

第1章 人工神经网络

1943 年，神经生理学家沃伦·麦卡洛克和数学家沃尔特·皮茨提出了神经元基本模型，首次将神经网络的概念引入计算机领域。1957 年美国学者 Frank Rosenblatt 提出了感知机，它是神经网络和支持向量机的基础，人工神经网络得以快速的发展。但此后神经网络经历了漫长的低谷，直到 20 世纪 80 年代，新的架构以及新的训练技术的出现，特别是反向传播技术的提出又重新激起了人们的兴趣，但进展缓慢。20 世纪 90 年代其他机器学习算法的出现，如支持向量机，取得了比神经网络更好的结果以及其更坚实的理论基础，神经网络又进入了低谷。

近些年随着大规模高质量标注数据集的出现；计算能力的提升，特别是 GPU 卡的出现；更好的非线性激活函数的使用；更多优秀的网络结构的发明；更多稳健的优化算法的出现以及深度学习开发平台的发展，人工神经网络技术得到快速的发展。人工神经网络在语音识别，机器翻译，图像识别，AI 等领域得到广泛的应用，吸引大量资金的投入使得人工神经网络的发展进入良性的循环。

与统计机器学习相比，深度学习不需要人工提取特征，可以自动学习有用特征。

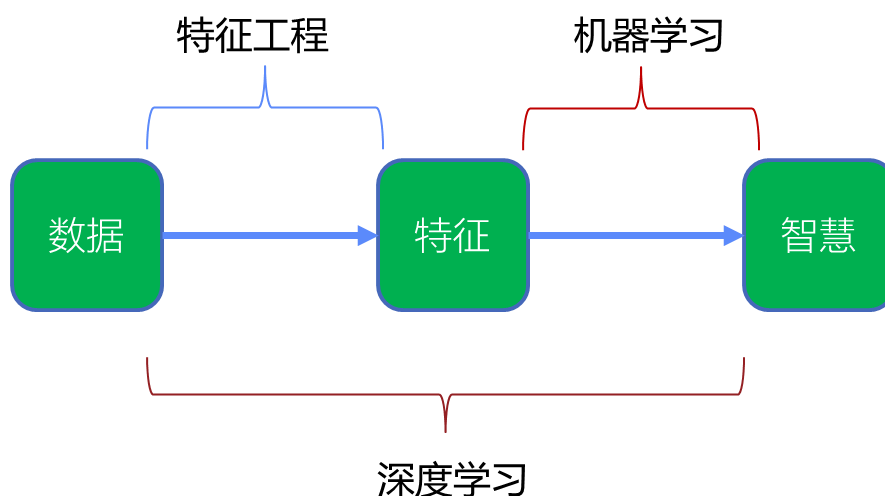


图 1-1 深度学习与统计机器学习

1.1 感知机

感知机是受到生物学上的启发，动物大脑是一个生物的神经网络，其中最小单元就是一个神经元(neuron)，这些神经元连接起来形成一个错综复杂的网络，称为神经网络。神经网络中树突组织接收到信号，产生一些微弱的生物电，就形成刺激，那么在细胞核里面对这些接收到的刺激进行综合的处理，当他的信号达到了一定的阈值之后，就会被激活，产生一个刺激的输出，形成进一步的信号。

人工神经网络正是模仿动物神经网络的机制实现信息的传递。

1.1.1 单层感知机

1943 年提出的人工神经元基本模型不含激活函数。1957 年美国学者 Frank Rosenblatt 提出的感知机，是现代人工神经网络和支持向量机的基础。感知机结构如图 1-2。

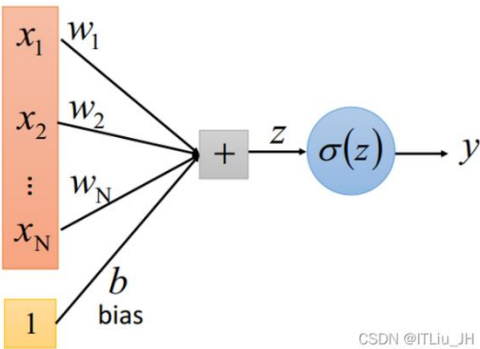


图 1-2 单层感知机结构图

感知机是最简单的人工神经网络（ANN）架构之一，输入空间到输出空间如下函数：

$$y = \sigma(z) \quad (1-1)$$

$$z = x^T w + b \quad (1-2)$$

其中： x 为 N 个输入， w 为输入对应的权重， b 为偏置， y 为输出。 σ 为阶跃函数，感知机中最常用的阶跃函数为 Heaviside 阶跃函数，通常也使用符号函数 sign 代替。

$$\text{Heaviside}(z) = \begin{cases} 0, & \text{if } z < 0 \\ 1, & \text{if } z \geq 0 \end{cases} \quad (1-3)$$

$$\text{sign}(z) = \begin{cases} -1, & \text{if } z < 0 \\ 0, & \text{if } z = 0 \\ 1, & \text{if } z > 0 \end{cases} \quad (1-4)$$

z 与 x 为线性关系，不论多少层，每层多少节点，输入与输出始终为线性关系。

感知机学习的目标是求得一个能够将训练数据集正例和负例完全分开的分离超平面。感知器的分类不同于逻辑回归，感知器基于阈值硬分类。

$$wx + b = 0 \quad (1-5)$$

为了求得分离超平面，需要构建一个损失函数，并通过极小化此损失函数来求得模型参数 w 和 b 。感知机的损失函数为所有误分类点到超平面的距离之和。

$$L(w, b) = -\sum_{x \in M} y_i (wx_i + b) \quad (1-6)$$

单层感知机可以很好解决线性可分问题，但对线性不可分问题如异或问题单层感知机无法解决。

1.1.2 多层感知机（MLP）

多层感知机包含一个输入层（直通层），一个输出层，中间一个或多个隐含层，此时又称为深度神经网络（DNN）。输入层不含偏置，层间的神经元相互连接，如图 1-3 所示。

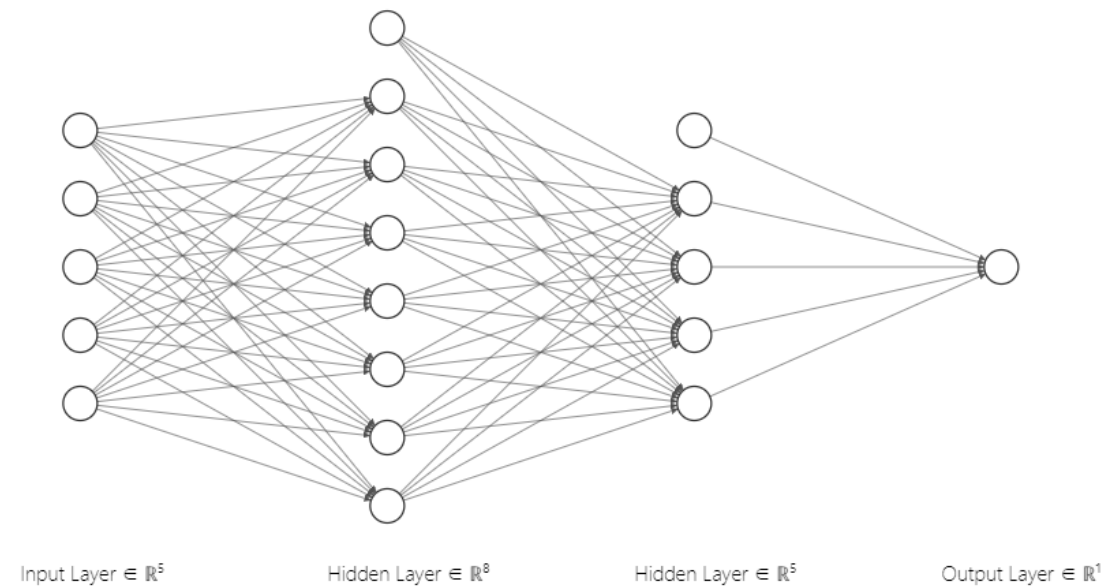


图 1-3 包含两个隐含层的多层感知机

$$y = F(x) = f_3(W_3, f_2(W_2, f_1(W_1, x))) \tag{1-7}$$

模型的参数为网络中边的权重和神经元中的偏置，记为 W 、 b 。模型的优化目标是获取最优的权重 w 和 $bias$ ，最小化误差 E 。寻找最优参数的过程就是训练神经网络模型的过程。

多层感知机的出现很好地解决了线性不可分等问题，但参数优化困难。直到 20 世纪 80 年代，反向传播(Back Propagation)算法提出，很好的解决了多层神经网络的参数计算问题，阶跃函数也被其它连续可导的激活函数替代，如 sigmoid 函数、relu 函数等。多层感知机不仅用于分类问题，还用于回归问题。

1.2 常用神经网络结构

1.2.1 全连接前馈神经网络

前馈神经网络，是一种最简单的神经网络，各神经元分层排列，每个神经元

与前一层的神经元相连，接收前一层的输出；与下一层的神经元相连并输出给下一层的神经元，各层间没有反馈。是应用最广泛、发展最迅速的人工神经网络之一。研究从 20 世纪 60 年代开始，理论研究和实际应用达到了很高的水平。

在此种神经网络中，第 0 层叫输入层，最后一层叫输出层，其他中间层叫做隐含层（或隐藏层、隐层）。隐层可以是一层。也可以是多层。当层间全连接时，称为全连接前馈神经网络。当输入特征较多时，全连接神经网络的容易导致参数爆炸。

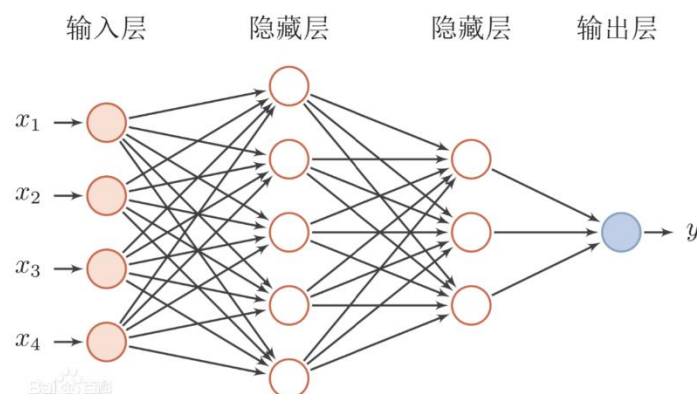


图 1-4 前馈神经网络

1.2.2 卷积神经网络

卷积神经网络（CNN）也是一种前馈神经网络，与全连接前馈神经网络相比，其层间连接不是全连接。大大减少了参数的个数。

如图像处理时由于像素多，容易导致参数爆炸，当图像大小为 1024×1024 ，第一层复杂度 $O(1024 \times 1024)$ ，全连接网络并不适用于图像。

卷积神经网络采用稀疏连接；参数共享；等变表示。适合处理网格型数据，在计算机视觉领域获得巨大成功：物体识别，图片分类，2 维网格。

全连接与稀疏连接的对比。（上：稀疏连接；下：全连接）

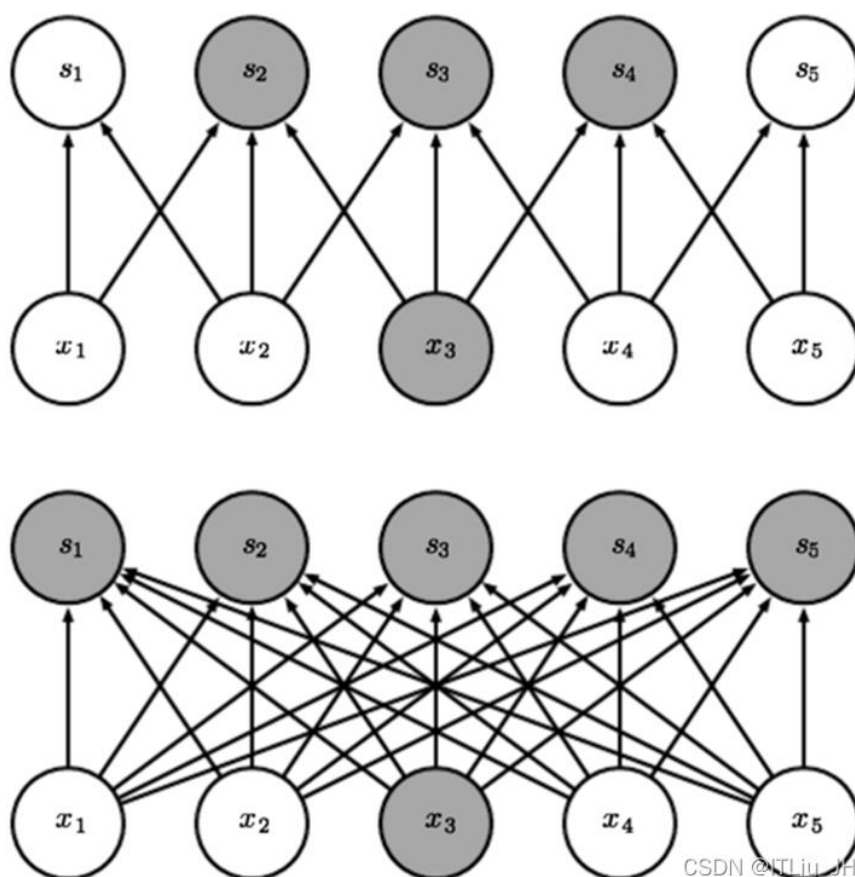


图 1-5 卷积神经网络

1.2.3 循环神经网络

在自然语言处理、时间序列分析等领域不仅需要考虑当前 t 时刻输入还要考虑 $t-i$ ($i=1-n$) 时刻的输入。RNN 在这方面得到成功应用，如机器翻译、词性标注、词向量、语音识别、图像描述生成、时间序列分析等。使用最成功的模型为长短期记忆模型（LSTM）

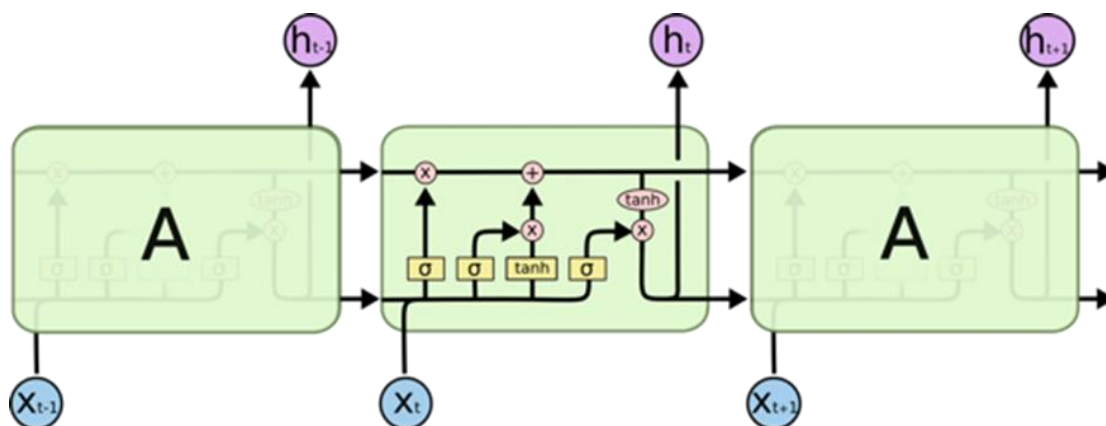


图 1-6 循环神经网络

1.2.4 其它神经网络

1) 自编码器和生成对抗网络

自编码器（Autoencoder）为无监督特征学习。

输出尽量逼近输入，隐层节点通常比输入少，可用于特征提取

非线性：表达能力比 PCA 更好

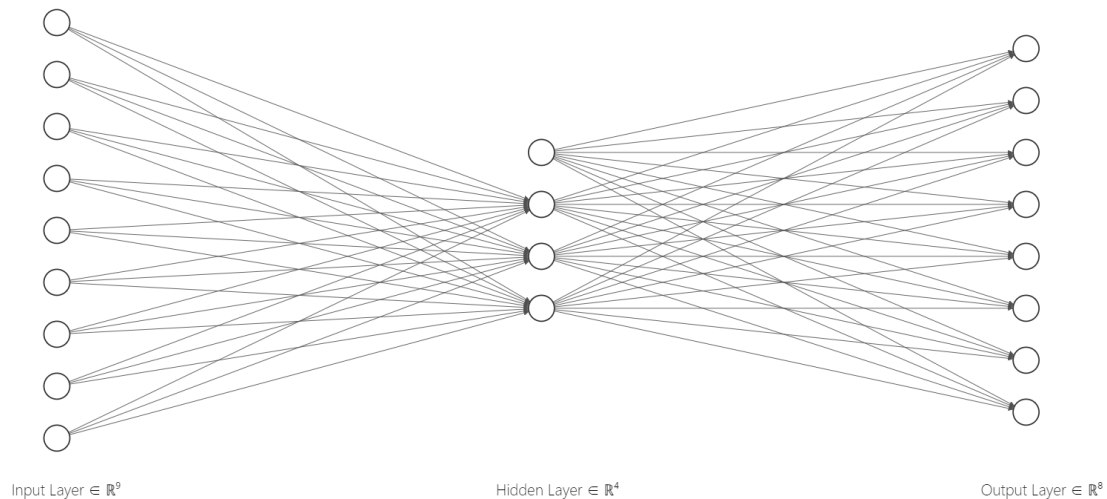


图 1-7 自编码器

2) 生成对抗网络

GAN 被评价为“20 年来机器学习领域最酷的想法”，其利用深度网络生成数据，也利用深度网络判别，其结构如图 1-8。

D 为判别网络，G 为生成网络。

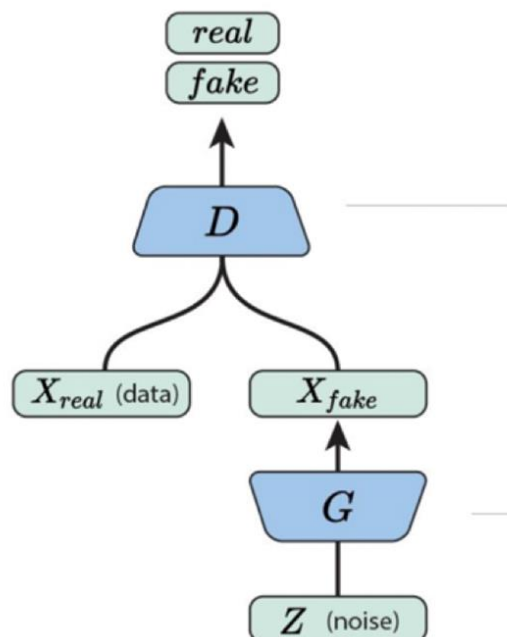


图 1-8 生成对抗网络

真实数据和 G 生成网络伪造的数据同时交由 D 判别网络判别。当判别网络识别出伪造的数据，优化生成网络，否则优化判别网络，这样反复优化，使得 G 与 D 网络都得到优化。

1.3 环境搭建

1.3.1 GPU 显卡驱动安装

第一步：安装 **cuda**

第二步：安装 **cuDNN**

1.3.2 Tensorflow2 安装

```
pip install tensorflow
```

```
pip install jupyter
```

有时需要 **Visual Studio 2013**。

启动：命令提示符输入：jupyter notebook

1.3.3 检验显卡安装

```
import tensorflow as tf
hello=tf.constant('Hello,TensorFlow!')
sess = tf.Session()
print (sess.run(hello))
    Hello,TensorFlow!
a = tf.constant(10)
b = tf.constant(83)
print (sess.run(b-a))
```