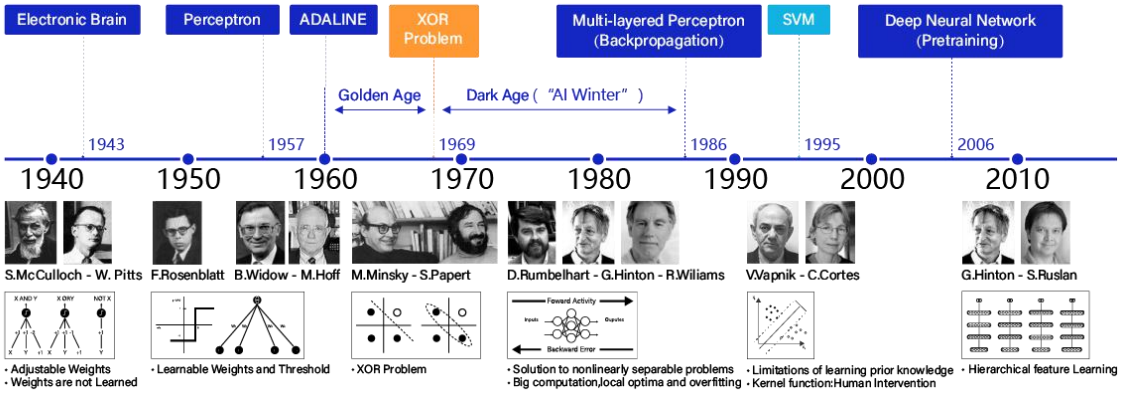
1. 深度学习发展历史



- 1940 年代：首次提出神经元的结构，但权重是不可学的。

- 50-60 年代：提出权重学习理论，神经元结构趋于完善，开启了神经网络的第一个黄金时代。

- 1969 年：提出异或问题（人们惊讶的发现神经网络模型连简单的异或问题也无法解决，对其的期望从云端跌落到谷底），神经网络模型进入了被束之高阁的黑暗时代。

- 1986 年：新提出的多层神经网络解决了异或问题，但随着 90 年代后理论更完备并且实践效果更好的 SVM 等机器学习模型的兴起，神经网络并未得到重视。

- 2010 年左右：深度学习进入真正兴起时期。随着神经网络模型改进的技术在语音和计算机视觉任务上大放异彩，也逐渐被证明在更多的任务，如自然语言处理以及海量数据的任务上更加有效。至此，神经网络模型重新焕发生机，并有了一个更加响亮的名字：深度学习。

  为何神经网络到 2010 年后才焕发生机呢？这与深度学习成功所依赖的先决条件：大数据涌现、硬件发展和算法优化有关。

- 大数据是神经网络发展的有效前提。神经网络和深度学习是非常强大的模型，需要足够量级的训练数据。时至今日，之所以很多传统机器学习算法和人工特征依然是足够有效的方案，原因在于很多场景下没有足够的标记数据来支撑深度学习。深度学习的能力特别像科学家阿基米德的豪言壮语：“给我一根足够长的杠杆，我能撬动地球！”。深度学习也可以发出类似的豪言：“给我足够多的数据，我能够学习任何复杂的关系”。但在现实中，足够长的杠杆与足够多的数据一样，往往只能是一种美好的愿景。直到近些年，各行业 IT 化程度提高，累积的数据量爆发式地增长，才使得应用深度学习模型成为可能。

- 依靠硬件的发展和算法的优化。现阶段，依靠更强大的计算机、GPU、autoencoder 预训练和并行计算等技术，深度学习在模型训练上的困难已经被逐渐克服。其中，数据量和硬件是更主要的原因。没有前两者，科学家们想优化算法都无从进行。

2. 人工智能、机器学习、深度学习有什么区别和联系

概括来说，人工智能、机器学习和深度学习覆盖的技术范畴是逐层递减的。

人工智能是最宽泛的概念。

机器学习是当前比较有效的一种实现人工智能的方式。

深度学习是机器学习算法中最热门的一个分支，近些年取得了显著的进展，并替代了大多数传统机器学习算法。三者的关系如图所示，即：人工智能 > 机器学习 > 深度学习。

区别：人工智能是一个大杂烩，机器学习（监督学习）则有更加明确的指代。

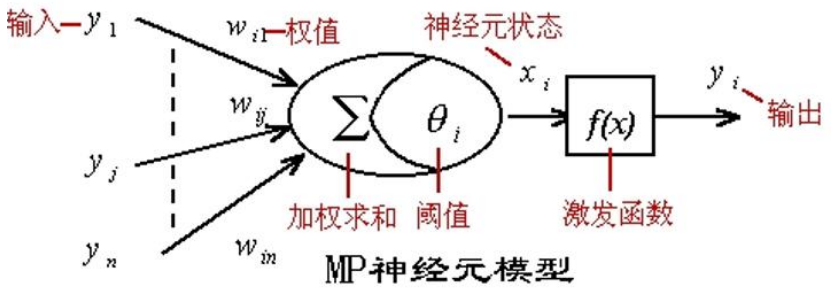
3. 神经元、单层感知机、多层感知机

神经元：

神经网络中每个节点称为神经元，由两部分组成：

加权和：将所有输入加权求和。

非线性变换（激活函数）：加权和的结果经过一个非线性函数变换，让神经元计算具备非线性的能力。

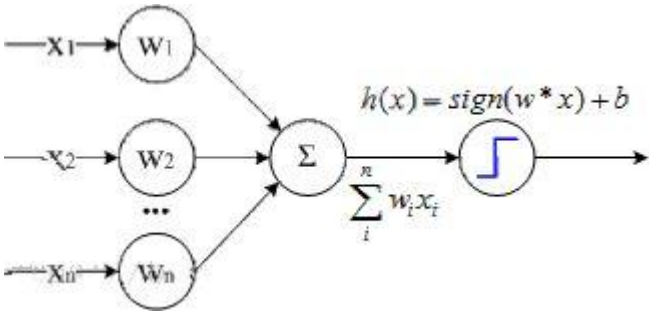


单层感知机：

单层感知机目标是将被感知数据集划分为两类的分离超平面，并计算出该超平面。单层感知机是二分类的线性分类模型，输入是被感知数据集的特征向量，输出时数据集的类别{+1,-1}。感知器的模型可以简单表示为：

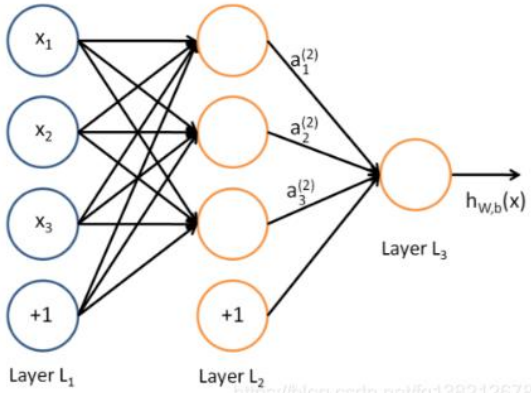
$$f(x)=sign(w.x+b)$$

该函数称为单层感知机，其中 w 是网络的 N 维权重向量，b 是网络的 N 维偏置向量，w.x 是 w 和 x 的内积，w 和 b 的 N 维向量取值要求在实数域。



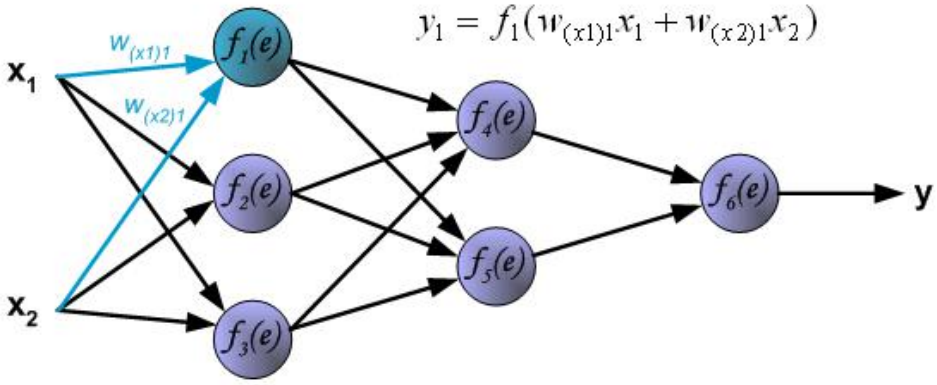
多层感知机：

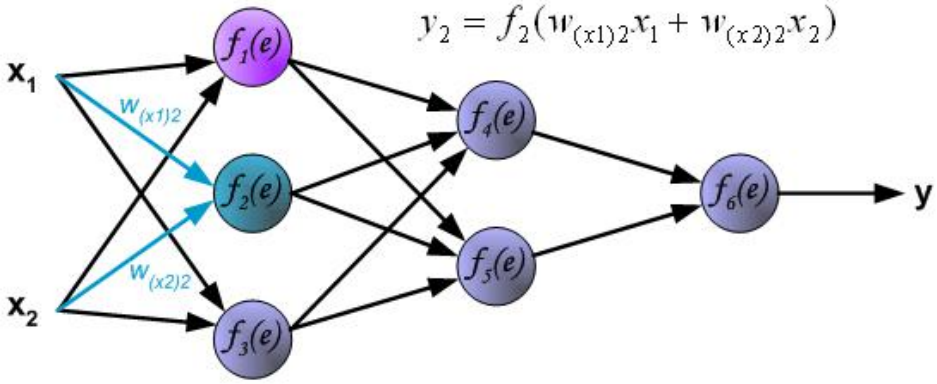
多层感知机在单层神经网络的基础上引入了一到多个隐藏层（hidden layer）。隐藏层位于输入层和输出层之间。多层感知机层与层之间是全连接的。多层感知机最底层是输入层，中间是隐藏层，最后是输出层。

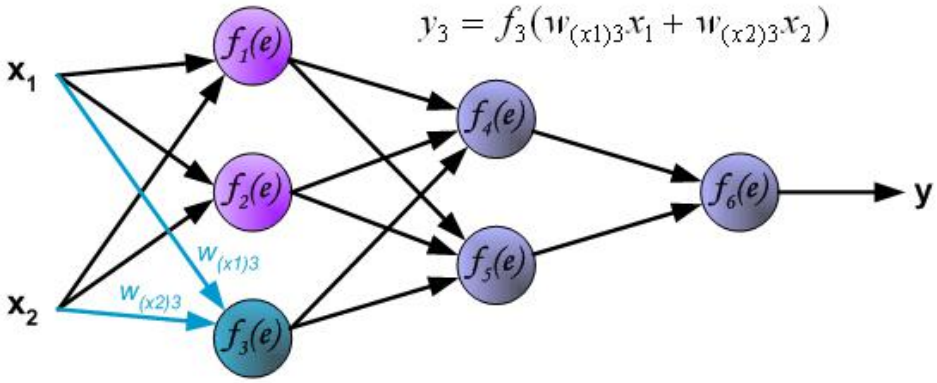


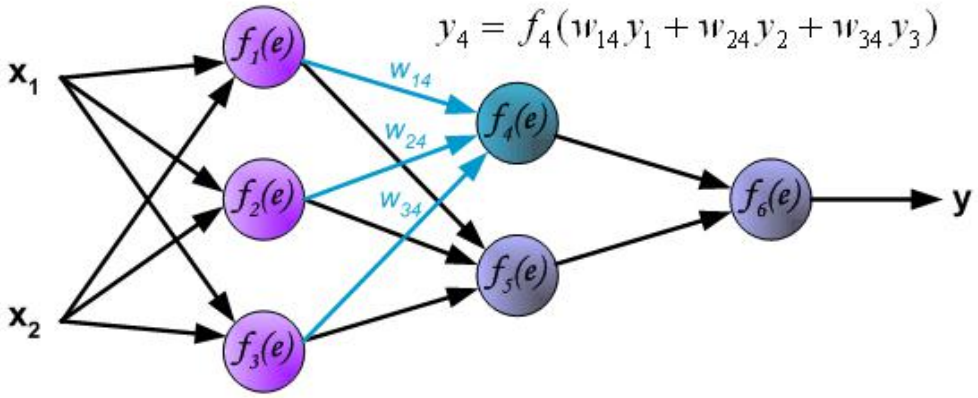
4. 什么是前向传播（包含图文示例）

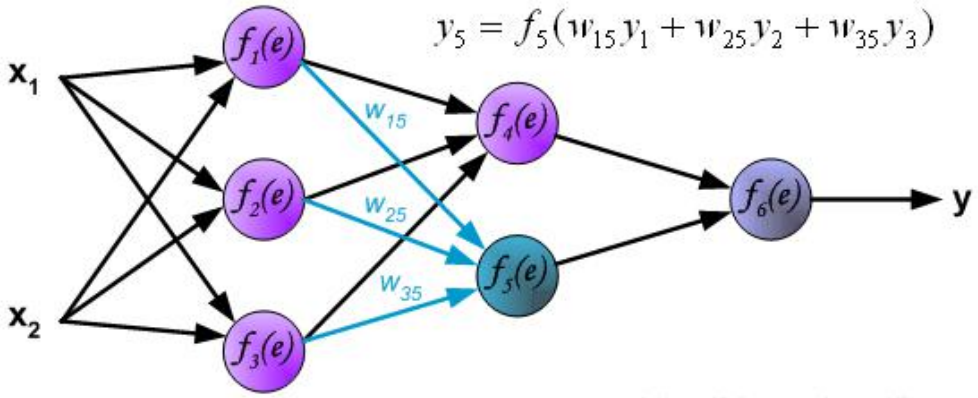
前向传播：从输入经过一层层隐层计算得到输出的过程即为前向过程，也称前向传播。

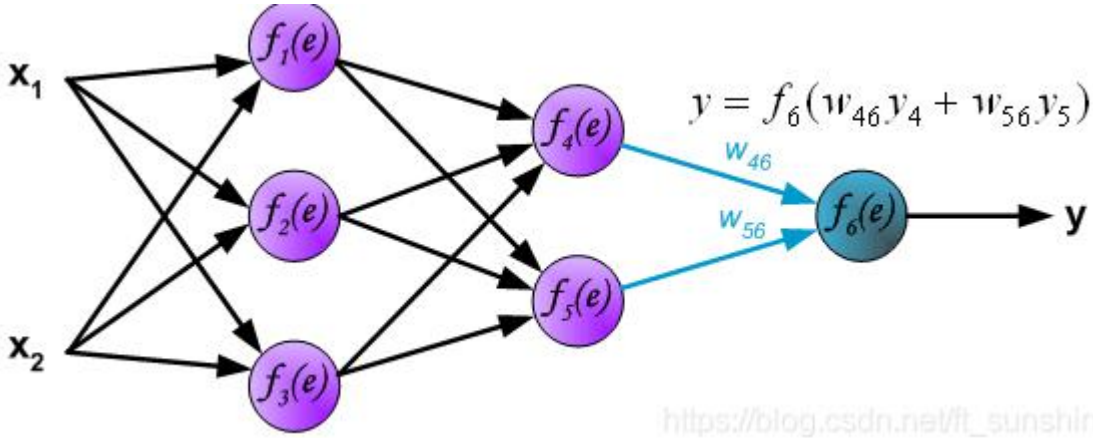








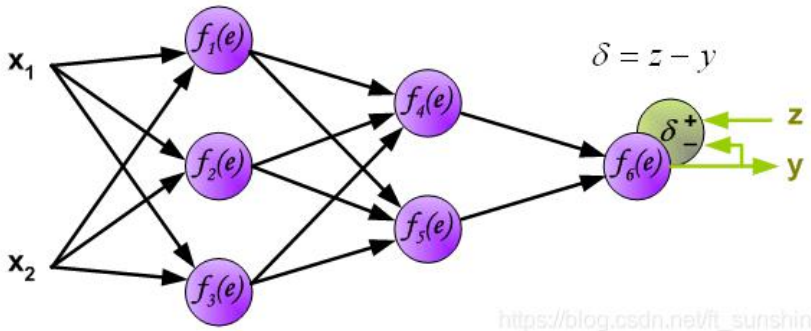




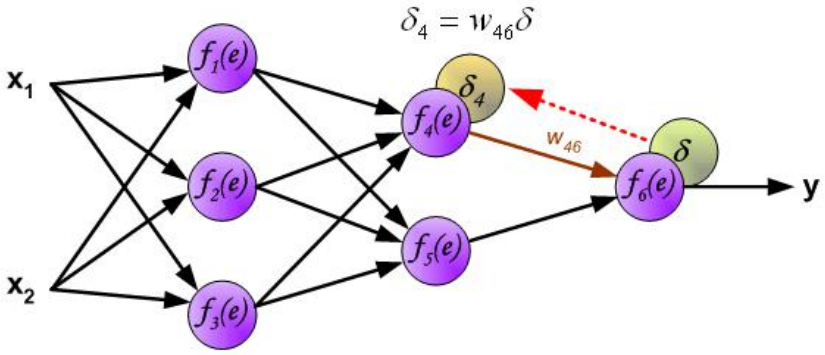
5. 什么是反向传播（包含图文示例）

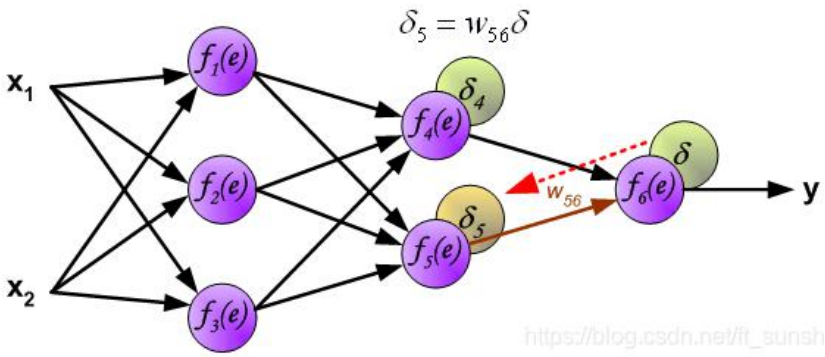
反向传播（Backward Propagation）则是与前向传播的计算方向相反，它是通过输出值与真实值之间的误差，来更新训练参数，具体来说反向传播是根据损失函数，通过网络反向流动来计算每一层参数的梯度（偏导数），从而更新参数。反向传播解决了神经网络中训练模型时参数的更新问题。

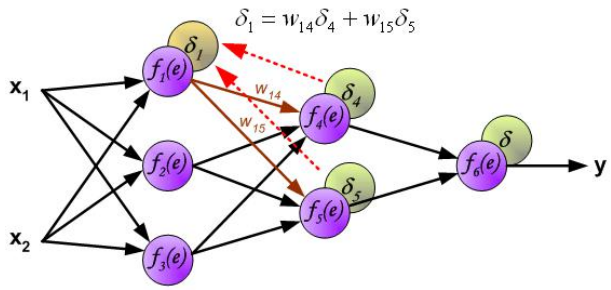
预测结果和真实的标签（z）相比较，计算预测结果和真实标签的误差（δ \delta δ），如下：

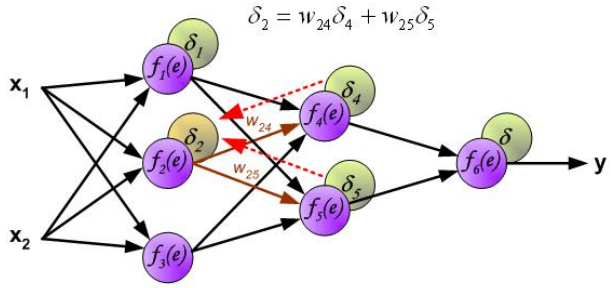


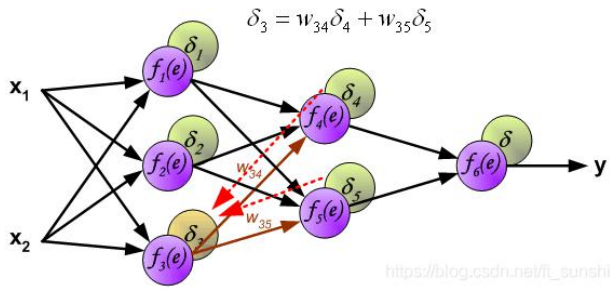
计算每个神经元的误差（δ \delta$\delta$）











利用反向传播的误差，计算各个神经元（权重）的导数，开始反向传播修改权重：