

# 第14章 人工神经网络和深度学习 概述

## 《人工智能算法》

清华大学出版社

2022年7月

# 提纲

- ◆ 人工神经网络
  - 神经元模型
  - 感知机
  - 多层神经网络
- ◆ 深度学习
- ◆ 总结

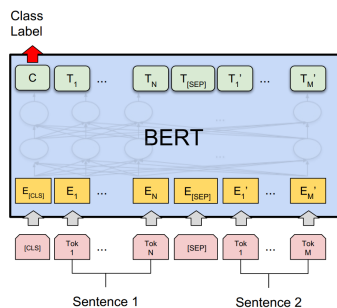
# 人工神经网络 (1)

## ◆ 人工神经网络 (Artificial Neural Network, ANN)

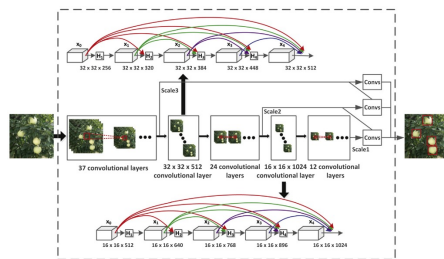
- 抽象人脑神经元 → 构建人工神经元。
- 连接人工神经元 → 模拟人脑神经网络。

数学模型

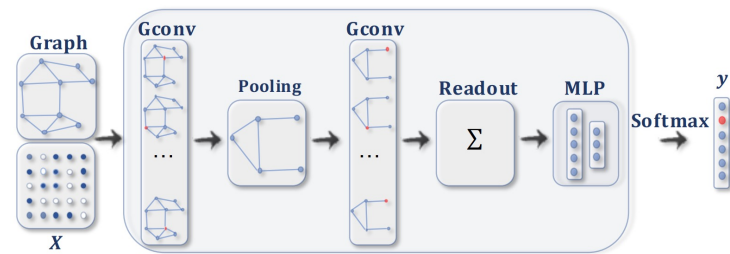
## ◆ 应用广泛



自然语言处理



目标检测

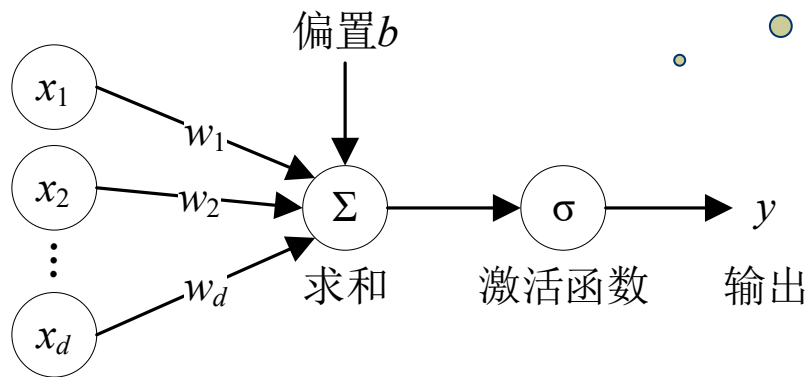


图分析

# 人工神经网络 (2)

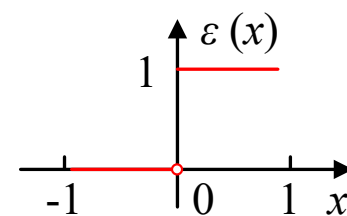
## ◆ 神经元模型

- 接收  $d$  个输入信号  $\mathbf{x} = [x_1; x_2; \dots; x_d]$ 。
- 对信号加权求和，再通过激活函数输出。



MP神经元模型

激活函数 $\sigma$ 为  
阶跃函数



1: 神经元兴奋

0: 神经元抑制

# 人工神经网络 (3)

## ◆ MP神经元模型的数学表示

- 权重参数:  $\mathbf{w} = [w_1; w_2; \dots; w_d]$ , 偏置:  $b$

$$y = \sigma \left( \sum_{i=1}^d w_i x_i + b \right) = \sigma(\mathbf{w}^T \mathbf{x} + b)$$

## ◆ MP神经元模型的数学理解

- 对复杂函数  $y = f(x_1, \dots, x_d)$  的一阶泰勒近似

$$y = f(x_1, \dots, x_d) = \boxed{f(0, \dots, 0)} + \sum_{i=1}^d \boxed{\left[ \frac{\partial f}{\partial x_i} (0, \dots, 0) \right]} x_i + \dots$$

偏置  $b$  权重参数  $w_i$

# 人工神经网络 (4)

## ◆ 感知机 (Perceptron)

- 输入：实例的特征向量  $\mathbf{x} = [x_1; \dots; x_d]$
- 输出：+1 或 -1

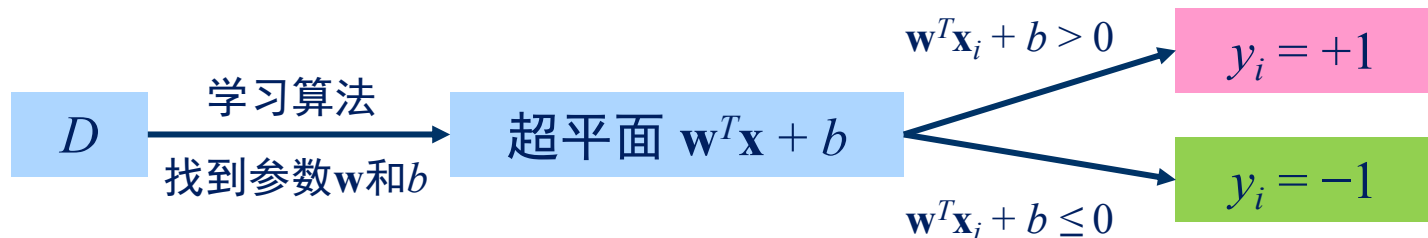
w和b为模型参数  
sign为符号函数

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

$$\text{sign}(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

## ◆ 被广泛使用的二分类模型

- 输入：数据集  $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$



# 人工神经网络 (5)

## ◆ 感知机学习算法

- 求解参数 $\mathbf{w}$ 和 $b$ 的损失函数:

$$\min_{\mathbf{w}, b} L(\mathbf{w}, b) = - \sum_{\mathbf{x}_i \in M} y_i (\mathbf{w}^T \mathbf{x}_i + b)$$

$M$ 为误分类的样本集合

- 梯度下降求解:

随机初始化 $\mathbf{w}$ 和 $b$

**Repeat**

在训练集中随机选取一个样本 $(\mathbf{x}_i, y_i)$

**If**  $y_i(\mathbf{w}^T \mathbf{x}_i + b) \leq 0$  **Then**

$$\mathbf{w} \leftarrow \mathbf{w} + \eta y_i \mathbf{x}_i, b \leftarrow b + \eta y_i$$

**End If**

**Until** 没有误分类的样本

**Return**  $\mathbf{w}$ 和 $b$

复杂度为 $O(td)$   
 $t$ : 迭代次数

# 人工神经网络 (6)

## ◆ 感知机的主要贡献

- 提供了一个通用的机器学习框架

给定数据集 $D$ ，寻找函数  $y = f(\mathbf{x}, \theta)$ ，预测新样本 $\mathbf{x}'$ 对应的  $y' = f(\mathbf{x}', \theta)$

感知机：  $\theta = (\mathbf{w}, b)$ ，  $f(\mathbf{x}, \theta) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$

- 占用计算和存储资源较少

仅存储参数  $\mathbf{w}$  和  $b$

每次更新只选择一小部分训练样本进行计算



# 人工神经网络 (7)

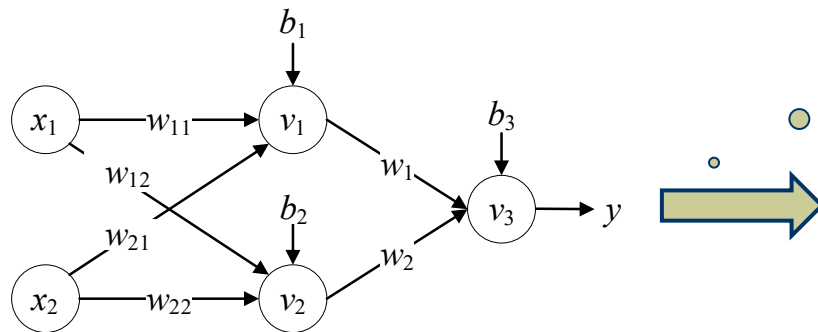
## ◆ 多层神经网络

- 按一定结构连接多个神经元，可处理非线性可分问题
- 多层前馈神经网络 (Multi-layer Feedforward Neural Network)

1个输入层、1个隐藏层、1个输出层

每层包含多个神经元

每层神经元与下一层神经元全连接



输出:  $y = v_3$

$$v_1 = \sigma(w_{11}x_1 + w_{21}x_2 + b_1)$$

$$v_2 = \sigma(w_{12}x_1 + w_{22}x_2 + b_2)$$

$$v_3 = \sigma(w_1x_1 + w_2x_2 + b_3)$$

训练网络: 确定参数  $\mathbf{w} = [w_{11}; w_{12}; w_{21}; w_{22}; w_2]$  和  $\mathbf{b} = [b_1; b_2; b_3]$

# 人工神经网络 (8)

## ◆ 多层神经网络的特点

- 激活函数 $\sigma$ 必须是非线性函数
- 能以任意精度逼近任意复杂的连续函数

表示能力强

## ◆ 训练神经网络的步骤

(1) 设计神经网络的结构 (多少层? 每层多少神经元? 如何连接?)

简单问题/训练数据少→简单结构; 复杂问题/训练数据多→复杂结构

(2) 求解网络中的待求参数

调整参数 $\mathbf{w}$ 和 $\mathbf{b}$ , 使网络输出 $\hat{y}$ 与样本标签 $y$ 尽可能接近 ( $n$ 为样本数量)

$$\min E(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

# 人工神经网络 (9)

## ◆ 训练神经网络的步骤

(2) 求解网络中的待求参数

梯度下降求解: 
$$\min E(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

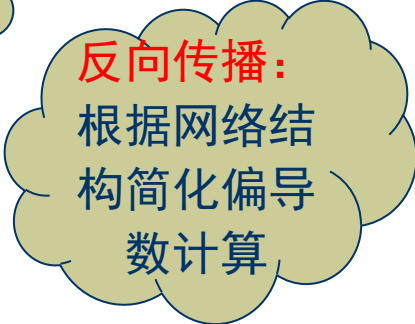
随机初始化  $\mathbf{w}$  和  $\mathbf{b}$

迭代更新:

$$w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i}$$

$$b_i \leftarrow b_i - \eta \frac{\partial E}{\partial b_i}$$

学习率:  $0 < \eta < 1$



反向传播:  
根据网络结  
构简化偏导  
数计算

# 人工神经网络 (10)

## ◆ 基于反向传播的神经网络学习算法

输入：训练数据集 $D$ ，学习率 $\eta$

输出： $w$  和  $b$

随机初始化  $w$  和  $b$

**Repeat**

反向传播计算各参数的偏导数  $\frac{\partial E}{\partial w_i}, \frac{\partial E}{\partial b_i}$

$$w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i}$$

$$b_i \leftarrow b_i - \eta \frac{\partial E}{\partial b_i}$$

**Until**  $\frac{\partial E}{\partial w_i} = 0$  且  $\frac{\partial E}{\partial b_i} = 0$

**Return**  $w$  和  $b$

时间复杂度：

$O(tm)$

$t$ ：迭代次数

$m$ ：参数数量

# 人工神经网络 (11)

## ◆ 神经网络学习算法的改进

### (1) 激活函数

使用Sigmoid、ReLU等函数

### (2) 损失函数

交叉熵损失 (Cross-entropy) :  $E(\mathbf{y}, \hat{\mathbf{y}}) = - \sum_{i=1}^k y_i \log(\hat{y}_i)$

### (3) 随机梯度下降

输入一批样本 (Batch)，利用这批样本的梯度平均值更新参数

将训练数据按Batch Size划分为多个Batch

模型训练需要多个Epoch，每个Epoch都随机划分训练样本

更新参数与训练集大小无关

# 提纲

- ◆ 人工神经网络
  - 神经元模型
  - 感知机
  - 多层神经网络
- ◆ 深度学习
- ◆ 总结

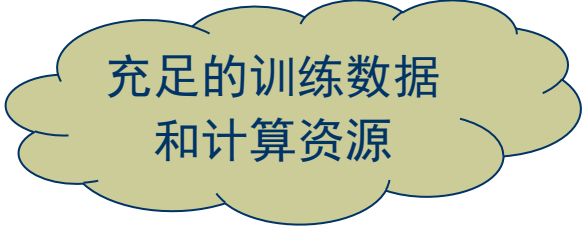
# 深度学习 (1)

## ◆ 多层神经网络存在许多不足

- 梯度下降法只能获取**局部最优值**，而非全局最优
- 模型**可解释性差**
- 需调整的参数太多，训练模型的**工作量太大**
- 需**大量的训练数据** .....

## ◆ 深度学习发展背景

- 计算能力得到了显著提升（GPU等硬件加速设备普及）
- 海量数据被采集和存储



充足的训练数据  
和计算资源

# 深度学习 (2)

## ◆ 深度学习的产生及发展

- 2006年，Hinton通过“预训练+微调”训练出超过7层的神经网络，并将这类学习方法称为深度学习（Deep Learning）
- 2009年，微软将深度神经网络引入语音识别系统，大幅提升了连续词汇的语音识别率
- 2013年，Hinton的学生使用深度神经网络AlexNet在图像识别比赛ImageNet上夺冠
- 2016年，Google利用基于深度学习的AlphaGo打败了围棋世界冠军李世石
- .....



# 深度学习 (3)

## ◆ 深度学习的关键

- 构建“深度”神经网络模型（层数多、每层的神经元也多）
- 利用学习算法从数据中自动产生较好的特征

## ◆ 常用的深度学习算法

- 自编码器（Autoencoder）：高维数据降维
- 卷积神经网络（Convolutional Neural Network）：图像特征提取
- 循环神经网络（Recurrent Neural Network）：文本特征提取
- 图神经网络（Graph Neural Network）：图数据分析
- .....

# 提纲

- ◆ 人工神经网络
  - 神经元模型
  - 感知机
  - 多层神经网络
- ◆ 深度学习
- ◆ 总结

# 总结

## ◆ 人工神经网络

- MP神经元模型：模拟生物神经元的结构和功能
- 感知机：二分类模型，为机器学习提供了一个通用框架
- 多层神经网络：表示能力强，基于反向传播进行训练

## ◆ 深度学习

- 深度神经网络模型
- 能提取输入数据更好的特征



结语



谢谢！