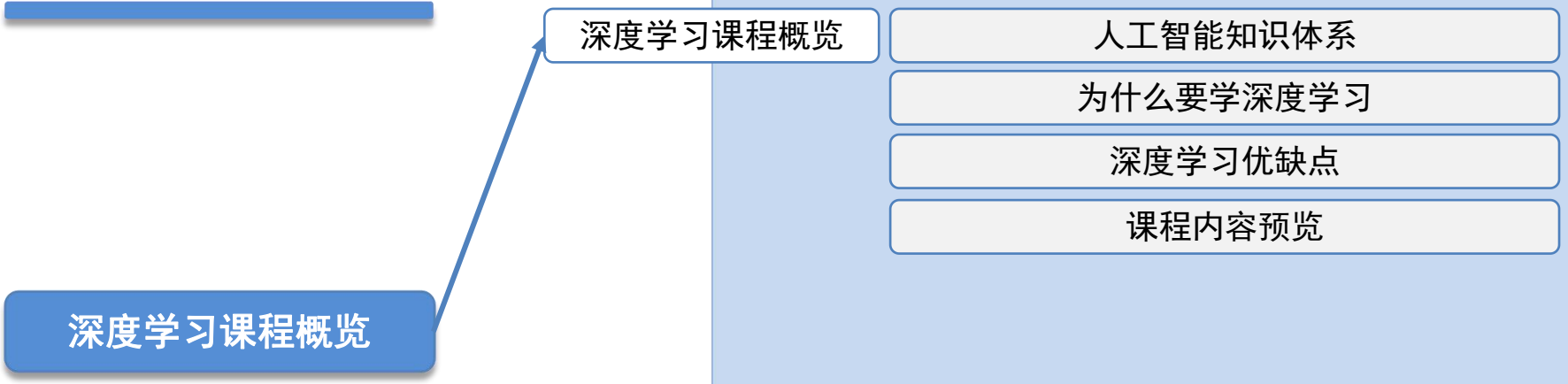
Tedu.cn
达内教育

深度学习

DEEP LEARNING

DAY01

深度学习课程概览



深度学习课程概览

引入

这个世界正在发生一些不可思议的事，而且完全超出你的想象

深度学习应用示例（一）

- 照片上色

人工智能技术对1937年的旷工照片上色

深度学习应用示例（二）

- 换脸

人脸更换

深度学习应用示例（三）

- 图像风格转换

生成毕加索、梵高、莫奈风格的蒙娜丽莎

深度学习应用示例（四）

- 虚拟主播

采用语音图像合成技术虚拟的新闻主播

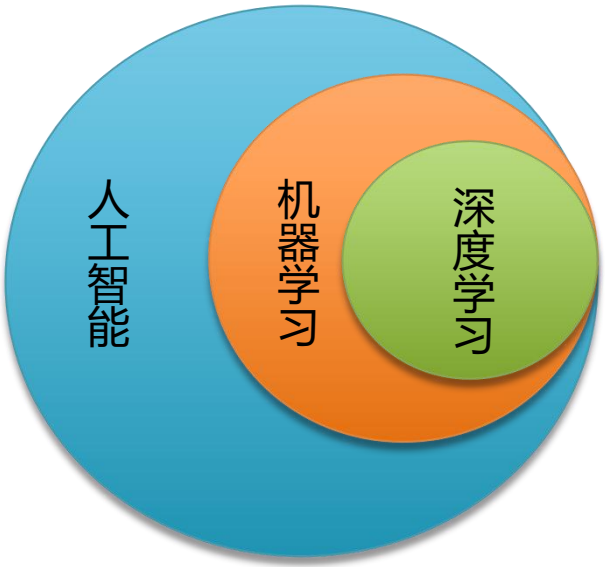
深度学习应用示例（五）

- **声音模仿**

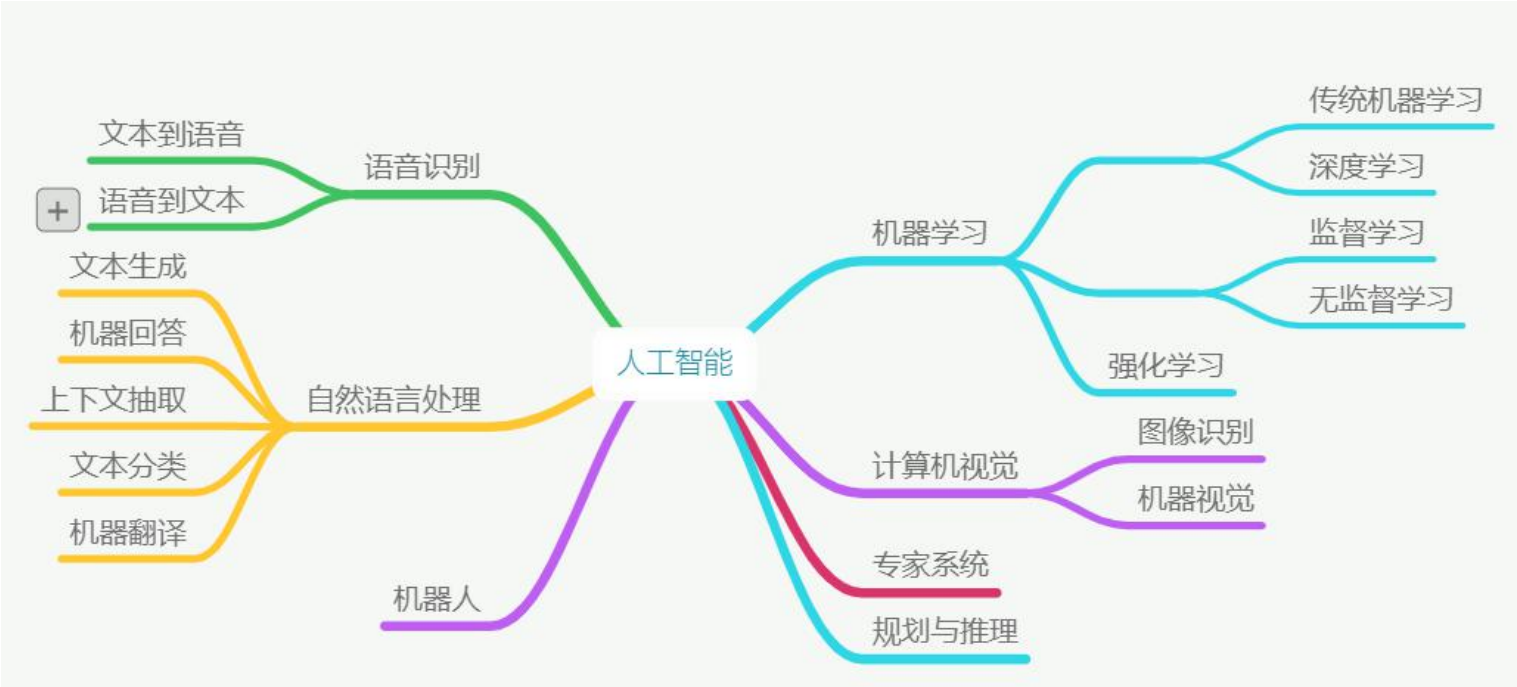
机器模仿歌星声音

深度学习

- 这些应用背后都有一项技术：深度学习
- 深度学习属于机器学习，是机器学习“高级阶段”



人工智能知识体系


人工智能
语音识别
文本到语音
语音到文本
自然语言处理
文本生成
机器回答
上下文抽取
文本分类
机器翻译
机器人
机器学习
传统机器学习
深度学习
监督学习
无监督学习
强化学习
计算机视觉
图像识别
机器视觉
专家系统
规划与推理

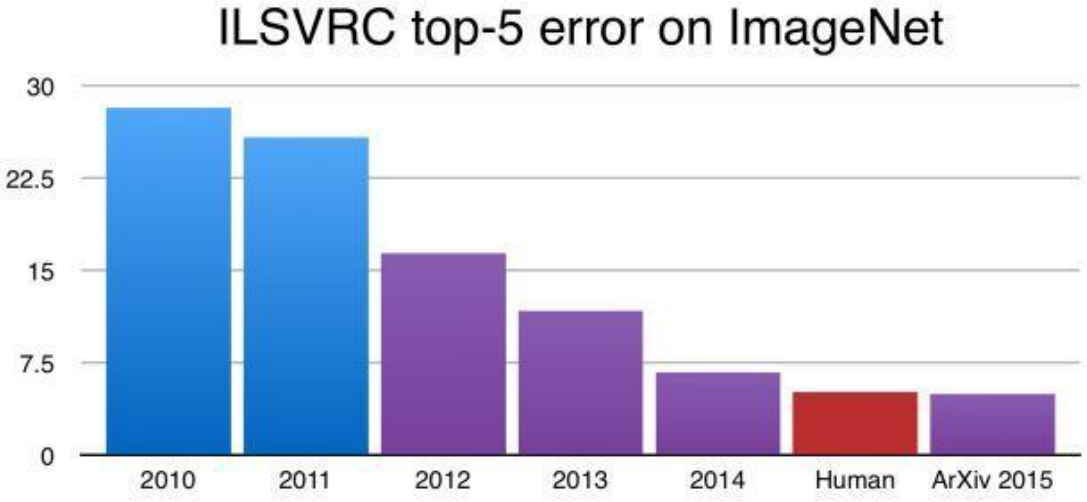
为什么要学习深度学习

- 深度学习具有更强的解决问题能力（例如图像识别准确率明显超过机器学习，甚至超过了人类）
- 掌握深度学习具有更强的职业竞争力
- 深度学习在行业中应用更广泛

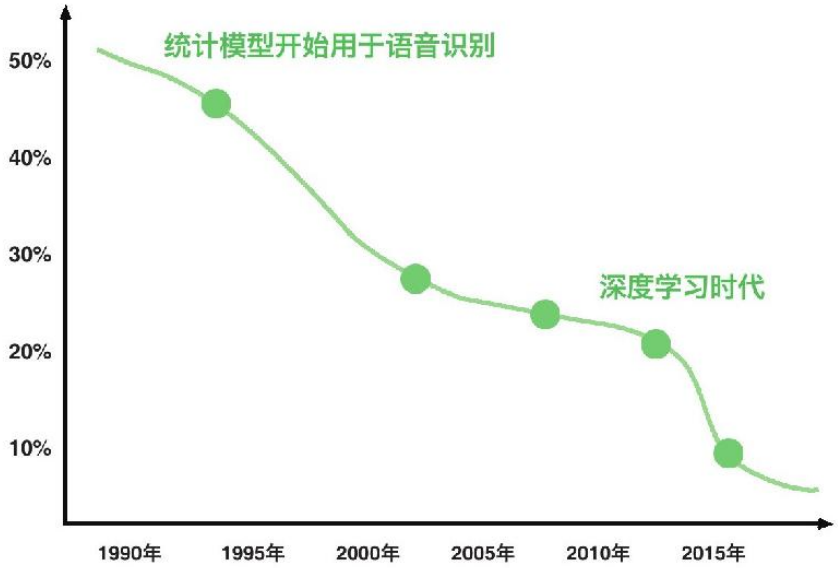
深度学习优点（一）

- **深度学习性能更优异**

下图是历年ImageNet大规模视觉识别挑战（ILSVRC）的分类精度，其中蓝色是经典机器学习方法，其它为深度学习方法。2015年比赛成绩，识别率超过了人类



同样，在语音识别领域，进入深度学习时代后，识别率有了明显的提高。Google在2015年5月举办的Google I/O年度开发者大会上宣布，其语音识别系统已将识别错误率降低到了惊人的8%；而后IBM的Watson智能系统很快就将语音识别的错误率降低到了6.9%；2016年9月，微软最新的基于深度学习的语音识别系统已经成功地将识别错误率降低到了6.3%



近20年来语音识别错误率的下降趋势图

深度学习优点（二）

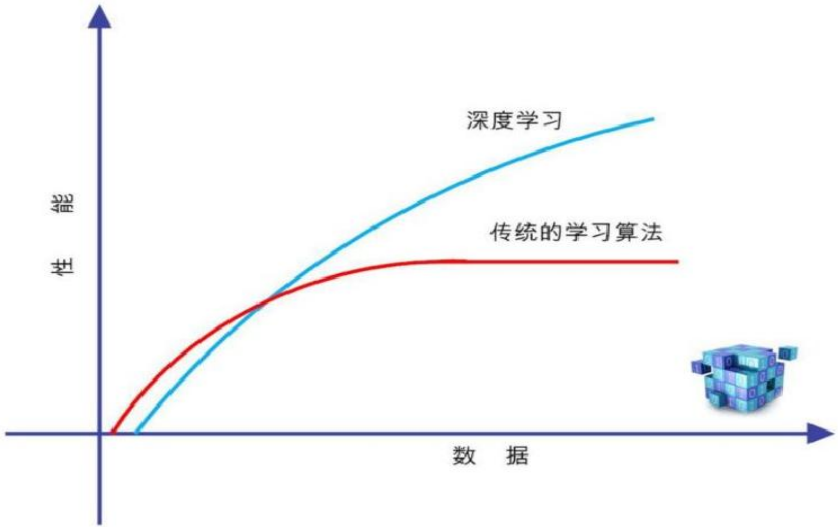
- **深度学习不需要特征工程**

传统机器学习需要人进行特征提取（特征工程），机器性能高度依赖于特征工程的质量。在特征很复杂的情况下，人就显得无能为力。而深度学习不需要这样的特征工程，只需将数据直接传递给深度学习网络，由机器完成特征提取。

例如：波士顿房价预测案例中，考虑了犯罪率（CRIM）、住宅用地占比（ZN）、非商业用地所占尺寸（INDUS）、查尔斯河虚拟变量（CHAS）、环保指数（NOX）、每栋住宅的房间数（RM）、1940年以前建成的自建单位比例（AGE）、距离5个波士顿就业中心的加权距离（DIS）、距离高速公路便利指数（RAD）、每一万元不动产税率（TAX）、教师学生比（PTRATIO）、黑人比例（B）、房东属于中低收入比例（LSTAT）等13个特征

深度学习优点（三）

- 深度学习在大样本数据下有更好的扩展性和性能



深度学习优点（四）

- 深度学习能解决传统机器学习无法解决的问题（如深度特征提取、特征复杂、数据量大）

深度学习缺点

- 深度学习在小数据上性能不如传统机器学习
- 深度学习网络结构复杂、构建成本高
- 传统机器学习比深度学习具有更好的解释性

深度学习与传统机器学习对比

比较项	传统机器学习	深度学习
特征提取	人提取特征	机器自己发现特征
准确度/精度	低	高
过程是否可知	可知	不可知
训练数据	小	大
模型结构	简单	复杂

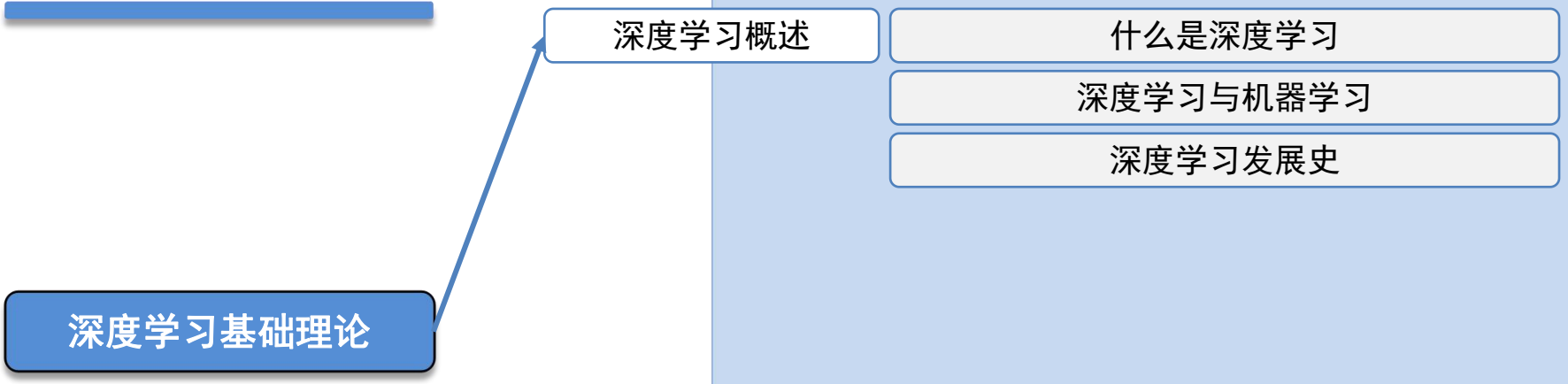
课程内容


深度学习
基本理论与算法
感知机
激活函数 *
神经网络 *
损失函数与梯度下降 *
反向传播算法 *
卷积神经网络（CNN） *
卷积函数与卷积运算
CNN网络结构
卷积层
池化层
全连接
激活层
循环神经网络（RNN）
长短期记忆（LSTM）
深度学习平台及框架
百度：PaddlePaddle
Google: Tensorflow
案例
动物图像分类
明星人脸识别
新闻资讯分类
目标物体检测

课程特点

- 概念术语多，理论复杂，学习曲线陡峭，需要长期、反复学习、理解、体会、实践
- 需要部分数学知识（记住结论、会使用接口、理解公式）
- 案例复杂度高
- 与传统程序差异较大

深度学习基础理论



深度学习概述

人工智能划时代事件

- 2016年3月，Google公司研发的AlphaGo以4:1击败世界围棋顶级选手李世石。次年，AlphaGo2.0对战世界最年轻的围棋四冠王柯洁，以3:0击败对方。背后支撑AlphaGo具备如此强大能力的，就是“深度学习”（Deep Learning）。
- 一时间，“深度学习”这个本专属于计算机学科的术语，成为包括学术界、工业界、风险投资界等众多领域的热词。

深度学习巨大影响

- 除了博弈，深度学习在计算机视觉（computer vision）、语音识别、自动驾驶等领域，表现与人类一样好，甚至有些地方超过了人类。**2013年**，深度学习就被麻省理工学院的《MIT科技评论》**评为世界10大突破性技术之一。**
- 深度学习不仅是一种算法升级，还是一种全新的思维方式，它的颠覆性在于，将人类过去痴迷的算法问题，演变成数据和计算问题，**以前“算法为核心竞争力”正在转换为“数据为核心竞争力”**。

深度学习巨大影响（续）

- 在人工智能领域，**深度学习**之所以备受瞩目，是因为从原始输入层开始，到中间每一个隐含层的数据抽取、变换，到最终输出层的判断，**所有的特征提取**，全程是一个**没有人工干预**的训练过程。这个自主过程，在机器学习领域是革命性的。
- 著名深度学习专家吴恩达（Andrew Ng）层表示：“我们没有像通常（机器学习那样），自己来框定边界，而是直接把海量数据投放到算法中去，让数据自己说话，系统会自动从数据中学习。”
- Google大脑项目计算机科学家杰夫.迪恩（Jeff Dean）则说，在训练时候，我们从来不会告诉机器说：这是一只猫。实际上，是系统自己发明或领悟了“猫”的概念。

什么是学习

- 1975年图灵奖获得者、1978年诺贝尔经济学奖获得者、著名学者赫伯特.西蒙（Herbert Simon）层下过一个定义：**如果一个系统，能够通过执行某个过程，就此改进了它的性能，那么这个过程就是学习。**由此可看出，学习的目的就是改善性能。

什么是学习（续）

- 卡耐基梅隆大学机器学习和人工智能教授汤姆.米切尔（Tom Mitchell）在他的经典教材《机器学习》中，给出了更为具体的定义：

对于某类任务（Task，简称T）和某项性能评价准则（Performance，简称P），如果一个计算机在程序T上，以P作为性能度量，随着经验（Experience，简称E）的积累，不断自我完善，那么我们称计算机程序从经验E中进行了学习。

特征工程

- **经典机器学习**，通常是用人类的先验知识，把原始数据预处理成各种特征（Feature），然后对特征进行分类。然而，这种分类的**效果，高度取决于特征选取的好坏。**传统的机器学习专家们，把大部分时间都花在如何寻找更加合适的特征上。因此，早期的机器学习专家非常辛苦。传统的机器学习，其实可以有一个更合适的称呼—特征工程（Feature Engineering）。通过人去发现特征，容易遭遇性能的瓶颈。

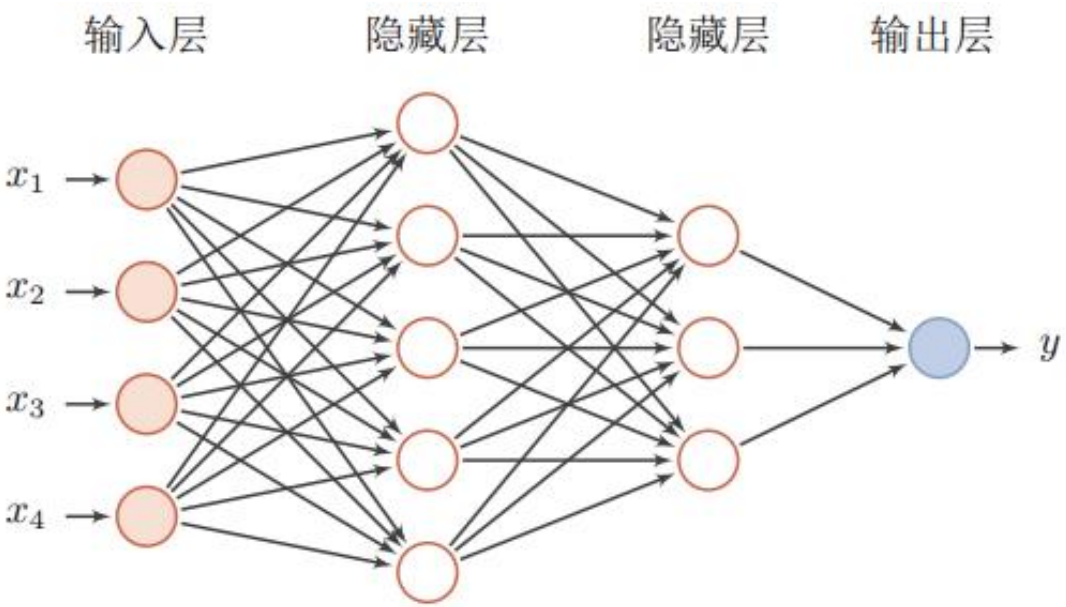
什么是深度学习

- 随着研究的进行，机器学习的专家们发现，**可以让神经网络自己学习如何抓取数据的特征**，这种学习方式的**效果似乎更佳。**于是兴起了特征表示学习（Feature Representation Learning）的风潮。这种学习方式，对数据的拟合也更加灵活好用。于是，人们终于从自寻特征的痛苦生活中解脱了出来。
- 再后来，网络进一步加深，出现了多层次的“表示学习”，它把学习的性能提升到另一个高度。这种学习的层次多了，其实也就是套路深了。于是，人们就给它取了一个特别的名称—Deep Learning（深度学习）。

什么是深度学习（续）

- 简单来说，深度学习就是一种包括多个隐含层（越多即为越深）的多层感知机。它通过组合低层特征，形成更为抽象的高层表示，用以描述被识别对象的高级属性类别或特征。能自生成数据的中间表示（虽然这个表示并不能被人类理解），是深度学习区别于其它机器学习算法的独门绝技。
- 所以，深度学习可以总结成：**通过加深网络，提取数据深层次特征**

深度神经网络示意图



深度学习发展历史（一）

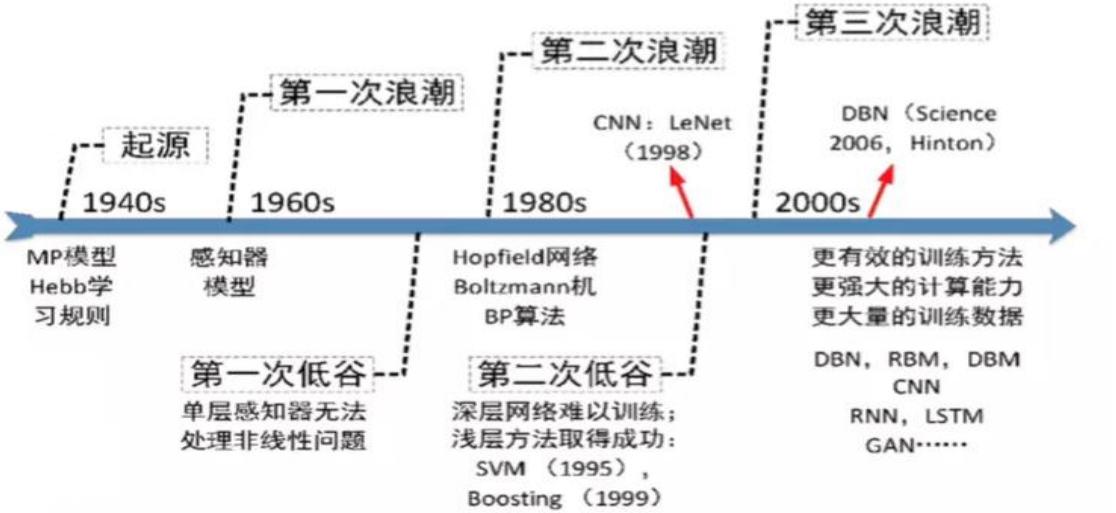
- 从1940年起，首先提出了MP模型Hebb(海布)学习规则.这是神经网络的起源，也奠定了神经网络的基础模型。
- 1960年，提出了感知机模型，感知机模型可以对简单的数据节点进行分类，这个发现引起了第一波的AI浪潮，因为人们认为简单的感知机可以实现分类功能，那通过组合可以实现更复杂的功能，但后面发现感知机无法模拟异或运算，无法处理非线性的问题，第一波浪潮就这样沉入了低谷。

深度学习发展历史（二）

- 1980年Hopfiled网络，Boltzmann机和BP算法的提出，人们发现可以增加网络的深度来实现非线性的功能，所以开始了第二次浪潮。但是在80年代，计算机的计算能力十分有限，很难训练出一个有效的模型来使用，所以导致了这种方式始终处于鸡肋的状态。再加上同一时期浅层方法的成功，如SVM(1995)， 使得人们转为研究浅层的方法。
- 1998年CNN被提出，也应用到了邮政局的邮政编码识别，但是因为当时并不重视这种深度网络，导致并没有火起来。

深度学习发展历史（三）

- 2006年，Hinton提出了DBN（深度信念网络），解决了更深层次的网络是可以通过一些策略更好的训练和实现，所以就引起了现在深度学习的第三次浪潮。

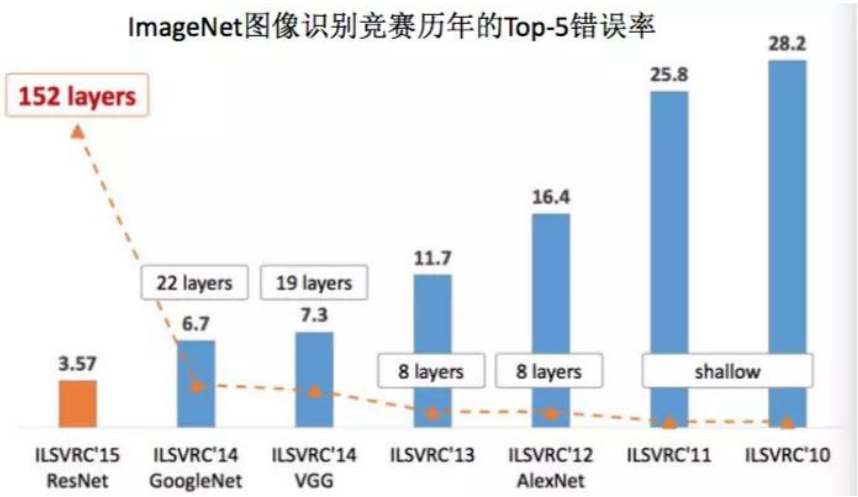


深度学习发展历史（四）

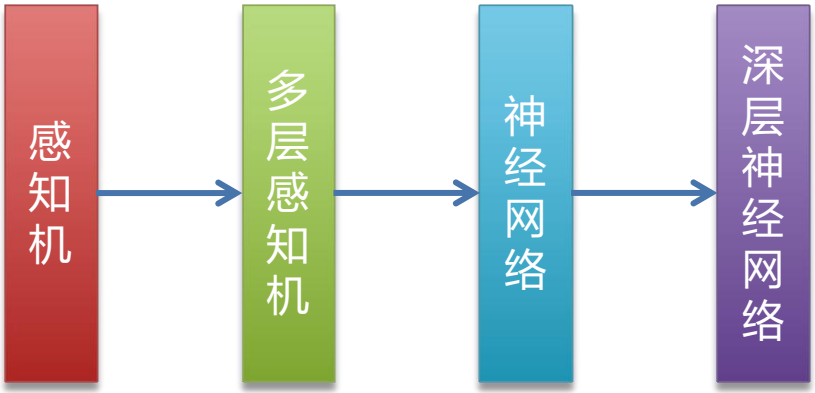
- 相比而言，区别于传统的浅层学习，深度学习强调模型结构的深度，隐含层远远不止一层。通常来说，层数更多的网络，通常具有更强的抽象能力（即数据表征能力），也就能够产生更好的分类识别的结果。
- 2012年，杰弗里·辛顿（Geoffery Hinton）教授团队在ImageNet中首次使用深度学习完胜其他团队，那时网络层深度只有个位数。2014年，谷歌团队把网络做了22层，问鼎当时的ImageNet冠军。到了2015年，微软研究院团队设计的基于深度学习的图像识别算法ResNet，把网络层做到了152层。很快，在2016年，商汤科技更是叹为观止地把网络层做到了1207层。

ImageNet Top5错误率和网络深度

- 2012年冠军（**AlexNet**, top-5错误率16.4%，使用额外数据可达到15.3%，8层神经网络）
- 2014年亚军（**VGGNet**，top-5错误率7.3%，19层神经网络），2014年冠军（InceptionNet，top-5错误率6.7%，22层神经网络）
- 2015年的冠军（**ResNet**，top-5错误率3.57%，152层神经网络）



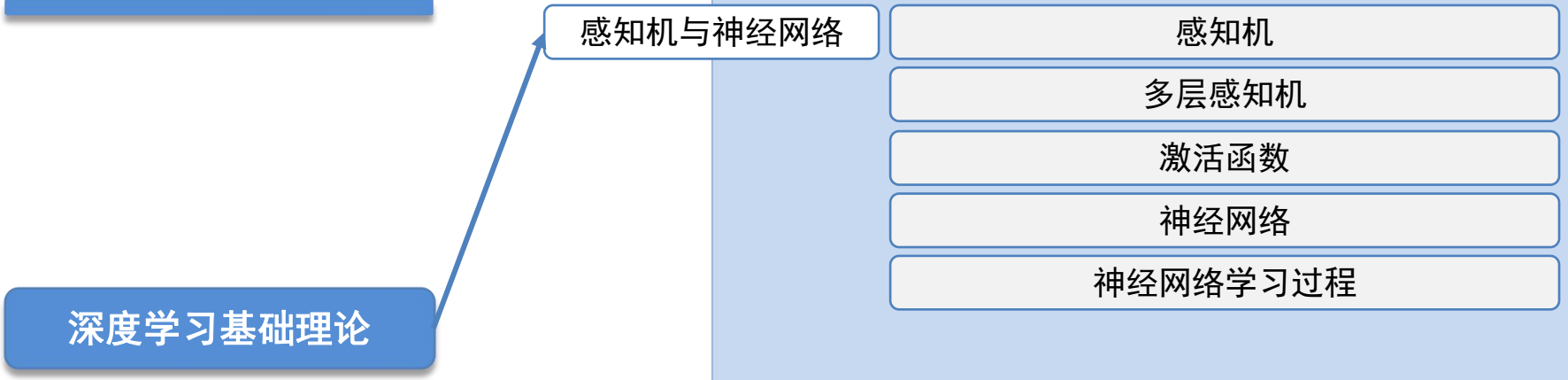
深度网络进化过程



小结

- 时至今日，深度学习网络越来越深，应用越来越广，解决的问题越来越难，扮演的角色越来越重要。但万丈高楼平地起，让我们追根溯源，探索如何深度学习究竟是如何由一个简单的“单细胞”演化成复杂神经网络系统的。

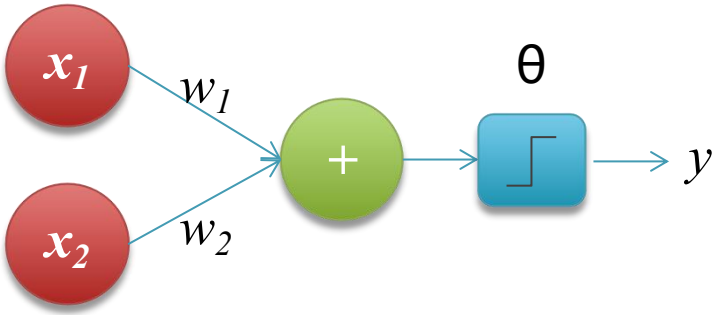
深度学习基础理论



感知机与神经网络

感知机

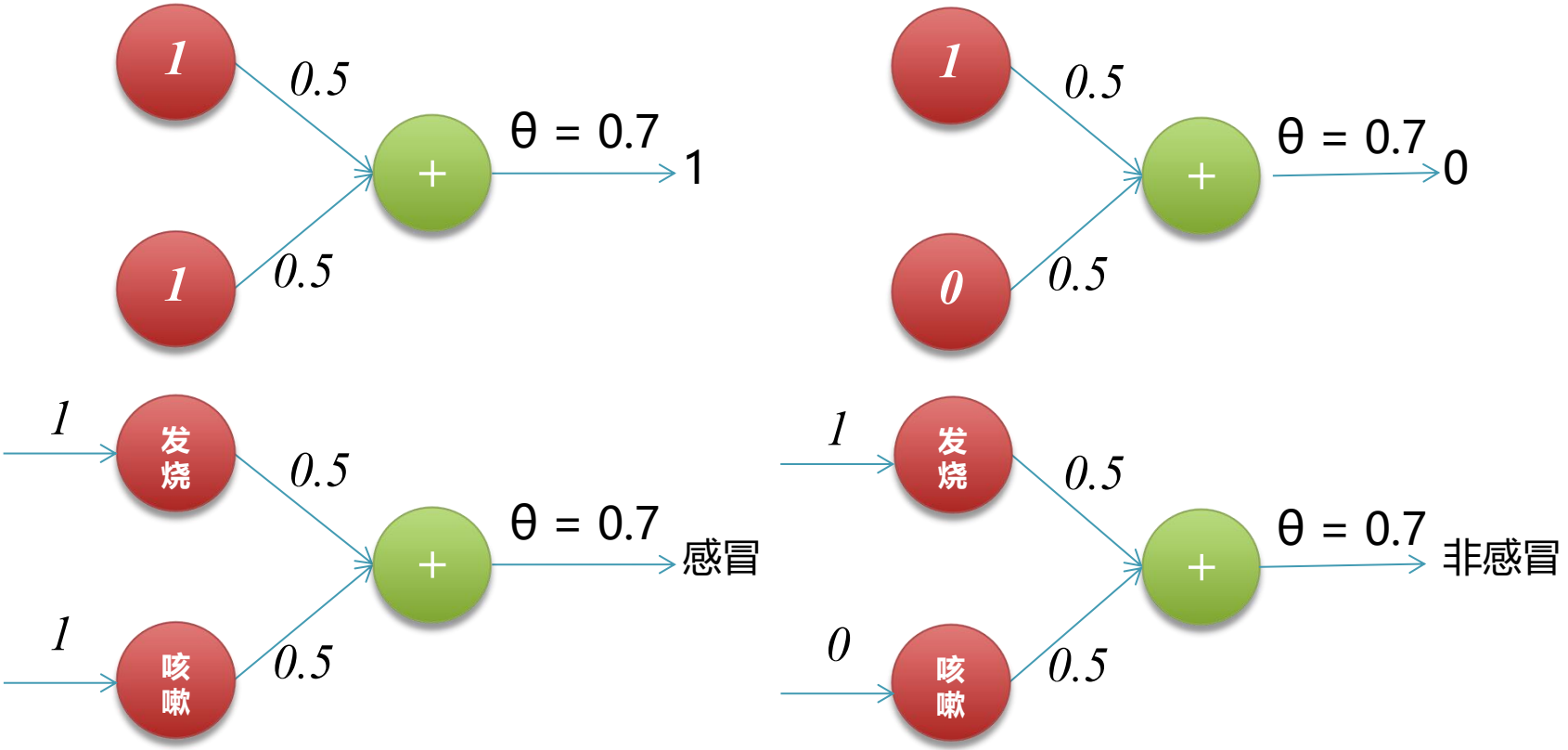
- 感知机（Perceptron）是神经网络（深度学习）的起源算法，学习感知机的构造是通向神经网络和深度学习的一种重要思想，它是1958年由康奈尔大学心理学教授弗兰克·罗森布拉特（Frank Rosenblatt）提出来的。
- 感知机接收多个输入信号，产生一个输出信号。



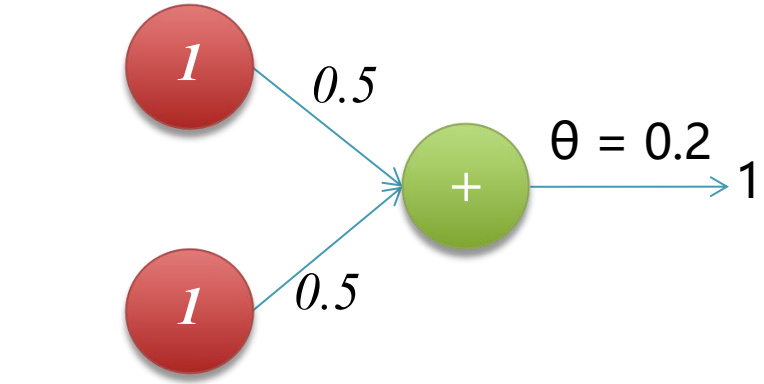
$$y = \begin{cases} 0 & (w_1x_1 + w_2x_2 <= \theta) \\ 1 & (w_1x_1 + w_2x_2 > \theta) \end{cases}$$

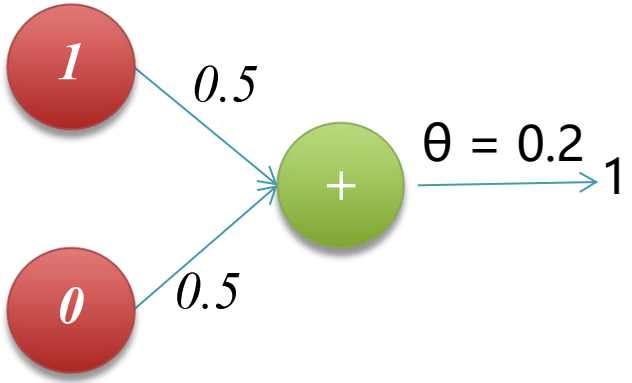
其中，x_1和x_2称为输入，w_1和w_2为权重，y为输出

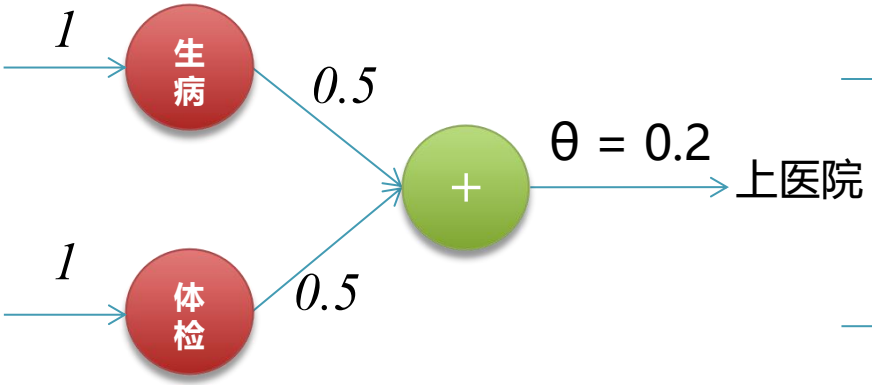
感知机实现逻辑和（AND）计算

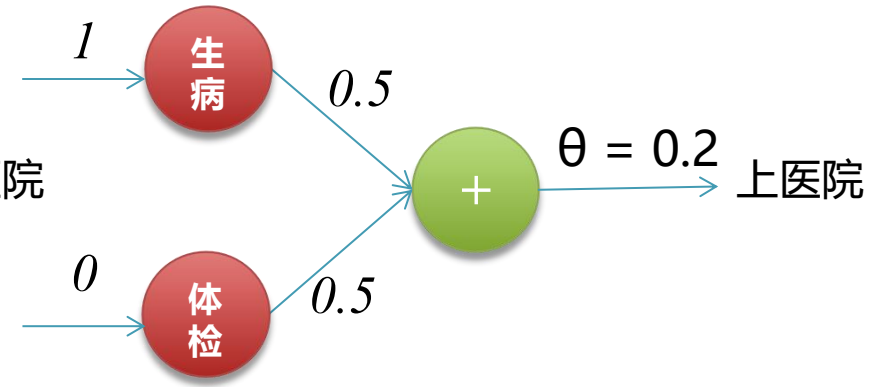


感知机实现逻辑或（OR）计算







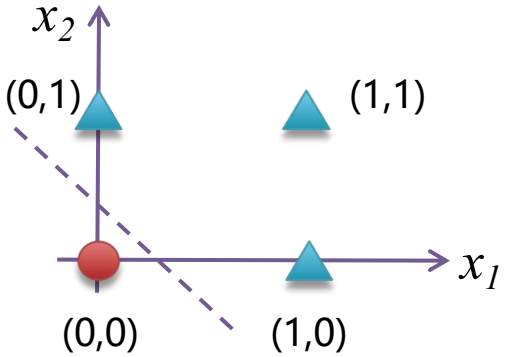


练习：编写感知机实现（待补）

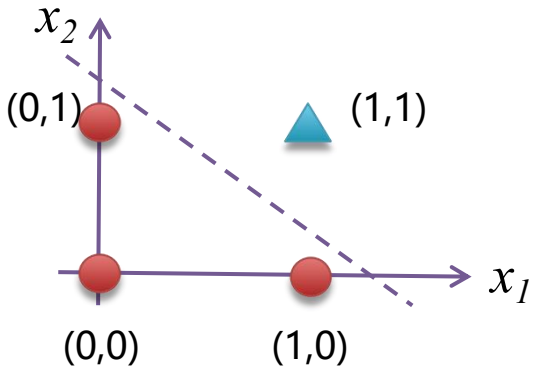
x_1

感知机的局限性

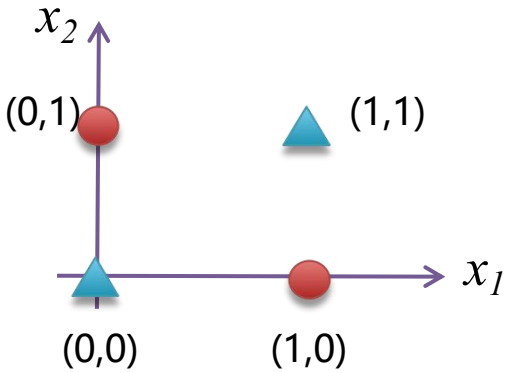
- 感知机的局限在于无法处理“异或”问题



或门



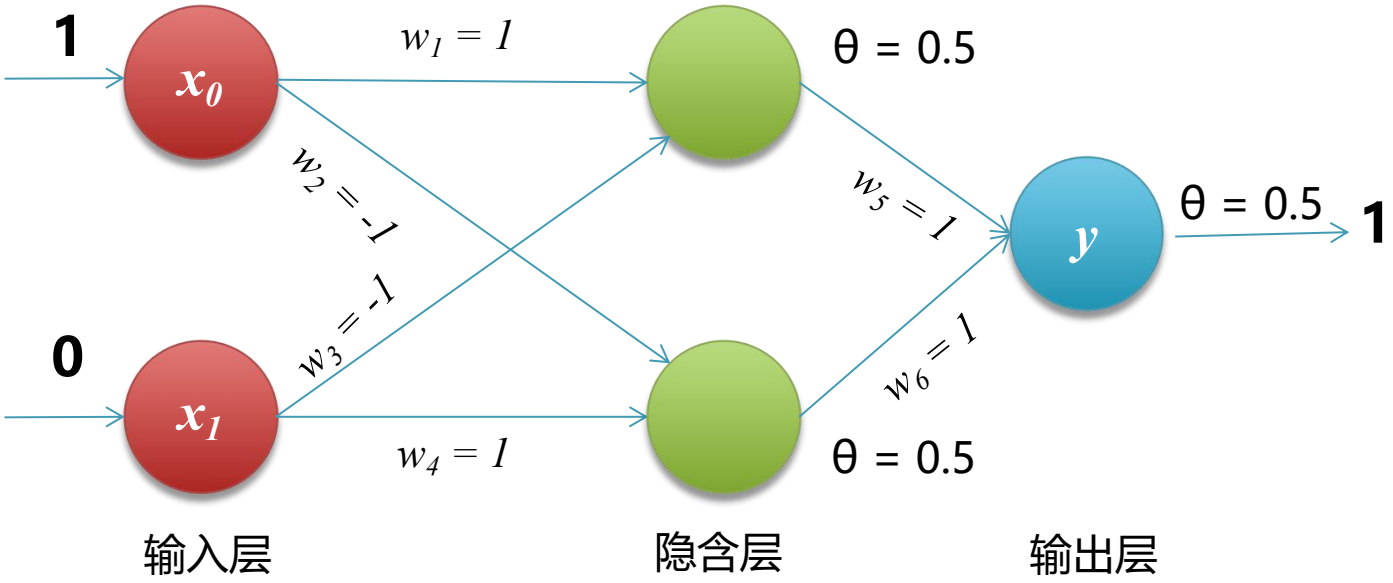
与门



异或门

感知机解决异或问题

- 1975年，感知机的“异或”难题才被理论界彻底解决，即通过多个感知机组合来解决该问题，这种模型也叫多层感知机（Multi-Layer Perceptron，MLP）。如下图所示，神经元节点阈值均设置为0.5

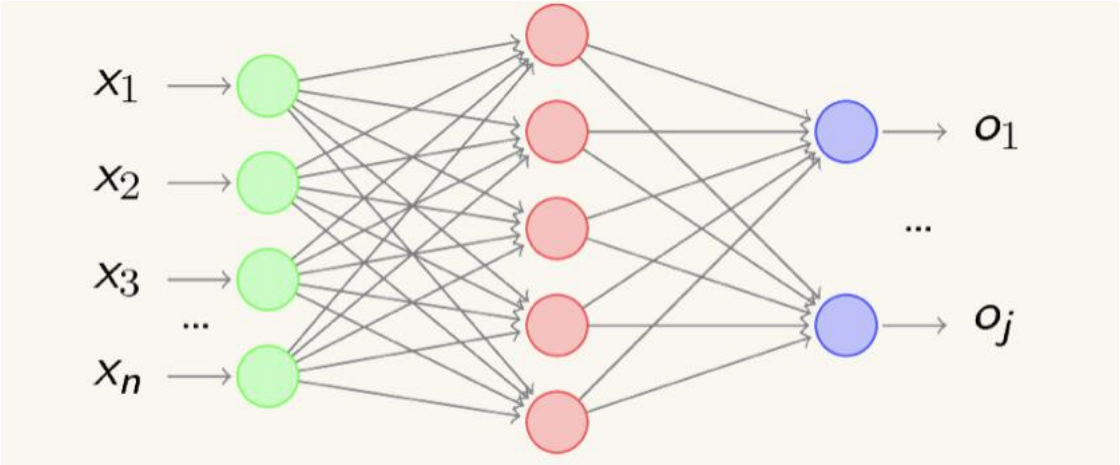


练习：编写多层感知机实现异或门（待补）

x_1

多层前馈网络

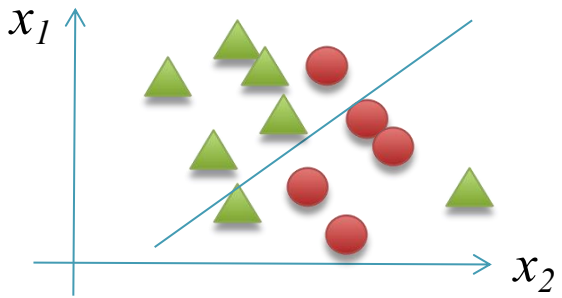
- 感知机由于结构简单，完成的功能十分有限。可以将若干个感知机连在一起，形成一个级联网络结构，这个结构称为“多层前馈神经网络”（Multi-layer Feedforward Neural Networks）。所谓“前馈”是指将前一层的输出作为后一层的输入的逻辑结构。每一层神经元仅与下一层的神经元全连接。但在同一层之内，神经元彼此不连接，而且跨层之间的神经元，彼此也不相连。

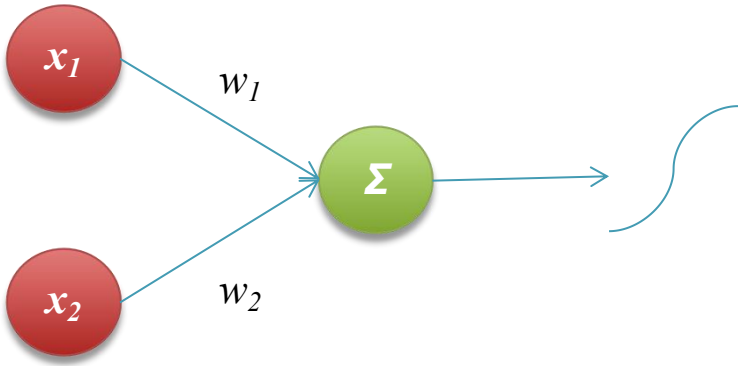


多层前馈网络强大的表达能力

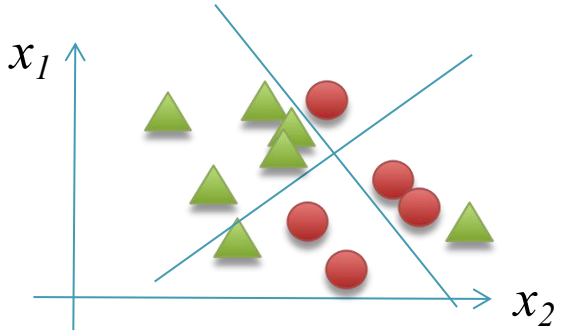
- 多层前馈网络能干什么？能力又如何？1989年，奥地利学者库尔特·霍尼克（Kurt Hornik）等人发表论文证明，**对于任意复杂度的连续波莱尔可测函数（Borel Measurable Function）f，仅仅需要一个隐含层，只要这个隐含层包括足够多的神经元，前馈神经网络使用挤压函数（Squashing Function）作为激活函数，就可以以任意精度来近似模拟f。**
- 如果想增加f的近似精度，单纯依靠增加神经元的数目即可实现。
- 这个定理也被称为**通用近似定理**（Universal Approximation Theorem），该定理表明，前馈神经网在理论上可近似解决任何问题。

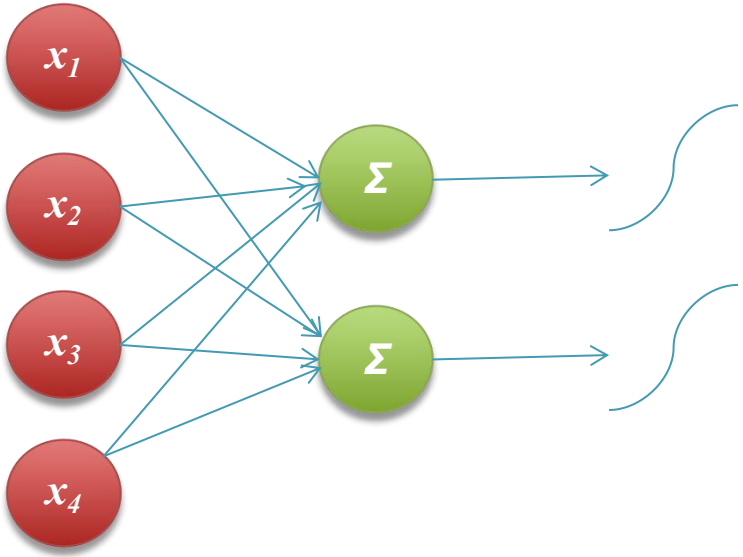
增加神经元提高分类准确率（一）



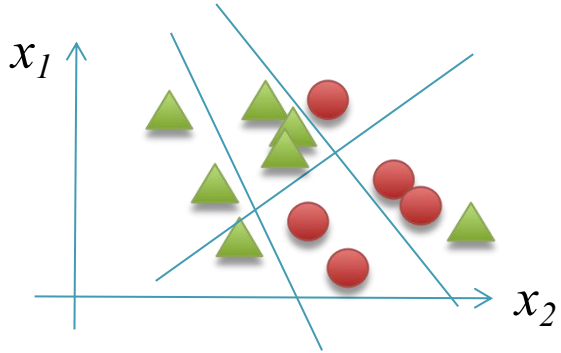


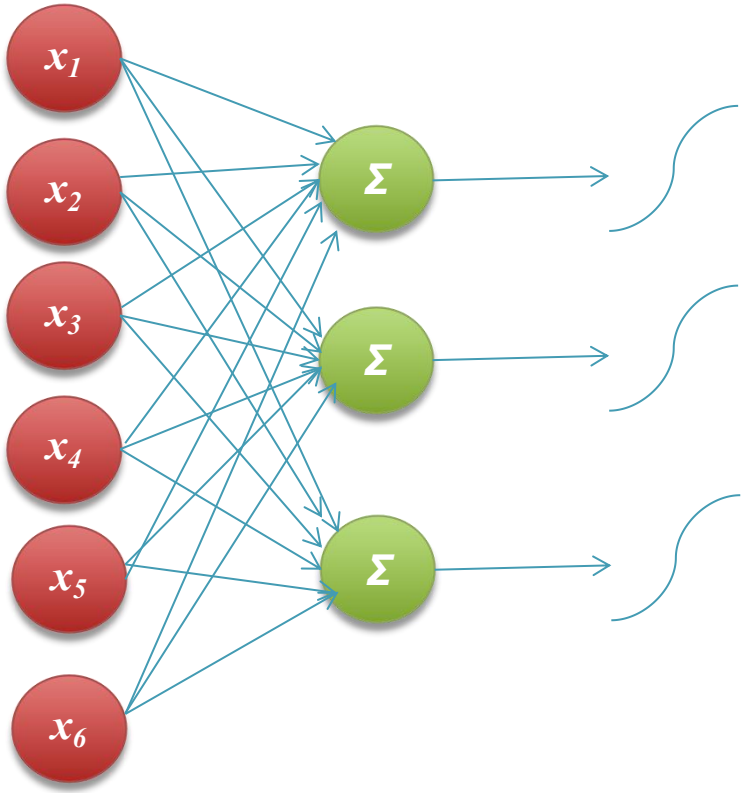
增加神经元提高分类准确率（二）





增加神经元提高分类准确率（三）





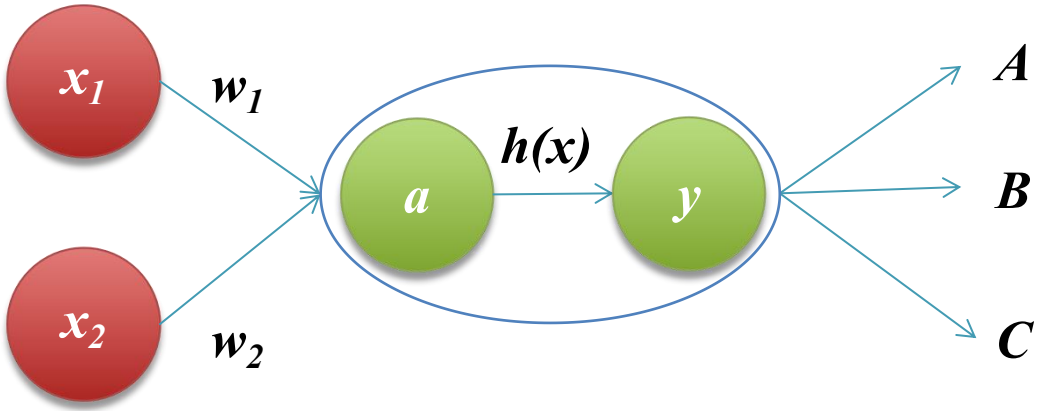
我们通过增加神经元的方式，使得分类趋于更加细腻、准确。理论上，这个过程可以一直推行下去，直到所有样例的分类都是准确的。但这样一来，隐含层就变得越来越“胖”。

“深”而“瘦”的网络

- 其实，神经网络的结构还有另外一个“进化”方向，那就是朝着“纵深”方向发展，也就是说，减少单层的神经元数量，而增加神经网络的层数，也就是“深”而“瘦”的网络模型。
- 微软研究院的科研人员就以上两类网络性能展开了实验，实验结果表明：增加网络的层数会显著提升神经网络系统的学习性能，这从某种角度也证明了深度学习朝着“纵深方向”发展的策略是正确的。

激活函数

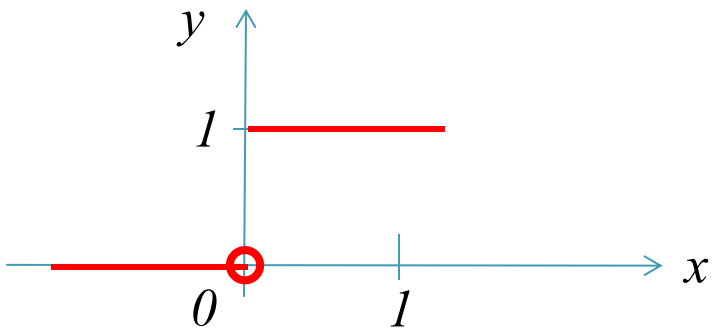
- 激活函数负责将输入信号的总和转换为输出信号



常见激活函数（一）：阶跃函数

- 阶跃函数（Step Function）是一种特殊的连续时间函数，是一个从0跳变到1的过程，函数形式与图像：

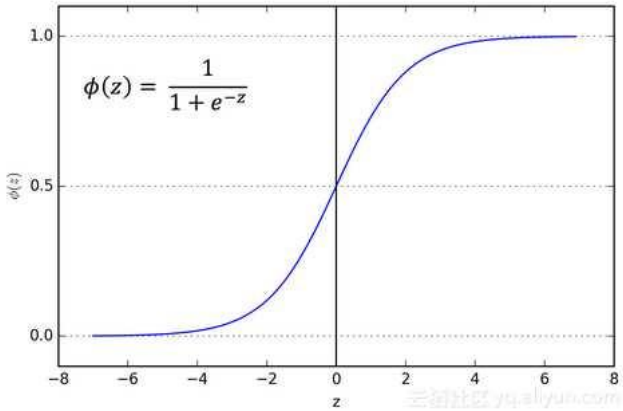
$$f(x) = \begin{cases} 1 & (x >= 0) \\ 0 & (x < 0) \end{cases}$$



常见激活函数（二）：sigmoid函数

- sigmoid函数也叫Logistic函数，用于隐层神经元输出，取值范围为(0,1)，它可以将一个实数映射到(0,1)的区间，可以用来做二分类

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



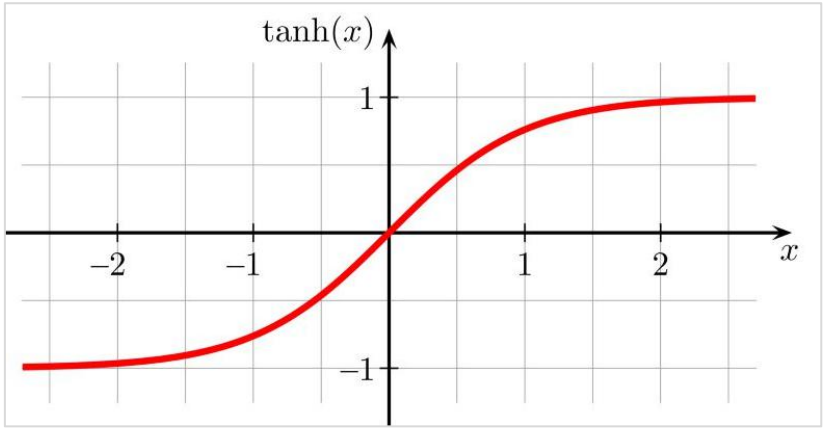
优点：平滑、易于求导

缺点：激活函数计算量大，反向传播求误差梯度时，求导涉及除法；反向传播时，很容易就会出现梯度消失的情况，从而无法完成深层网络的训练

常见激活函数（三）：tanh函数

- tanh是双曲函数中的一个，tanh()为双曲正切函数。

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$



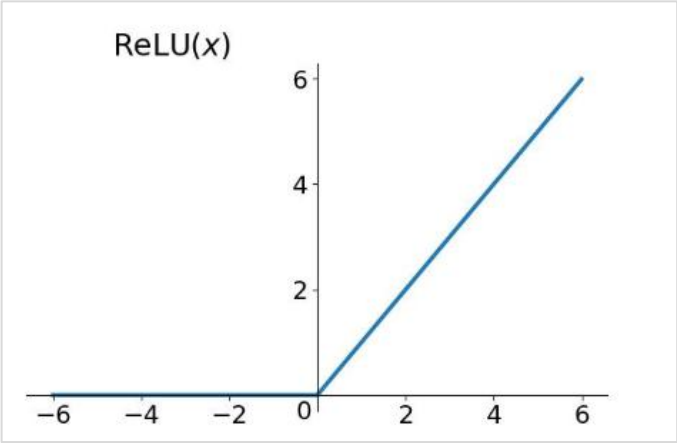
优点：平滑、易于求导；输出均值为0，收敛速度要比sigmoid快，从而可以减少迭代次数

缺点：梯度消失

常见激活函数（四）：ReLU函数

- ReLU全称为修正线性单元（Rectified Linear Units）

$$f(x) = \begin{cases} x & (x > 0) \\ 0 & (x <= 0) \end{cases}$$



优点：（1）更加有效率的梯度下降以及反向传播，避免了梯度爆炸和梯度消失问题

（2）计算过程简单

常见激活函数（五）：softmax函数

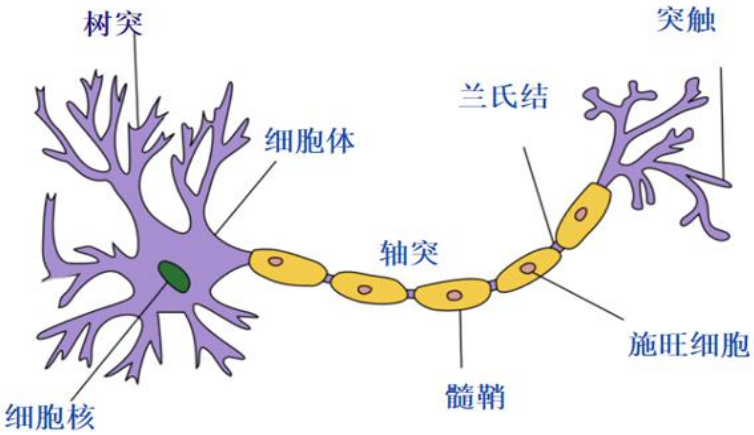
- 待补

神经网络

- 神经网络（Neural Network）的网络结构、神经元的多层连接的构造、信号的传递方法等，基本上和感知机是一样的。区别在于使用的激活函数与多层感知机不同，多层感知机使用阶跃函数作为激活函数，而神经网络使用连续函数作为激活函数。可以说，如果将激活函数从阶跃函数换成其它函数，就可以进入神经网络的世界了。

生物神经元

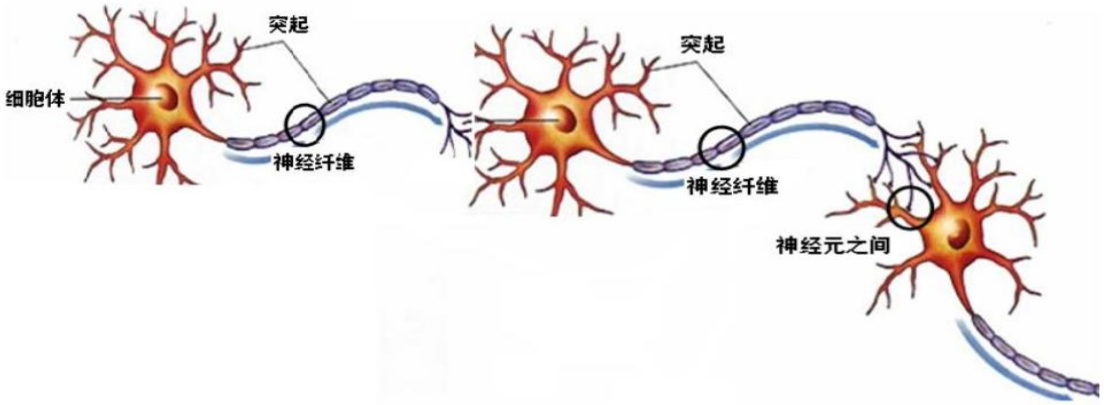
- 计算机中的神经网络是收到生物神经网络启发。神经元是神经系统最基本的结构和功能单位。分为细胞体和突起两部分。细胞体由细胞核、细胞膜、细胞质组成，具有联络和整合输入信息并传出信息的作用。



突起有树突和轴突两种。树突短而分枝多，直接由细胞体扩张突出，形成树枝状，其作用是接受其他神经元轴突传来的信息并传给细胞体。当这些信息超过一定的值（阈值）以后，这个神经元激活，然后再由轴突将刺激传递出去。

生物神经元兴奋传递

- 每个神经元都是一个信息处理单元，且具有多输入单输出特性。神经元的输入可分为兴奋性输入和抑制性输入两种类型。突触前膜借助化学信号或电信号，使下一个神经元产生兴奋效应的为兴奋性突触，使下一个神经元禅师抑制效应的为抑制性突触。因此看来，突触的主要作用是在神经元细胞传递信息。

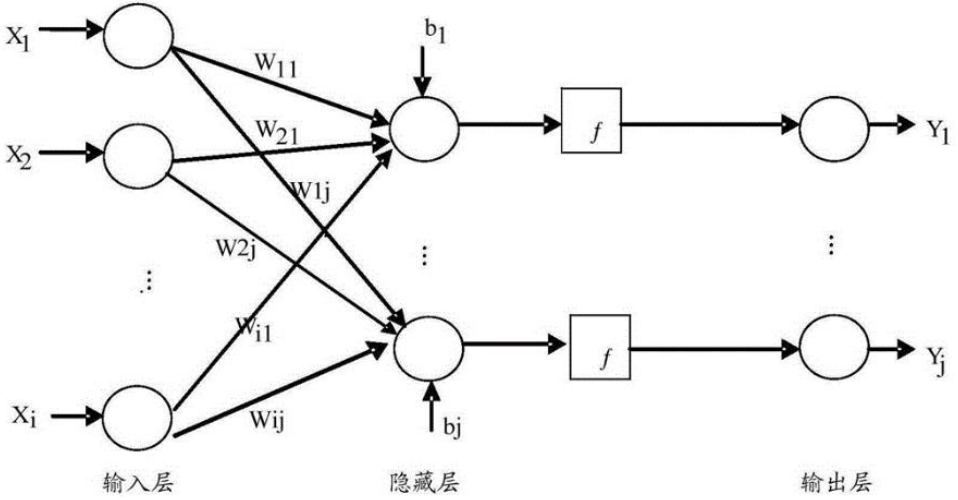


人工神经网络

- 人工神经网络（Artificial Neural Network，即ANN）从信息处理角度对人脑神经元网络进行抽象，通过将多个感知机（神经元）连接起来，针对不同的输入进行处理、传递，并在激活函数的计算下，产生一定的输出。
- 自20世纪80年代起，人工神经网络（Artificial Neural Network，ANN）开始兴起，而且在很长一段时间内都是人工智能领域的研究热点。

人工神经网络（续）

- 作为处理数据的一种新模式，人工神经网络的强大之处在于，它拥有很强的学习能力。
- 在得到一个训练集合之后，通过学习，提取到所观察事物的各个部分的特征，特征之间用不同网络节点链接，通过训练链接的网络权重，改变每一个链接的强度，直到顶层的输出得到正确的答案。



人工神经网络特点

（1）非线性。非线性关系是自然界的普遍特性之一，大脑的活动就属于一种非线性现象。人工神经元可处于抑制或激活两种状态，这种行为在数学上表现为一种非线性关系。它们可以通过具有阈值（或称偏置）的激活函数来完成该功能。具有阈值的神经元，可构成性能更佳的神经网络，可提高整个网络的容错性和存储容量。

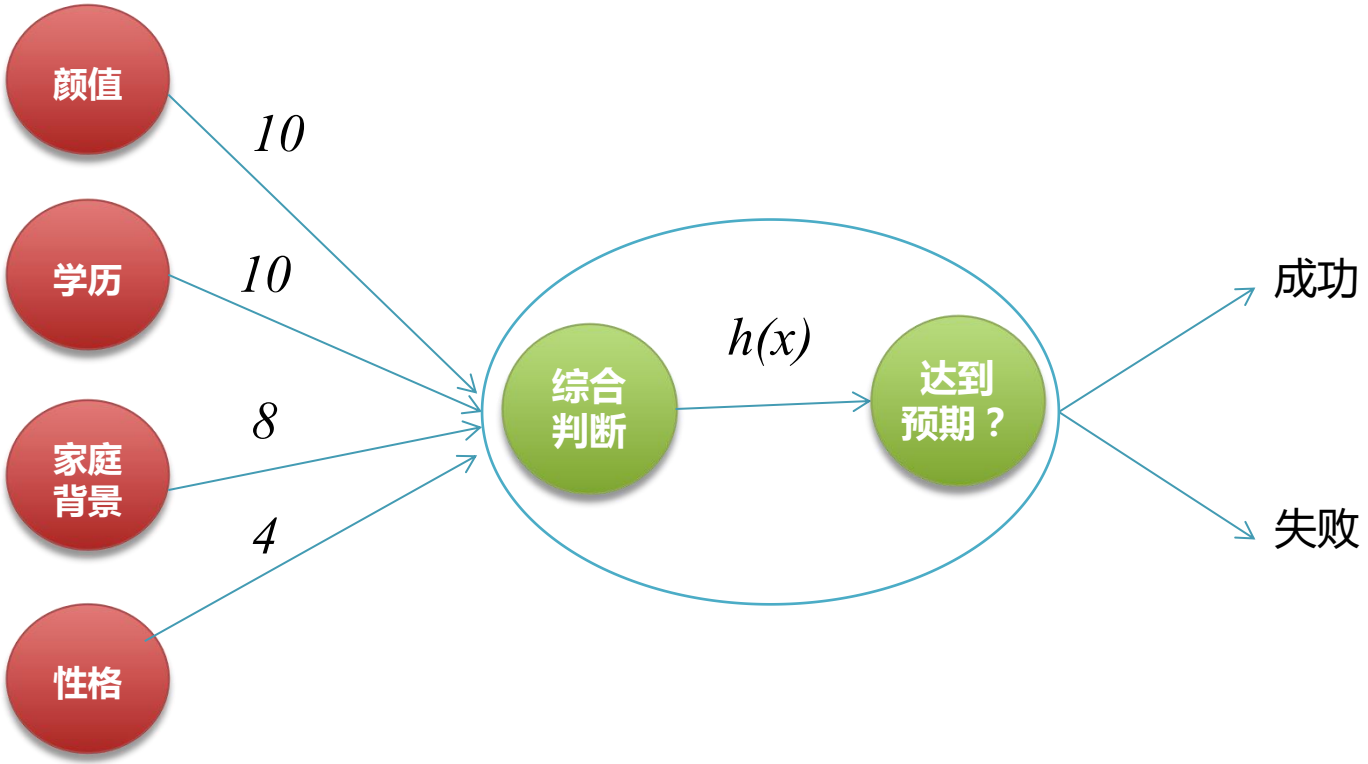
（2）非局限性。神经网络通常由多个神经元广泛连接而成，神经元之间阡陌纵横。因此，系统的整体行为，不仅取决于单个神经元的特征，而且高度依赖神经元之间的相互作用关系。任何一个神经元的“作用域”都不是局部的，而是可能通过网络链接波及全网，联想记忆就是非局限性的典型例子。

人工神经网络特点（续）

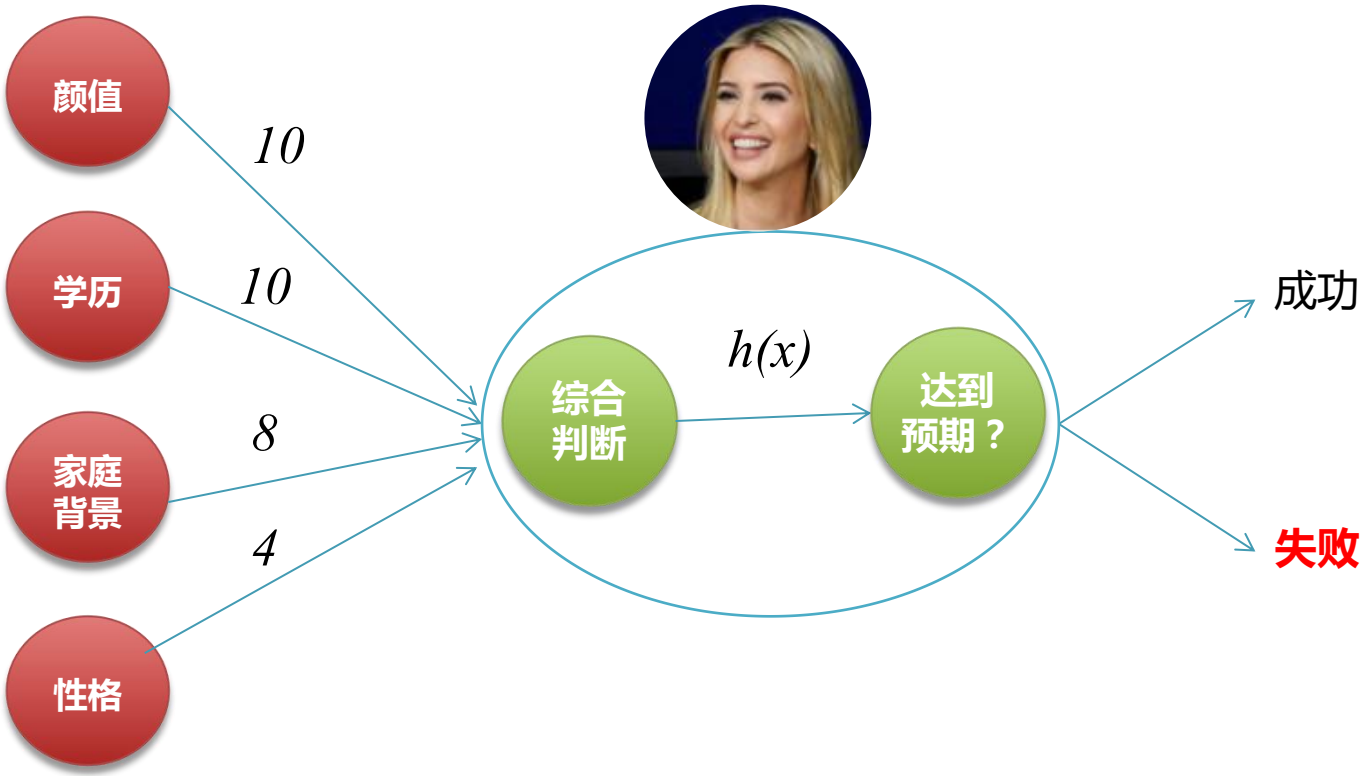
（3）非常定性。人工神经网络一直处于“更新”状态。这是因为，它具有强大的自适应、自组织、自学习能力。在神经网络中，不但处理的信息可以是变化多端的，而且在处理信息的同时，非线性动力系统本身可能也在演化（比如网络连续权值的迭代更新）。

（4）非凸性。一个系统的演化方向，在一定条件下取决于某个特定的状态函数，如目标函数和激活函数。当前的神经网络，基本都放弃了线性激活函数，通常采用诸如Sigmoid、Tanh、ReLU等非线性激活函数，这就导致神经网络的目标函数具有非凸性。所谓非凸性，是指函数可能有多个极值。极值通常对应于系统比较稳定的状态，多极值表明系统具备多个较稳定的平衡态，而多个平衡态将导致系统演化出多样性。

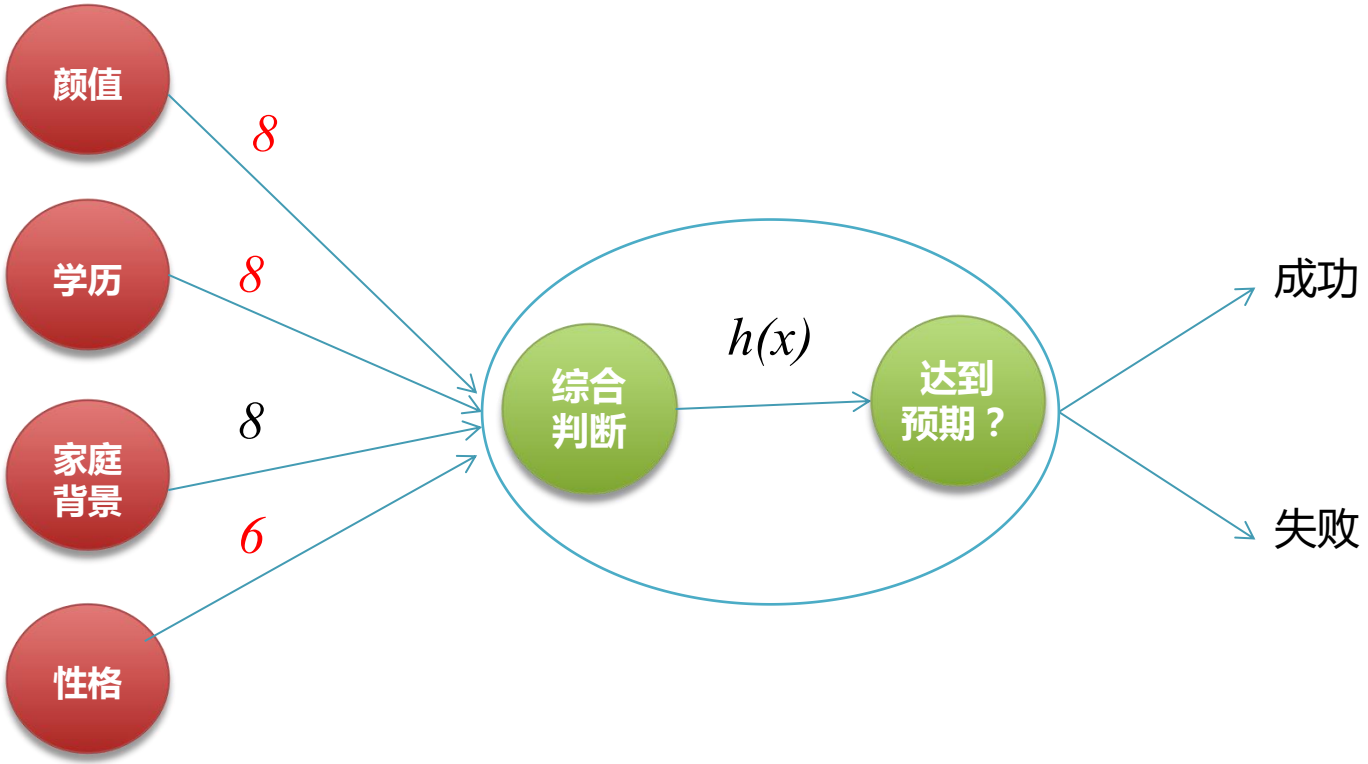
人工神经网络学习示例（一）



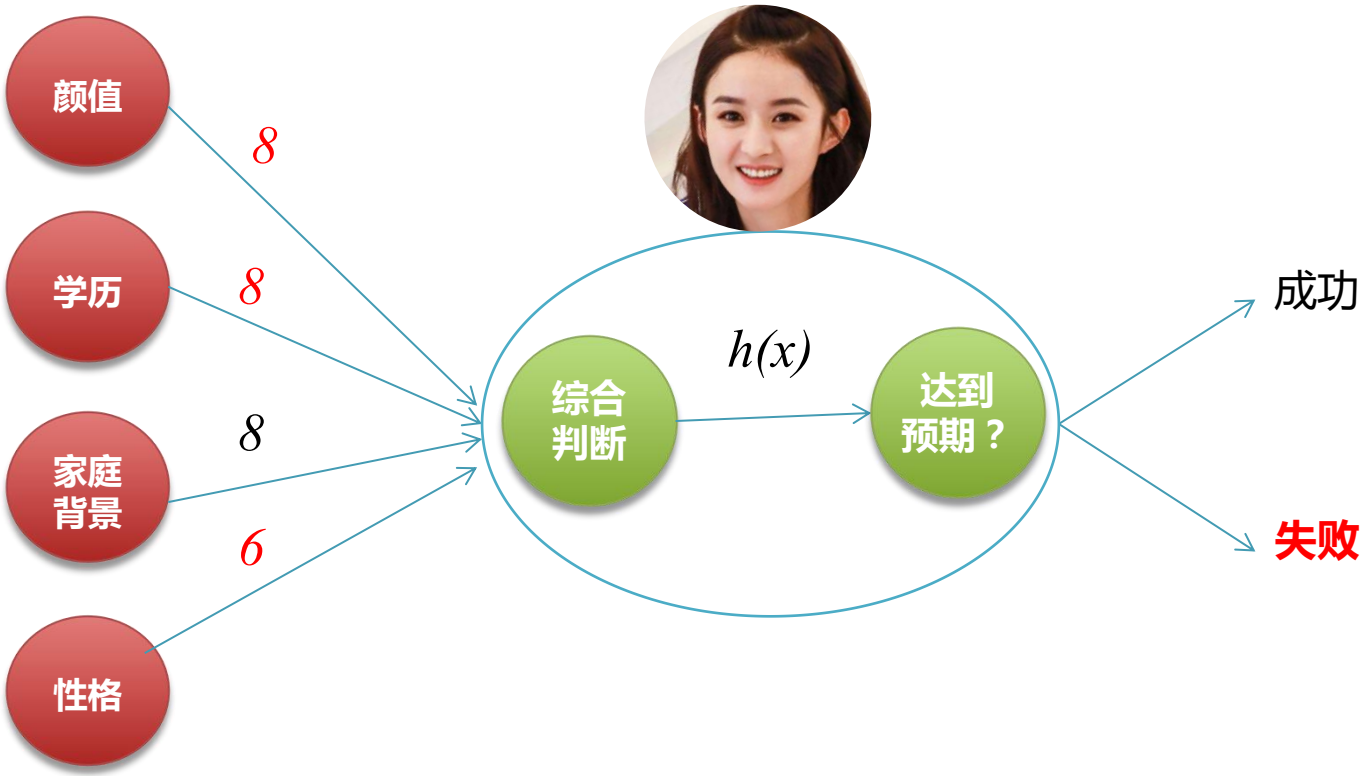
人工神经网络学习示例（二）



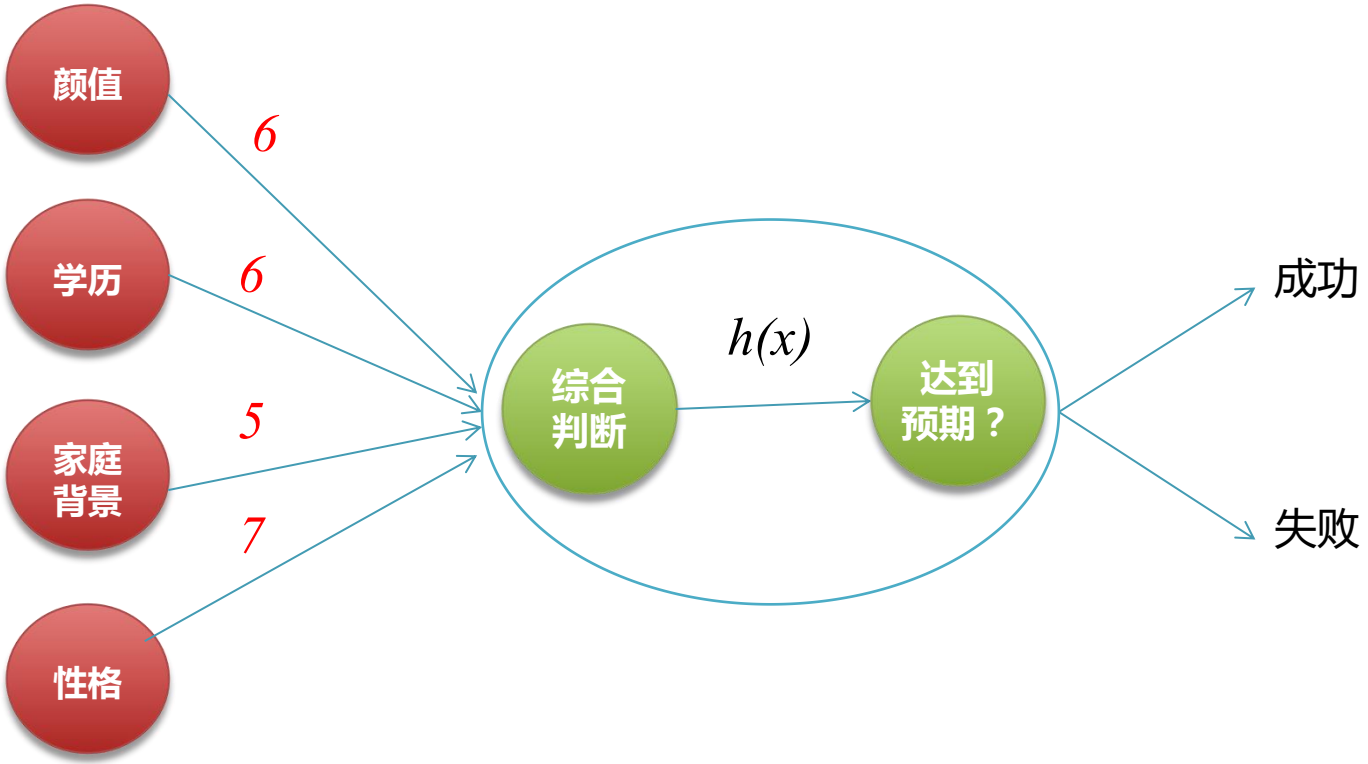
人工神经网络学习示例（三）



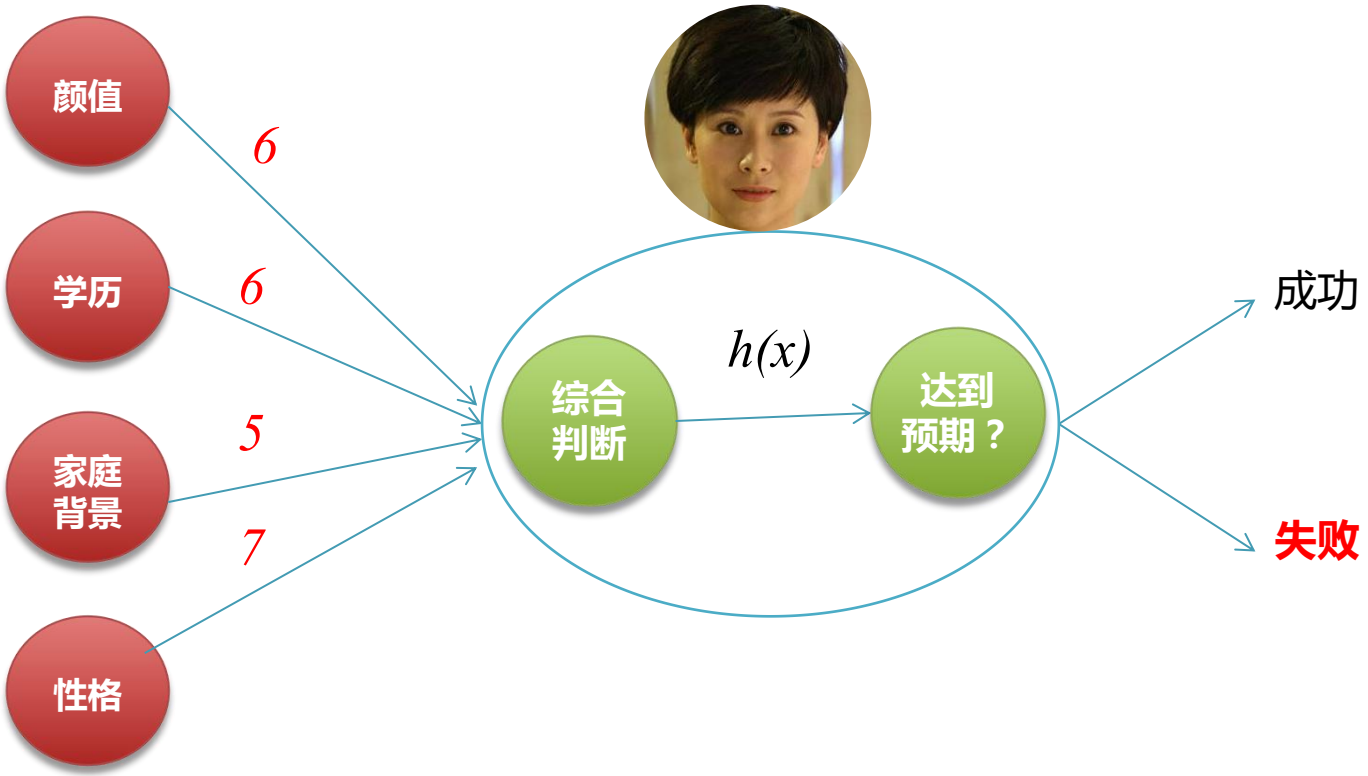
人工神经网络学习示例（四）



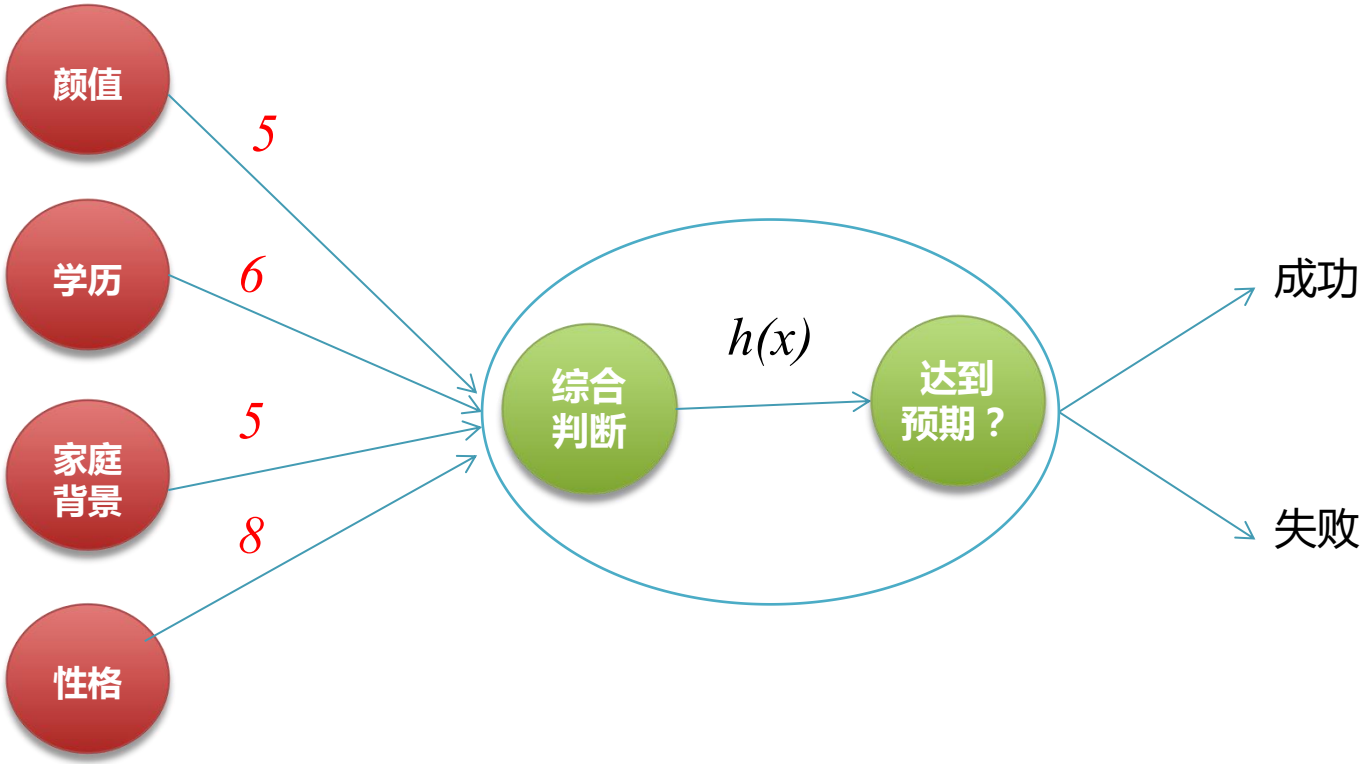
人工神经网络学习示例（五）



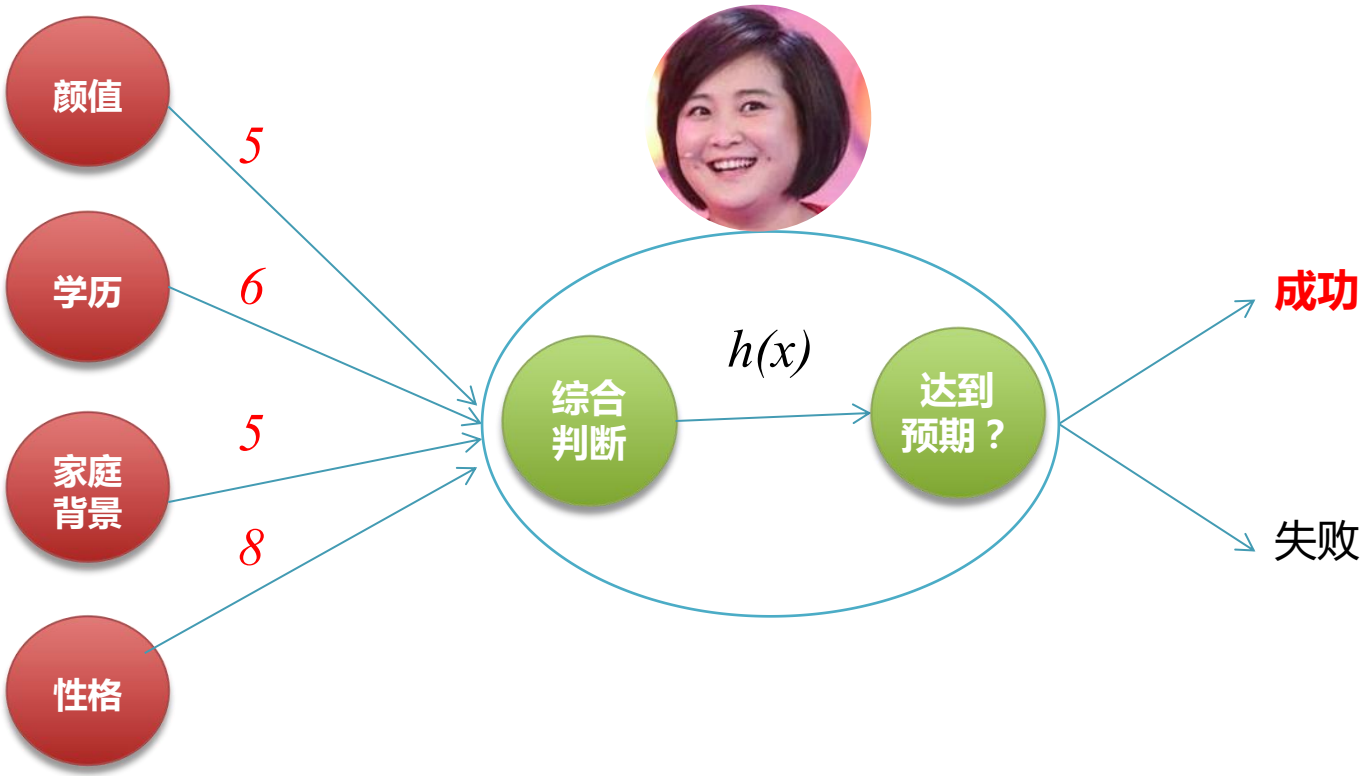
人工神经网络学习示例（六）



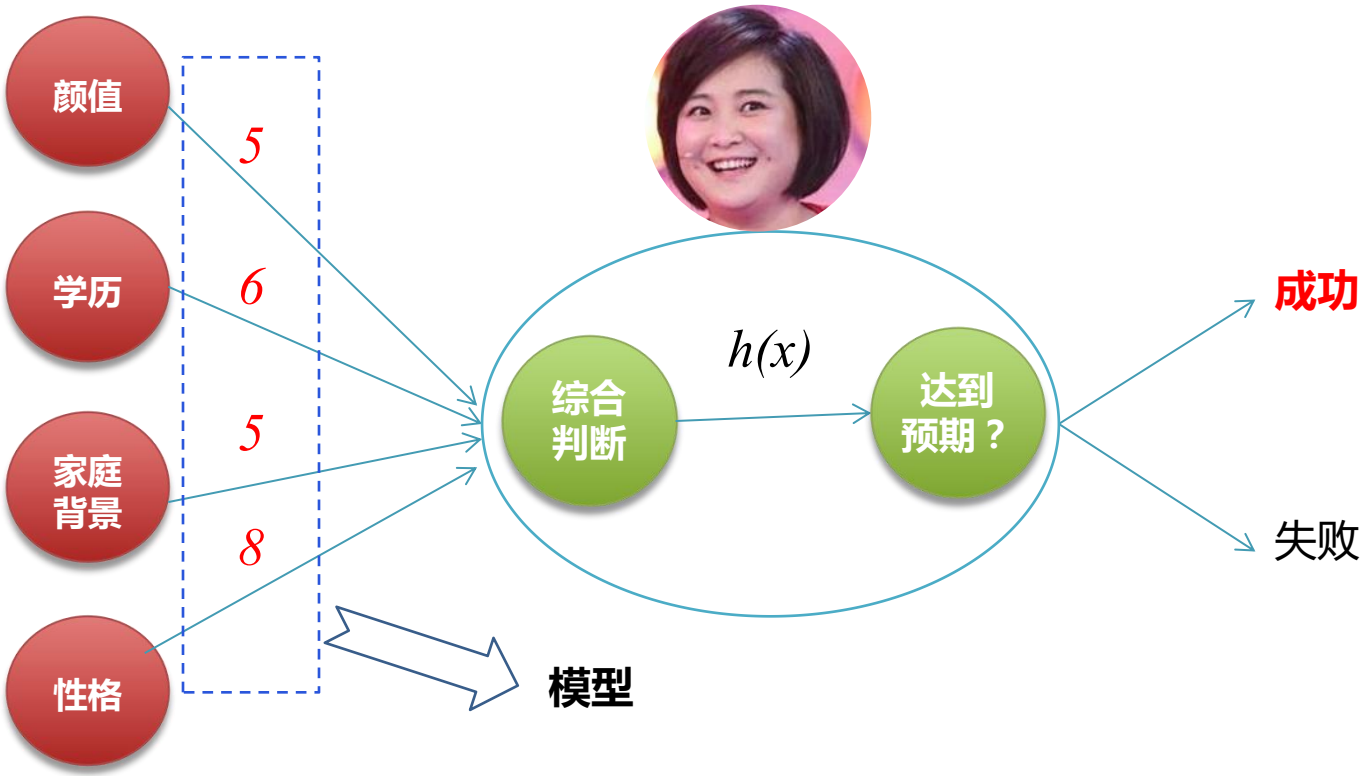
人工神经网络学习示例（七）



人工神经网络学习示例（八）

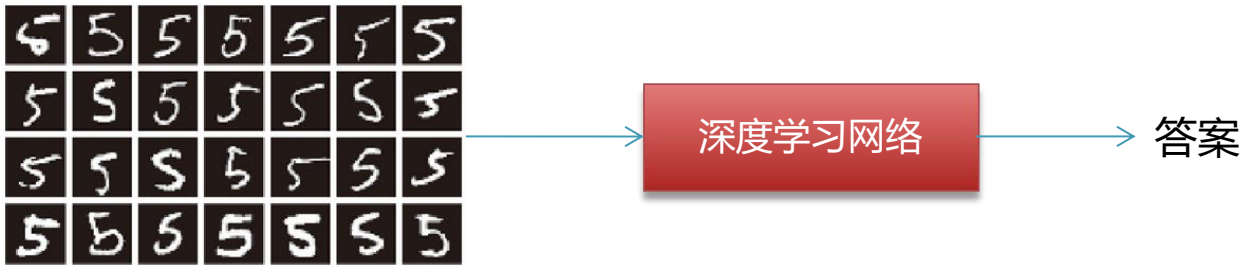


人工神经网络学习示例（九）



从数据中学习

- 神经网络的特征就是可以从数据中学习。所谓“从数据中学习”，是指可以由数据自动决定权重参数的值。在实际的神经网络中，参数的数量成千上万，在层数更深的深度学习中，参数的数量甚至可以上亿，想要人工决定这些参数的值是不可能的。
- 所以，机器必须具备从海量数据中，自己发现模式、寻找答案，数据是机器学习的核心。这种数据驱动的方法，也可以说脱离了过往以人为中心的方法。



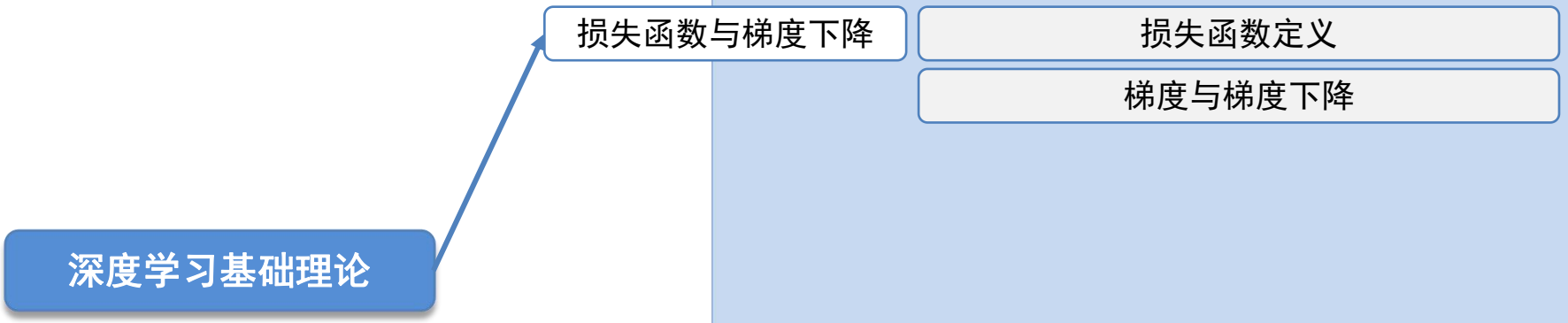
训练数据和测试数据

- 机器学习中，一般将数据分为训练数据和测试数据两部分来进行学习和实验等。首先，使用训练数据进行学习，寻找最优的参数；然后，使用测试数据评价训练得到的模型的实际能力。
- 为了正确评价模型的泛化能力，就必须划分训练数据和测试数据。另外，训练数据也可以称为监督数据。

小结

- 本章节介绍了深度学习一些重要的基本概念，需要理解并熟练掌握。同时，还介绍了如何从简单的感知机逐步演化到复杂的神经网络。重要的概念有：
  - ✓ **感知机。**接收多个输入信号，产生一个输出信号，无法解决异或问题
  - ✓ **多层感知机。**将多个感知机组合
  - ✓ **多层前馈网络。**若干个感知机组合成若干层的网络，上一层输出作为下一层输入
  - ✓ **激活函数。**将计算结果转换为输出的值，包括阶跃函数、sigmoid、tanh、ReLU
  - ✓ **人工神经网络。**将多层感知机激活函数由阶跃函数更换为其它函数

深度学习基础理论



损失函数与梯度下降

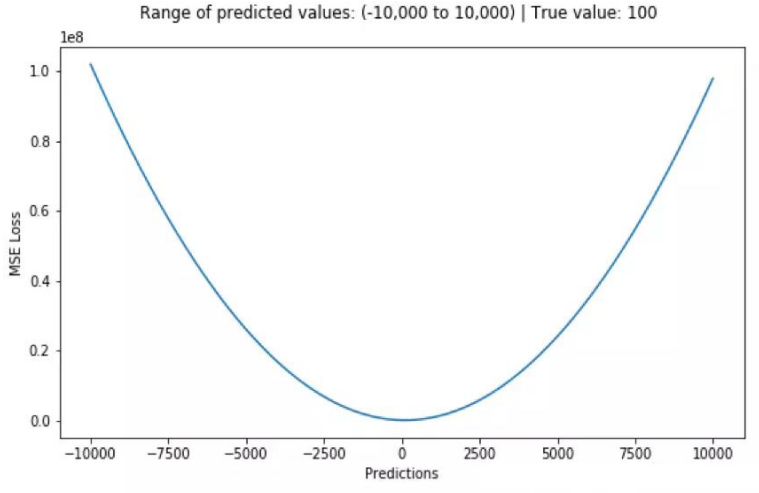
损失函数

- 在机器学习中的过程中，如何**保障参数始终朝着最优的方向调整**？这就需要用到损失函数。“有监督学习”算法里，构造一个决策函数f，对于给定的输入x，由f (x)给出相应的输出，这个实际输出值和原先预期值Y可能不一致，所以需要定义一个损失函数（Loss Function），也有称之为代价函数（Cost Function）来度量这二者之间的“差异”程度。
- **所以，损失函数是用来度量预测值和实际值之间的差异的。**
- 损失函数值越小，说明预测输出和实际结果（也称期望输出）之间的差值就越小，也就说明我们构建的模型越好。学习的过程，就是不断通过训练数据进行预测，不断调整预测输出与实际输出差异，使的损失值最小的过程。

常见损失函数（一）：均方误差

- 均方误差（Mean square error）损失函数。均方误差是**回归问题常用的误差函数**，它是预测值与目标值之间差值的平方和，其公式和图像如下所示：

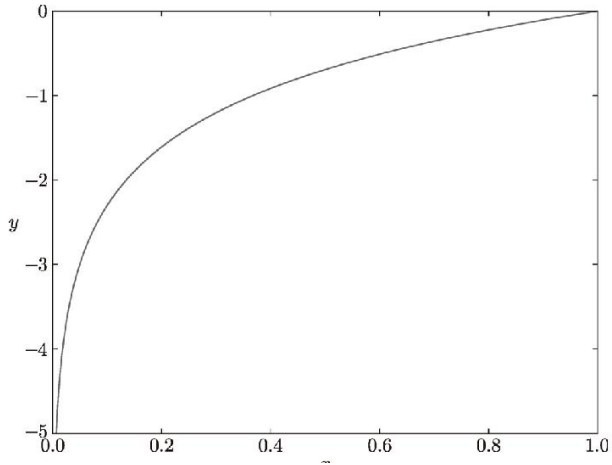
$$MSE = \frac{\sum_{i=1}^{n} (y_i - y_i^p)^2}{n}$$



常见损失函数（二）：交叉熵

- 交叉熵（Cross Entropy）。交叉熵是Shannon信息论中一个重要概念，主要用于**度量两个概率分布间的差异性信息**，在机器学习中用来作为分类问题的损失函数。假设有两个概率分布，t_k与y_k,其交叉熵函数公式及图形如下所示：

$$E = -\sum_k t_k \log y_k$$



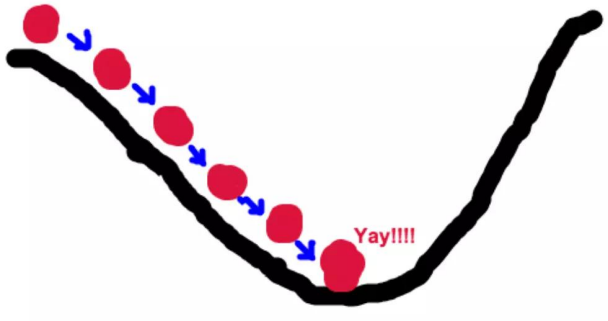
寻找损失函数最小值

- **通过损失函数，我们将“寻找最优参数”问题，转换为了“寻找损失函数最小值”问题。**寻找步骤：

（1）损失是否足够小？如果不是，计算损失函数的梯度。

（2）按梯度的反方向走一小步，以缩小损失。

（3）循环到（1）。



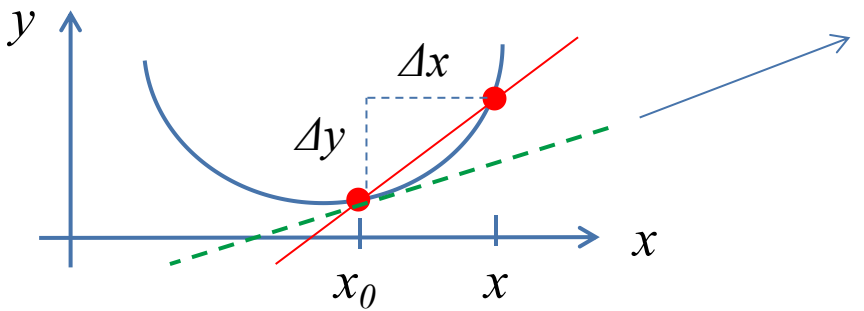
这种按照负梯度不停地调整函数权值的过程就叫作“***梯度下降法***”。通过这样的方法，改变每个神经元与其他神经元的连接权重及自身的偏置，让损失函数的值下降得更快，进而将值收敛到损失函数的某个极小值。

函数的导数

- 所谓导数，就是用来分析函数“变化率”的一种度量。其公式为：

$$f'(x_0)=\lim_{\Delta x\to 0}\frac{\Delta y}{\Delta x}=\lim_{\Delta x\to 0}\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x}$$

图像可表示为：



曲线切线即为该点梯度方向

偏导数

- “偏导”的英文本意是“partial derivatives“（表示局部导数）。对于多维变量函数而言，当求某个变量的导数时，就是把其他变量视为常量，然后对整个函数求其导数（相比于全部变量，这里只求一个变量，即为“局部”）。例如有函数：

$$f = x^2 + 3xy + y^2 + z^3$$

则，对x, y, z分别求偏导公式为：

$$\frac{\partial f}{\partial x} = 2x + 3y$$

↓

y, z为常量

$$\frac{\partial f}{\partial y} = 3x + 2y$$

↓

x, z为常量

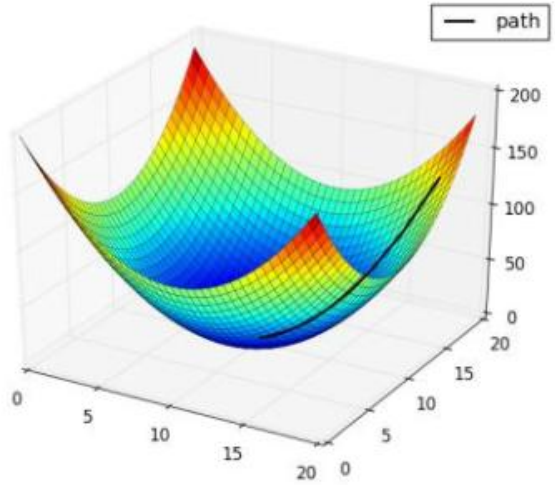
$$\frac{\partial f}{\partial z} = 3z^2$$

↓

x, y为常量

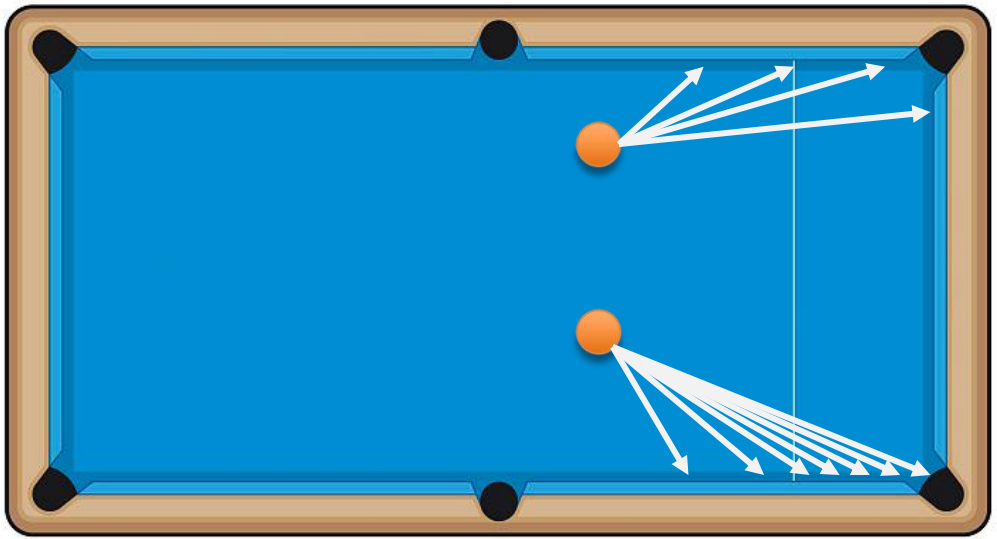
什么是梯度

- 梯度（gradient）是一个向量（矢量，有方向），表示**某一函数在该点处的方向导数沿着该方向取得最大值，即函数在该点处沿着该方向（此梯度的方向）变化最快，变化率最大。**损失函数沿梯度相反方向收敛最快（即能最快找到极值点）。当梯度向量为零（或接近于零），说明到达一个极值点，这也是梯度下降算法迭代计算的终止条件。



学习率

- 如果在梯度下降过程中，每次都按照相同的步幅收敛，则可能错过极值点（如下图），所以每次在之前的步幅减小一定比率，这个比率称之为“学习率”。



梯度递减训练法则

- 神经网络中的权值参数是非常多的，因此针对损失函数E的权值向量的梯度如以下公式所示：

$$\nabla E(\vec{w}) \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_n} \right]$$

$\nabla E(\vec{w})$ 表示损失函数E的梯度，它本身也是一个向量，它的多个维度分别由损失函数E对多个权值参数w_i求偏导所得。当梯度被解释为权值空间中的一个向量时，它就确定了E陡峭上升的方向，那么梯度递减的训练法则就如下公式所示：

$$w_i \leftarrow w_i + \Delta w_i \qquad \Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

梯度下降算法（一）：批量梯度下降

- 批量梯度下降法（Batch Gradient Descent，BGD）是最原始的形式，它是指在每一次迭代时使用所有样本来进行梯度的更新。
- 优点：
  - ✓ 一次迭代是对所有样本进行计算，此时利用矩阵进行操作，实现了并行。
  - ✓ 由全数据集确定的方向能够更好地代表样本总体，从而更准确地朝向极值所在的方向。当目标函数为凸函数时，BGD一定能够得到全局最优。
- 缺点：
  - ✓ 当样本数目 m 很大时，每迭代一步都需要对所有样本计算，训练过程会很慢。

梯度下降算法（二）：随机梯度下降

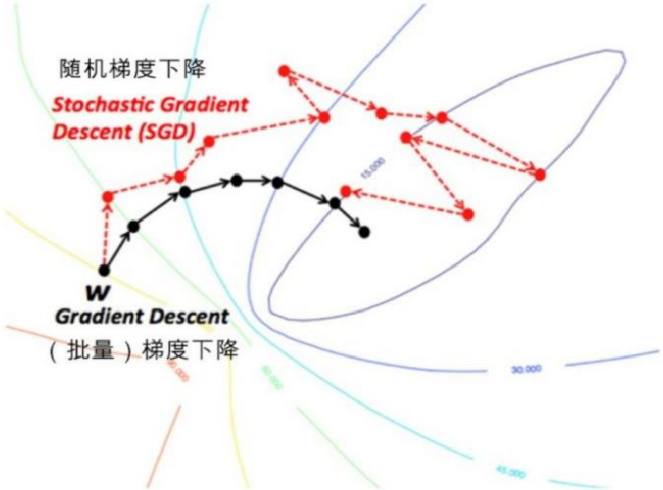
- 随机梯度下降法（Stochastic Gradient Descent，SGD）每次迭代使用一个样本来对参数进行更新，使得训练速度加快。
- 优点：
  - ✓ 由于不是在全部训练数据上的损失函数，而是在每轮迭代中，随机优化某一条训练数据上的损失函数，这样每一轮参数的更新速度大大加快。
- 缺点：
  - ✓ 准确度下降。由于即使在目标函数为强凸函数的情况下，SGD仍旧无法做到线性收敛。
  - ✓ 可能会收敛到局部最优，由于单个样本并不能代表全体样本的趋势。
  - ✓ 不易于并行实现。

梯度下降算法（三）：小批量梯度下降

- 小批量梯度下降（Mini-Batch Gradient Descent, MBGD）是对批量梯度下降以及随机梯度下降的一个折中办法。其思想是：每次迭代 使用指定个（batch_size）样本来对参数进行更新。
- 优点：
  - ✓ 通过矩阵运算，每次在一个batch上优化神经网络参数并不会比单个数据慢太多。
  - ✓ 每次使用一个batch可以大大减小收敛所需要的迭代次数，同时可以使收敛到的结果更加接近梯度下降的效果。
- 缺点：
  - ✓ batch_size的不当选择可能会带来一些问题。

几种梯度下降算法收敛比较

- 批量梯度下降稳健地向着最低点前进的
- 随机梯度下降震荡明显，但总体上向最低点逼近
- 小批量梯度下降位于两者之间



小结

- 本章节介绍了损失函数与梯度下降概念与算法
  - ✓ **损失函数。**用于度量预测值和期望值之间的差异，根据该差异值进行参数调整
  - ✓ **梯度下降。**用于以最快的速度、最少的步骤快速找到损失函数的极小值

今日总结

- 深度学习概念、与机器学习区别、发展历史
- 感知机、神经网络
- 损失函数与梯度下降