

深度学习的发展历史

1.深度学习的起源阶段

1943 年，心里学家麦卡洛克和数学逻辑学家皮兹发表论文《神经活动中内在思想的逻辑演算》，提出了 MP 模型。MP 模型是模仿神经元的结构和工作原理，构成出的一个基于神经网络的数学模型，本质上是一种“模拟人类大脑”的神经元模型。MP 模型作为人工神经网络的起源，开创了人工神经网络的新时代，也奠定了神经网络模型的基础。

1949 年，加拿大著名心理学家唐纳德·赫布在《行为的组织》中提出了一种基于无监督学习的规则——海布学习规则(Hebb Rule)。海布规则模仿人类认知世界的过程建立一种“网络模型”，该网络模型针对训练集进行大量的训练并提取训练集的统计特征，然后按照样本的相似程度进行分类，把相互之间联系密切的样本分为一类，这样就把样本分成了若干类。海布学习规则与“条件反射”机理一致，为以后的神经网络学习算法奠定了基础，具有重大的历史意义。

20 世纪 50 年代末，在 MP 模型和海布学习规则的研究基础上，美国科学家罗森布拉特发现了一种类似于人类学习过程的学习算法——感知机学习。并于 1958 年，正式提出了由两层神经元组成的神经网络，称之为“感知器”。感知器本质上是一种线性模型，可以对输入的训练集数据进行二分类，且能够在训练集中自动更新权值。感知器的提出吸引了大量科学家对人工神经网络研究的兴趣，对神经网络的发展具有里程碑式的意义。

但随着研究的深入，在 1969 年，“AI 之父”马文·明斯基和 LOGO 语言的创始人西蒙·派珀特共同编写了一本书籍《感知器》，在书中他们证明了单层感知器无法解决线性不可分问题（例如：异或问题）。由于这个致命的缺陷以及没有及时推广感知器到多层神经网络中，在 20 世纪 70 年代，人工神经网络进入了第一个寒冬期，人们对神经网络的研究也停滞了将近 20 年。

2.深度学习的发展阶段

1982 年，著名物理学家约翰·霍普菲尔德发明了 Hopfield 神经网络。Hopfield 神经网络是一种结合存储系统和二元系统的循环神经网络。Hopfield 网络也可以模拟人类的记忆，根据激活函数的选取不同，有连续型和离散型两种类型，分别用于优化计算和联想记忆。但由于容易陷入局部最小值的缺陷，该算法并未在当时引起很大的轰动。

直到 1986 年，深度学习之父杰弗里·辛顿提出了一种适用于多层感知器的反向传播算法——BP 算法。BP 算法在传统神经网络正向传播的基础上，增加了误差的反向传播过程。反向传播过程不断地调整神经元之间的权值和阈值，直到输出的误差达到减小到允许的范围之内，或达到预先设定的训练次数为止。BP 算法完美的解决了非线性分类问题，让人工神经网络再次的引起了人们广泛的关注。

但是由于八十年代计算机的硬件水平有限，如：运算能力跟不上，这就导致当神经网络的规模增大时，再使用 BP 算法会出现“梯度消失”的问题。这使得 BP 算法的发展受到了很大的限制。再加上 90 年代中期，以 SVM 为代表的其它浅层机器学习算法被提出，并在分类、回归问题上均取得了很好的效果，其原理又明显不同于神经网络模型，所以人工神经网络的发展再次进入了瓶颈期。

3.深度学习的爆发阶段

2006 年，杰弗里·辛顿以及他的学生鲁斯兰·萨拉赫丁诺夫正式提出了深度学习的概念。他们在世界顶级学术期刊《科学》发表的一篇文章中详细的给出了“梯度消失”问题的解决方案——通过无监督的学习方法逐层训练算法，再使用有监督的反向传播算法进行调优。该深

度学习方法的提出，立即在学术圈引起了巨大的反响，以斯坦福大学、多伦多大学为代表的众多世界知名高校纷纷投入巨大的人力、财力进行深度学习领域的相关研究。而后又在迅速蔓延到工业界中。

2012 年，在著名的 ImageNet 图像识别大赛中，杰弗里·辛顿领导的小组采用深度学习模型 AlexNet 一举夺冠。AlexNet 采用 ReLU 激活函数，从根本上解决了梯度消失问题，并采用 GPU 极大的提高了模型的运算速度。同年，由斯坦福大学著名的吴恩达教授和世界顶尖计算机专家 Jeff Dean 共同主导的神经网络——DNN 技术在图像识别领域取得了惊人的成绩，在 ImageNet 评测中成功的把错误率从 26%降低到了 15%。深度学习算法在世界大赛的脱颖而出，也再一次吸引了学术界和工业界对于深度学习领域的关注。

随着深度学习技术的不断进步以及数据处理能力的不断提升，2014 年，Facebook 基于深度学习技术的 DeepFace 项目，在人脸识别方面的准确率已经能达到 97%以上，跟人类识别的准确率几乎没有差别。这样的结果也再一次证明了深度学习算法在图像识别方面的一骑绝尘。

2016 年，随着谷歌公司基于深度学习开发的 AlphaGo 以 4:1 的比分战胜了国际顶尖围棋高手李世石，深度学习的热度一时无两。后来，AlphaGo 又接连和众多世界级围棋高手过招，均取得了完胜。这也证明了在围棋界，基于深度学习技术的机器人已经超越了人类。

2017 年，基于强化学习算法的 AlphaGo 升级版 AlphaGo Zero 横空出世。其采用“从零开始”、“无师自通”的学习模式，以 100:0 的比分轻而易举打败了之前的 AlphaGo。除了围棋，它还精通国际象棋等其它棋类游戏，可以说是真正的棋类“天才”。此外在这一年，深度学习的相关算法在医疗、金融、艺术、无人驾驶等多个领域均取得了显著的成果。所以，也有专家把 2017 年看作是深度学习甚至是人工智能发展最为突飞猛进的一年。

人工智能、机器学习、深度学习有什么区别和联系



机器学习是一种实现人工智能的方法，深度学习是一种实现机器学习的技术。在应用场景中

人工智能的研究领域在不断的扩大，包括专家系统、机器学习、进化计算、模糊逻辑、计算机视觉、自然语言处理、推荐系统等。并且目前的科研工作都集中在弱人工智能这部分。

机器学习直接来源于早期的人工智能领域，传统的算法包括决策树、聚类、贝叶斯分类、支持向量机、EM、Adaboost 等等。从学习方法上来分，机器学习可以分为监督学习(如分类问题)、无监督学习(如聚类问题)、半监督学习、集成学习、深度学习和强化学习。传统的机器学习算法在指纹识别、人脸检测、特征物体检测等领域的应用基本达到了商业化的要求或特定场景的商业化水平。

深度学习本来并不是一种独立的学习方法，其本身也会用到监督学习和无监督学习方法来训练深度神经网络，但由于近年来该领域发展迅猛，一些特有的学习手段相继被提出(如残差网络)，因此越来越多的人将其单独看作一种学习方法。主要应用在互联网、安防、金融、智能硬件、医疗、教育等行业，在人脸技术、图象识别、智能监控、文字识别、语义分析等领域。

神经元、单层感知机、多层感知机

神经元：

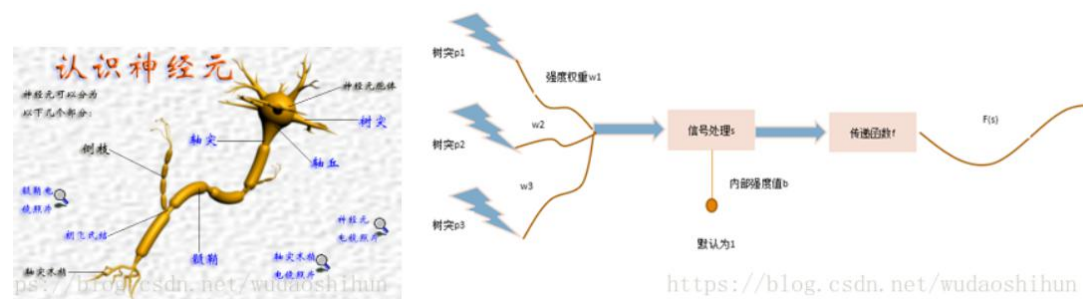
人工神经元模拟了生物神经元的仿生工程。是模仿生物神经元组成的神经网络在接收刺激及信号传递等行为。具体可查看第二章第一节人工神经元构造。

我们可把神经元归纳为：

感觉神经元（传入神经元）：其树突的末端分布于身体的外周部，接收来自体内外的刺激，将兴奋传至脊髓和脑。

运动神经元（传出神经元）：其轴突大于肌肉和腺体。

联络神经元（中间神经元）：结余上述 2 种神经元之间，起着神经元之间技能联系的作用，多存在于脑和脊髓里。



单层感知机：

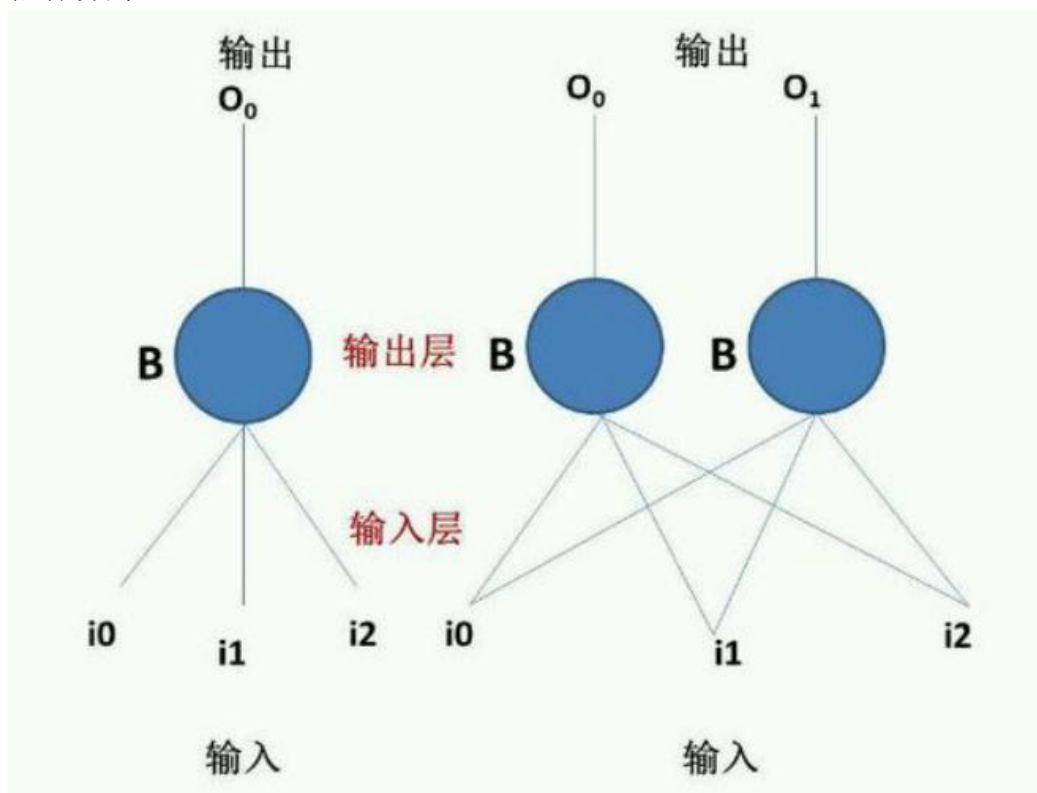
单层感知器可以用来区分线性可分的数据，并且一定可以在有限的迭代次数中收敛。

从使用过程来看，单层的感知器其实看起来是相当简单的：对假设函数进行训练，直到对所有的样本都满足经过假设函数计算出来的结果与预期结果一致，就结束了。

单层感知器能够用来模拟逻辑函数，例如逻辑非 NOT、逻辑或非 XOR、逻辑或 OR、逻辑与 AND 和逻辑与非 NAND 等，但是不能用来模拟逻辑异或 XOR（对于这个函数，必须用两层神经元）。稍后将讨论这个问题。

除了输入之外，偏置（bias）也经常用于每个神经元，它在通过转换函数之前被加到输入的加权和上。权值也经常被应用于偏置上。偏置决定多少输入激活（输入的加权和）才能激发神经元进入兴奋状态。偏置往往被设定为 1，而偏置的权值可以通过学习算法加以调整。

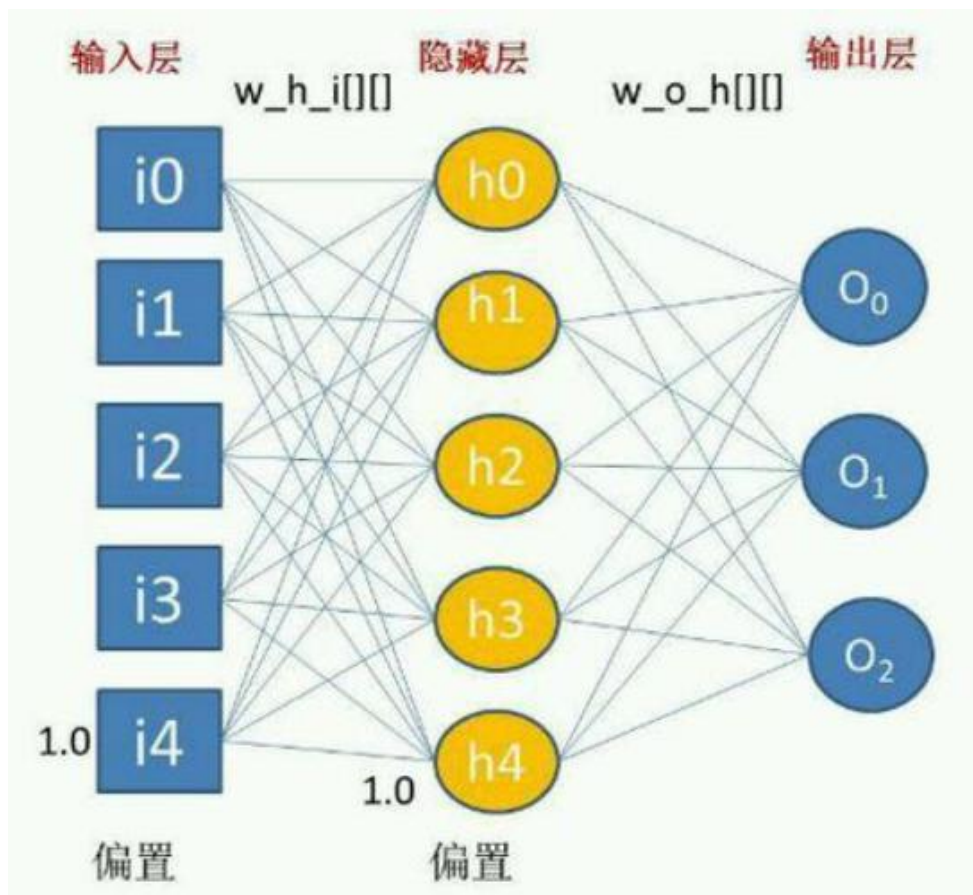
不应将单层感知器和单个神经元相混淆。如图所示的网络就是一个单层感知器，因为它只由一层构成。对于多维问题，必须采用多层感知器。表示单层感知器非常简单。由于输入和权值存在一一对应关系，因此计算输出是很容易的。



多层感知机：

Minsky 和 Papert 在他们的 *Perception* 一书中揭示：单层感知器有一个缺点，即它们只能对线性可分的数据集进行分类。然而，此后不久人们就发现，如果见单层感知器叠加成多层感知器，则理论上就具备对任何分类问题的解决能力。事实上，多层感知器可以模拟成任意复杂的函数，其中函数的复杂性取决于网络输入个数和隐层个数。

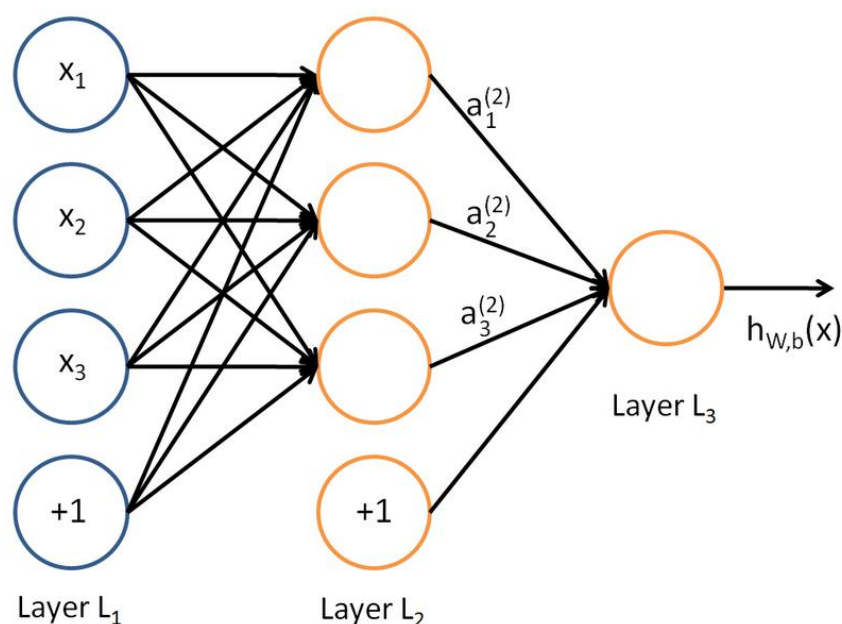
多层感知器中神经元的性质和单层感知器中的一样（例如偏置等）。然而由于存在多层，多层感知器中某一层的输出就是下一层的输入。因此多层感知器的实现比单层感知器略显复杂，但还是很直观的。注意，在这里对隐层结点和输出层结点使用相同的激励函数，即 sigmoid 函数。Sigmoid 函数可以将神经元的输出值规格化到区间 $[0, 1]$ 。



该多层感知器有 4 个输入单元，4 个隐层单元和 3 个输出单元。偏置作为输入单元和隐层单元来处理，但偏置具有常数值 1（虽然可以通过调整权值来改变偏置的影响）。

程序的执行流程如下：首先，计算隐层单元的输出（利用输入单元和隐层单元到输入层单元的权值）；然后，继续计算下一层（这里指输出层）单元的输出。通常这个过程被称为前向传送数据，或简单的称之为前馈（feed forward）。可以把神经网络看成并行计算系统。每个神经元是一个处理元件，该处理元件接受一个或多个输入并产生一个输出。输入和输出均可以看成信息。在多层感知器架构中，输出可以被送往多个其他处理单元，做进一步的处理。神经网络并行性的特性来自于同一层多个神经元之间的并行性。所有的神经元能够同时处理它们各自的输入，使得在多处理器系统中同一层次上具有大量神经元的神经网络的运行速度更快。

什么是前向传播？



对于第 2 层第 1 个节点的输出：

$$a_1^2 = \sigma(z_1^2) = \sigma(w_{11}^2 x_1 + w_{12}^2 x_2 + w_{13}^2 x_3 + b_1^2)$$

对于第 3 层第 1 个节点的输出：

$$a_1^3 = \sigma(z_1^3) = \sigma(w_{11}^3 a_1^2 + w_{12}^3 a_2^2 + w_{13}^3 a_3^2 + b_1^3)$$

一般化的，假设 1-1 层有 m 个神经元，对于 a_j^l 有：

$$a_j^l = \sigma(z_j^l) = \sigma\left(\sum_{k=1}^m w_{jk}^l a_k^{l-1} + b_j^l\right)$$

也就是第 1 层第 j 个神经元的输入为与它相连的上一层每个神经元的输出加权求和后加上该神经元对应的偏置，该神经元所做的工作只是把这个结果做一个非线性激活。

什么是反向传播？

反向传播算法（Backpropagation）是目前用来训练人工神经网络（Artificial Neural Network, ANN）的最常用且最有效的算法。其主要思想是：

（1）将训练集数据输入到 ANN 的输入层，经过隐藏层，最后达到输出层并输出结果，这是 ANN 的前向传播过程；

（2）由于 ANN 的输出结果与实际结果有误差，则计算估计值与实际值之间的误差，并将该误差从输出层向隐藏层反向传播，直至传播到输入层；

（3）在反向传播的过程中，根据误差调整各种参数的值；不断迭代上述过程，直至收敛。