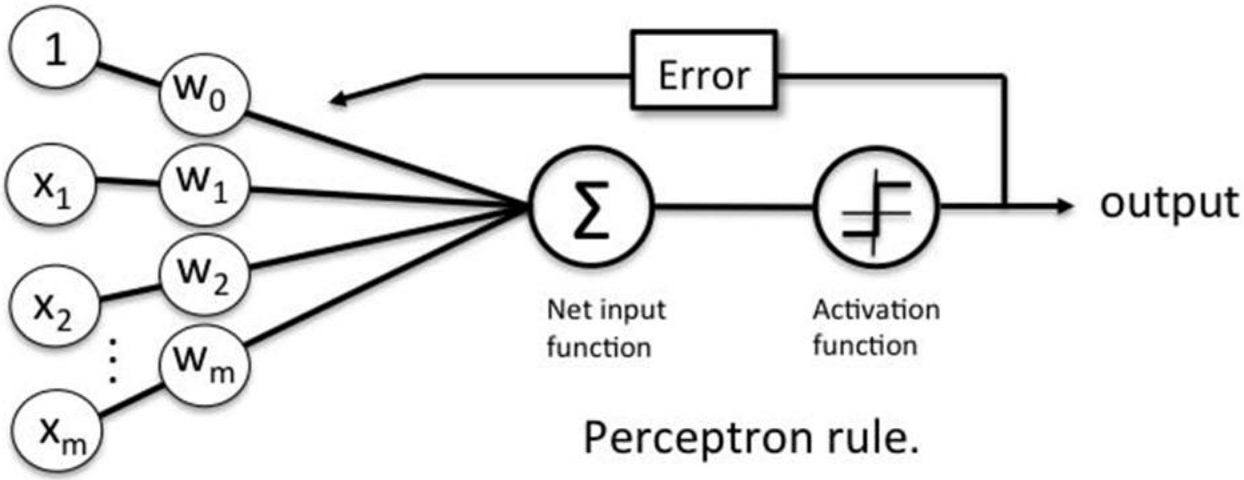




感知机模型

- 模型假设

$$h(\boldsymbol{x}) = \mathrm{sgn}\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}\right) = \begin{cases} 1 & \text{if } \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x} \geq 0 \\ 0 & \text{if } \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x} < 0 \end{cases}$$



Perceptron rule.

感知机学习算法

- 感知机损失函数

$$l_P(\boldsymbol{\omega}) = \sum_{\boldsymbol{x}^{(h)} \epsilon M_0} \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x}^{(h)} - \sum_{\boldsymbol{x}^{(l)} \epsilon M_1} \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x}^{(l)}$$

M_0：真实标签为负类、预测标签为正类

M_1：真实标签为正类、预测标签为负类

$$= \sum_{k=1}^{N} \left(\left(1 - y^{(k)}\right) h\left(\boldsymbol{x}^{(k)}\right) - y^{(k)} \left(1 - h\left(\boldsymbol{x}^{(k)}\right)\right) \right) \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x}^{(k)}$$

$$= \sum_{k=1}^{N} \left(h\left(\boldsymbol{x}^{(k)}\right) - y^{(k)}\right) \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x}^{(k)}$$

- 梯度计算

$$\frac{dl_P(\boldsymbol{\omega})}{d\boldsymbol{\omega}} = \frac{1}{N} \frac{d}{d\boldsymbol{\omega}} \sum_{k=1}^{N} \left(h\left(\boldsymbol{x}^{(k)}\right) - y^{(k)}\right) \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x}^{(k)}$$

$$= \frac{1}{N} \sum_{k=1}^{N} \boxed{\left(h\left(\boldsymbol{x}^{(k)}\right) - y^{(k)}\right) \boldsymbol{x}^{(k)}}$$

误差 · 输入

（随机）梯度下降

- 梯度下降(GD)

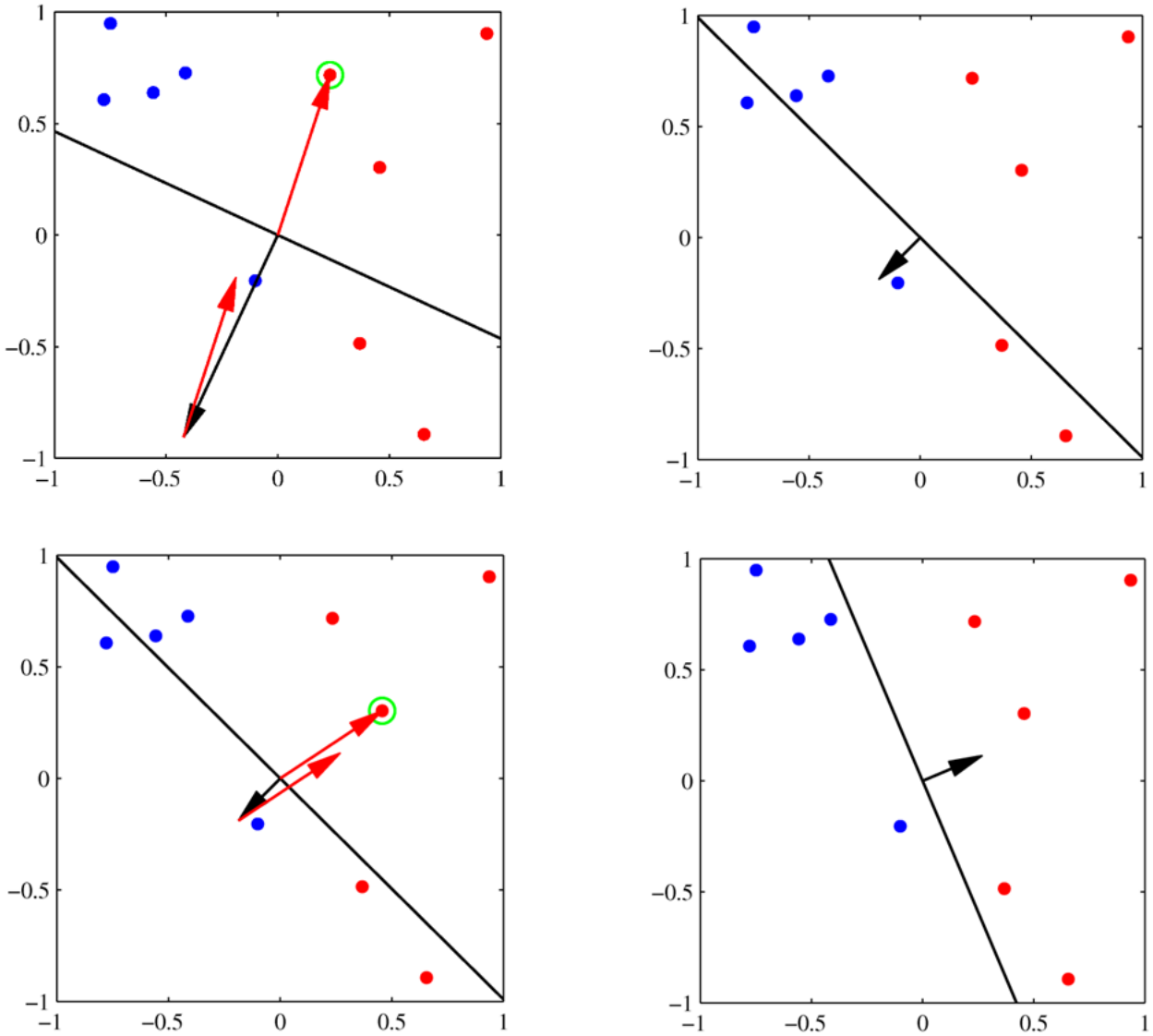
$$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \alpha \frac{d}{d\boldsymbol{\omega}} l_P(\boldsymbol{\omega}) = \boldsymbol{\omega} - \alpha \frac{1}{N} \sum_{k=1}^{N} \left(h\left(\boldsymbol{x}^{(k)}\right) - y^{(k)} \right) \boldsymbol{x}^{(k)}$$

- 随机梯度下降(SGD)

$$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \alpha (h(\boldsymbol{x}) - y)\boldsymbol{x}$$

$$= \begin{cases} \boldsymbol{\omega} + \alpha \boldsymbol{x}, & \text{if } y = 1 \text{ and } h(\boldsymbol{x}) = 0 \\ \boldsymbol{\omega} - \alpha \boldsymbol{x}, & \text{if } y = 0 \text{ and } h(\boldsymbol{x}) = 1 \\ \boldsymbol{\omega}, & \text{otherwise} \end{cases}$$

感知机算法图释



感知机手算

- 训练集

样本序号	特征向量 (x_1, x_2)	真实类别 y
1	$(-0.8, 0.6)$	0
2	$(-0.55, 0.7)$	0
3	$(-0.4, 0.7)$	0
4	$(-0.1, -0.2)$	0
5	$(-0.75, 0.95)$	0
6	$(0.2, 0.75)$	1
7	$(0.9, 0.9)$	1
8	$(0.45, 0.3)$	1
9	$(0.4, -0.5)$	1
10	$(0.6, -0.9)$	1

- 运行一个感知机学习算法 （不含偏置项）

$$h(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \omega_1 x_1 + \omega_2 x_2 \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

感知机手算

- **初始化：**设置初始权重为 $\boldsymbol{w}^{(0)} = (-0.5, -1)$，并且固定学习率 $\alpha = 1$。
- **预测阶段：**根据 $\boldsymbol{w}^{(0)}$ 进行预测。

样本序号	特征向量 (x_1, x_2)	真实类别 y	预测类别 $h(\boldsymbol{x})$
1	$(-0.8, 0.6)$	0	
2	$(-0.55, 0.7)$	0	
3	$(-0.4, 0.7)$	0	
4	$(-0.1, -0.2)$	0	
5	$(-0.75, 0.95)$	0	
6	$(0.2, 0.75)$	1	
7	$(0.9, 0.9)$	1	
8	$(0.45, 0.3)$	1	
9	$(0.4, -0.5)$	1	
10	$(0.6, -0.9)$	1	

观察表格，我们可以发现样本 _____ 分类错误。

感知机手算

- **权重更新阶段：**我么随机选择一个错分样本，如第6个样本 $(0.2, 0.75)$，用于更新权重。权重更新公式：

$$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \alpha(h(\boldsymbol{x}) - y)\boldsymbol{x}$$

$$= \begin{cases} \boldsymbol{\omega} + \boldsymbol{x}, & \text{if } y = 1 \text{ and } h(\boldsymbol{x}) = 0 \\ \boldsymbol{\omega} - \boldsymbol{x}, & \text{if } y = 0 \text{ and } h(\boldsymbol{x}) = 1 \\ \boldsymbol{\omega}, & \text{otherwise} \end{cases}$$

我们可以得到新的权重 $\boldsymbol{w}^{(1)} =$_____。

感知机手算

- **预测阶段：**根据 $\boldsymbol{w}^{(1)}$ 进行预测。

样本序号	特征向量 (x_1, x_2)	真实类别 y	预测类别 $h(\boldsymbol{x})$
1	$(-0.8, 0.6)$	0	
2	$(-0.55, 0.7)$	0	
3	$(-0.4, 0.7)$	0	
4	$(-0.1, -0.2)$	0	
5	$(-0.75, 0.95)$	0	
6	$(0.2, 0.75)$	1	
7	$(0.9, 0.9)$	1	
8	$(0.45, 0.3)$	1	
9	$(0.4, -0.5)$	1	
10	$(0.6, -0.9)$	1	

观察表格，我们可以发现样本 _____ 分类错误。

- **权重更新阶段：**我们随机选择一个错分样本，比如第8个样本 $(0.45, 0.3)$ 用于更新权重，得到新的权重 $\boldsymbol{w}^{(2)}$ =_____。

感知机手算

- **预测阶段：** 根据$\boldsymbol{w}^{(2)}$进行预测。

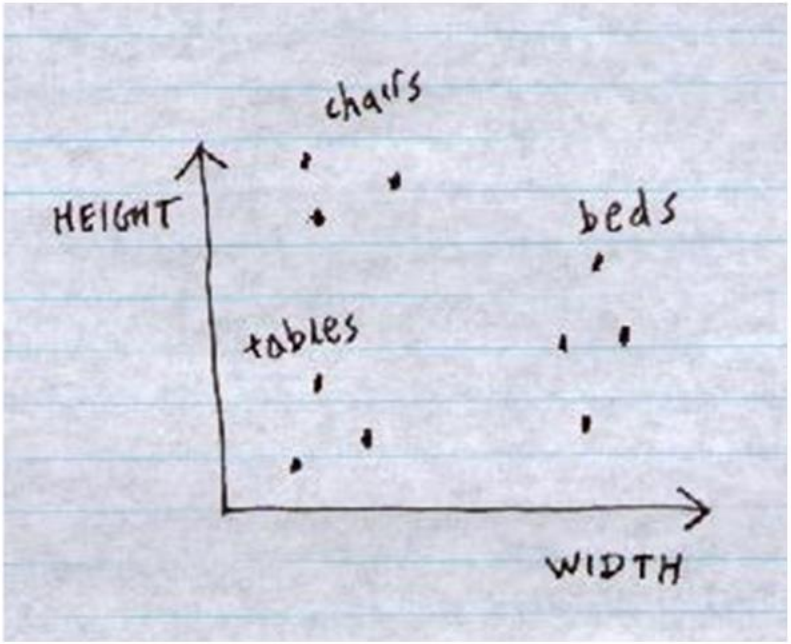
样本序号	特征向量 (x_1, x_2)	真实类别 y	预测类别 $h(\boldsymbol{x})$
1	$(-0.8, 0.6)$	0	
2	$(-0.55, 0.7)$	0	
3	$(-0.4, 0.7)$	0	
4	$(-0.1, -0.2)$	0	
5	$(-0.75, 0.95)$	0	
6	$(0.2, 0.75)$	1	
7	$(0.9, 0.9)$	1	
8	$(0.45, 0.3)$	1	
9	$(0.4, -0.5)$	1	
10	$(0.6, -0.9)$	1	

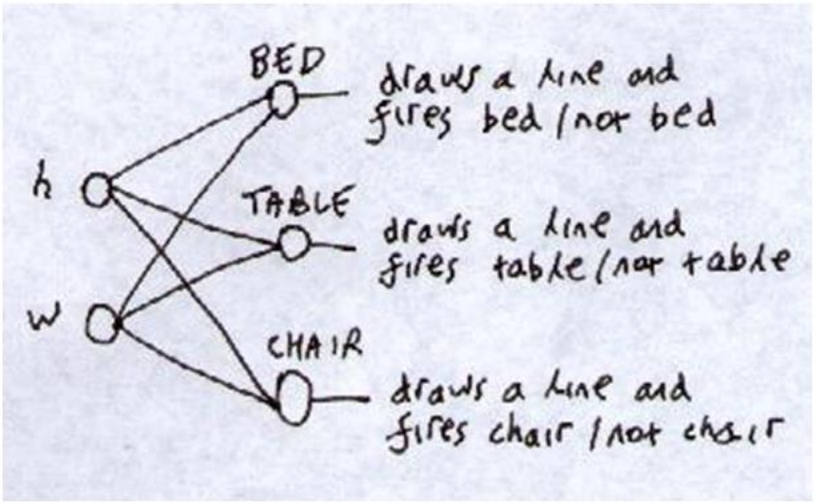
感知机算法现在收敛了吗？

多分类感知机

多分类感知机模型

- 多类感知机（multi-class perceptron）是标准感知机的扩展，用于解决多类分类问题；
- 多类感知机广泛应用于 NLP。





假设与学习

- 假设

$$h(\boldsymbol{x}) = \arg \max_{j=1,\ldots,C} \boldsymbol{\omega}_j^{\mathrm{T}} \boldsymbol{x}$$

- 损失函数

$$l_p(\boldsymbol{\omega}) = \sum_{k=1}^{N} \left(\max_{j=1,\ldots,C} \boldsymbol{\omega}_j^{\mathrm{T}} \boldsymbol{x}^{(k)} - \boldsymbol{\omega}_{y^{(k)}}^{\mathrm{T}} \boldsymbol{x}^{(k)} \right) = \sum_{k=1}^{N} \left(\boldsymbol{\omega}_{h(\boldsymbol{x}^{(k)})}^{\mathrm{T}} \boldsymbol{x}^{(k)} - \boldsymbol{\omega}_{y^{(k)}}^{\mathrm{T}} \boldsymbol{x}^{(k)} \right)$$

- 随机梯度下降参数更新

$$\boldsymbol{\omega}_j \leftarrow \boldsymbol{\omega}_j - \alpha \left(I\big(j = h(\boldsymbol{x})\big) - I(j = y) \right) \boldsymbol{x}$$

$$= \begin{cases} \boldsymbol{\omega}_j - \alpha \boldsymbol{x}, & \text{if } j = h(\boldsymbol{x}) \neq y \\ \boldsymbol{\omega}_j + \alpha \boldsymbol{x}, & \text{if } j = y \neq h(\boldsymbol{x}) \\ \boldsymbol{\omega}_j, & \text{otherwise} \end{cases}$$

线性模型回顾

模型假设（二分类）

- 线性回归

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}$$

- 感知机

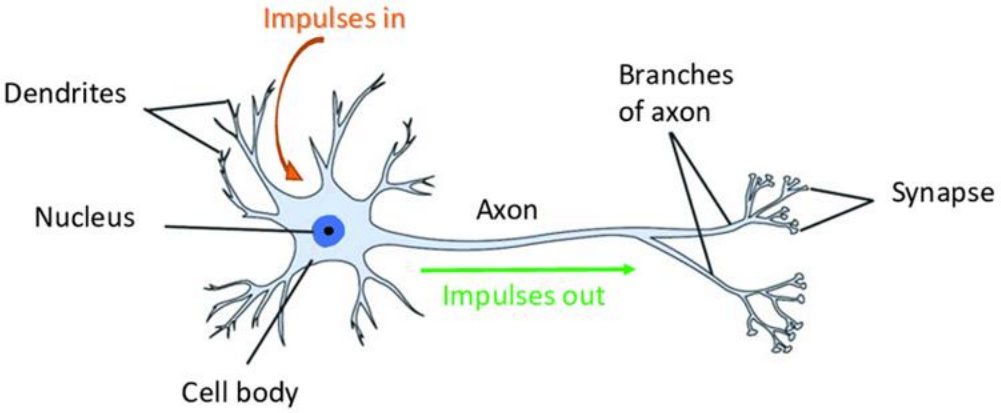
$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \begin{cases} 1, & if \ \ \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x} \geq 0 \\ 0, & if \ \ \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x} < 0 \end{cases}$$

- Logistic回归

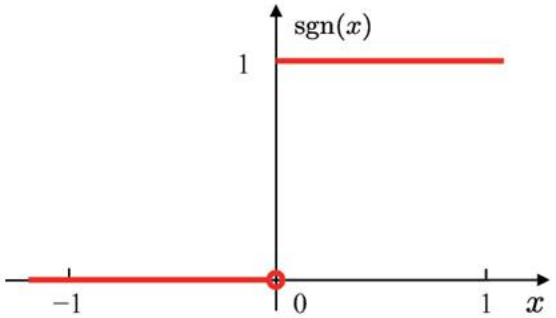
$$P(y = 1|\boldsymbol{x}; \boldsymbol{\theta}) = \delta\left(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}\right) = \frac{1}{1 + e^{-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}}}$$

$$P(y = 0|\boldsymbol{x}; \boldsymbol{\theta}) = 1 - \delta\left(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}\right)$$

从激活函数角度的理解



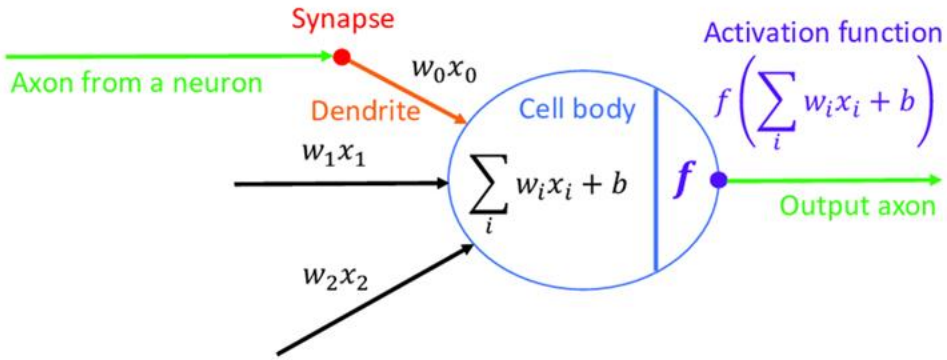
(a)



$$\mathrm{sgn}(x) = \begin{cases} 1, & x \geqslant 0; \\ 0, & x < 0. \end{cases}$$

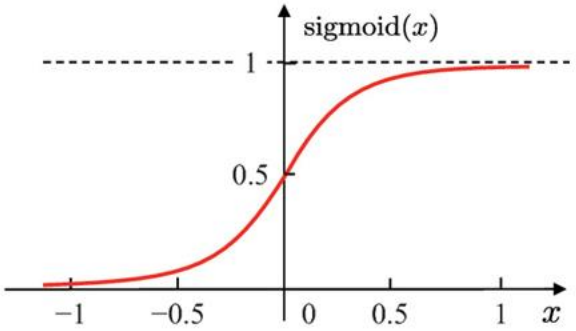
(a) Step function.

感知机的激活函数



(b)

单层人工神经网络



$$\mathrm{sigmoid}(x) = \frac{1}{1+e^{-x}}$$

(b) Sigmoid function.

Logistic回归的激活函数

模型学习 – 损失函数

- 线性回归

$$L_{lr}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{k=1}^{N}\left(h\left(\boldsymbol{x}^{(k)}\right) - y^{(k)}\right)^2$$

Least Square = Maximum Likelihood

- 感知机

$$L_p(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}^{(k)} \in M_0} \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}^{(k)} - \sum_{\boldsymbol{x}^{(k)} \in M_1} \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}^{(k)} = \sum_{\boldsymbol{x}^{(k)} \in M} \left|\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}^{(k)}\right|$$

Perceptron Criterion

- Logistic回归

$$L(\boldsymbol{\theta}) = \sum_{k=1}^{N} y^{(k)} \log h\left(\boldsymbol{x}^{(k)}\right) + \left(1 - y^{(k)}\right) \log\left(1 - h\left(\boldsymbol{x}^{(k)}\right)\right)$$

Maximum Likelihood = Minimum Cross Entropy

模型学习－最优化

- 线性回归

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \frac{d}{d\boldsymbol{\theta}} J_l(\boldsymbol{\theta}) = \boldsymbol{\theta} - \alpha \sum_{k=1}^{N} \left(h\left(\boldsymbol{x}^{(k)}\right) - y^{(k)} \right) \boldsymbol{x}^{(k)}$$

- 感知机

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha(h(\boldsymbol{x}) - y)\boldsymbol{x} = \begin{cases} \boldsymbol{\theta} + \alpha \boldsymbol{x}, & \text{if } y = 1 \text{ and } h(\boldsymbol{x}) = 0 \\ \boldsymbol{\theta} - \alpha \boldsymbol{x}, & \text{if } y = 0 \text{ and } h(\boldsymbol{x}) = 1 \\ \boldsymbol{\theta}, & \text{otherwise} \end{cases}$$

- Logistic回归

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \sum_{k=1}^{N} \left(y^{(k)} - h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(k)}\right) \right) \boldsymbol{x}^{(k)}$$

模型假设（多分类）

- Softmax回归

$$p(y=j|\boldsymbol{x};\boldsymbol{\theta})=h_j(\boldsymbol{x})=\frac{e^{\boldsymbol{\theta}_j{}^{\mathrm{T}}\boldsymbol{x}}}{\sum_{j'=1}^{C}e^{\boldsymbol{\theta}_{j'}{}^{\mathrm{T}}\boldsymbol{x}}},j=1,2,\ldots,C$$

- 多分类感知机

$$h(\boldsymbol{x})=\arg\max_{j=1,\ldots,C}\boldsymbol{\theta}_j^{\mathrm{T}}\boldsymbol{x}$$

模型学习－损失函数

- Softmax回归

$$L(\boldsymbol{\theta}) = \sum_{k=1}^{N} \log p(y^{(k)}|\boldsymbol{x}^{(k)};\boldsymbol{\theta}) = \sum_{k=1}^{N}\sum_{j=1}^{C} I(y^{(k)}=j)\log\left(\frac{e^{\boldsymbol{\theta}_j{}^{\mathrm{T}}\boldsymbol{x}}}{\sum_{j'=1}^{C} e^{\boldsymbol{\theta}_{j'}{}^{\mathrm{T}}\boldsymbol{x}}}\right)$$

- 多分类感知机

$$L_{mp}(\boldsymbol{\theta}) = \sum_{k=1}^{N}\left(\max_{j=1,\ldots,C} \boldsymbol{\theta}_j^{\mathrm{T}}\boldsymbol{x}^{(k)} - \boldsymbol{\theta}_{y^{(k)}}^{\mathrm{T}}\boldsymbol{x}^{(k)}\right)$$

模型学习－最优化

- Softmax回归

$$\boldsymbol{\theta}_j \leftarrow \boldsymbol{\theta}_j + \alpha \sum_{k=1}^{N} \left(I(y^{(k)} = j) - h_j\left(\boldsymbol{x}^{(k)}\right) \right) \boldsymbol{x}^{(k)}$$

- 多分类感知机

$$\boldsymbol{\theta}_j \leftarrow \boldsymbol{\theta}_j - \alpha \left(I\left(j = h(\boldsymbol{x})\right) - I(j = y) \right) \boldsymbol{x} = \begin{cases} \boldsymbol{\theta}_j - \alpha \boldsymbol{x}, & \text{if } j = h(\boldsymbol{x}) \neq y^{(k)} \\ \boldsymbol{\theta}_j + \alpha \boldsymbol{x}, & \text{if } j = y^{(k)} \neq h(\boldsymbol{x}) \\ \boldsymbol{\theta}_j, & \text{otherwise} \end{cases}$$

本讲结束 欢迎提问