

The background features two wireframe hands, one in the upper right and one in the lower left, reaching towards the center. The background is a solid blue color with faint, concentric circular patterns and small star-like sparkles.

第7讲 神经网络学习

周文晖

杭州电子科技大学



- ✓ 学习的本质
学什么？ 怎么学？ ...
- ✓ 从生物神经元到人工神经元
生物神经元、人工神经元， ...
- ✓ 神经网络发展历史
发展历史 ...
- ✓ 神经网络基本概念
激活函数、学习规则、优化算法、 ...
- ✓ 经典人工神经网络
前向神经网络、BP神经网络， Hopfield神经网络， ...



学习的本质

学什么？怎么学？ ...



从生物神经元到人工神经元

生物神经元、人工神经元， ...



神经网络发展历史

发展历史 ...



神经网络基本概念

激活函数、学习规则、优化算法、 ...



经典人工神经网络

前向神经网络、BP神经网络、Hopfield神经网络， ...

写在神经网络学习之前的事

学习什么？

寻找一个映射函数

$f(\text{Image of a cat}) = \text{"Cat"}$

图像识别

语音识别

$f(\text{Audio waveform}) = \text{"How are you"}$

人机对战

$f(\text{Go board state}) = \text{"5-5" (next move)}$

人机对话

$f(\text{"Hi"}) = \text{"Hello"}$
(what the user said) (system response)

学习什么？

图像识别: $f(\text{) = \text{"cat"}$

一组函数

模型

$f_1, f_2, \dots$

$f_1(\text{) = \text{"cat"}$

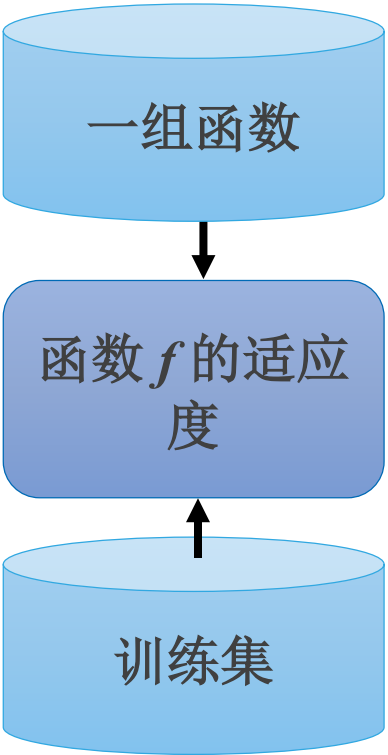
$f_2(\text{) = \text{"monkey"}$

$f_1(\text{) = \text{"dog"}$

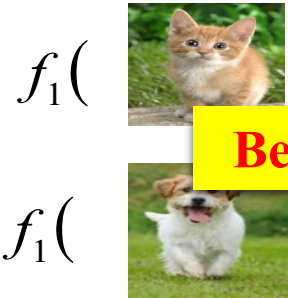
$f_2(\text{) = \text{"snake"}$

学习什么？

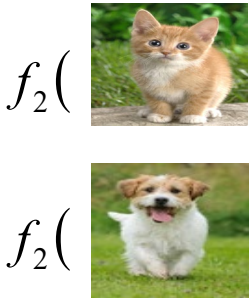
图像识别: $f(\text{img_cat}) = \text{"cat"}$



模型
 $f_1, f_2, \dots$



$f_1(\text{img_cat}) = \text{"cat"}$
 $f_1(\text{img_dog}) = \text{"dog"}$



$f_2(\text{img_cat}) = \text{"monkey"}$
 $f_2(\text{img_dog}) = \text{"snake"}$

有监督学习

函数输入:

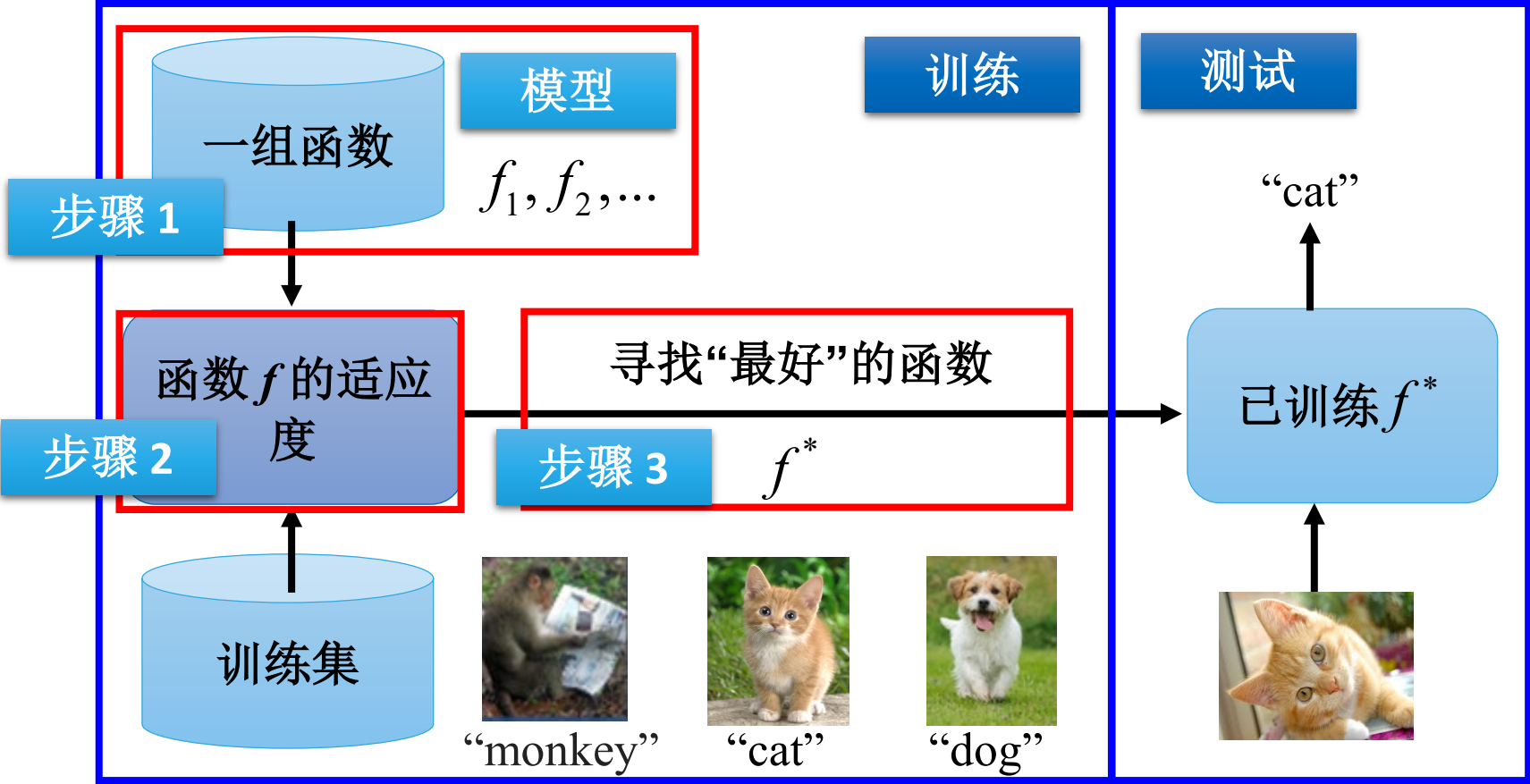


函数输出:

"monkey""cat""dog"

学习什么？

图像识别: $f(\text{img}) = \text{"cat"}$





学习的本质

学什么？怎么学？ ...



从生物神经元到人工神经元

生物神经元、人工神经元， ...



神经网络发展历史

发展历史 ...



神经网络基本概念

激活函数、学习规则、优化算法、 ...



经典人工神经网络

前向神经网络、BP神经网络、Hopfield神经网络， ...

人类智能——研究的发展史

探索人类智能有“宏观”与“微观”两个途径。

如同物理学研究物质宏观和微观结构一样。

脑功能（智能）研究的微观方法针对生物的神经网络研究。

是一种“自下而上”的研究方法：

- 首先关注单个神经元如何工作
- 再分析多神经元构成的神经系统的工作机理
- 理解智能与意识是如何从中产生。

脑是由什么组成的？

两位脑科学先驱：

- 意大利解剖学家、病理学家高尔基（C. Golgi）
- 西班牙解剖学家卡加尔（R. Cajál）

1873 年高尔基发明了一种染色技术，卡加尔将其应用于观察大脑中的神经元，提出了“神经元学说”。

神经元既是神经系统的结构单元，也是其功能、发育、营养和病变单元。

他们的研究有力地推动了神经科学的发展， 1906年两位学者同时获得诺贝尔生理学或医学奖。

詹姆斯的神经网络（1890年）

从多细胞低级动物到人类都有神经系统。

由神经元组成的神经系统，其功能是调节生物体内各器官的活动和适应外部环境。

随着动物的进化，从遍及躯体的简单神经网络，发展为向头部集中的复杂系统。

联想的基本原则：

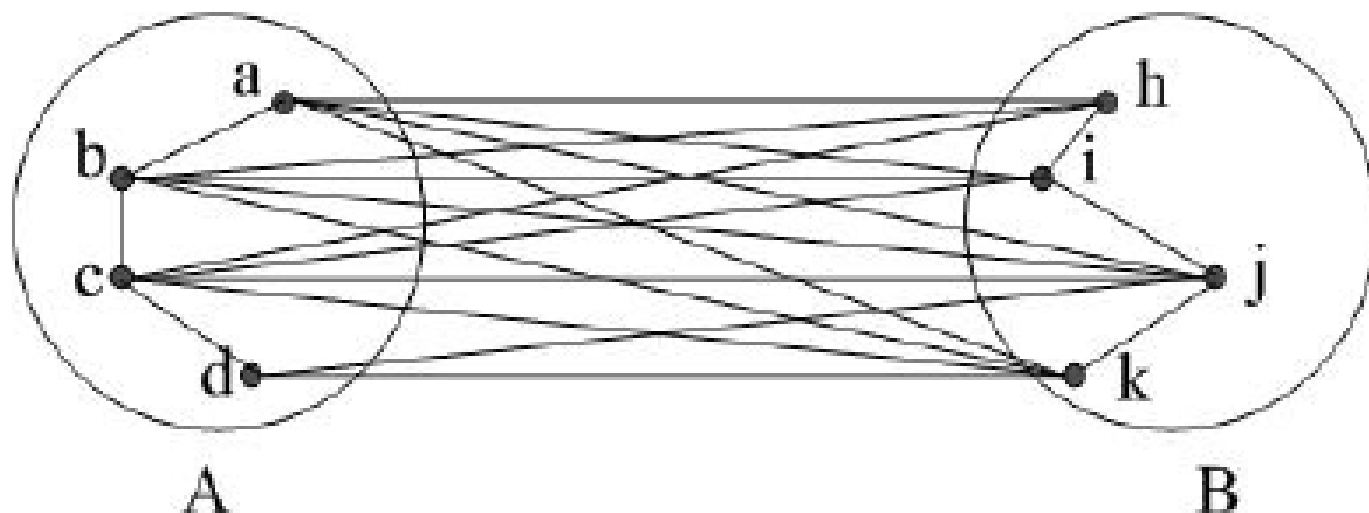
- 当两神经元作用在一起，或相互作用时，其中之一就会把它的刺激反复传向另一个内部。

詹姆斯的神经网络——联想

人类的计算能力远不如计算机。但动物和人类在其智能性远优于任何机器。

这主要由于人类感知（视觉、听觉、嗅觉、味觉、触觉等）与模式识别、联想、运动与机体控制的高度灵活性等。

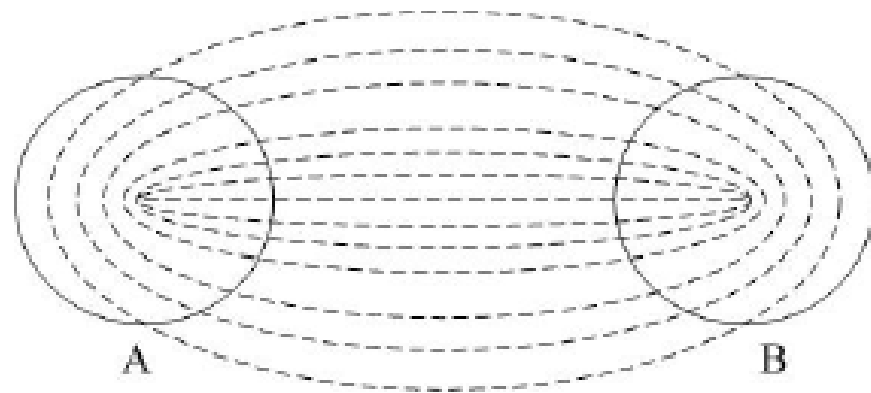
詹姆斯对联想的解释：
由事件 A 联想起事件 B 的过程，是大脑皮层活动区 A → 去激活皮层活动区 B 的过程。
图中：a, b, c, d 与 h, i, j, k 为神经束。



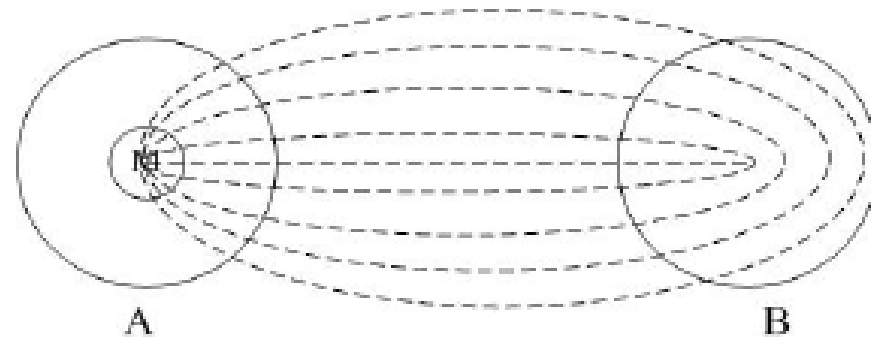
詹姆斯的神经网络——回忆

詹姆斯对回忆的解释

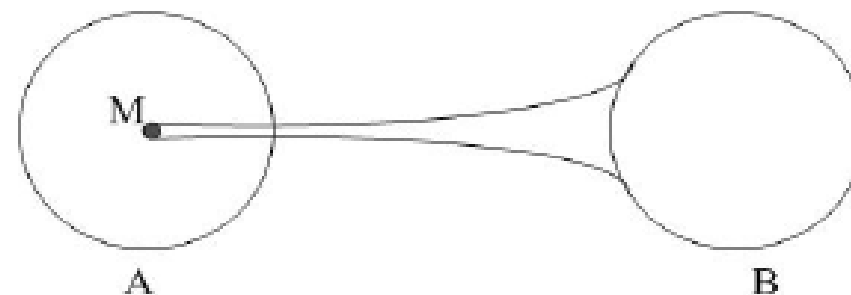
完全回忆：又叫全面联想、完善重整。由 A 区全体联想起 B 区全体。



部分回忆：又叫习惯性联想、混合联想。由 A 区部分 M 联想起 B 区全体。

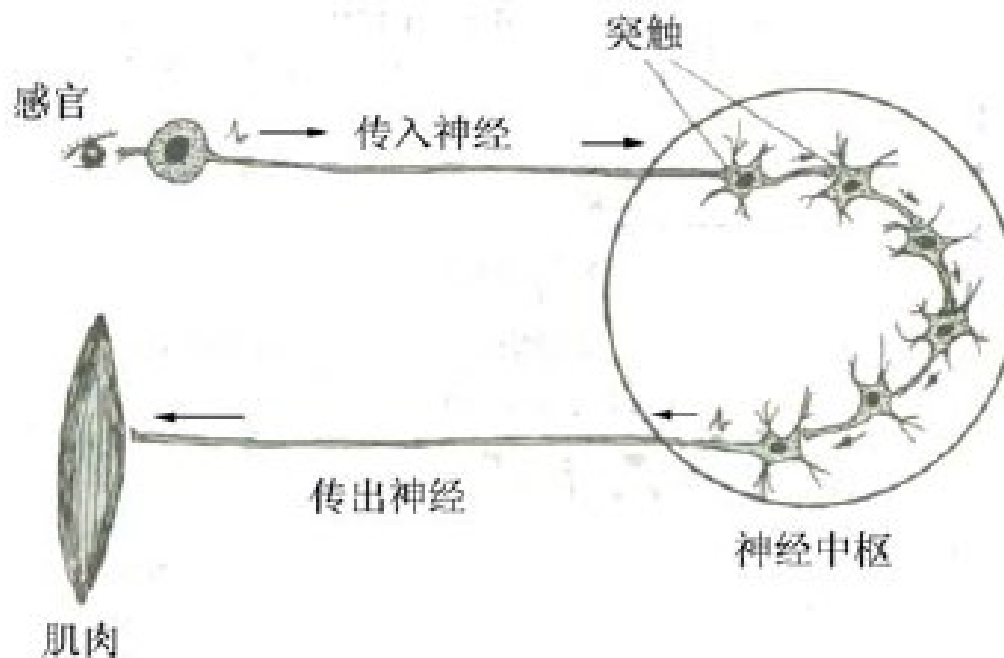


小域回忆：又叫相似性联想。由 A 区的小域 M 联想起 B 区并逐步扩展到 B 全体。



神经元到底什么样？

从感官到执行的神经传输过程。



神经元到底什么样？
与普通细胞有什么区别？

神经元结构

与其他细胞显著不同的是，除胞体外，神经元还有树突、轴突和突触。

树突：树枝状，接受其他神经元传来的刺激。

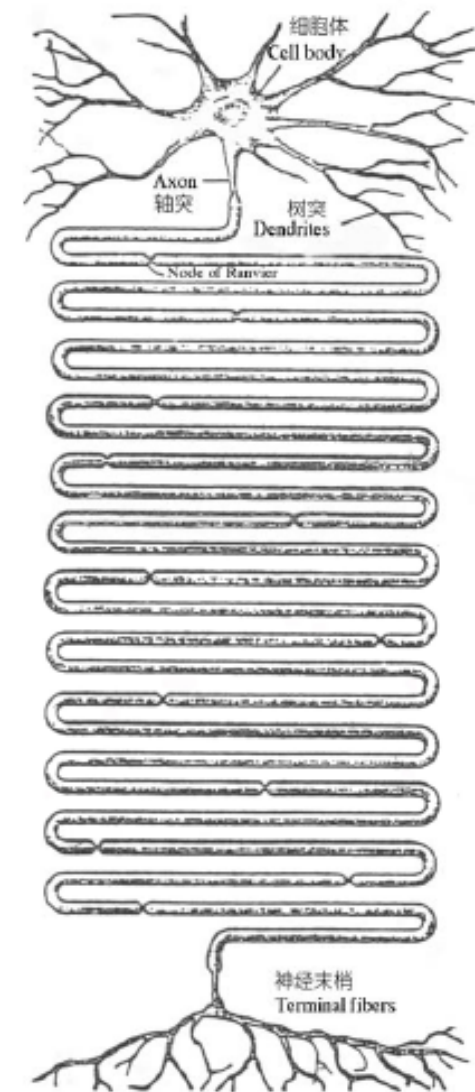
细胞体：包含神经元生存所必须的细胞物质。来自众多树突收集的信号汇集于此，满足一定条件后，胞体产生一个输出电信号。

轴突：胞体的输出电信号沿着蜿蜒的轴突传送到目标神经元。

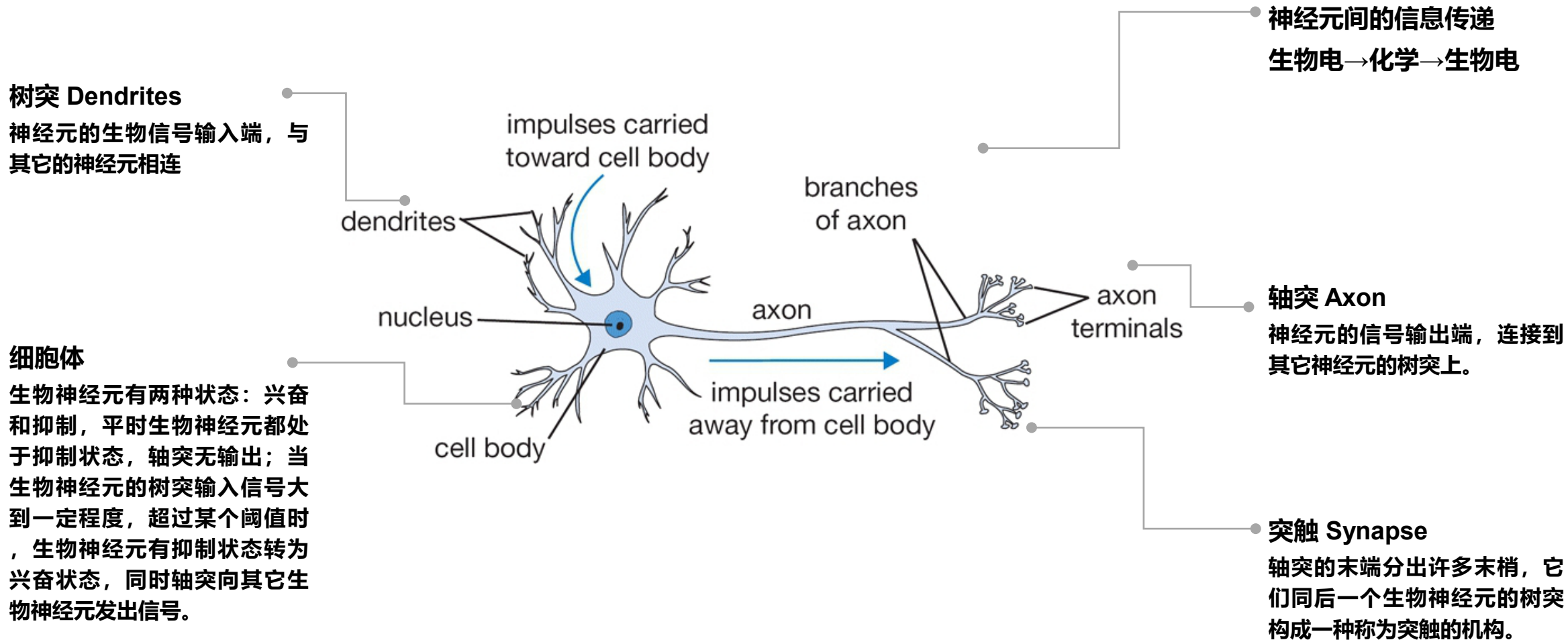
- 轴突只有一支，短的仅几微米，长的可达1米以上。
- 如大象坐骨神经元的胞体在脊髓，轴突却延伸到后脚跟。

神经束：由若干（可能多达几十万乃至百万）根轴突组成。

突触：突触前神经元的轴突神经末端与突触后神经元树突之间的连接处。



生物神经元



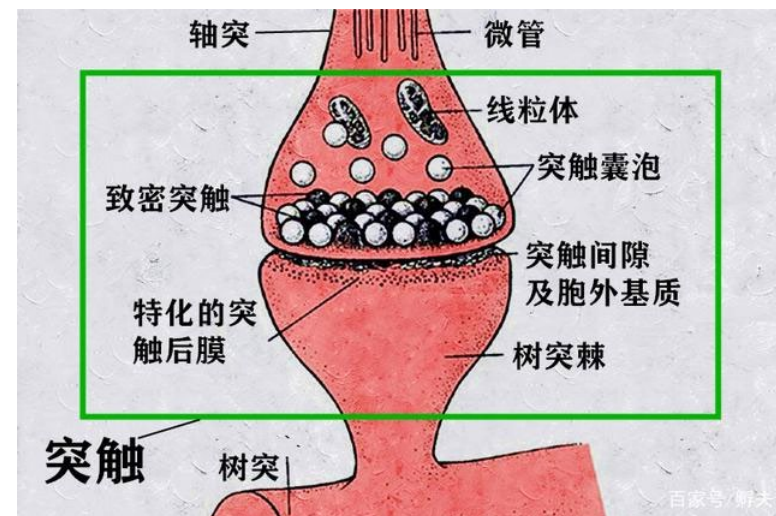
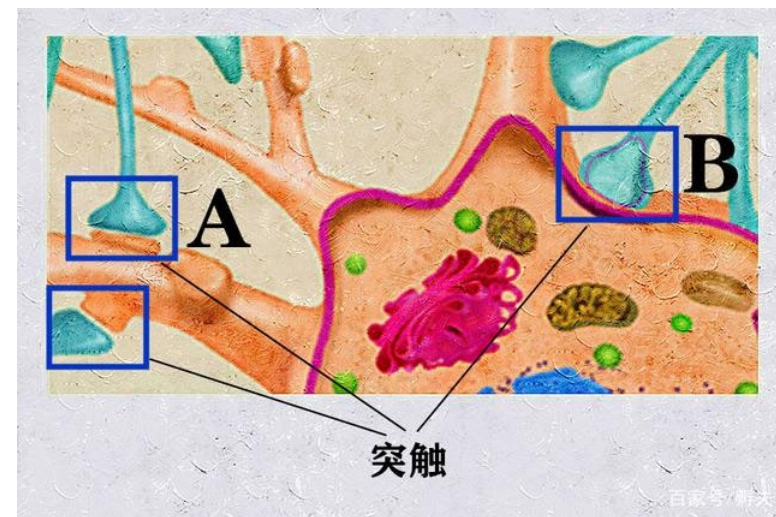
神经元连接的复杂性

突触是两神经元间的连接，是传输神经元的轴突与目标神经元的树突间的连接。

神经元内的电脉冲信号以每小时352千米（每秒98米）的速度沿轴突到达突触，经间隙（突触），到达接收神经元的树突。

信号渡过突触靠的是化学物质，生物化学作用到达目标神经元树突以后，再改为电信号，沿树突到达胞体。

生物电→化学→生物电



脑的可塑性

1939年英国学者康纳尔指出：

- 脑的可塑性就是脑皮层神经元网络的演变。

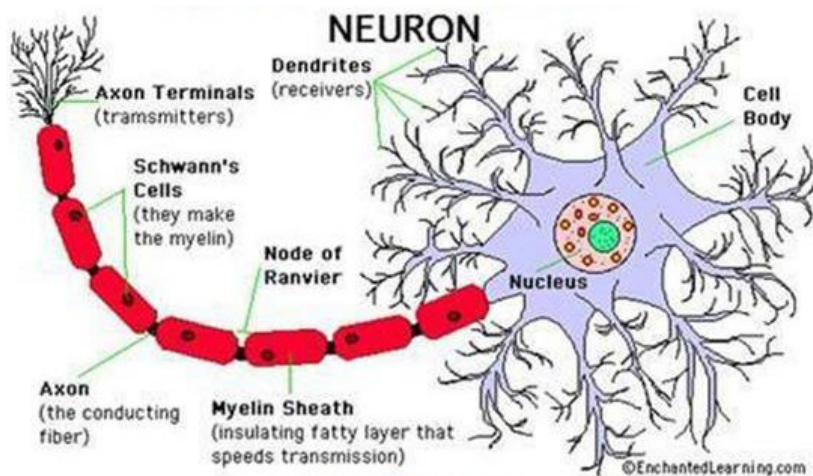
尽管当时还没有神经元学习的概念，但已经知道，初生婴儿脑皮层神经元网络处于一种欠组织状态，成熟的神经元和相互连接还不多。

随着婴儿长大，在外界环境刺激之下，神经系统不断地加强它的结构，调节那众多的突触连接强度，网络迅速复杂化和结构化。

婴儿脑皮层神经元网络发育

哈佛大学行为学家理查德教授

儿童时期大脑的能力主要是来源于神经联结的数量多少。从新生儿到18个月的孩子，其大脑发生着翻天覆地的变化，每秒钟约有700个新神经联结产生。

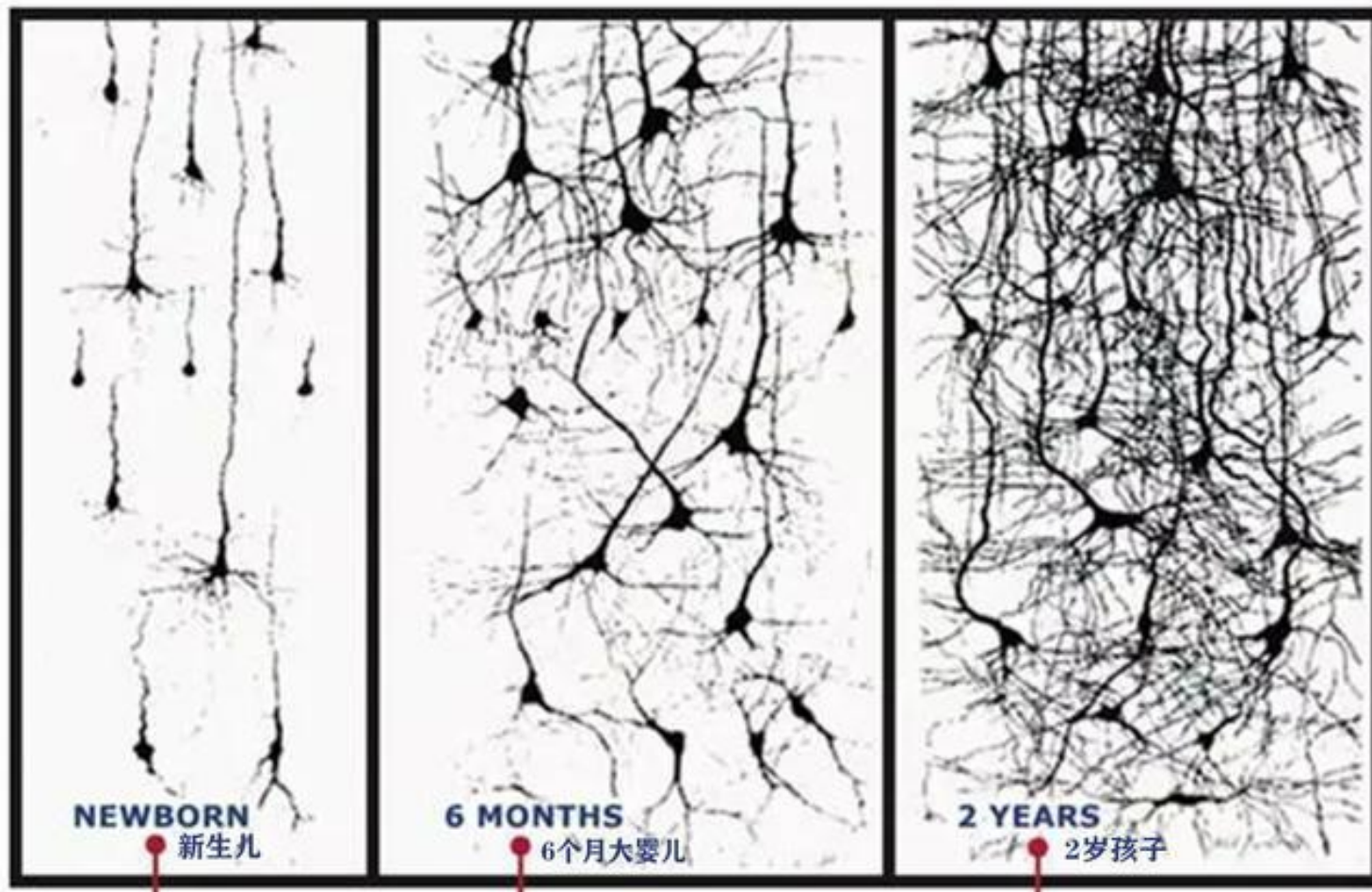


突触是靠体验培养

3岁之前
5岁之前

神经突触完成50%
神经突触完成90%

700 700 NEW NEURAL CONNECTIONS PER SECOND 每秒新生神经连接数量



脑模型（1943 年）

美国神经心理学家、神经病学家麦克卡洛和数学家、逻辑学家匹茨,把生物脑神经系统研究、数理逻辑思想以及图灵的计算理论相结合。

可视为计算智能的起步。

他们提出由突触与神经纤维连接构成系统，从而形成MP脑模型的思想。

1943年在《数学生物物理》杂志上发表了著名论文，提出了以后被称为MP模型的脑模型。

人工神经元模型：呈现“开”和“关”两种状态。

人工神经元模型

麦克卡洛和匹茨指出：

- 任何可计算函数、所有的逻辑关系，都可以用神经网络（神经计算）来实现；
- 所有逻辑连接符（与、或、非等）都可以用简单的神经网络实现。

匹茨与维纳一起开创了用电子装置模仿人脑结构和功能的新途径。

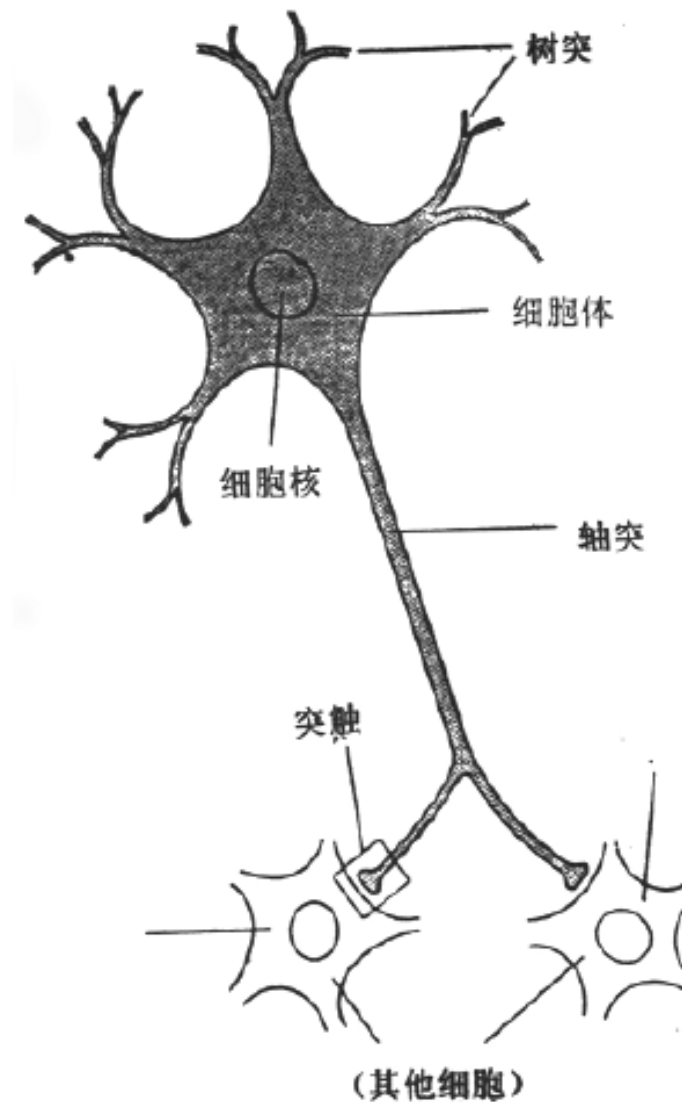
基于卡加尔—高尔基神经元学说，麦克卡洛和匹茨从一种简化的生物神经元模型，设计出人工神经元模型。

人工神经元模型

迄今为止,该人工神经元模型一直是人工神经网络设计的基础。

包括:

- 突触连接
- 加法器 Σ
- 激活函数 $f(\cdot)$



人工神经元模型

突触连接

- 其大小用权值(weight, 连接强度) w_{ki} 表示。
- 即第 i 个输入信号 x_i 与神经元 k (编号) 的连接。
- 权值越大表明所连接的输入信号影响越大。

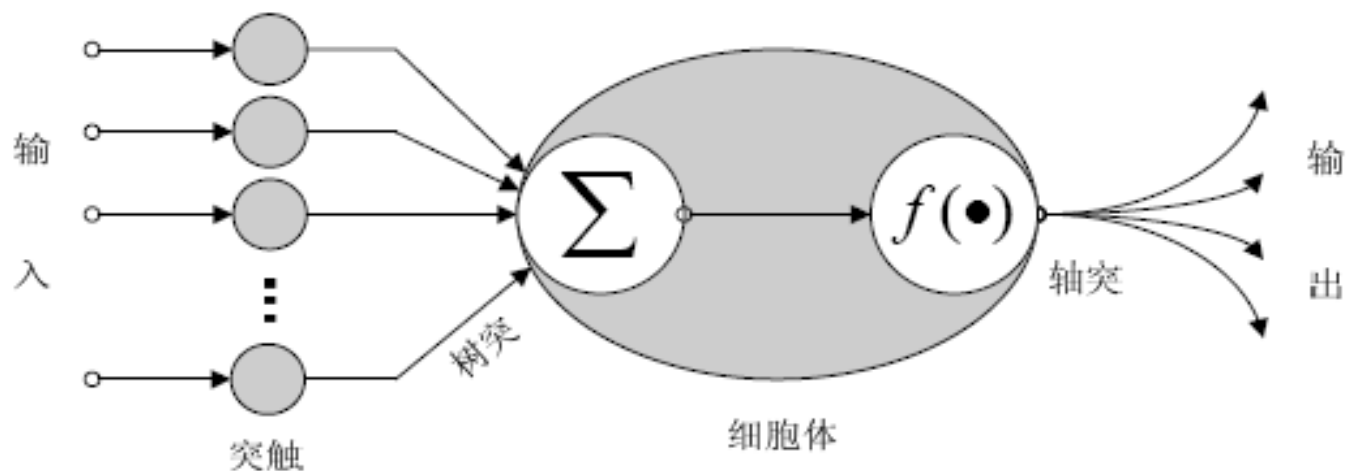
加法器 Σ

- 将所有的输入信号与相应突触权值的乘积相加，作为该神经元的总输入 net 。

激活函数 $f(\cdot)$

- 作用于总输入 net ，产生神经元输出 $y_k = f(net)$ 。

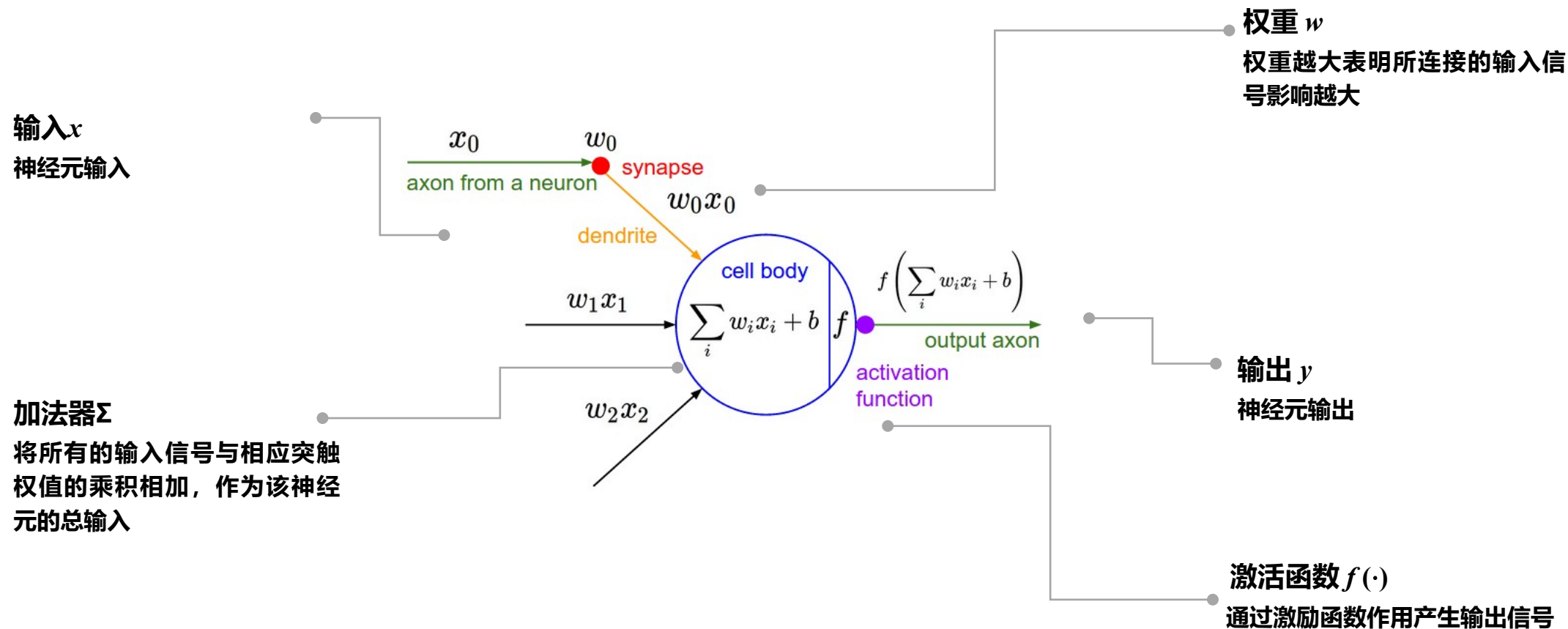
人工神经元模型



生物神经元→人工神经元模型：

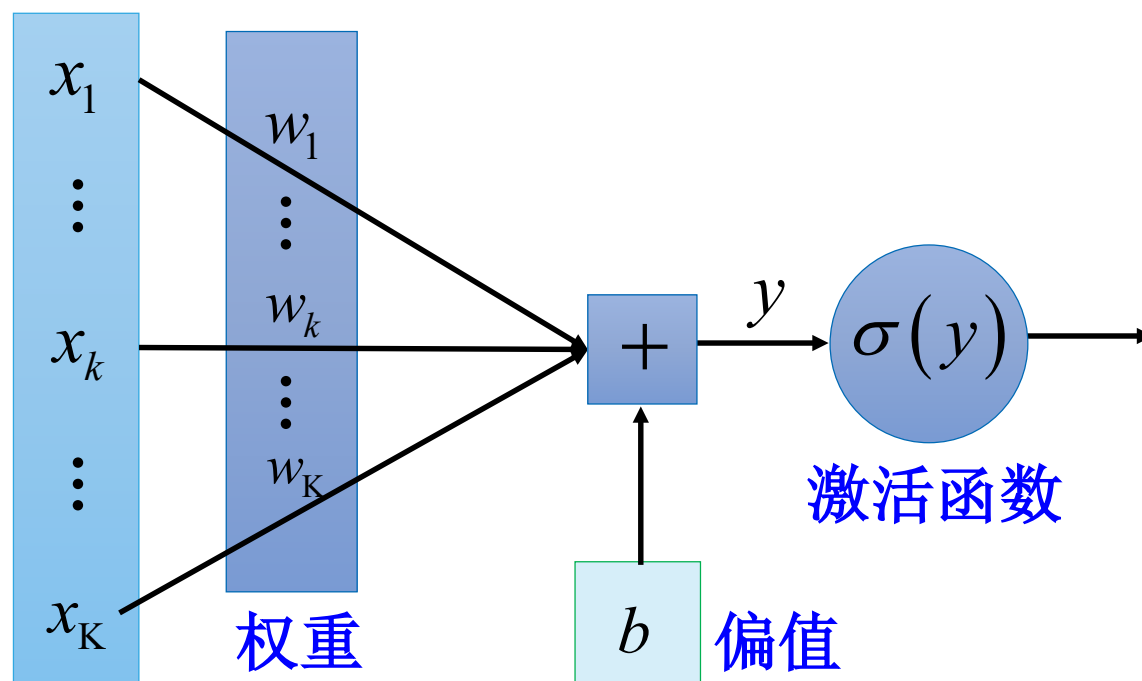
- 多个输入，通过突触连接到树突，进入细胞体。
- 在细胞体内相加。
- 通过激励函数作用产生输出信号，经轴突向外输出。

人工神经元模型



人工神经元模型

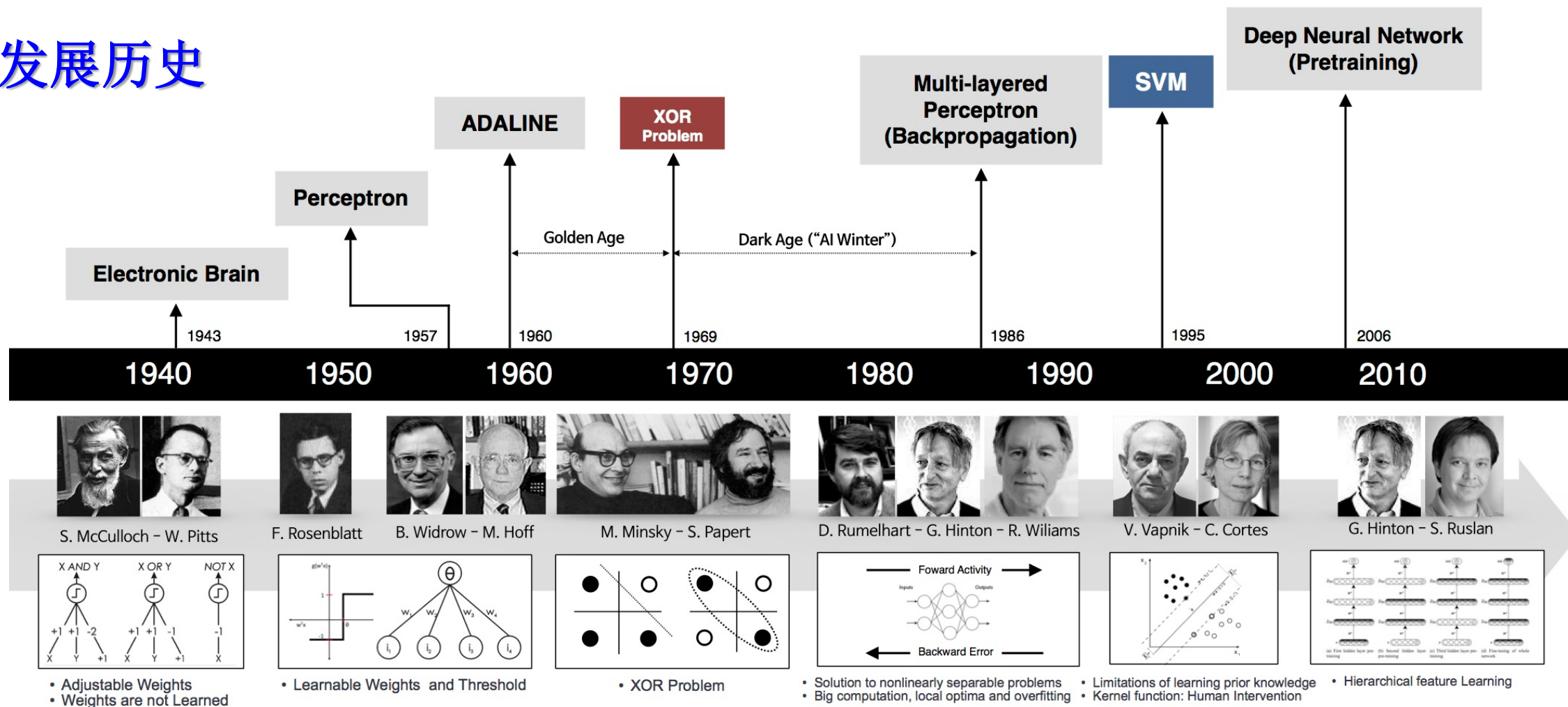
$$y = x_1 w_1 + \dots + x_k w_k + \dots + x_K w_K + b$$





- ✓ 学习的本质
学什么？ 怎么学？ ...
- ✓ 从生物神经元到人工神经元
生物神经元、人工神经元， ...
- ✓ **神经网络发展历史**
发展历史 ...
- ✓ 神经网络基本概念
激活函数、学习规则、优化算法、 ...
- ✓ 经典人工神经网络
前向神经网络、BP神经网络， Hopfield神经网络， ...

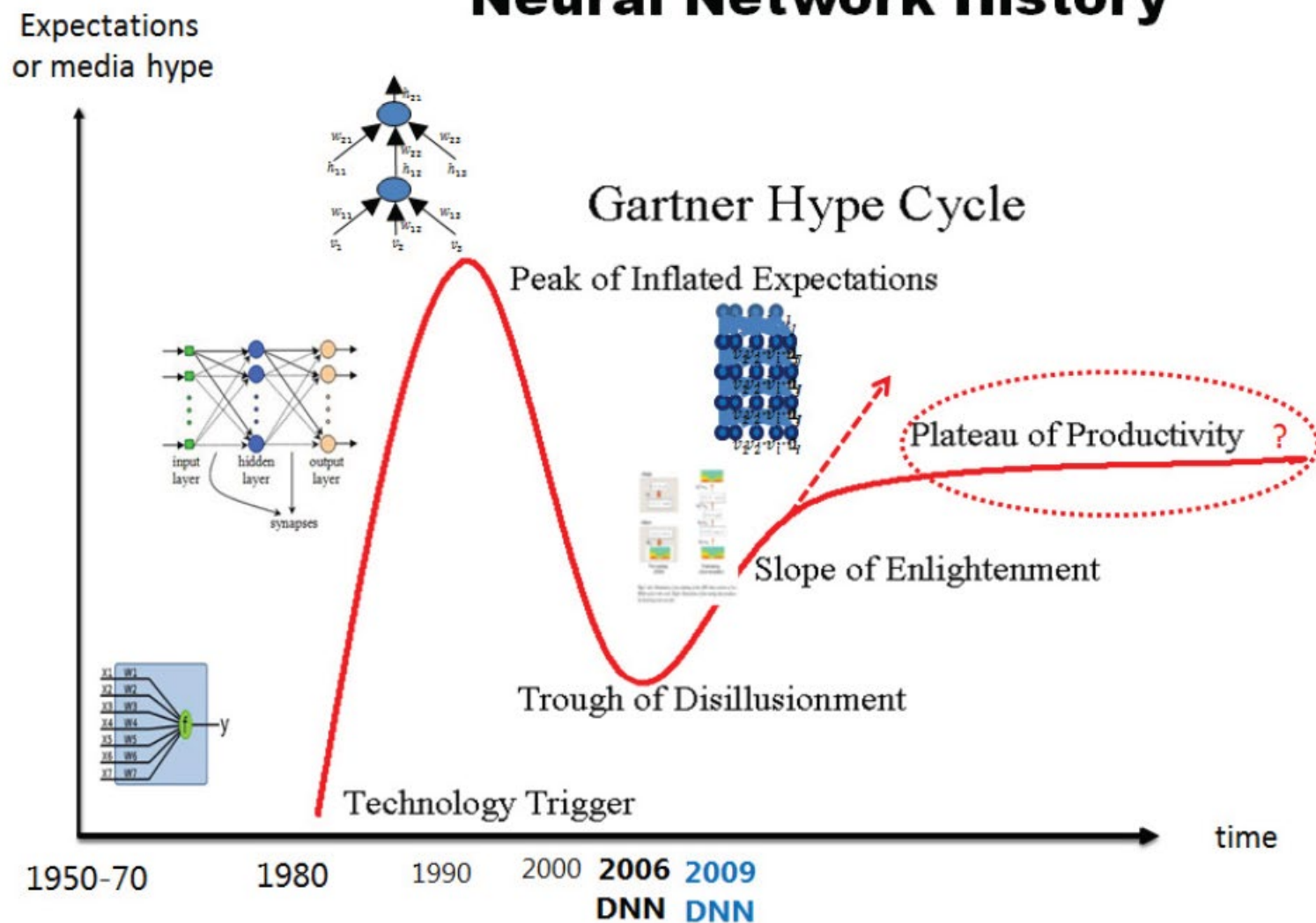
发展历史



http://beamandrew.github.io/deeplearning/2017/02/23/deep_learning_101_part1.html

发展历史

Neural Network History



诞生期



奥地利医生 Franz Joseph Gall (1758-1828) 推测人类的精神活动是由脑的功能活动实现的，认识到意识和精神活动具有物质基础。



意大利细胞学家 Camillo Golgi (1843~1926) 徒手将脑组织切成薄片，用重铬酸钾 - 硝酸银浸染法染色，第一次在显微镜下观察到了神经细胞和神经胶质细胞。这为神经科学的研究提供了最为基本的组织学方法。

诞生期



西班牙神经组织学家 Santiago Ramón y Cajal (1852~1934) 在掌握了 Golgi 染色法后, 进一步改良了 Golgi 染色法, 并发明了独创的银染法——还原硝酸银染色法, 可显示神经纤维的微细结构。他发现神经细胞之间没有原生质的联系, 因而提出神经细胞是整个神经活动最基本的单位 (故称神经元)



The Nobel Prize in Physiology or Medicine 1906

"in recognition of their work on the structure of the nervous system"



Camillo Golgi

1/2 of the prize

Italy

Pavia University
Pavia, Italy

b. 1843
d. 1926



Santiago Ramón y Cajal

1/2 of the prize

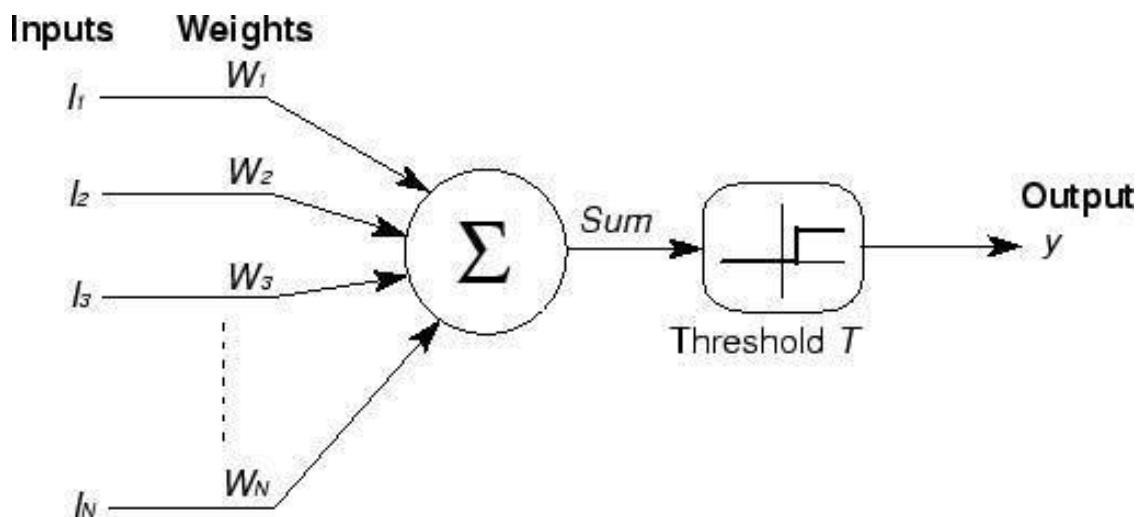
Spain

Madrid University
Madrid, Spain

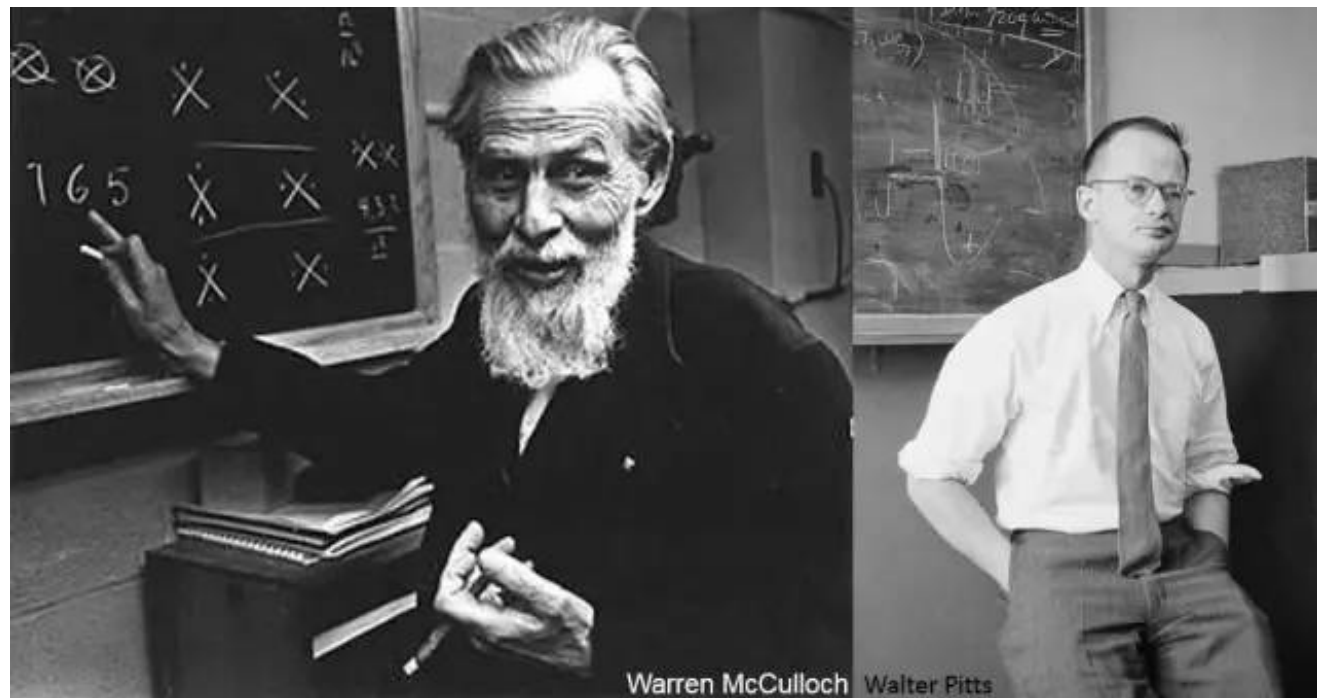
b. 1852
d. 1934

Santiago Ramón y Cajal 和 Camillo Golgi 两人共享了 1906 年诺贝尔生理学或医学奖

萌芽期



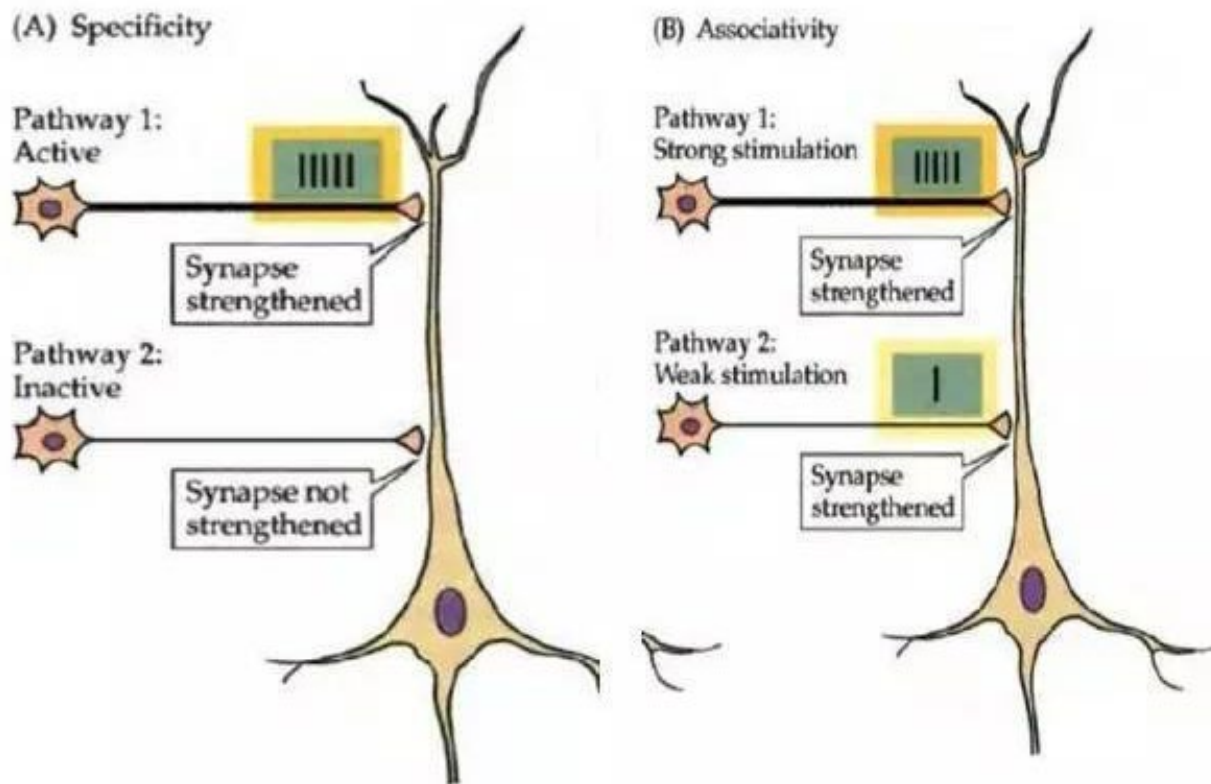
M-P模型



Warren McCulloch 和 Walter Pitts

1943年，Warren McCulloch 和 Walter Pitts 发表题为《A Logical Calculus of the Ideas Immanent in Nervous Activity》的论文，首次提出神经元的M-P模型。它借鉴已知的生物神经细胞机理，是第一个神经元数学模型，是人类史上对大脑工作原理描述的第一次尝试。M-P模型工作原理是神经元输入信号加权求和，与阈值比较再决定神经元是否输出，这从原理上证明了人工神经网络可以计算任何算术和逻辑函数。

萌芽期

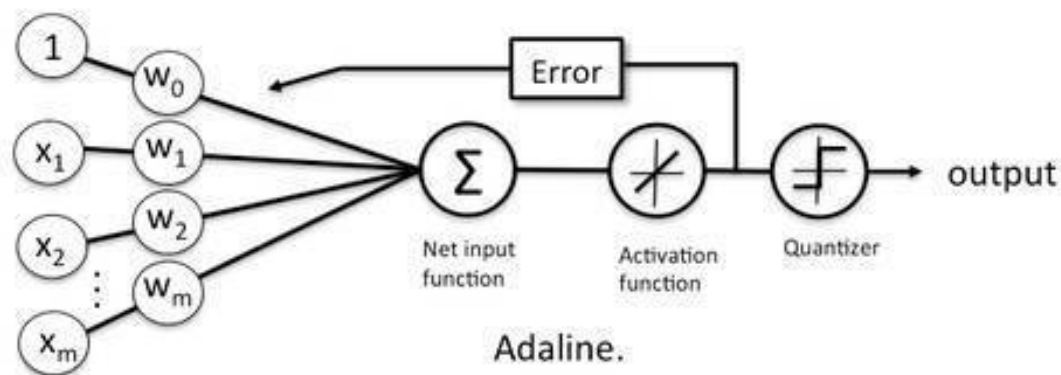
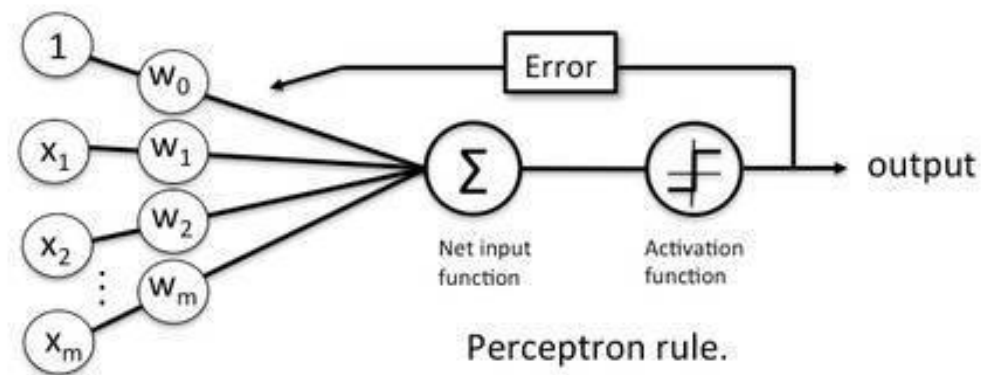


20世纪40年代末，Donald Olding Hebb在《The Organization of Behavior》中对神经元之间连接强度的变化进行了分析，首次提出来一种调整权值的方法，称为Hebb学习规则。受巴普洛夫的条件反射实验启发，Hebb认为如果两个神经元在同一时刻被激发，则它们间的联系应被强化。这就是Hebb提出的生物神经元的学习机制，在这种学习中，对神经元的重复刺激会使得神经元之间的突触强度增加。

成长期

1958年，Cornell航空实验室的Frank Rosenblatt提出了感知器（Perceptron）。它被视为是一种最简单形式的前馈神经网络，是一种二元线性分类器（激活函数为 $\text{sign}(x)$ ）。是人工神经网络的第一个实际应用，标志着神经网络进入了新的发展阶段。

1960年，斯坦福大学教授Bernard Widrow教授和他的研究生Ted Hoff开发了Adaline（Adaptive Linear Neuron 或 Adaptive Linear Element）和最小均方滤波器（LMS）。Adaline网络和感知机的区别就是将感知机的Step函数换为Linear线性函数。



冰河期



Marvin Minsky 和 Seymour Papert

1969年，Marvin Minsky 和 Seymour Papert 发表《Perceptrons: an introduction to computational geometry》一书，从数学角度证明了单层神经网络的功能有限，甚至在面对简单的“异或”逻辑问题时也显得无能为力。此后，神经网络的研究陷入了很长一段时间的低迷期。

冰河期 —> 繁荣期

1972年，芬兰的KohonenT.教授，提出了自组织神经网络SOM(Self-Organizing feature map)。

1974 年，Paul Werbos在哈佛大学攻读博士学位期间，就在其博士论文中发明了影响深远的著名**BP神经网络学习算法**。但没有引起重视。

1976年，美国Grossberg教授提出了著名的自适应共振理论ART(Adaptive Resonance Theory)，其学习过程具有自组织和自稳定的特征。

1982年，David Parker重新发现了**BP神经网络学习算法**。

繁荣期

1982年, John Hopfield 提出了连续和离散的Hopfield神经网络模型, 采用全互联型神经网络尝试对非多项式复杂度的旅行商问题进行了求解, 促进神经网络的研究再次进入了蓬勃发展的时期。

1983年, G. E. Hinton 和 T. J. Sejnowski 设计了玻尔兹曼机, 首次提出了“隐单元”的概念。在全连接的反馈神经网络中, 包含了可见层和一个隐层。

层数的增加可以为神经网络提供更大的灵活性, 但参数的训练算法一直是制约多层神经网络发展的一个重要瓶颈。



John Hopfield

繁荣期

1986年，David E. Rumelhart, Geoffrey E. Hinton 和 Ronald J. Williams发表文章《Learning representations by back-propagating errors》，使得BP神经网络学习算法受到重视。BP算法引入了可微分非线性神经元或Sigmoid函数神经元，克服了早期神经元的弱点，为多层神经网络的学习训练与实现提供了一种切实可行的解决途径。

1988年，继BP算法之后，David Broomhead 和 David Lowe 将径向基函数引入到神经网络的设计中，形成了径向基神经网络（RBF）。RBF网络是神经网络真正走向实用化的一个重要标志。

1989年，一系列文章对BP神经网络的非线性函数逼近性能进行了分析，证明对于具有单隐层，传递函数为Sigmoid的连续型前馈神经网络可以以任意精度逼近任意复杂的连续映射。这样，BP神经网络凭借能够保证对复杂函数连续映射关系的刻画能力（只要引入隐层神经元的个数足够多），打开了Marvin Minsky 和 Seymour Papert 早已关闭的研究大门。

沉寂期

此后的近十年时间，神经网络由于其浅层结构，容易过拟合以及参数训练速度慢等原因，曾经火热的神经网络又慢慢的淡出了人们的视线。

由于神经网络等学习方法在理论上缺乏实质性进展，统计学习理论开始受到越来越广泛的重视。

统计学习理论是一种专门研究小样本情况下机器学习规律的理论。V. N. Vapnik 等人从六、七十年代开始致力于此方面研究。到九十年代中期，其理论的不断发展和成熟。在这一理论基础上发展了一种新的通用学习方法——支持向量机（SVM），表现出很多优于已有方法的性能。

爆发期之前

1997年，Sepp Hochreiter和Jurgen Schmidhuber首先提出长短期记忆（LSTM）模型。

20世纪发表的神经网络论文中，目前引用最多的两篇论文就是LSTM和反向传播。

2019年12月LSTM论文引用达到2.6w次。超过了此前一直排名第一的反向传播算法的一系列文章。

Long short-term memory

[S Hochreiter](#), [J Schmidhuber](#) - *Neural computation*, 1997 - MIT Press

Learning to store information over extended time intervals by recurrent backpropagation takes a very long time, mostly because of insufficient, decaying error backflow. We briefly review Hochreiter's (1991) analysis of this problem, then address it by introducing a novel, efficient, gradient based method called long short-term memory (LSTM). Truncating the gradient where this does not do harm, LSTM can learn to bridge minimal time lags in excess of 1000 discrete-time steps by enforcing constant error flow through constant error carousels ...

☆ 被引用次数: 26166 相关文章 所有 46 个版本



Learning internal representations by error propagation

[DE Rumelhart](#), [GE Hinton](#), [RJ Williams](#) - 1985 - [apps.dtic.mil](#)

This paper presents a generalization of the perception learning procedure for learning the correct sets of connections for arbitrary networks. The rule, called the generalized delta rule, is a simple scheme for implementing a gradient descent method for finding weights that minimize the sum squared error of the system's performance. The major theoretical contribution of the work is the procedure called error propagation, whereby the gradient can be determined by individual units of the network based only on locally available information ...

☆ 被引用次数: 25398 相关文章 所有 31 个版本



爆发期之前



早在1989年，Yann Le Cun在贝尔实验室就开始使用卷积神经网络识别手写数字；

1998年，Yann Le Cun提出了用于字符识别的卷积神经网络LeNet5，并在小规模手写数字识别中取得了较好的结果。基于这些工作，Yann Le Cun也被称为卷积网络之父。

爆发期

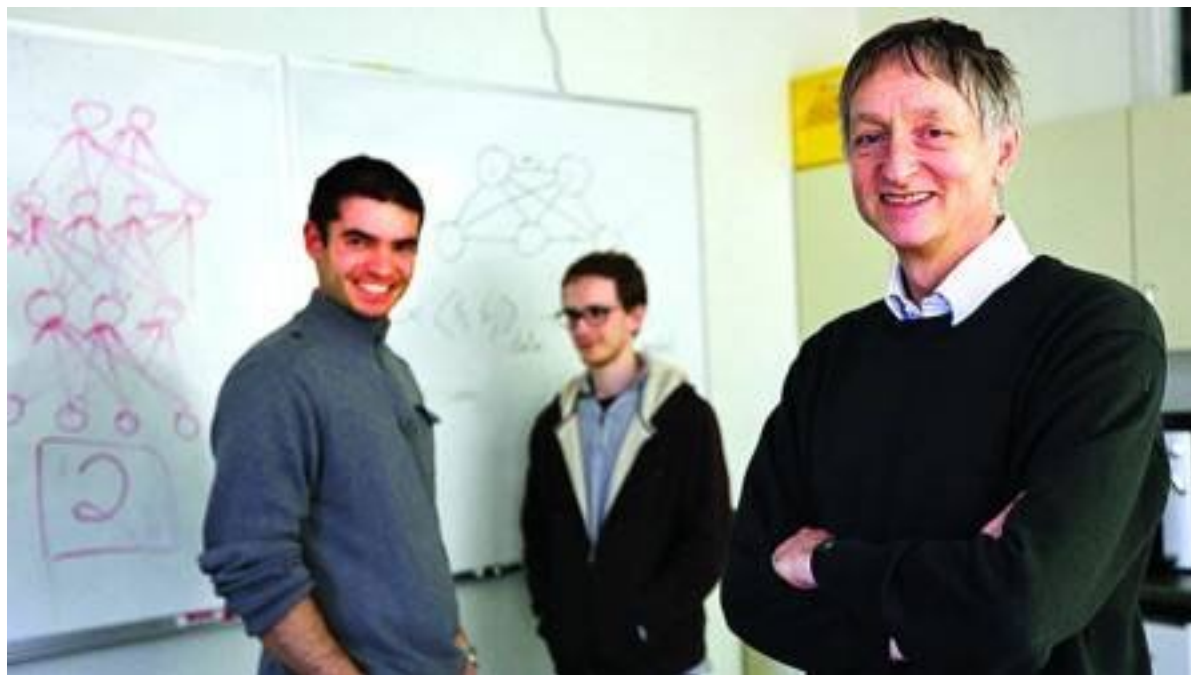
2006年，G. E. Hinton 和学生 R. R. Salakhutdinov 在《科学》上发表《Reducing the Dimensionality of Data with Neural Networks》。文章阐述了两个重要观点：一是多隐层的神经网络可以学习到能刻画数据本质属性的特征，对数据可视化和分类等任务有很大帮助；二是可以借助于无监督的“逐层初始化”策略来有效克服深层神经网络在训练上存在的难度。

这篇文章是一个分水岭，拉开了深度学习大幕，标志着深度学习的诞生。从感知机提出，到BP算法应用以及2006年以前的历史被称为浅层学习，以后的历史被称为深度学习。



G. E. Hinton（中）和他的学生 R. R. Salakhutdinov（左）

爆发期



From left to right: Ilya Sutskever, Alex Krizhevsky and Geoffrey Hinton

2012年，Alex Krizhevsky等使用采用卷积神经网络的AlexNet在ImageNet竞赛图像分类任务中取得了最好成绩，是卷积神经网络在图像分类中的巨大成功。

随后Alex Krizhevsky., Ilya Sutskever.和 Geoffrey Hinton.发表了文章《ImageNet Classification with Deep Convolutional Neural Networks》。

爆发期

2019年2019年3月27日 ——ACM宣布，深度学习的三位创造者Yoshua Bengio, Yann LeCun 以及 Geoffrey Hinton获得了2019年的图灵奖。





学习的本质

学什么？怎么学？ ...



从生物神经元到人工神经元

生物神经元、人工神经元， ...



神经网络发展历史

发展历史 ...



神经网络基本概念

激活函数、学习规则、优化算法、 ...



经典人工神经网络

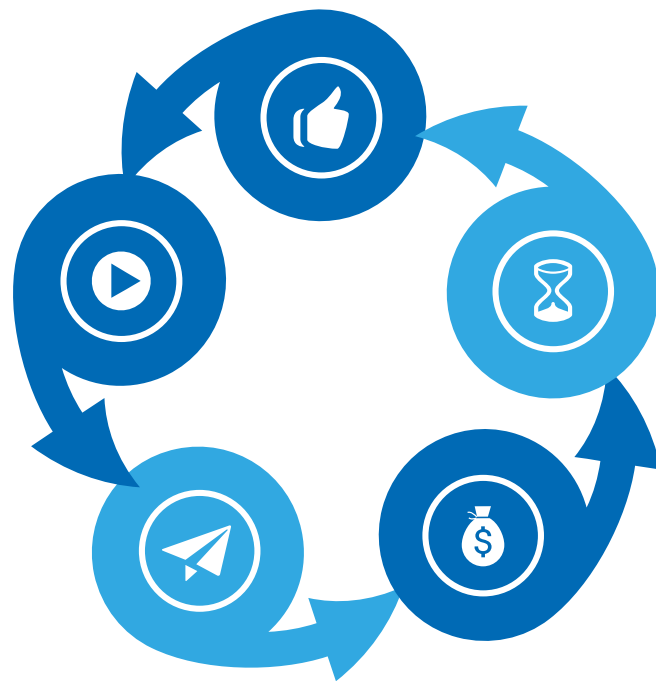
前向神经网络、BP神经网络、Hopfield神经网络， ...

感知器与线性分类

多层感知器

误差反传与梯度下降

损失函数



激活函数

优化算法

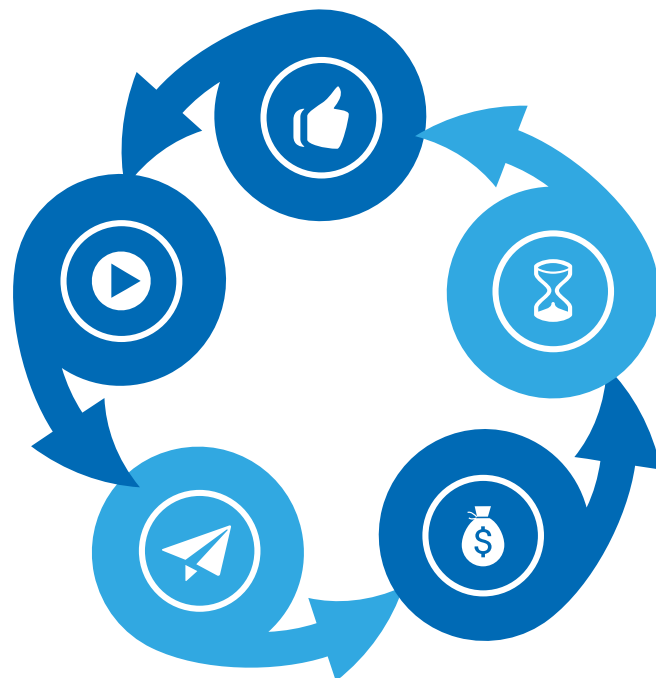
其他

感知器与线性分类

多层感知器

误差反传与梯度下降

损失函数



激活函数

优化算法

其他

罗森勃拉特感知器（1958 年）

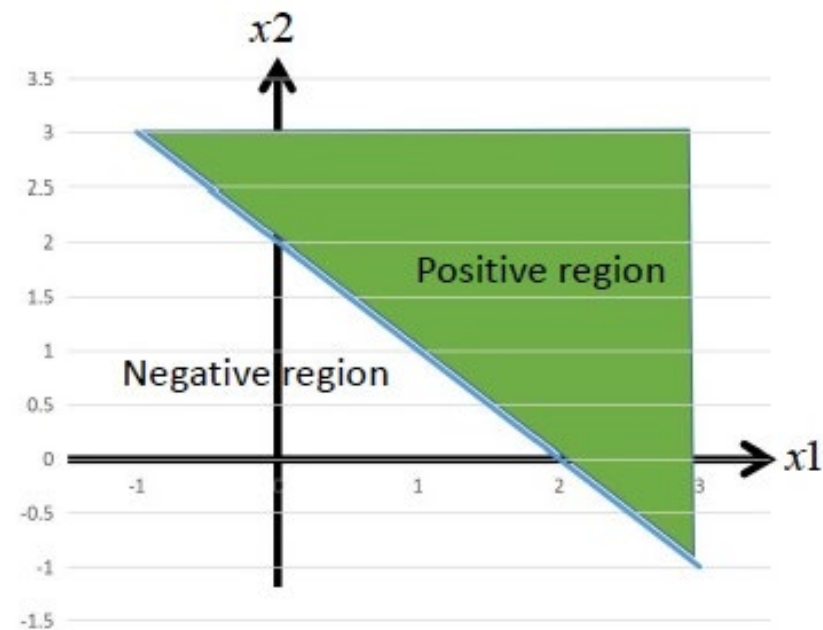
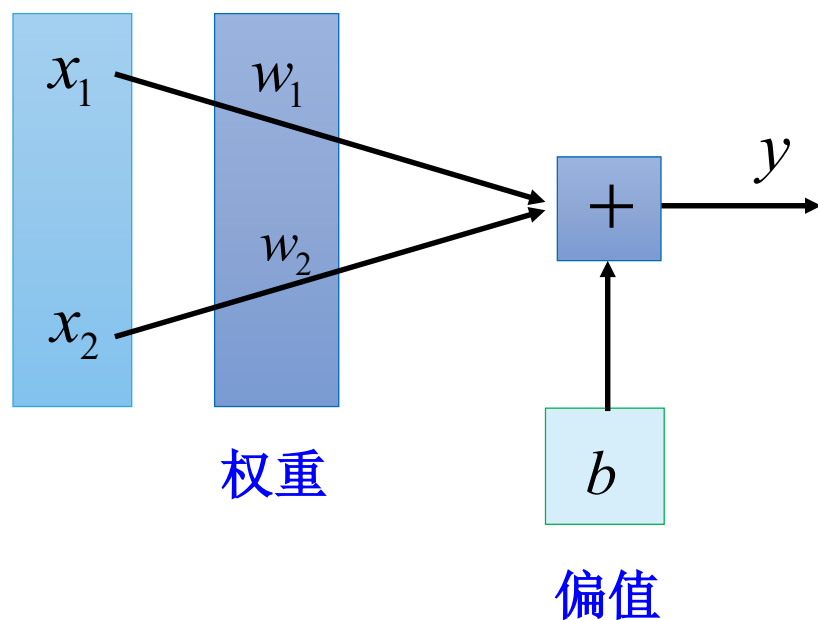
1958年，美国著名心理学家罗森勃拉特在《心理学评论》上发表了著名论文“感知器：脑的组织与信息存储的概率模型”。推广了MP模型，提出了著名的感知器模型（脑感知模型）。

罗森勃拉特感知器是有指导学习方式神经网络的第一个模型。涉及并开始解决神经网络线性模型、学习算法与收敛、分类等基本问题。

罗森勃拉特感知器是用于线性可分模式分类的、最简单的神经网络模型。

感知器(Perceptron)——线性分类器(Linear classifier)

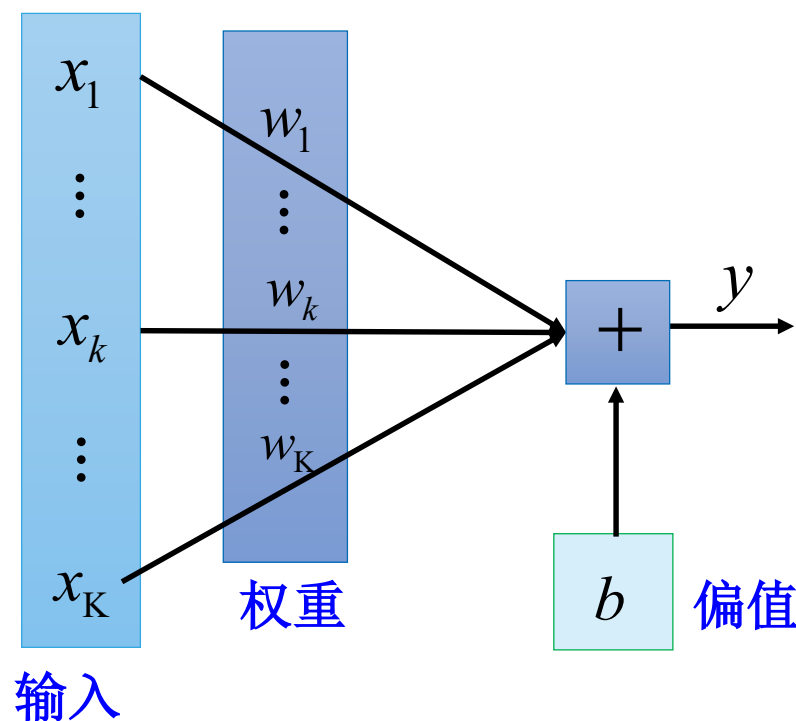
$$y = x_1 w_1 + x_2 w_2 + b$$



$$w_1 = 1, w_2 = 1, b = -2$$

感知器(Perceptron)——线性分类器(Linear classifier)

$$y = x_1 w_1 + \dots + x_k w_k + \dots + x_K w_K + b$$

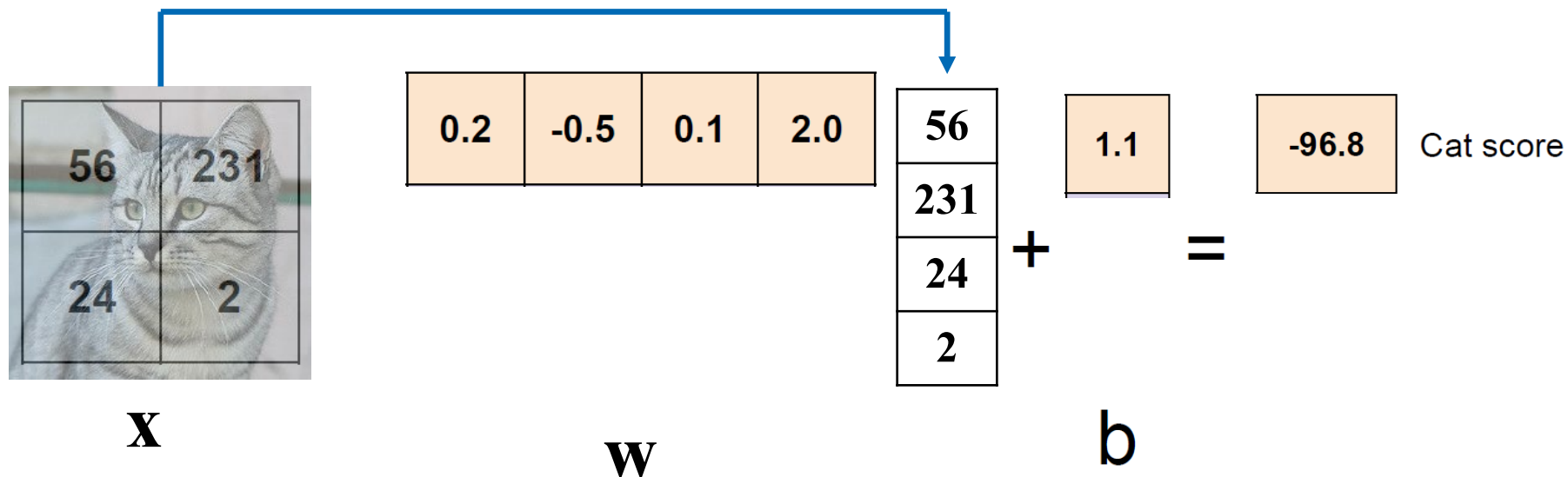


超平面上的线性分类

线性分类器 $f(\mathbf{x}, \mathbf{w}, \mathbf{b}) = \mathbf{w}\mathbf{x} + \mathbf{b}$

Example with an image with 4 pixels, and 3 classes (cat/dog/ship)

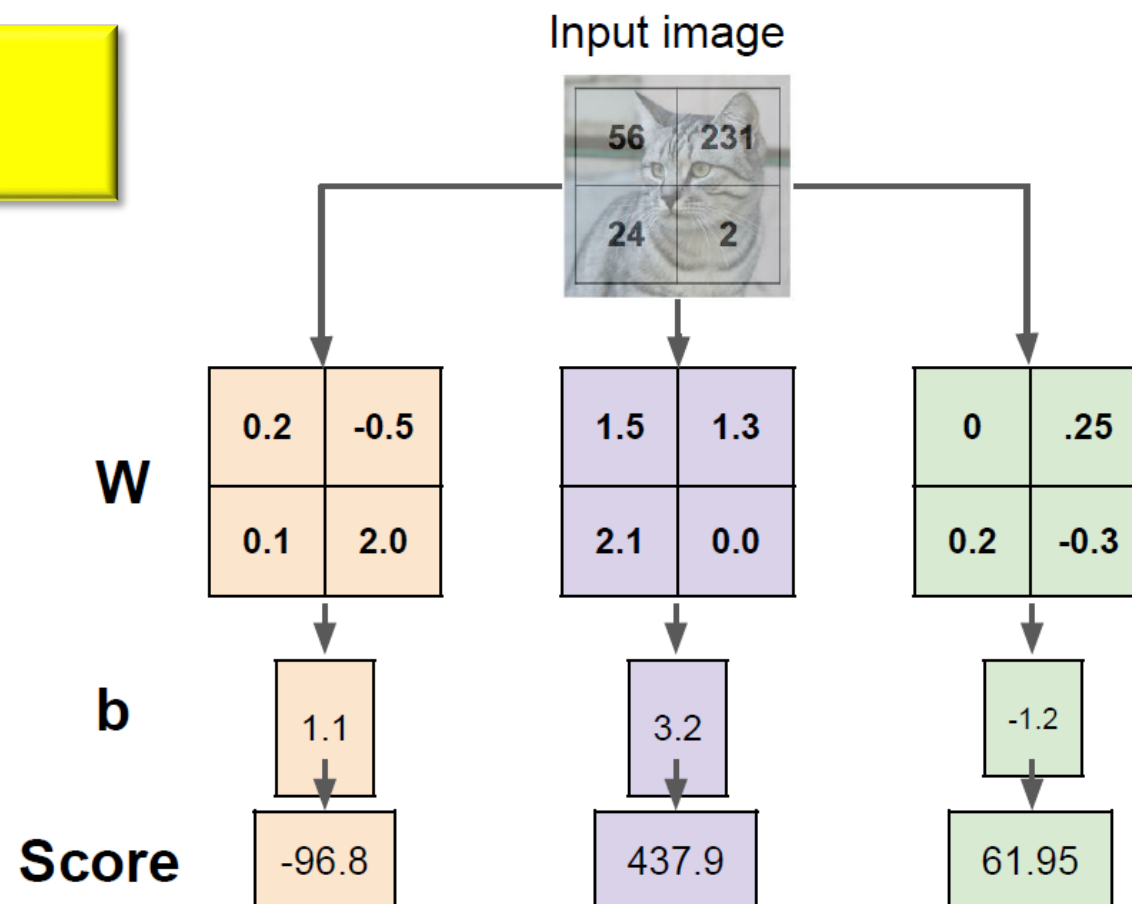
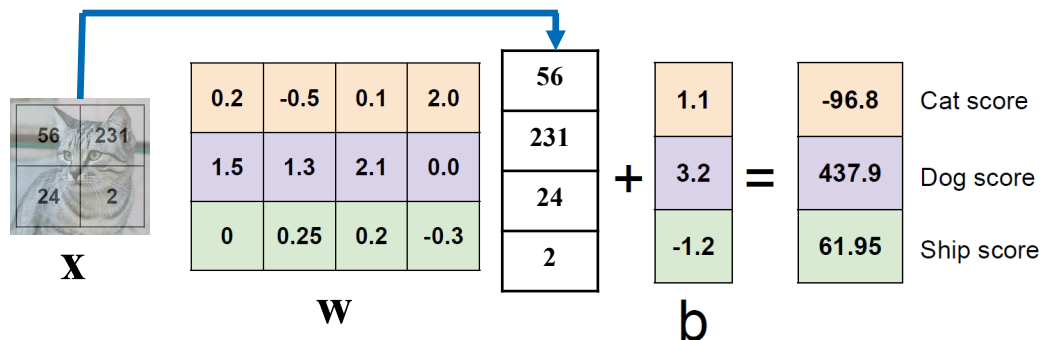
排列为一列



线性分类器 $f(x, w, b) = wx + b$

代数表示形式: $f(x, w, b) = wx + b$

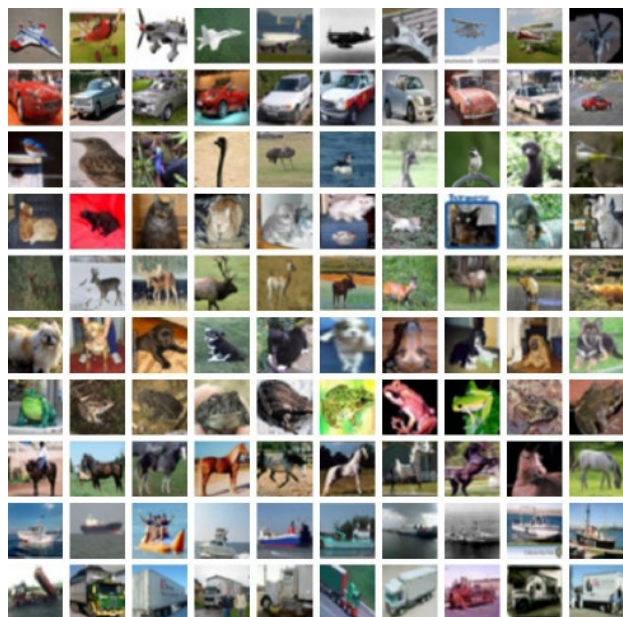
排列为一列



线性分类器 $f(x, w, b) = wx + b$

视觉表示形式：为每一类建立一个分类器

飞机
汽车
鸟
猫
鹿
狗
蛙
马
船
卡车



飞机

汽车

鸟

猫

鹿

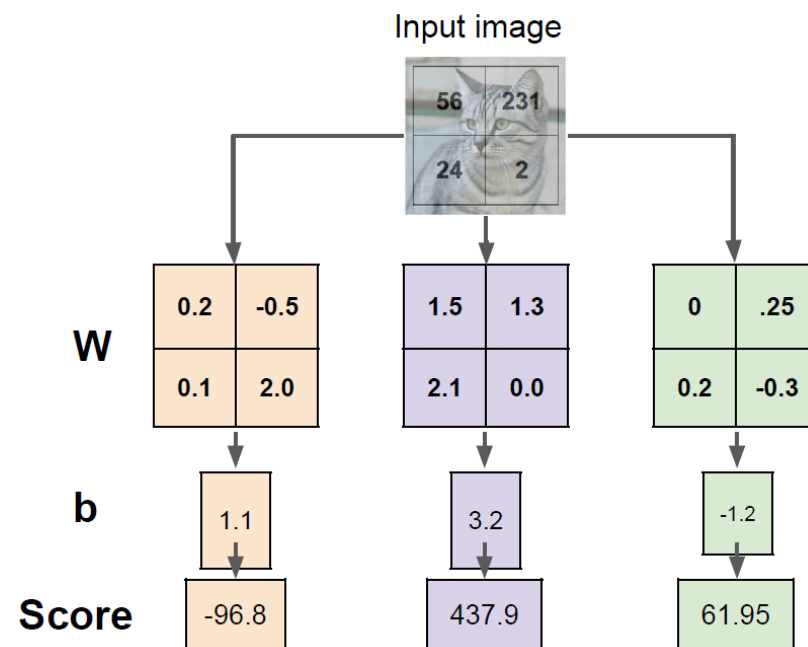
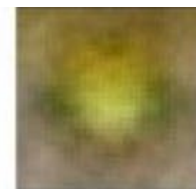
狗

蛙

马

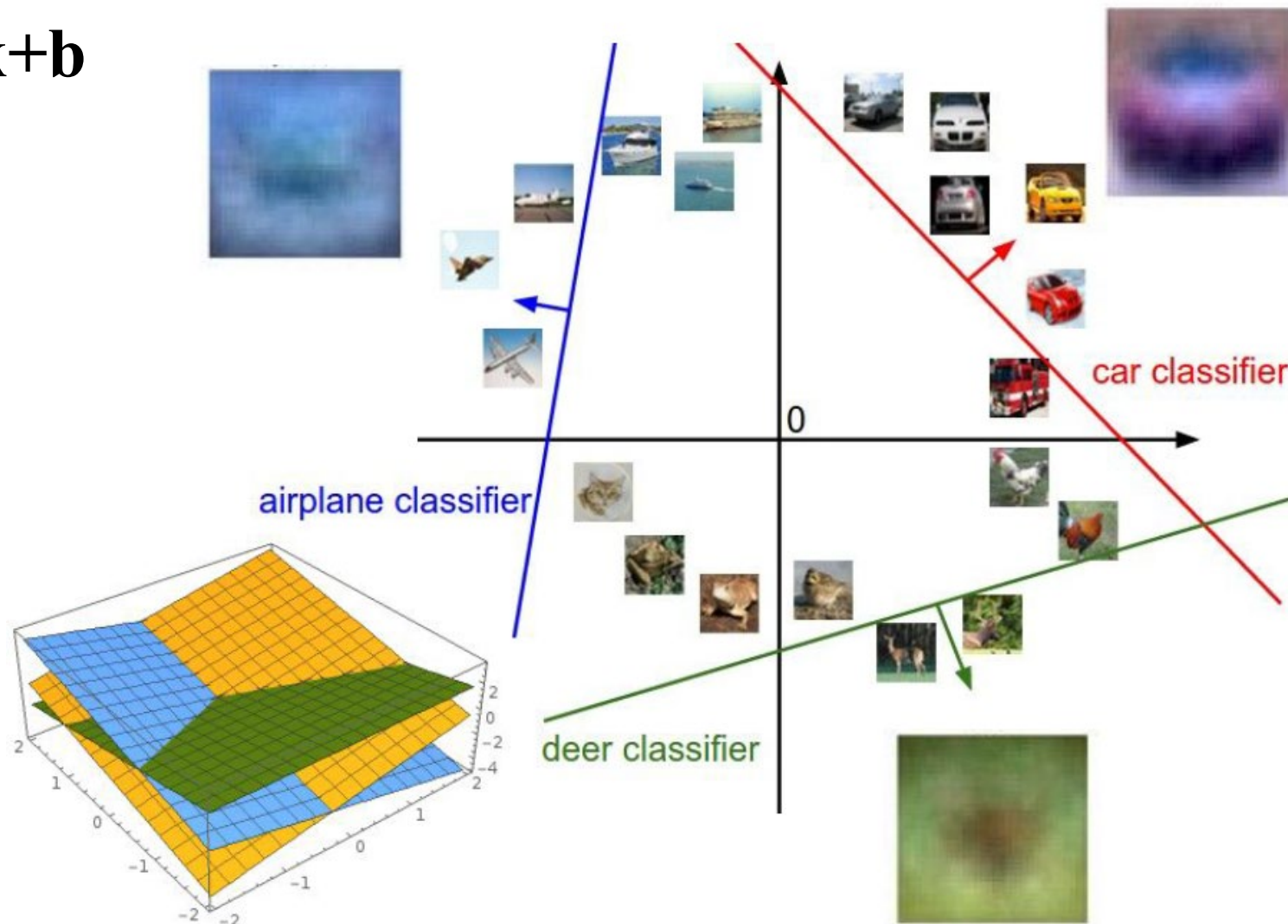
船

卡车



线性分类器 $f(\mathbf{x}, \mathbf{w}, \mathbf{b}) = \mathbf{w}\mathbf{x} + \mathbf{b}$

几何表示形式：采用
超平面切分目标空间

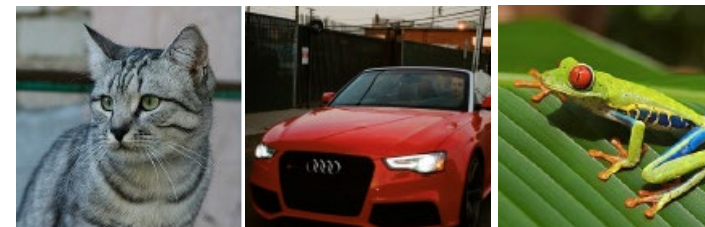


线性分类器 $y=f(\mathbf{x}, \mathbf{w}, \mathbf{b})=\mathbf{w}\mathbf{x}+\mathbf{b}$

如何判断权重好还是不好?

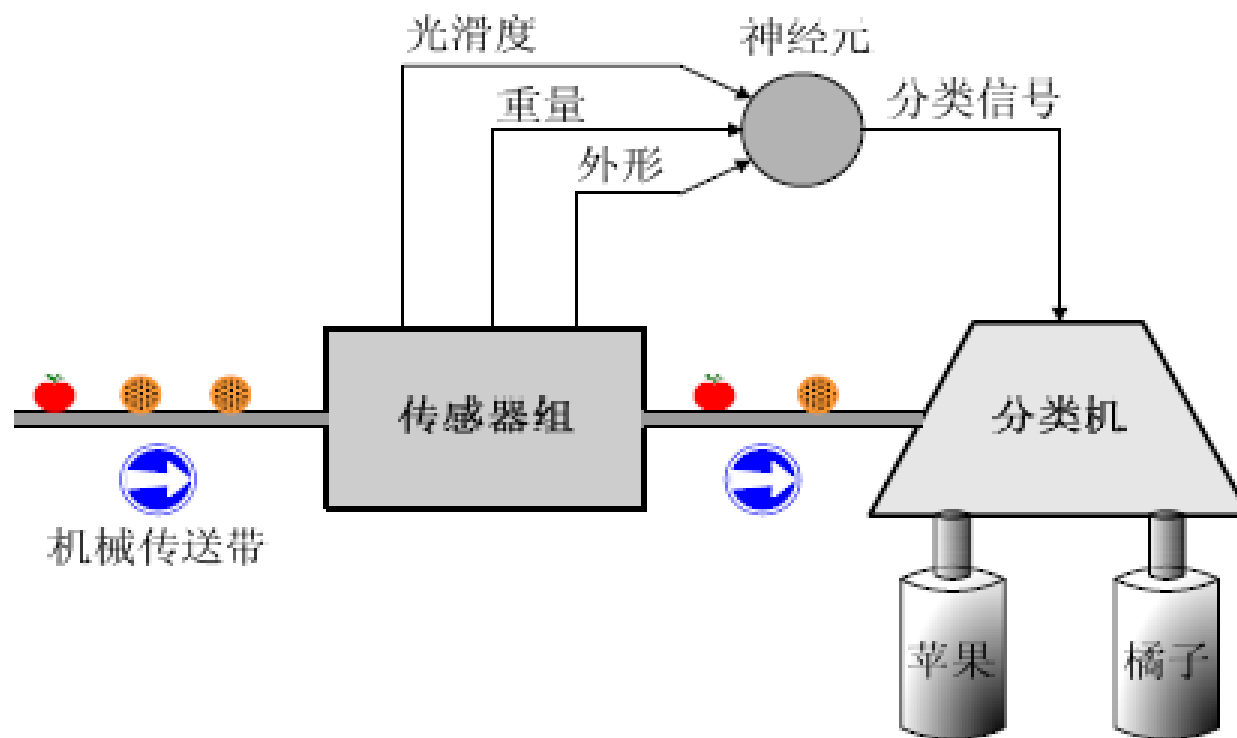
线性分类器直接输出有正负，难以和数据标注统一。

通常做法，线性分类器直接输出经Softmax转为[0,1]的数。



	猫	车	蛙
飞机	-3.45	-0.51	3.42
汽车	-8.87	6.04	4.64
鸟	0.09	5.31	2.65
猫	2.9	-4.22	5.1
鹿	4.48	-4.19	2.64
狗	8.02	3.58	5.55
蛙	3.78	4.49	-4.34
马	1.06	-4.37	-1.5
船	-0.36	-2.09	-4.79
卡车	-0.72	-2.93	6.14

感知器分类举例



感知器水果自动分类机

具有三种传感器： 外形传感器——接近圆形的为 1，接近椭圆形的为-1

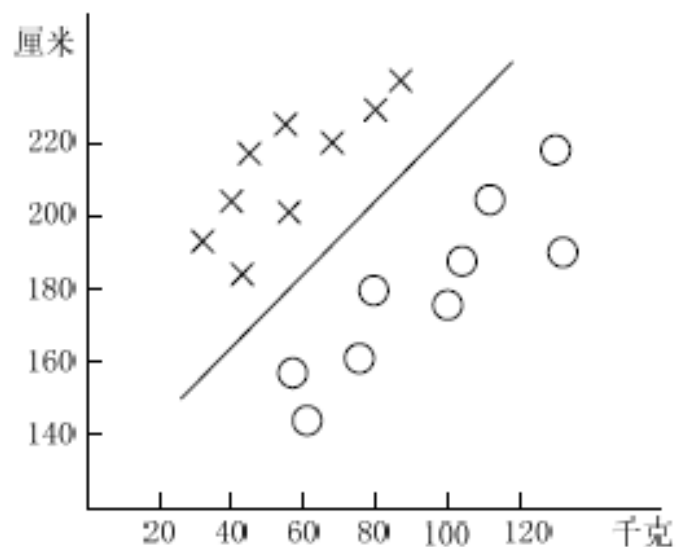
光滑度传感器——表面光滑的为 1，较粗糙的为-1

重量传感器——重量超过 1 磅的为 1，轻于 1 磅的为-1

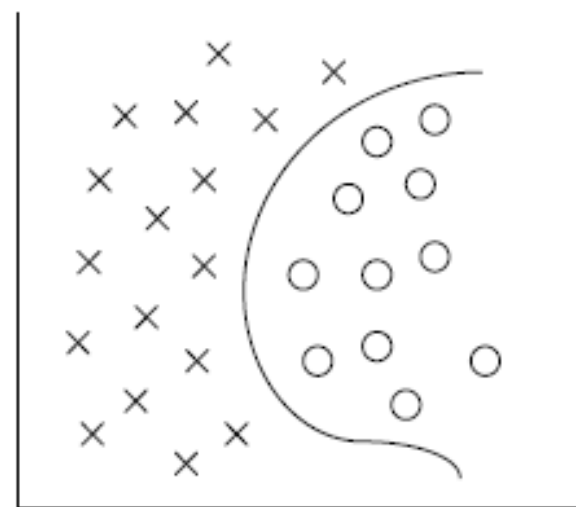
依此次序，标准橘子的模式为 $[-1, -1, 1]$ ，标准苹果的模式为 $[1, -1, 1]$ 。神经元完成橘子与苹果两个模式的识别，输出分类信号控制分类机。

线性分类器局限

线性可分的模式分类



两类模式（数据）线性可分的例子。

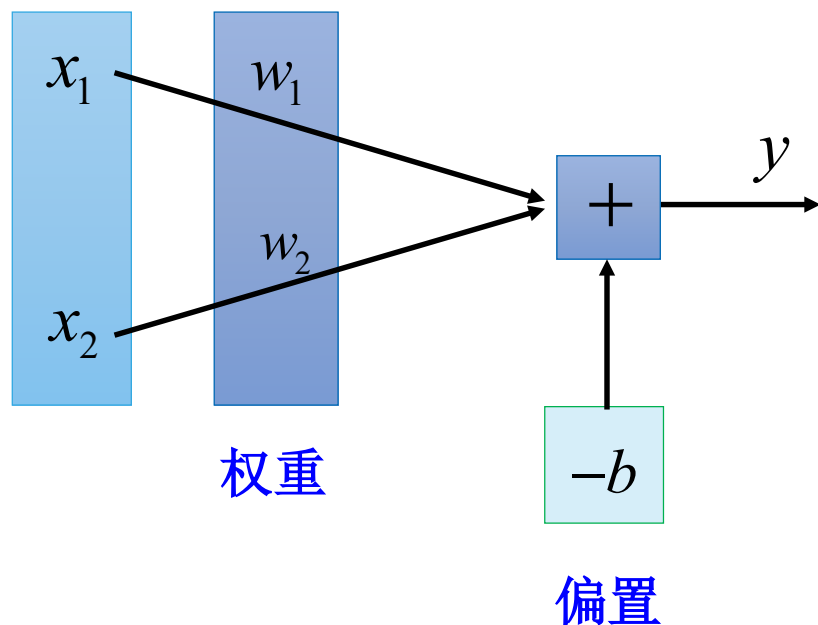
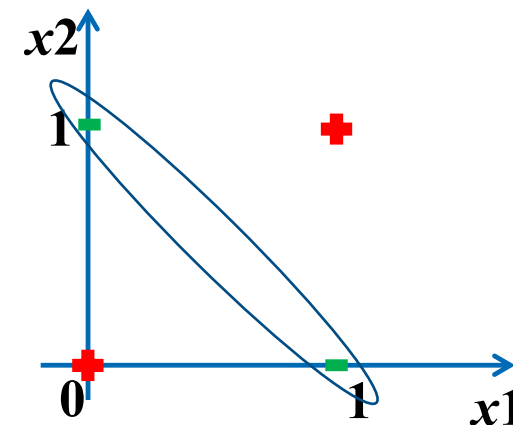


两类模式线性不可分的情况（非线性分类问题）。

线性分类器局限——异或问题

$$y = x_1 w_1 + x_2 w_2 - b$$

Input x1	Input x2	Output y
0	0	0
0	1	1
1	0	1
1	1	0



$$0 \cdot w_1 + 0 \cdot w_2 + b < 0 \Rightarrow b < 0$$

$$0 \cdot w_1 + 1 \cdot w_2 + b > 0 \Rightarrow b > -w_2$$

$$1 \cdot w_1 + 0 \cdot w_2 + b > 0 \Rightarrow b > -w_1$$

$$1 \cdot w_1 + 1 \cdot w_2 + b < 0 \Rightarrow b < (-w_1) + (-w_2)$$

出现矛盾!

[Minsky and Papert, "Perceptrons", 1969]

线性分类器局限

Input: x

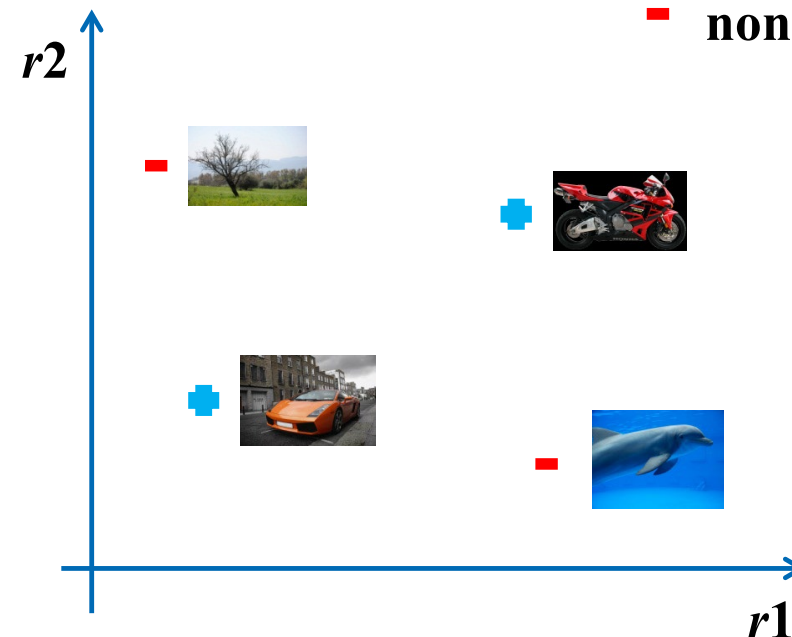


Output: y

Label “motorcycle”



⊕ car
- non-car



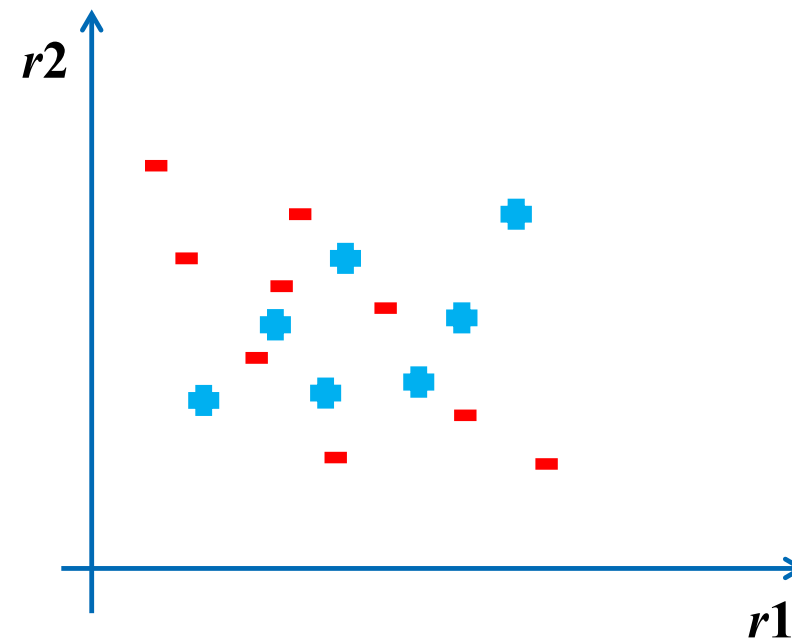
线性分类器局限

Input: x

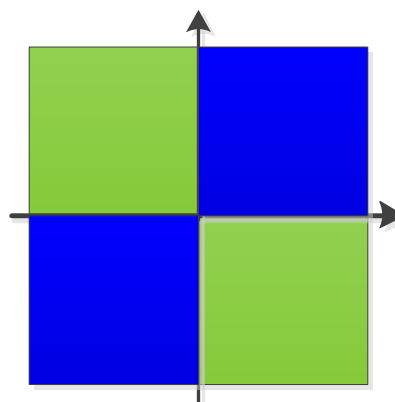
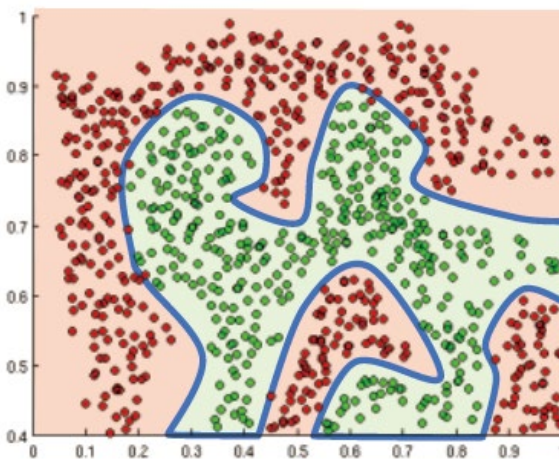
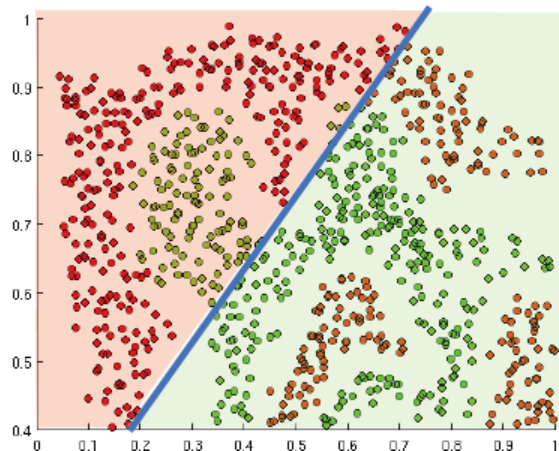


Output: y

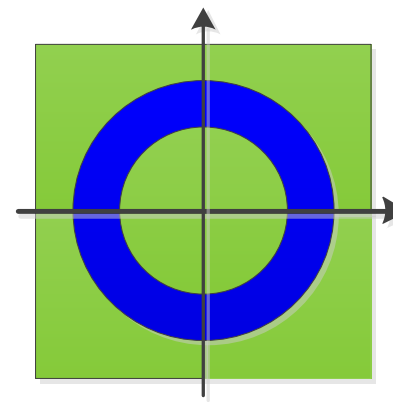
Label “motorcycle”



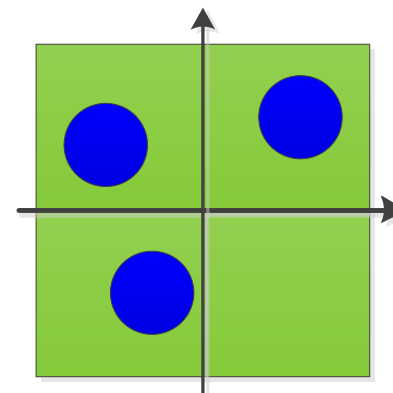
线性分类器局限



类1: 1、3象限
类2: 2、4象限



类1: $1 < L2 \text{范数} < 2$
类2: 其他



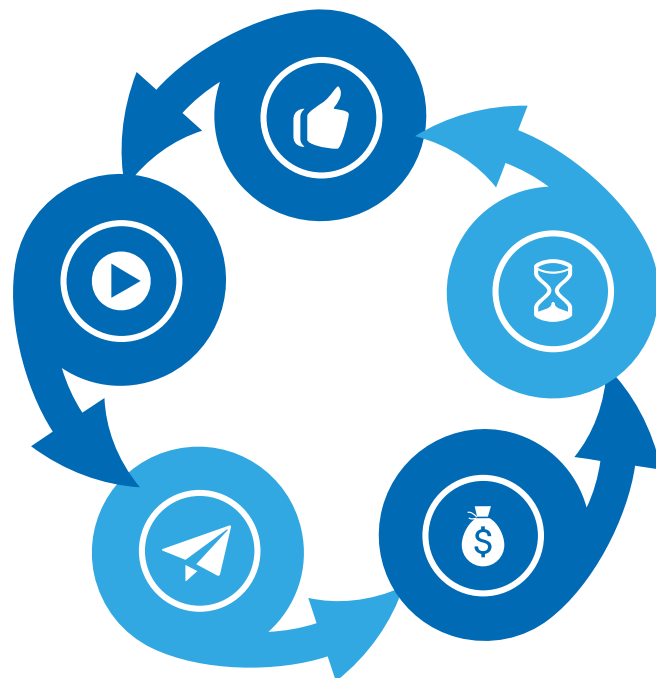
类1: 三种模式
类2: 其他

感知器与线性分类

多层感知器

误差反传与梯度下降

损失函数

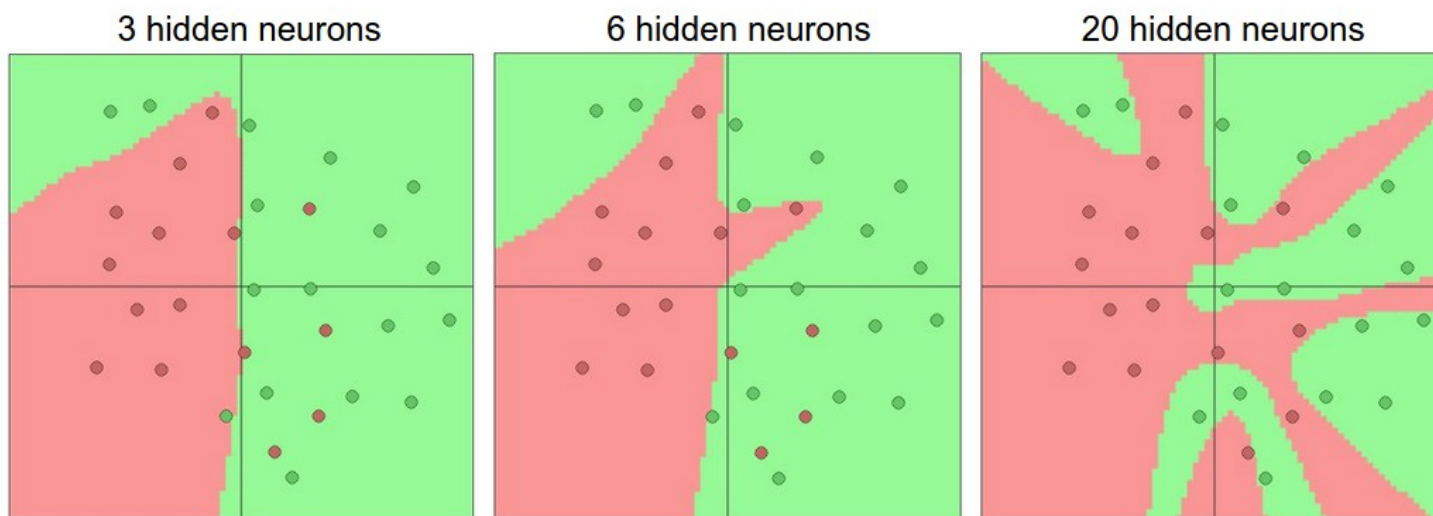
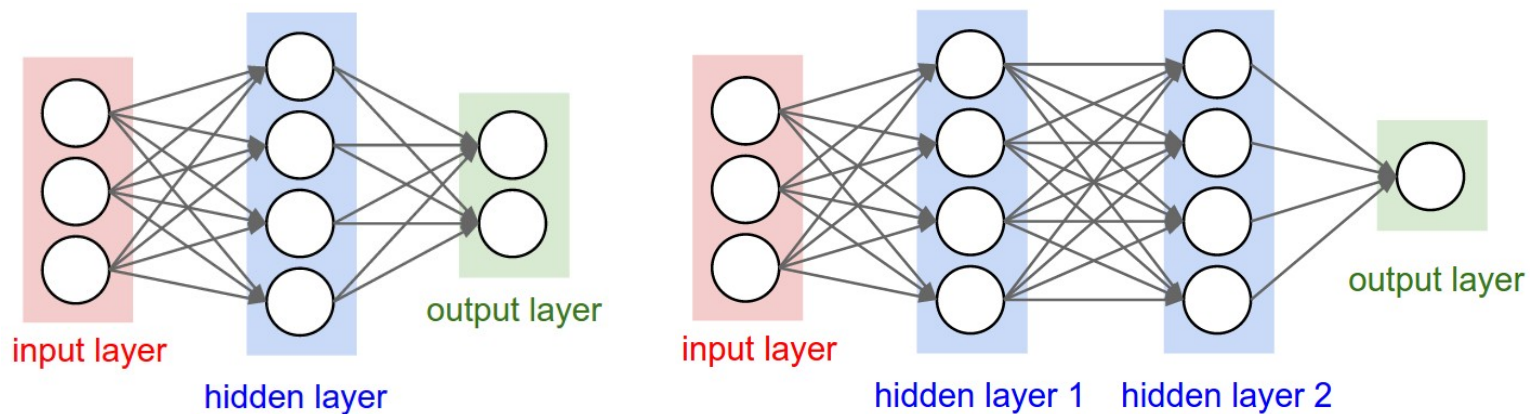


激活函数

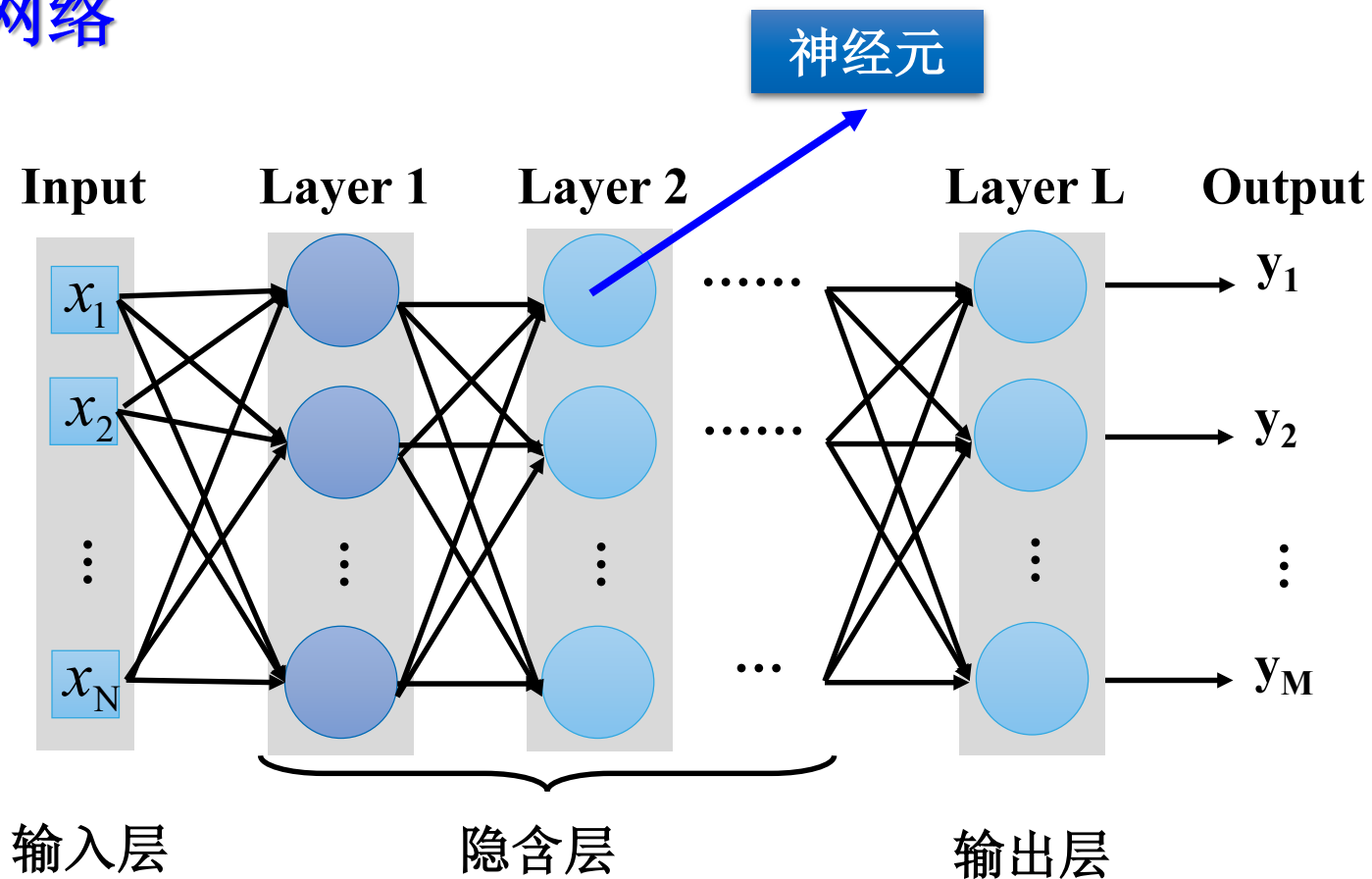
优化算法

其他

多层感知器



全连接神经网络

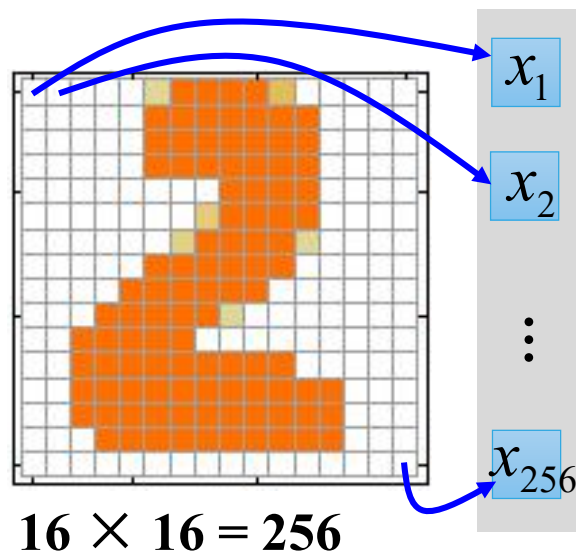


Deep means many hidden layers

全连接神经网络：举例



输入

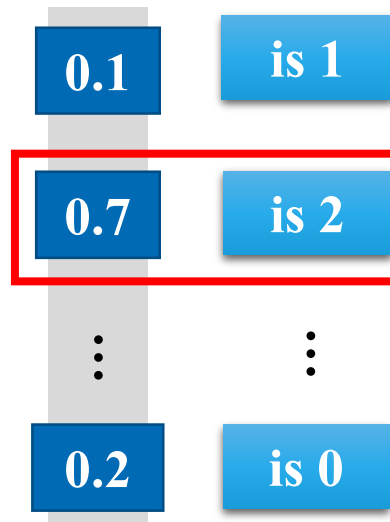


$16 \times 16 = 256$

ink $\rightarrow$ 1

no ink $\rightarrow$ 0

输出



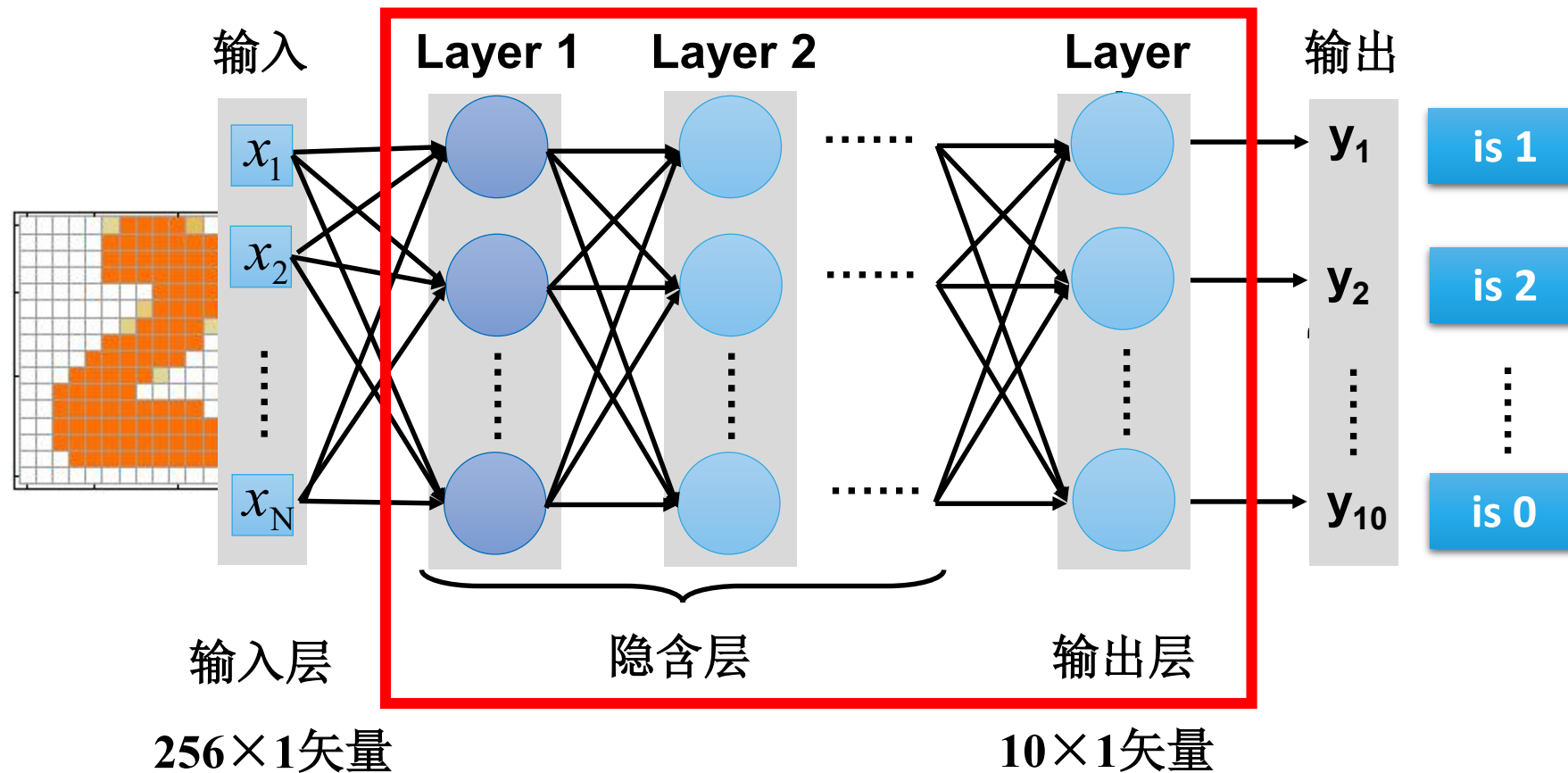
结论：输入
图像为 “2”

输出的各个维度代表了
对应数值的置信度

全连接神经网络：举例

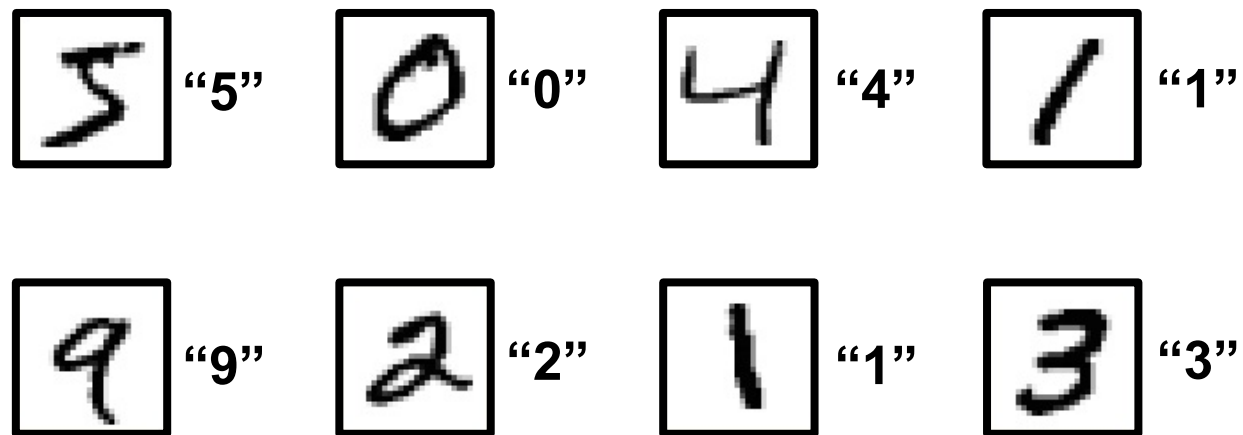


全连接神经网络：举例

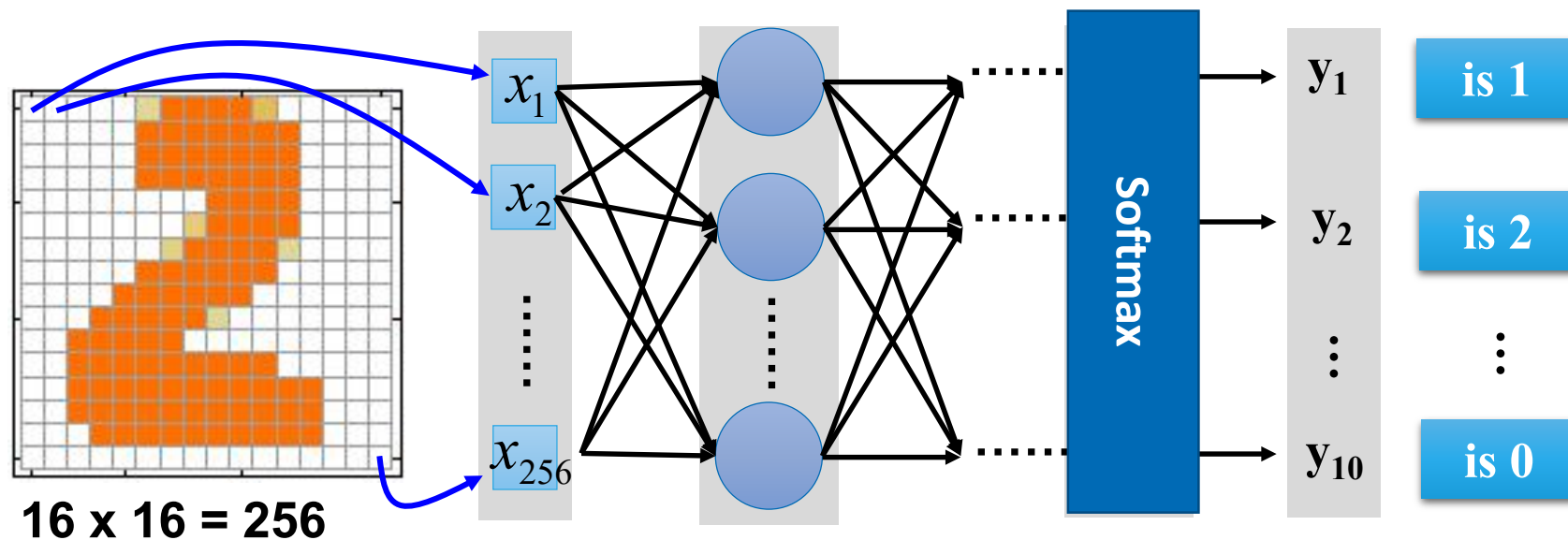


全连接神经网络：举例

训练集：



全连接神经网络：举例



Ink $\rightarrow$ 1

No ink $\rightarrow$ 0

训练目标:

输入:  $\rightarrow$ y_1 有最大值

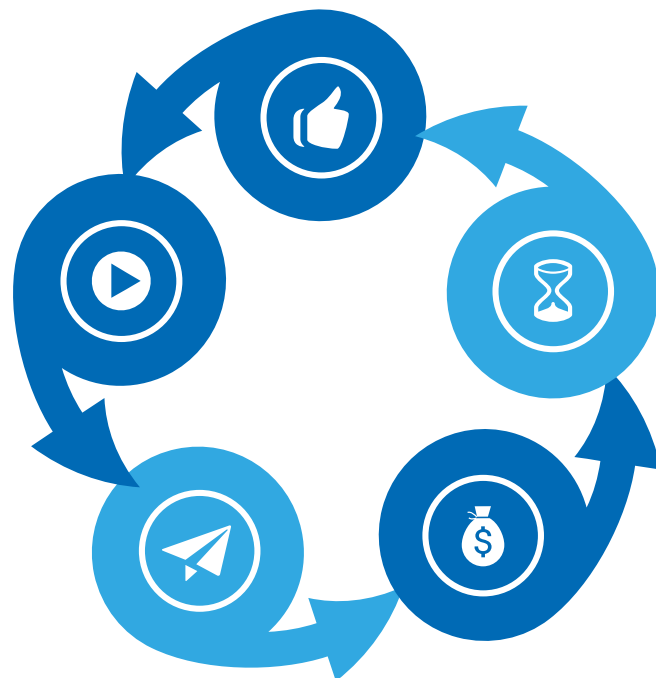
输入:  $\rightarrow$ y_2 有最大值

感知器与线性分类

多层感知器

误差反传与梯度下降

损失函数



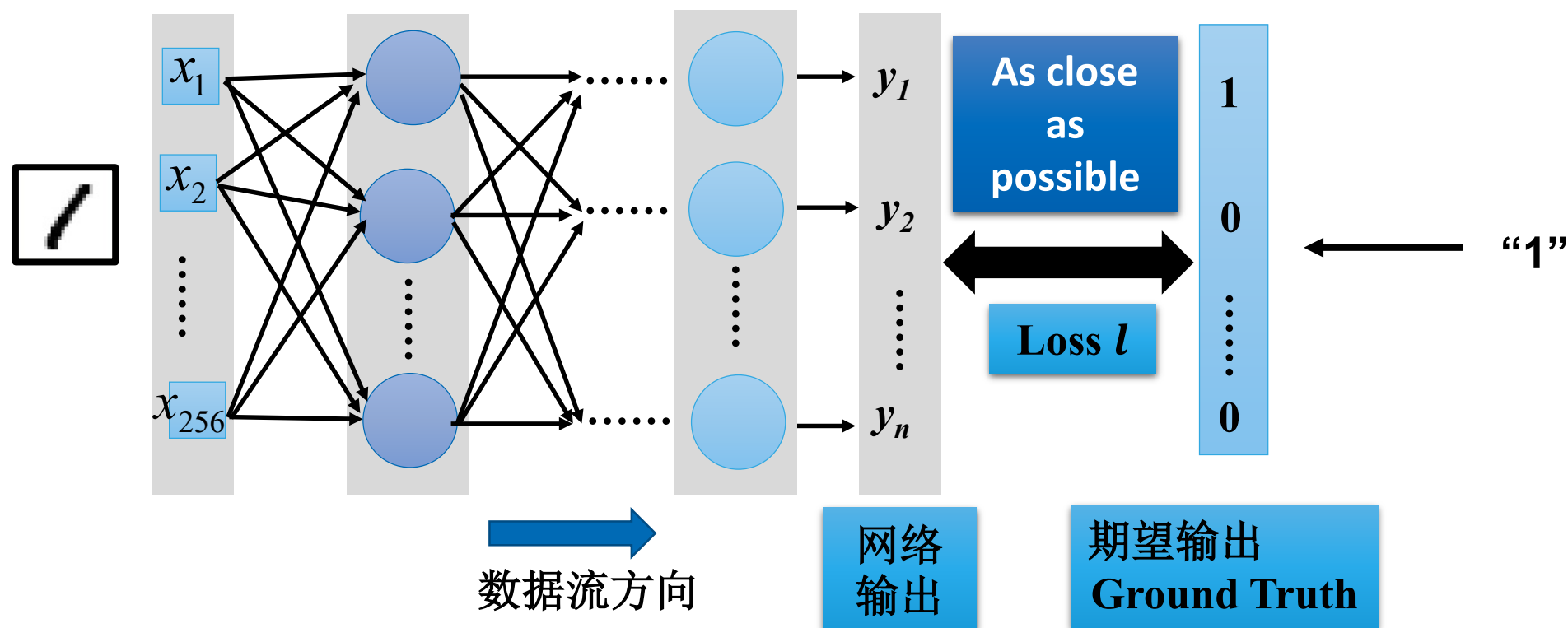
激活函数

优化算法

其他

神经网络学习目标

如何保证网络输出逼近期望输出？



神经元学习

生物神经系统中神经元互连（突触）强度的确定，是个很复杂的生物电—化学过程。

两个问题：

- 强度是如何形成的，形成规律是什么？
- 生物神经网络如何适应环境的变化？

人工神经网络的学习过程就是不断调整网络的连接权值,以获得期望输出的过程。

人工神经网络不断地从数据（样本）中自动获取知识，将这些知识以**网络结构**和**连接权值**的形式存储于网络之中。人工神经网络的学习规则就是权值修正规则。

简单神经元学习规则

无监督学习规则：Hebb学习规则

Hebb学习规则的核心思想：当两神经元同时处于激发状态时，两者间连接权值会被加强，否则被减弱。

例如：Hopfield神经网络、自组织映射神经网络、...

有监督学习规则：Delta学习规则

Delta学习规则是一种简单的有监督学习算法，根据神经元的实际输出与期望输出差别来调整连接权值。

例如：感知器学习、BP神经网络学习、...

Hebb神经元学习规则（1949 年）

1949年，加拿大神经心理学家、认知心理生物学之父赫伯，出版了他一生最重要的书《行为的组织：一种神经心理理论》。

将行为科学和神经生理学结合起来，成为现代神经科学的奠基石。

书中提出了以下假设：

- 当细胞 i 的轴突足够近地刺激细胞 k ，并反复或持续地激励它时，某种增长过程或新陈代谢变化就会在它们中间发生，使得作为激励 k 的诸细胞之一 i 细胞的激励效率增大。

Hebb学习规则与“条件反射”机理一致，并且已经得到了神经细胞学说的证实。

Hebb学习规则

若某个突触（连接）前后的两个神经元被彼此相关激活，那么该突触连接增强；

若它们的激活彼此不相关，则该突触连接被减弱或消除。

突触连接
强度改变量 = 学习率 × 前神经元输出 × 后神经元输出

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot y_i \cdot y_j$$

学习率是一个反映学习效率的参数：学习率大，每次学习引起的连接强度改变量就大；反之则小。

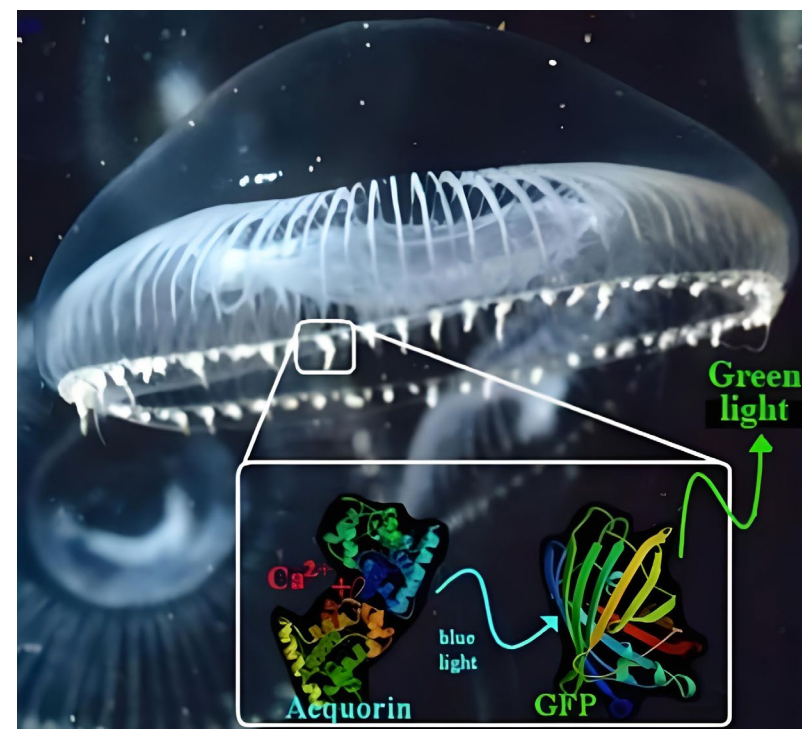
Hebb突触

符合Hebb学习规则的突触被称为Hebb突触。

突触是接受与储存信息的地方。

该观点已被证实，有报道称美国与日本神经科学家的联合研究组，从水母中提取荧光蛋白，植入神经细胞（神经元），然后用激光扫描显微镜观察大脑记忆过程，发现：当记忆事物时，神经元的突触变大，失去记忆时突触变小。

突触权值的调整（改变）相当于突触变大或变小，调整量与神经元活动信号的这种相关性是学习的基础。



Delta学习规则——感知器的学习算法

一个神经元构成的感知器，可完成两类模式的线性划分。

推广到多个神经元的感知器可完成多类分类。

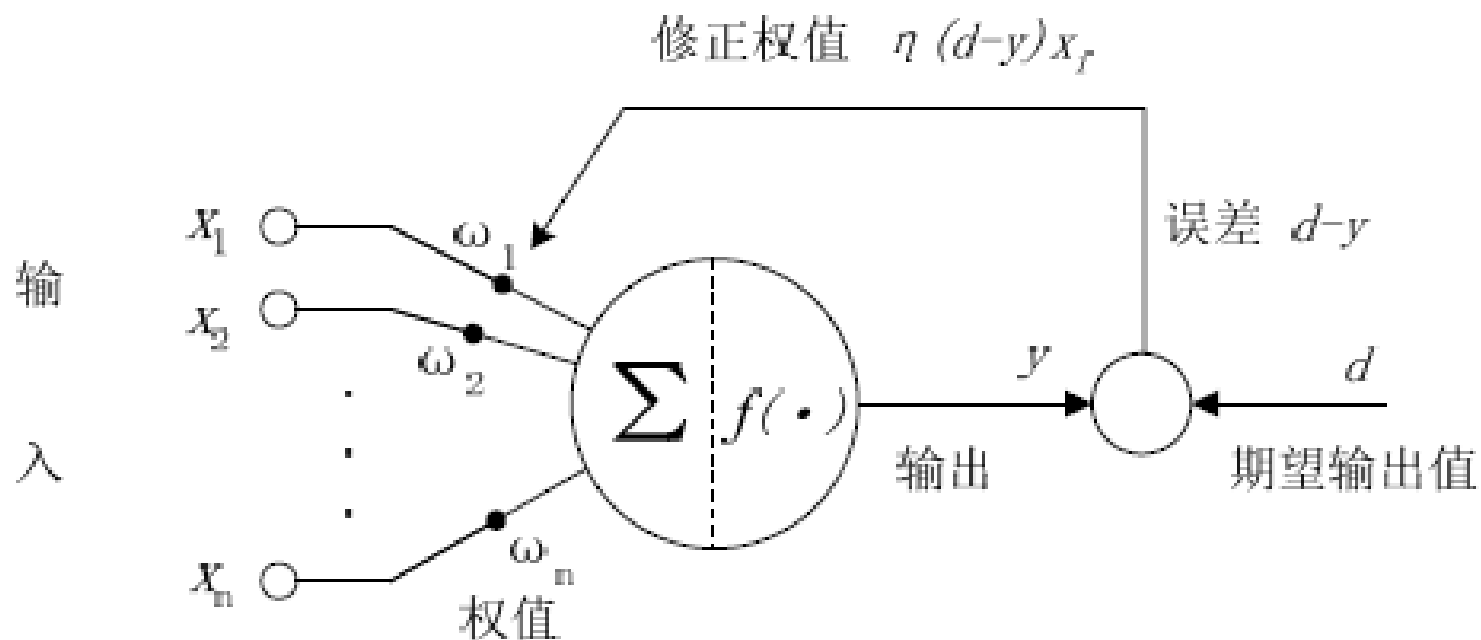
罗森勃拉特给出了感知器的一种学习(训练)算法，并证明该算法能得到稳定结果(收敛性)。

感知器收敛定理:

- 通过训练（学习过程），找到神经元的各个连接权值应该是多少，从而能实现类的自适应划分。

Delta学习规则——感知器的学习算法

$$w_{ij}(t+1) = w_{ij}(t) + \eta [d_j - y_j(t)] x_i(t)$$



一个神经元的感知器的学习原理

神经元的输出 y 与期望值 d 的差 $(d-y)$ 被用来修改连接强度(权值), 以便减少误差。
罗森勃拉特严格证明了这个算法的可行性。

Delta学习规则——感知器的学习步骤

神经元的MP模型:

- 神经元接收到多个输入，分别乘以权值，求和;
- 再由阈值函数(符号函数)产生输出(+1 或-1)。

神经元输出与期望分类结果（目标值：+1 表示一个类，-1 表示另一个类）比较，得到它们之间的误差信号；

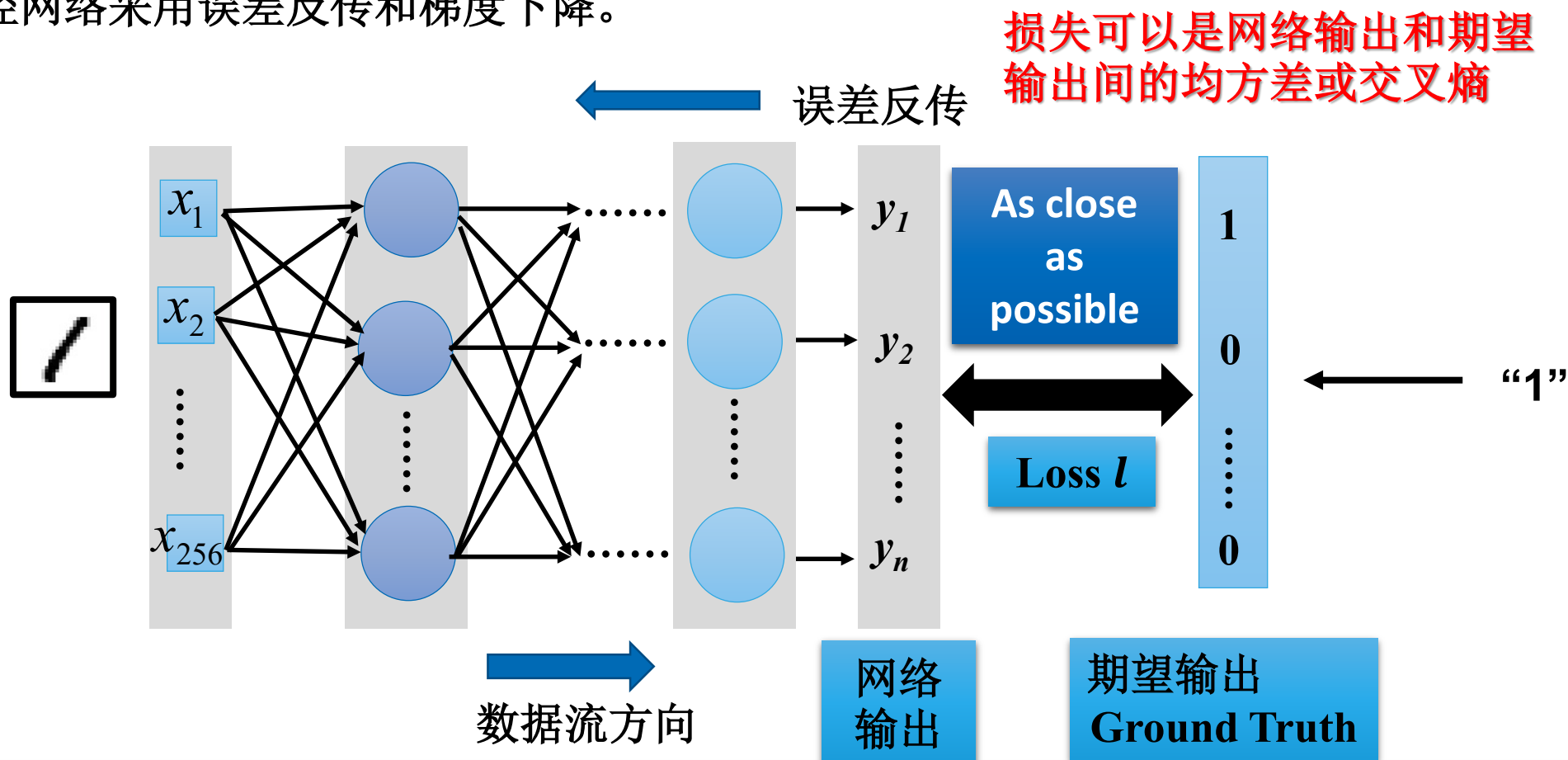
用误差信号修正权值，以减少误差。

如此反复，直到误差足够的小，达到收敛。

神经元能从所提供的训练样本中发现规律。

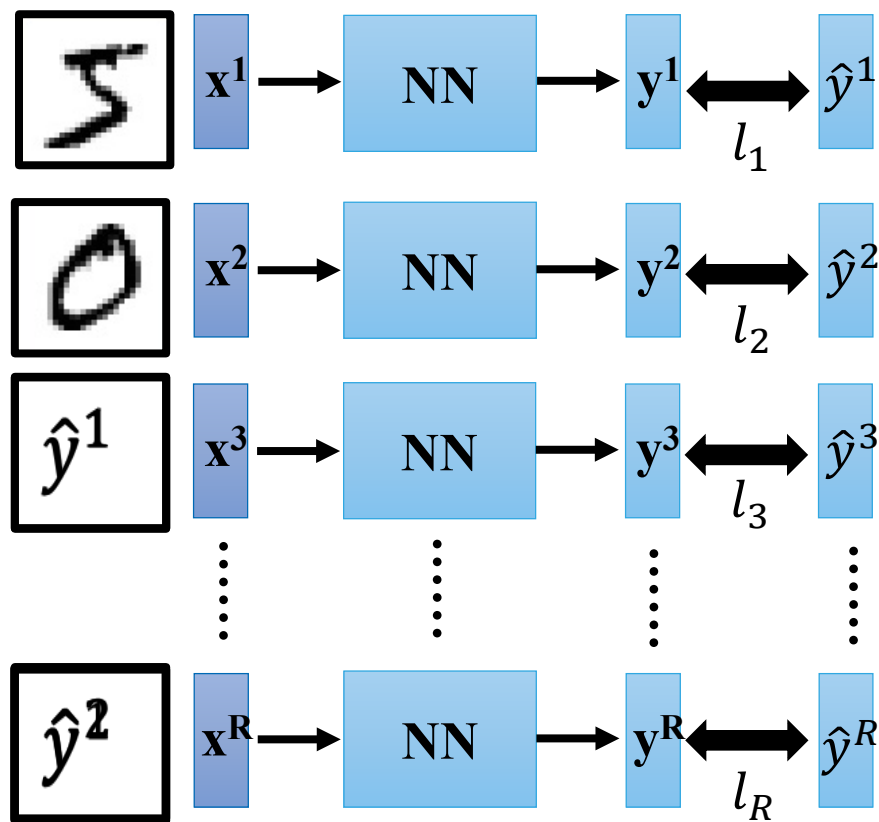
误差反传与梯度下降

对于多层神经网络采用误差反传和梯度下降。



误差反传与梯度下降

对于所有的训练数据…



总损失:

$$L = \sum_{r=1}^R l_r$$

尽可能地小

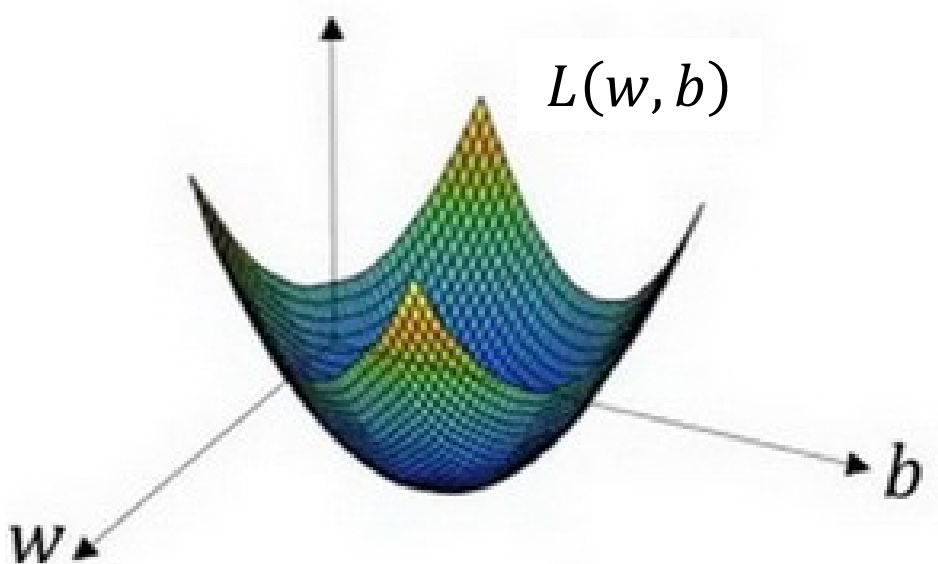
选择合适的损失函数

调整网络参数（权重）使得
总损失最小

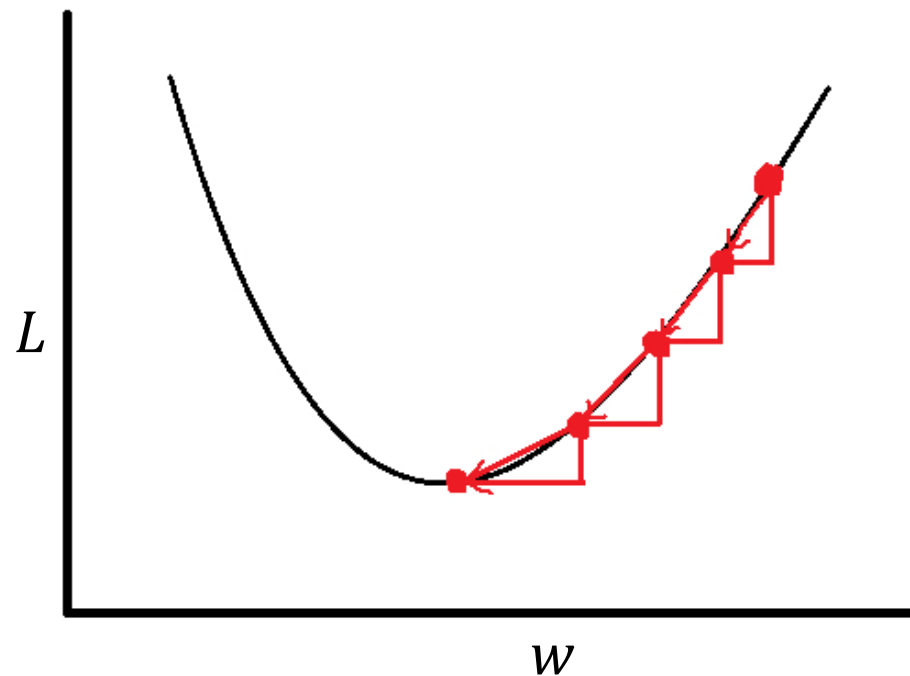
误差反传与梯度下降

总损失: $L(w, b)$

目标: $w, b = \arg \min_{w, b} L(w, b)$



沿着损失函数梯度下降最大的方向寻找



梯度下降——数学基础知识

导数 Derivation

设函数 $y = f(x)$ 在 $U(x_0, \delta)$ 内有定义,

如果极限 $\lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$ 存在,

则称这个极限值为 $y = f(x)$ 在点 x_0 处的导数。

记为 $y'|_{x=x_0}$

导数的意义:

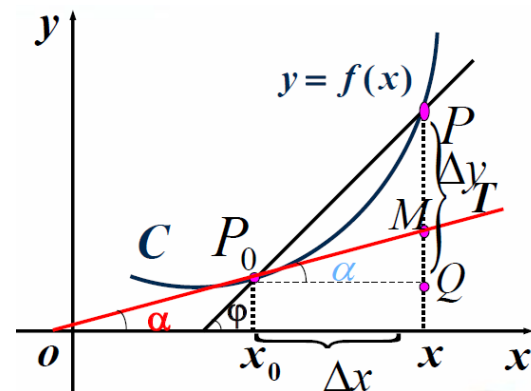
1、变化率、变化速度

反映变量 y 在点 x_0 处的变化率。

或是 y 随自变量 x 变化而变化的快慢程度。

如: 速度 $v = \Delta d / \Delta t$, 加速度 $a = \Delta v / \Delta t$

2、导数的几何意义——曲线上某一点处的切线斜率



$$\text{切线斜率 } \tan \alpha = \lim_{\varphi \rightarrow \alpha} \tan \varphi = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = f'(x_0)$$

梯度下降——数学基础知识

偏导数 Partial Derivation

二元偏导数定义：设二元函数 $z = f(x, y)$ 在 $P_0(x_0, y_0)$ 的某邻域内有定义，

如果极限 $\lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x, y_0) - f(x_0, y_0)}{\Delta x}$ 存在，

则称此极限值为函数 f 在 P_0 处对于 x 的偏导数，记为 $\left. \frac{\partial f}{\partial x} \right|_{P_0}$ 。

同理有对于 y 的偏导数

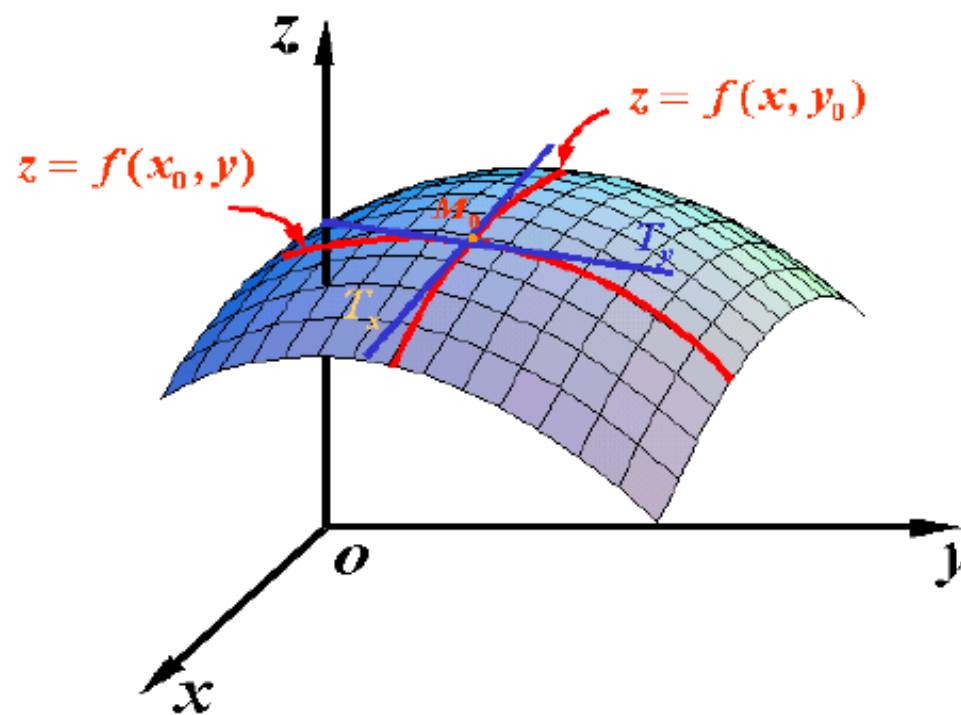
$$\left. \frac{\partial f}{\partial y} \right|_{P_0} = \lim_{\Delta y \rightarrow 0} \frac{f(x_0, y_0 + \Delta y) - f(x_0, y_0)}{\Delta y}$$

梯度下降——数学基础知识

偏导数的几何意义

偏导数 $f'_x(x_0, y_0)$ 就是曲线 $z = f(x, y_0)$ 在 x_0 处的 x 方向切线斜率

偏导数 $f'_y(x_0, y_0)$ 就是曲线 $z = f(x_0, y)$ 在 y_0 处的 y 方向切线斜率



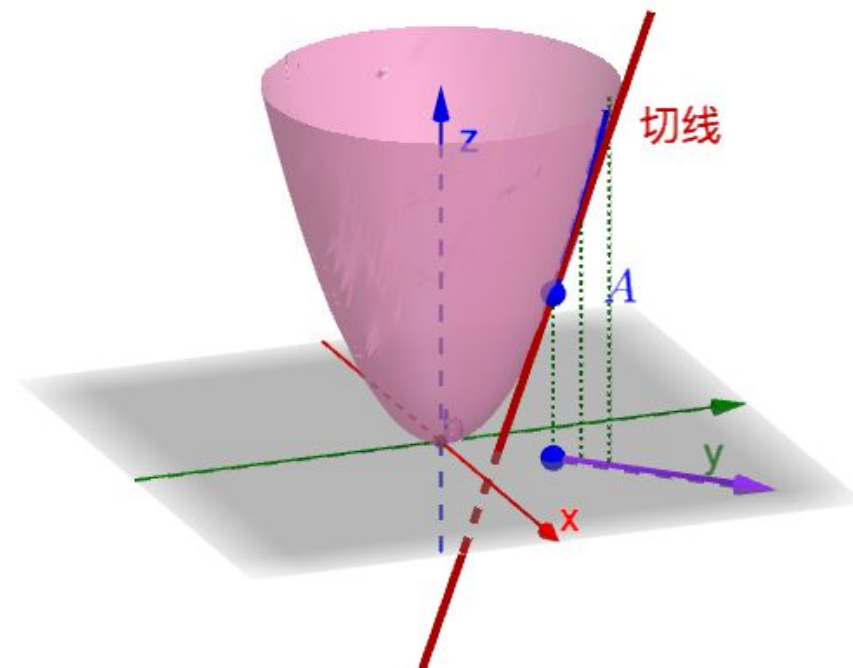
梯度下降——数学基础知识

方向导数 Directional Derivative

设 f 是定义于 R^n 中某区域 D 上的函数，点 $P_0 \in D$ ， l 为一给定的非零向量， P 为一动点，向量 P_0P 与 l 的方向始终一致，如果极限

$$\lim_{\|P_0P\| \rightarrow 0} \frac{f(P) - f(P_0)}{\|P_0P\|}$$

存在，则称此极限为函数 f 在 P_0 处沿 l 方向的方向导数，记为 $\left. \frac{\partial f}{\partial l} \right|_{P_0}$ 。



梯度下降——数学基础知识

梯度 Gradient

显然A点不止一个方向导数，而是360度都有方向导数。

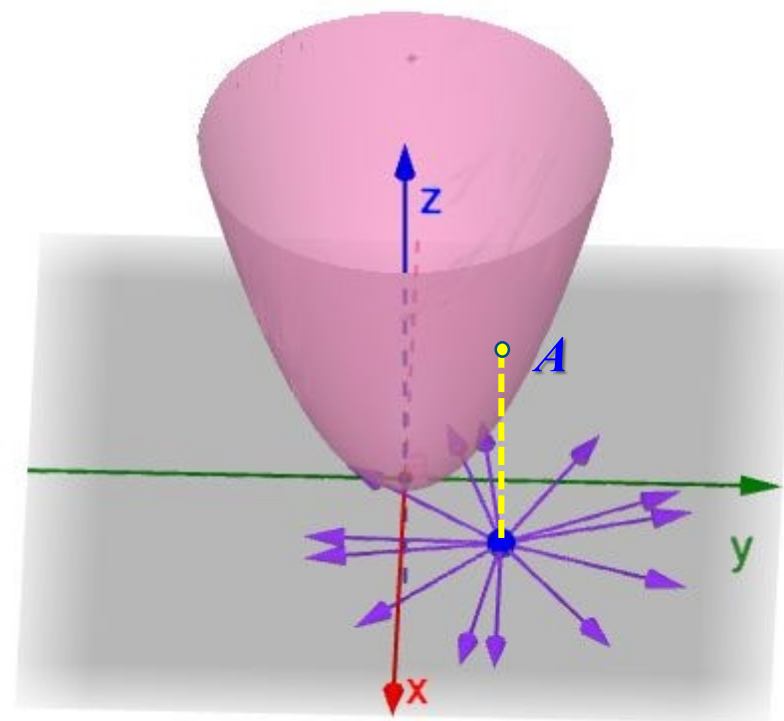
梯度的几何意义：是一个矢量，其方向上的方向导数最大，其大小正好是此最大方向导数。

二元函数梯度定义：设二元函数 $z = f(x, y)$ 在 $P_0(x_0, y_0)$ 点

存在偏导数，该点处梯度为： $\nabla G = \left(\frac{\partial f}{\partial x} \Big|_{P_0}, \frac{\partial f}{\partial y} \Big|_{P_0} \right)$

梯度方向为： $\theta = \tan^{-1} \left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right)$

梯度幅值为： $\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$



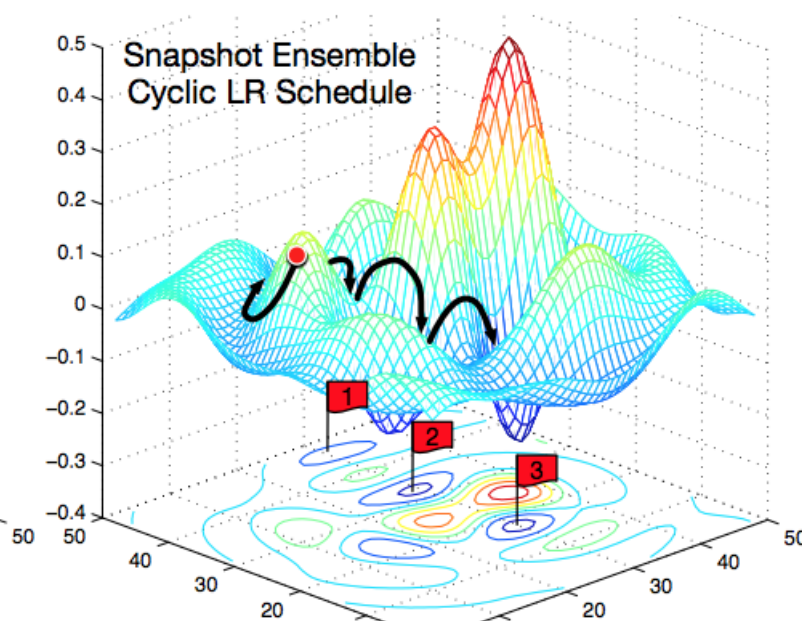
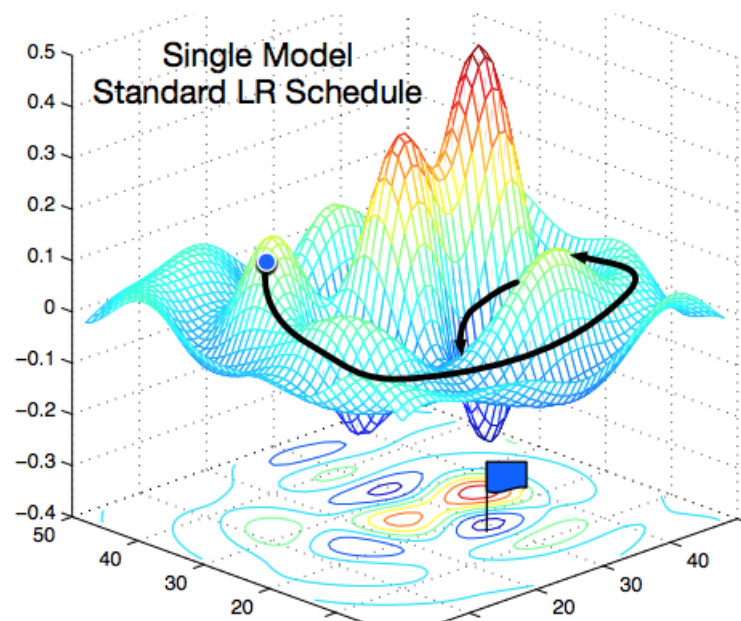
梯度下降——数学基础知识

梯度 Gradient

设 f 是 R_n 中区域 D 上的数量场, 如果 f 在 $P_0 \in D$ 处

可微, 称向量 $\left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \bigg|_{P_0}$ 为 f 在 P_0 处的梯度。

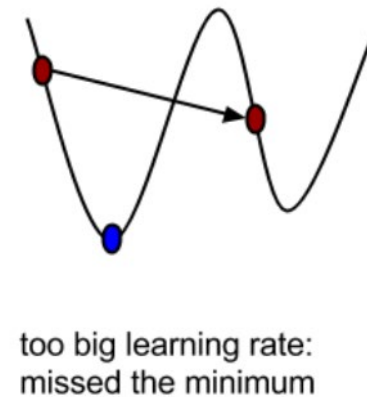
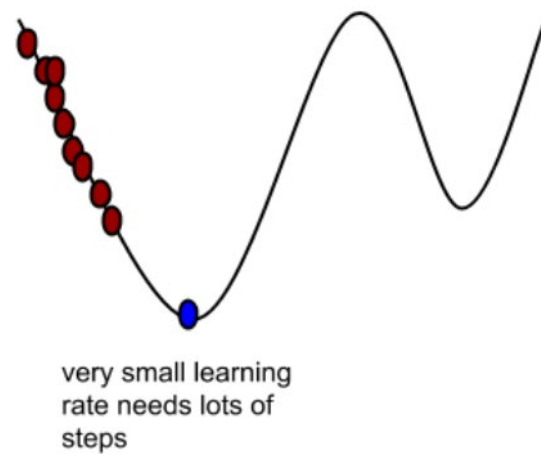
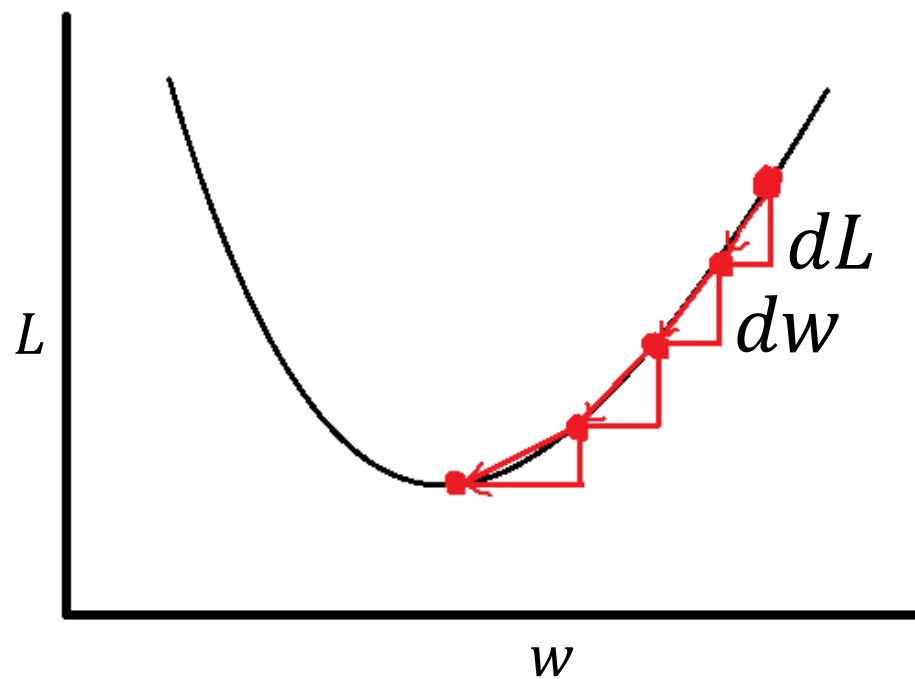
梯度下降



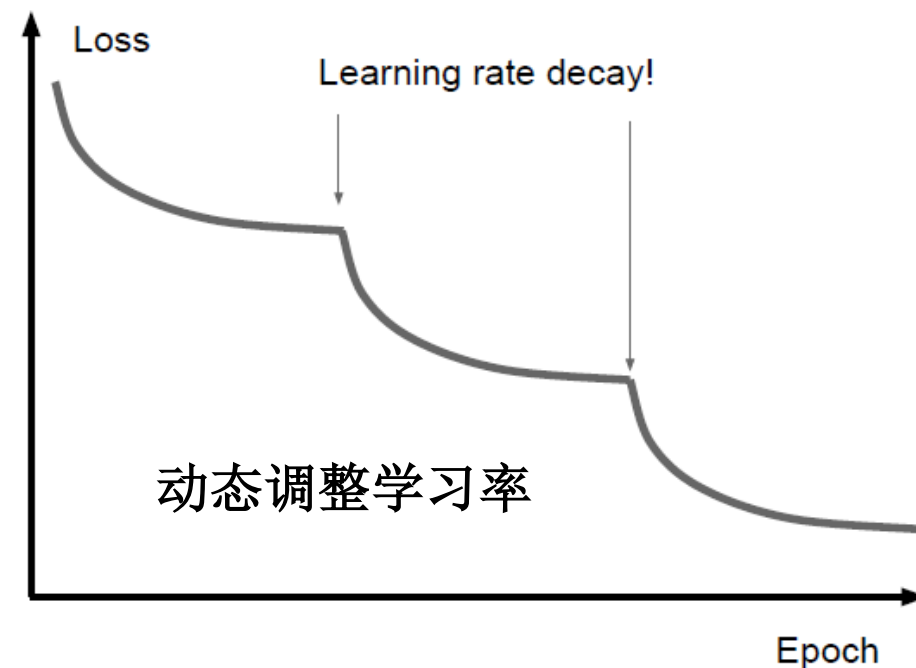
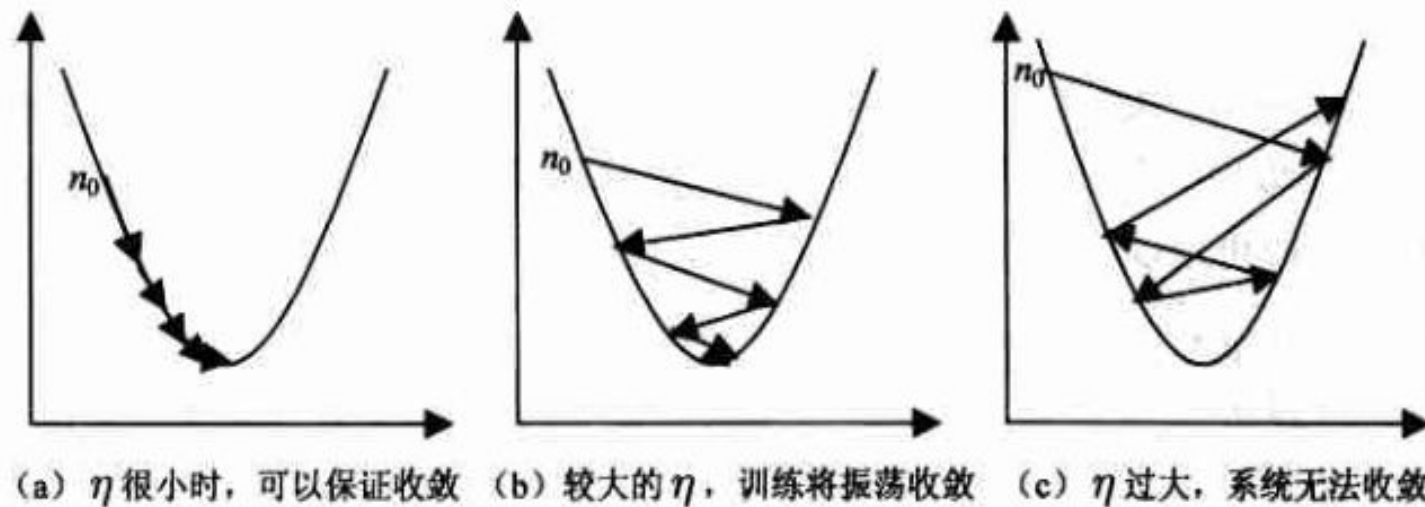
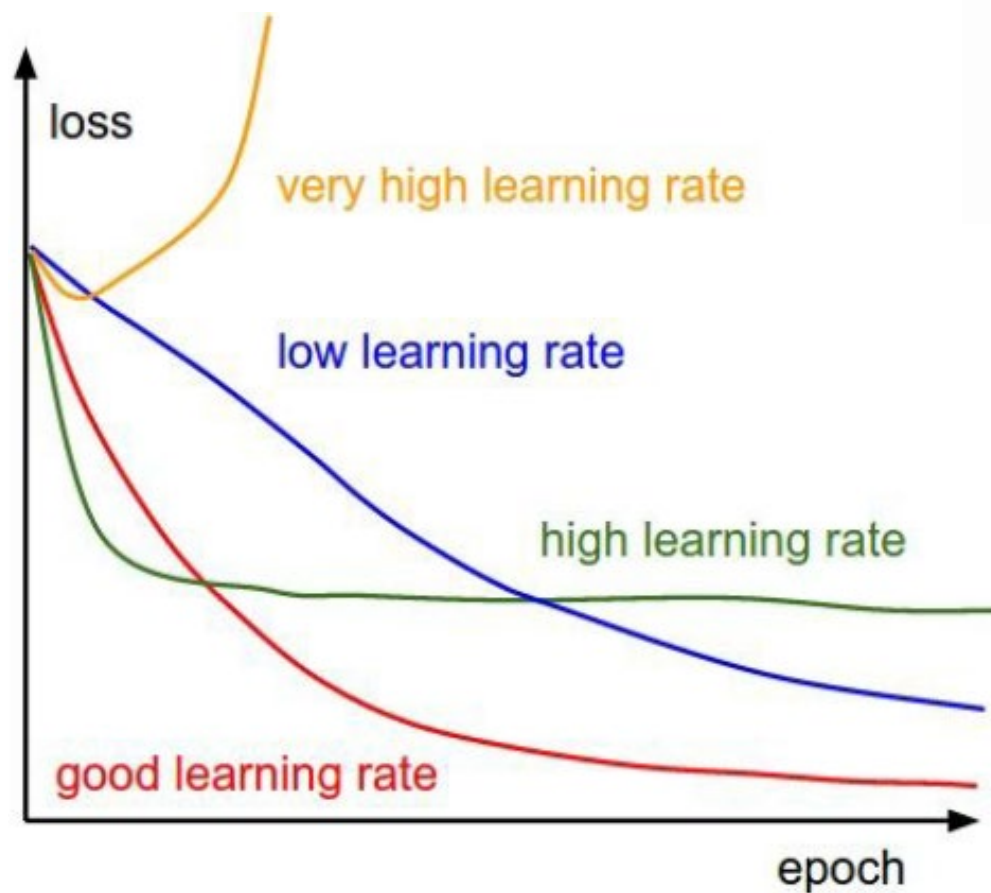
梯度下降与学习率

参数更新: $w \leftarrow w - \eta \frac{\partial L}{\partial w}$

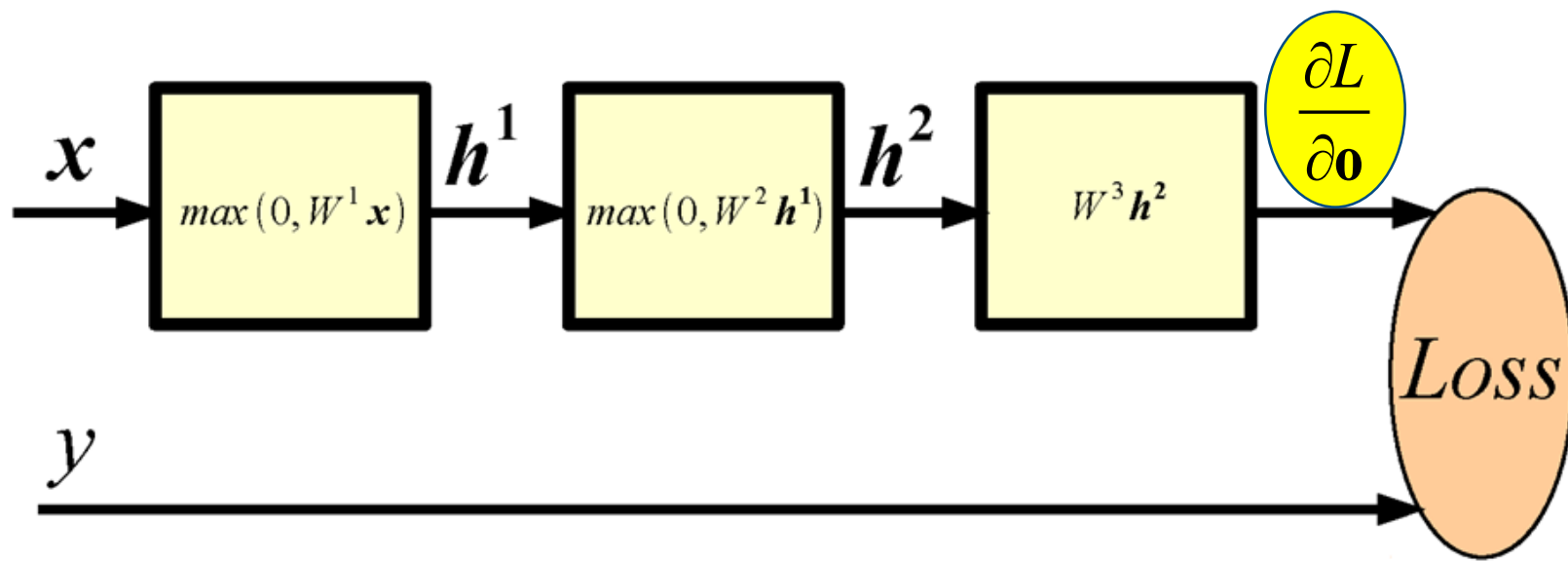
η 为学习率



梯度下降与学习率



误差反传与链式法则



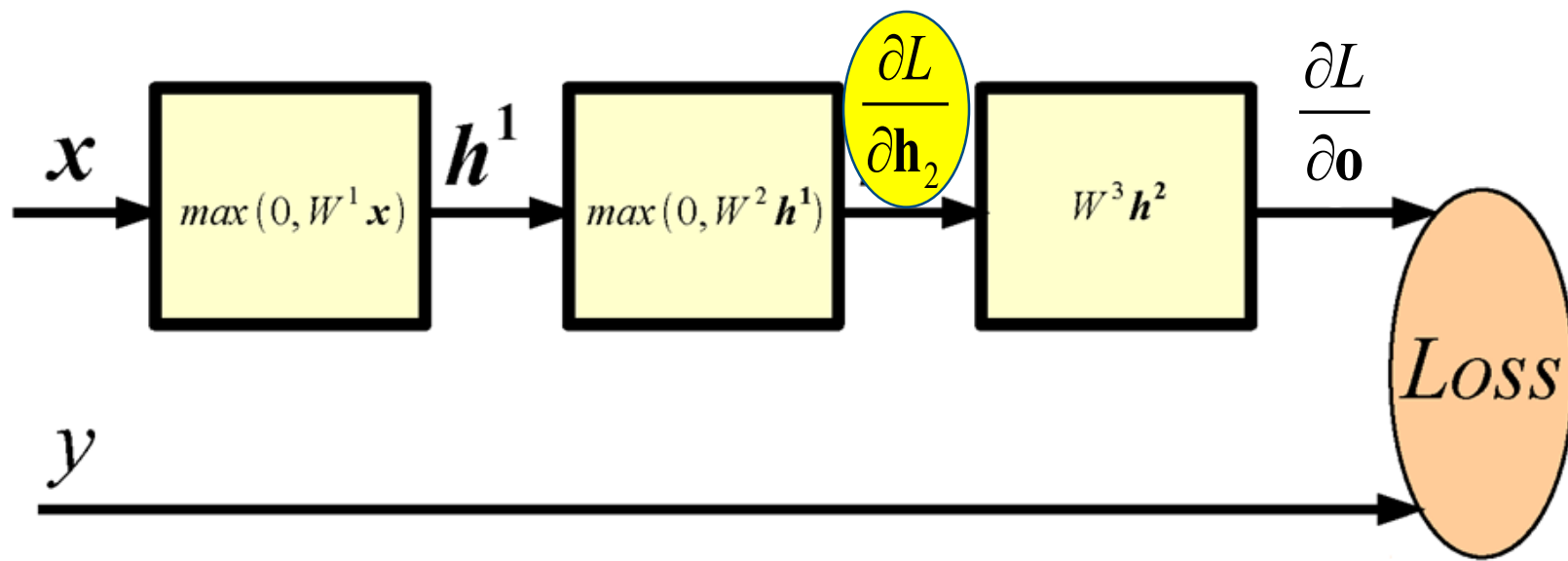
$$\frac{\partial L}{\partial \mathbf{W}_3} = \frac{\partial L}{\partial \mathbf{o}} \frac{\partial \mathbf{o}}{\partial \mathbf{W}_3}$$

$$\frac{\partial L}{\partial \mathbf{h}_2} = \frac{\partial L}{\partial \mathbf{o}} \frac{\partial \mathbf{o}}{\partial \mathbf{h}_2}$$

链式法则 Chain Rule

$$\frac{dy}{dx} = \frac{dy}{du_1} \cdot \frac{du_1}{du_2} \cdot \dots \cdot \frac{du_{m-1}}{du_m} \cdot \frac{du_m}{dx}$$

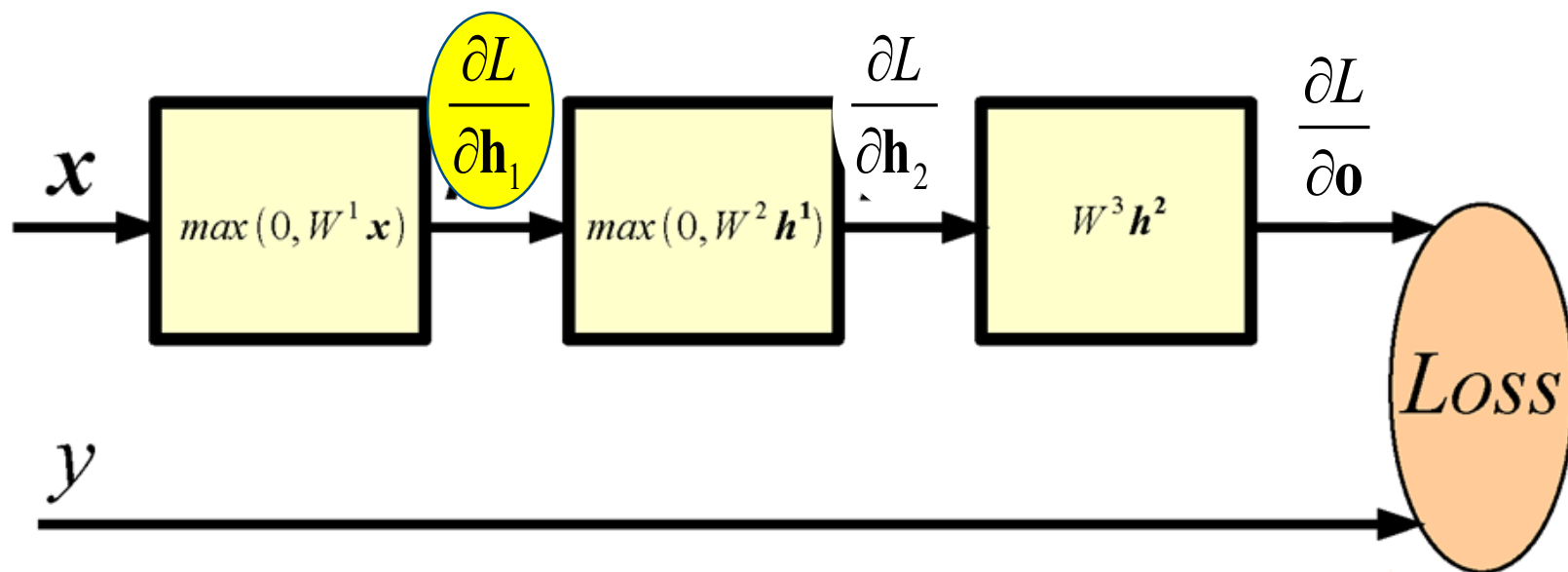
误差反传与链式法则



$$\frac{\partial L}{\partial \mathbf{W}_2} = \frac{\partial L}{\partial \mathbf{h}_2} \frac{\partial \mathbf{h}_2}{\partial \mathbf{W}_2} = \frac{\partial L}{\partial \mathbf{o}} \frac{\partial \mathbf{o}}{\partial \mathbf{h}_2} \frac{\partial \mathbf{h}_2}{\partial \mathbf{W}_2}$$

$$\frac{\partial L}{\partial \mathbf{h}_1} = \frac{\partial L}{\partial \mathbf{h}_2} \frac{\partial \mathbf{h}_2}{\partial \mathbf{h}_1} = \frac{\partial L}{\partial \mathbf{o}} \frac{\partial \mathbf{o}}{\partial \mathbf{h}_2} \frac{\partial \mathbf{h}_2}{\partial \mathbf{h}_1}$$

误差反传与链式法则



$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial h_1} \frac{\partial h_1}{\partial W_1} = \frac{\partial L}{\partial o} \frac{\partial o}{\partial h_2} \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W_1}$$

误差反传与链式法则

链式法则 Chain Rule

$$y = \ln \left(\sin \left(x^3 \right)^2 \right)$$

$$\begin{aligned} \begin{cases} u_1 = x^3 \\ u_2 = \sin(u_1) \\ u_3 = u_2^2 \\ y = \ln(u_3) \end{cases} & \Rightarrow \begin{cases} \frac{du_1}{dx} = \frac{d}{dx} x^3 = 3x^2 \\ \frac{du_2}{du_1} = \frac{d}{du_1} \sin(u_1) = \cos(u_1) \\ \frac{du_3}{du_2} = \frac{d}{du_2} u_2^2 = 2u_2 \\ \frac{dy}{du_3} = \frac{d}{du_3} \ln(u_3) = \frac{1}{u_3} \end{cases} \\ & \Rightarrow \begin{aligned} \frac{dy}{dx} &= \frac{dy}{du_3} \frac{du_3}{du_2} \frac{du_2}{du_1} \frac{du_1}{dx} \\ &= \frac{1}{u_3} \cdot 2u_2 \cdot \cos(u_1) \cdot 3x^2 \\ &= \frac{1}{u_2^2} \cdot 2u_2 \cdot \cos(x^3) \cdot 3x^2 = \frac{6x^2 \cdot \cos(x^3)}{\sin(x^3)} \end{aligned} \end{aligned}$$

误差反传与链式法则

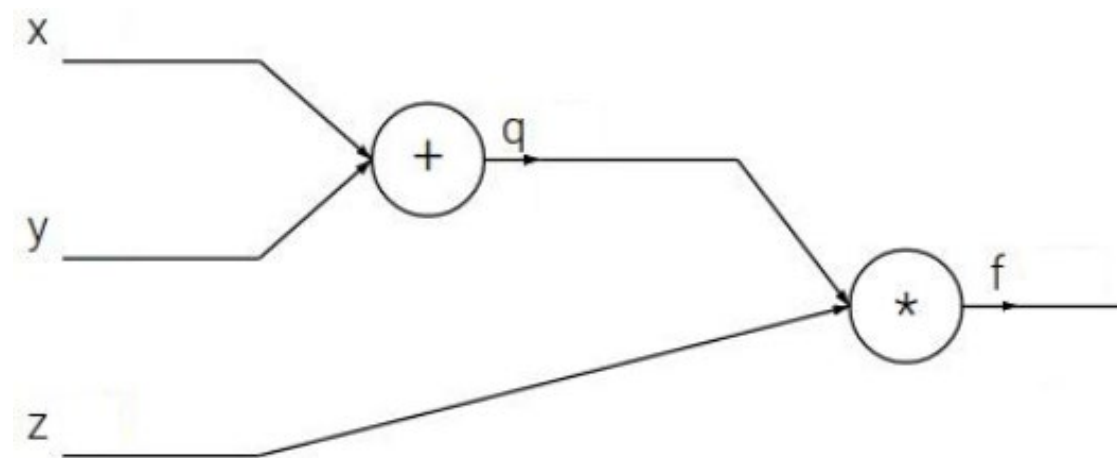
$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

两种梯度计算方法：

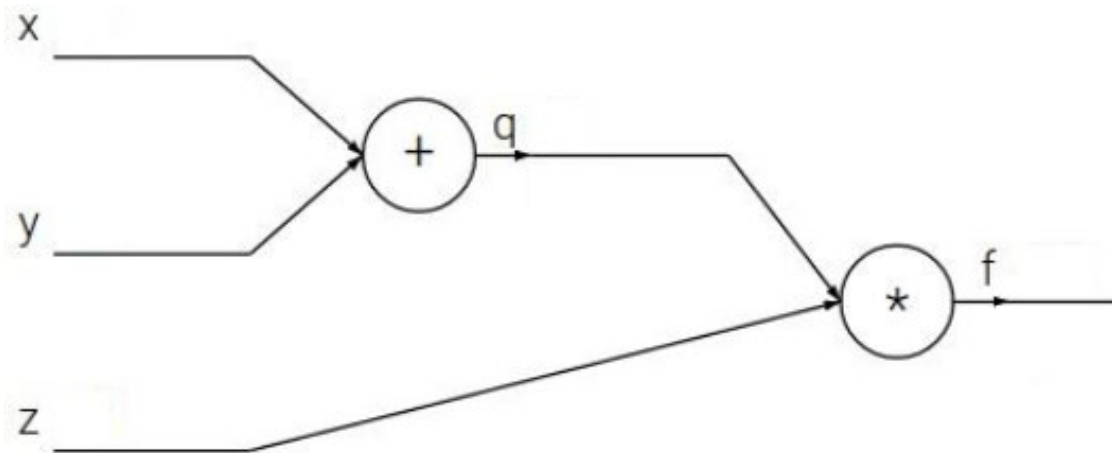
- 1) 数值梯度 (numerical gradient)：近邻数值之差
计算慢，只是近似值，但容易
- 2) 解析梯度 (analytic gradient)：求编导
计算快，精确值，但是容易出错

实际应用中，推导解析梯度，采用数值梯度检验

举例



误差反传与链式法则



$$q(x, y) = x + y$$

$$f(x, y, z) = q(x, y) \cdot z$$

求导:

$$\frac{\partial q}{\partial x} = 1, \quad \frac{\partial q}{\partial y} = 1$$

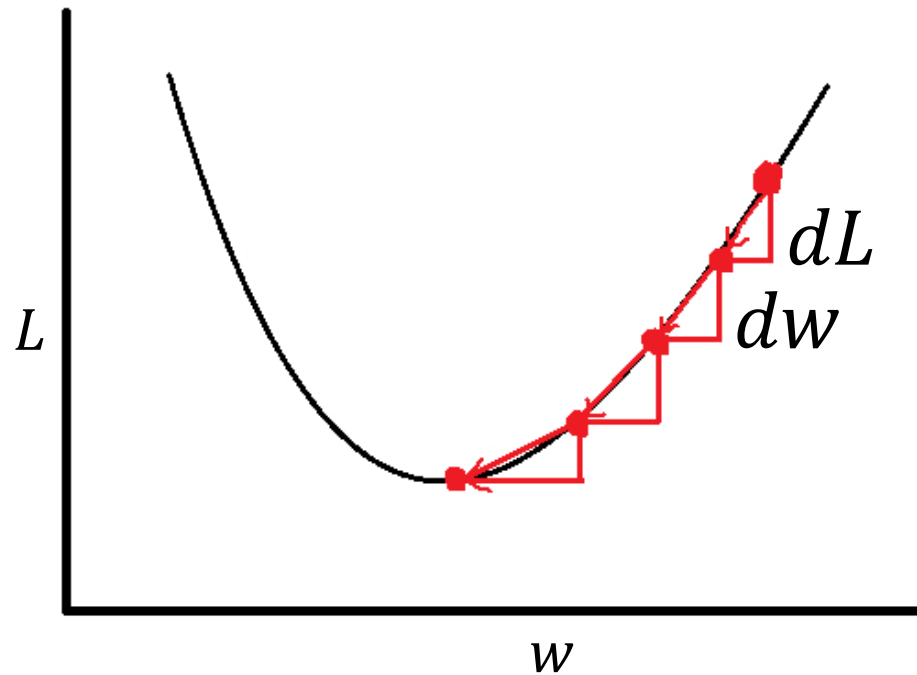
$$\frac{\partial f}{\partial z} = q(x, y) = x + y$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x} = z \frac{\partial q}{\partial x} = z$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y} = z \frac{\partial q}{\partial y} = z$$

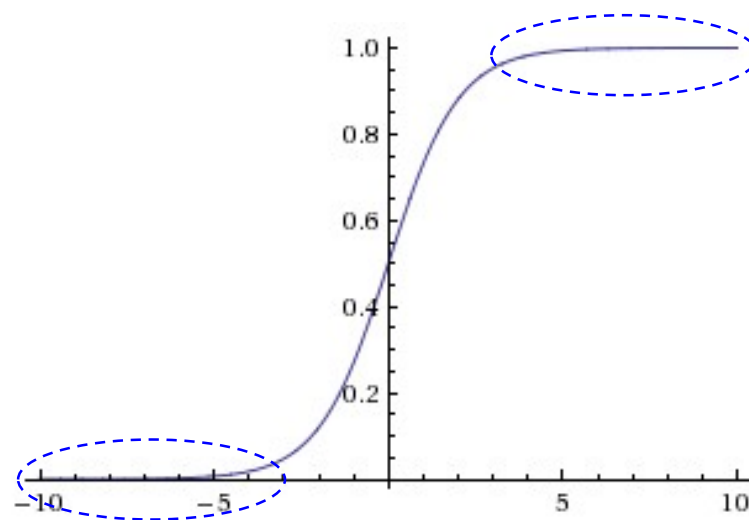
梯度消失问题

参数更新: $w \leftarrow w - \eta \frac{\partial L}{\partial w}$



梯度消失问题:

解决办法: 选择合适的损失函数!

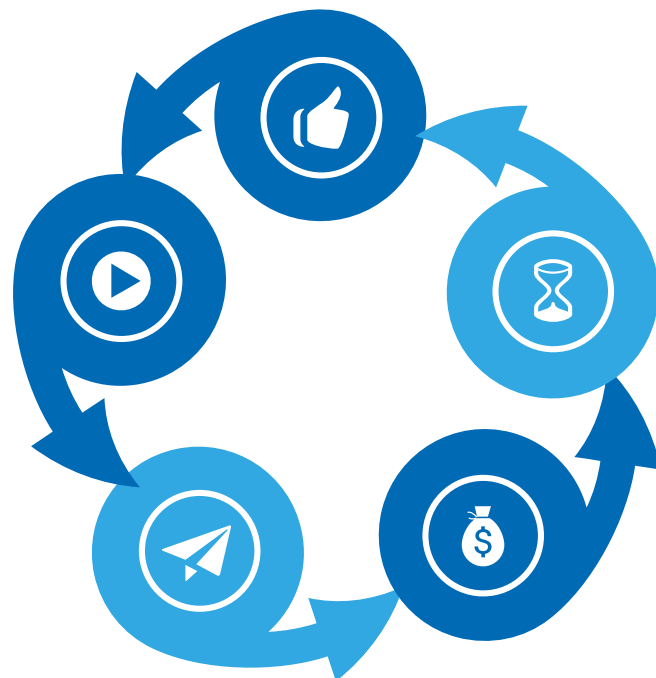


感知器与线性分类

多层感知器

误差反传与梯度下降

损失函数



激活函数

优化算法

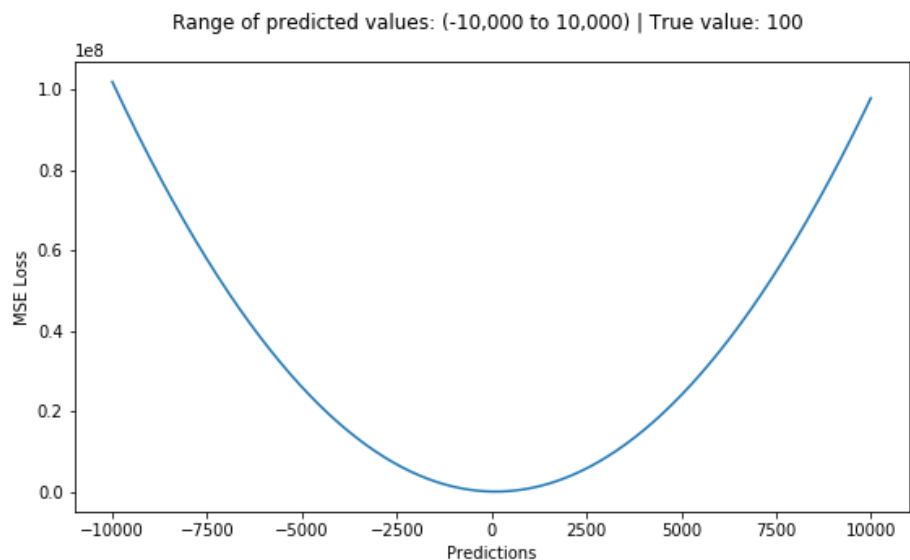
其他

误差→损失函数

均方误差

MSE(Mean Squared Error)

$$loss = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

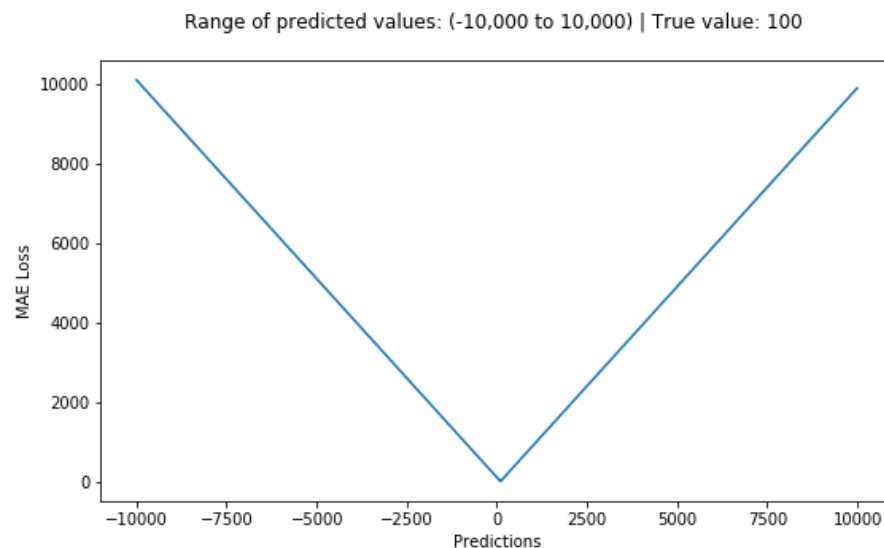


y_i 为网络输出
 $\hat{y}_i$ 为期望输出

平均绝对误差

MAE (Mean Absolute Error)

$$loss = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

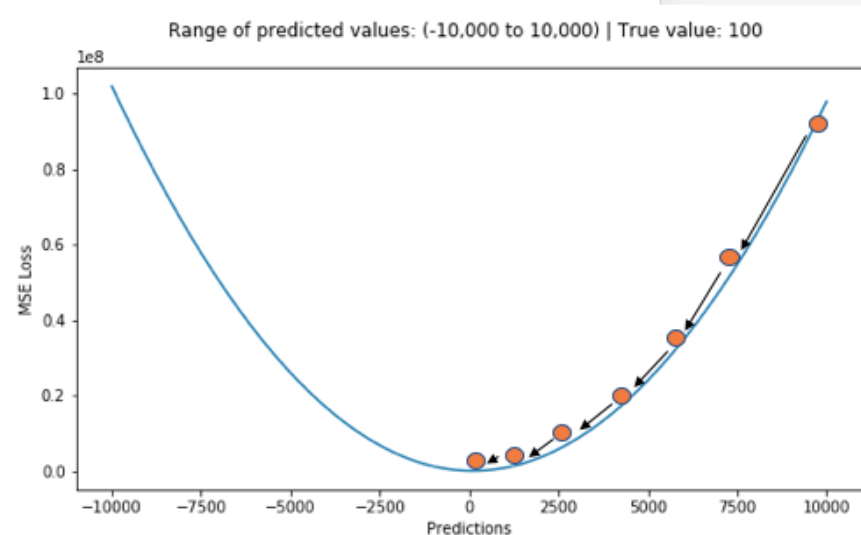
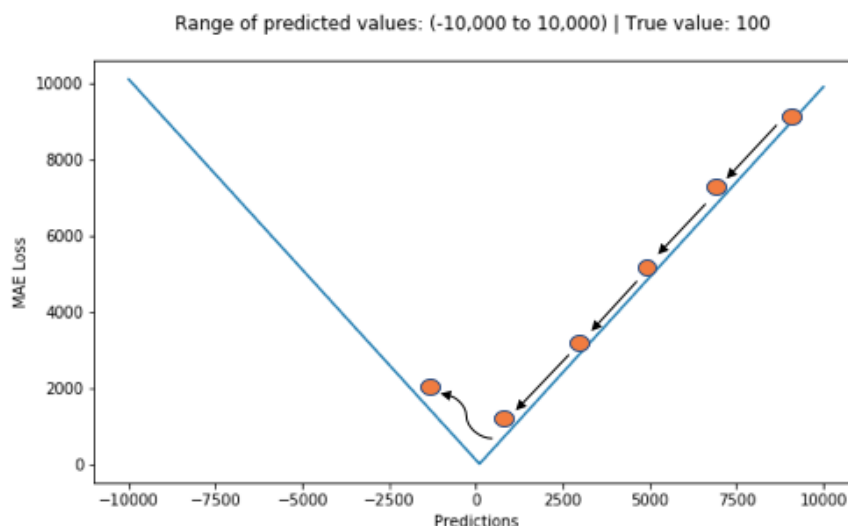


误差→损失函数

MSE vs. MAE (L2 loss vs L1 loss)

当训练集存在偏差时，采用MAE具有较好的鲁棒性

MAE的主要问题：当损失值很小时，其梯度值仍较大，不利于网络逼近最小值



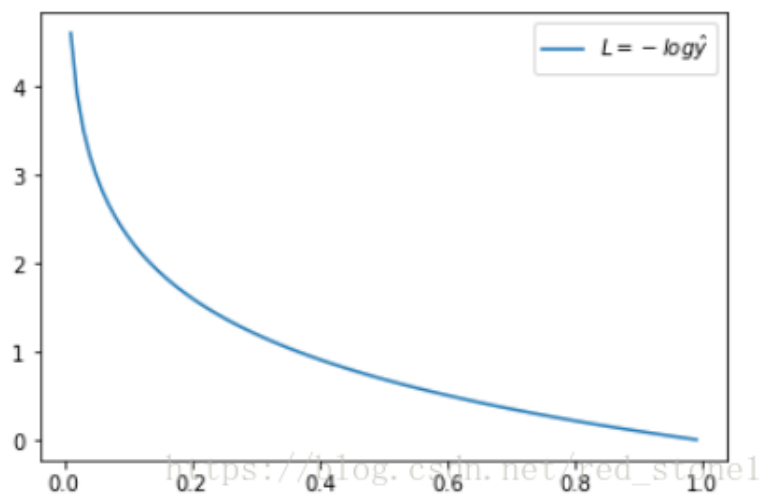
误差→损失函数

交叉熵误差 Cross Entropy Loss

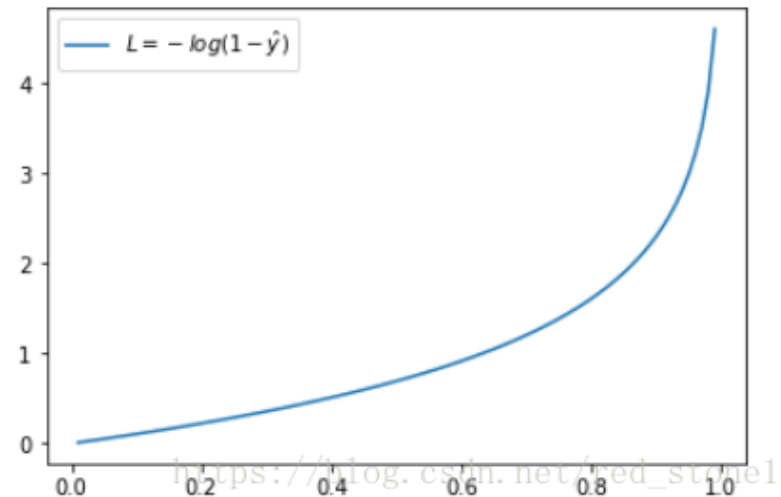
$$loss = -\frac{1}{N} \sum_{i=1}^N [\hat{y}_i \log y_i - (1 - \hat{y}_i) \log (1 - y_i)]$$

y_i 为网络输出, $\hat{y}_i$ 为期望输出

当 $\hat{y} = 1$ 时,
 $loss = -\log y$



当 $\hat{y} = 0$ 时,
 $loss = -\log (1 - y)$



误差→损失函数

MSE vs. 交叉熵

$$\begin{cases} L = \frac{1}{2}(y_i - \hat{y}_i)^2 \\ y_i = f(wx + b) \end{cases}$$

求导数:

$$\frac{\partial L}{\partial w} = (y_i - \hat{y}_i) x \cdot f'(wx + b) = c_1 \cdot f'$$

$$\frac{\partial L}{\partial b} = (y_i - \hat{y}_i) \cdot f'(wx + b) = c_2 \cdot f'$$

神经网络训练时, 采用梯度下降算法
更新权重 w 和偏置 b

更新权重 w 和偏置 b

$$w' = w - \eta \cdot \frac{\partial L}{\partial w} = w - \eta \cdot c_1 \cdot f'$$

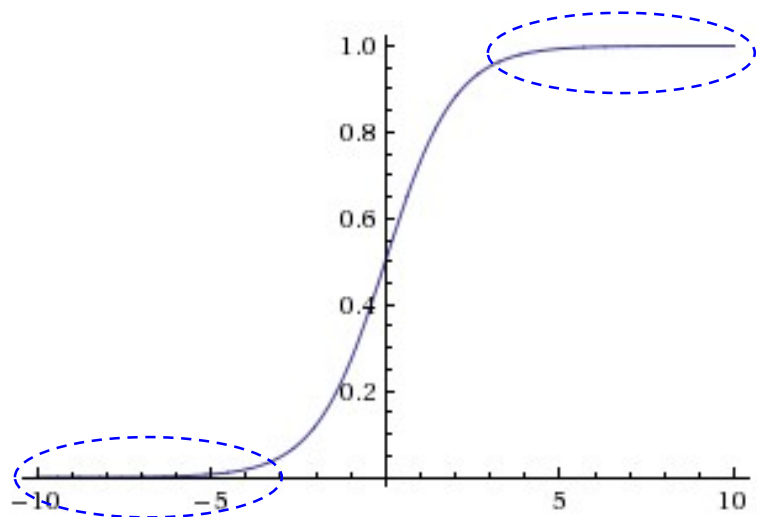
$$b' = b - \eta \cdot \frac{\partial L}{\partial b} = b - \eta \cdot c_2 \cdot f'$$

误差→损失函数

MSE vs. 交叉熵

MSE存在的问题:

当激活函数 f 为sigmoid函数时, 两端梯度 (导数) 趋于0, 导致无法有效学习。



$$L = -\frac{1}{N} \sum_{i=1}^N [\hat{y}_i \log y_i - (1 - \hat{y}_i) \log (1 - y_i)]$$

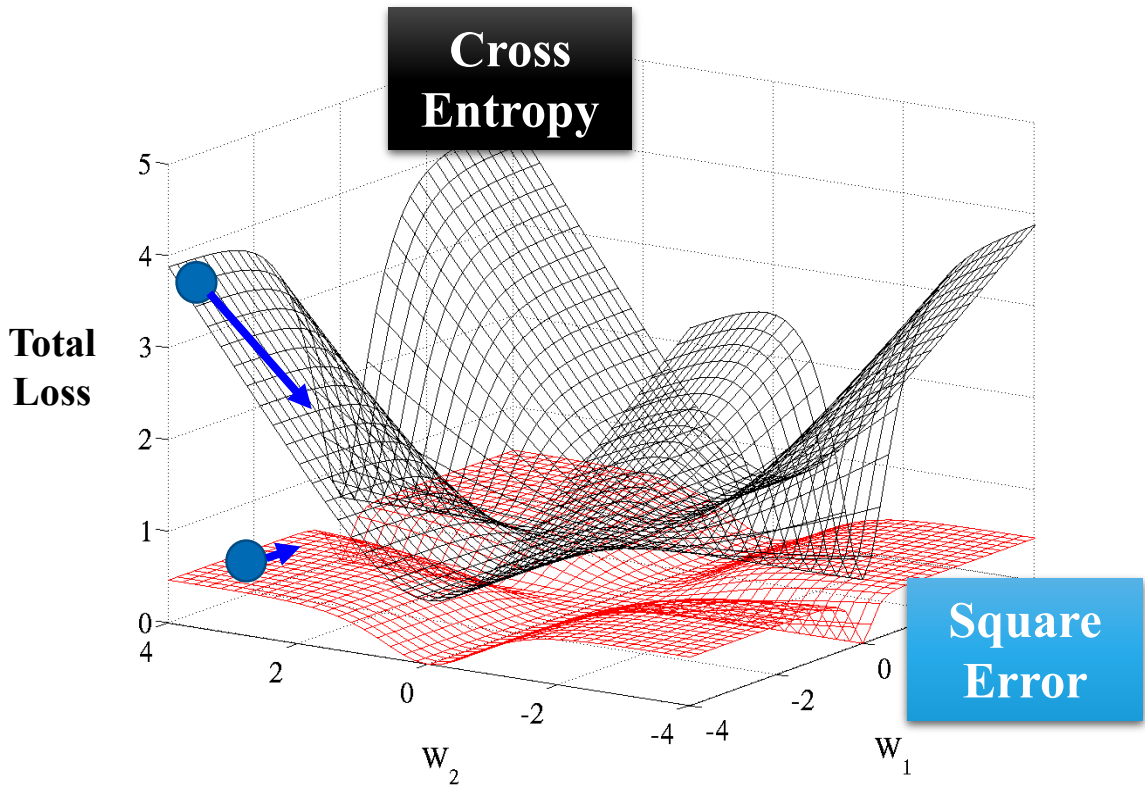
求导数:

$$\frac{\partial L}{\partial w} = -\frac{1}{N} \sum_{i=1}^N x_j (y_i - \hat{y}_i)$$

$$\frac{\partial L}{\partial b} = -\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)$$

无激活函数 f 导数项

误差→损失函数



若输出层激活函数选择Sigmoid或Softmax，
建议采用交叉熵

Task	Output layer activation	Loss function	Expression of loss function
Regression	Linear	Mean squared error (MSE)	$L(\mathbf{y}, \mathbf{t}) = \ \mathbf{y} - \mathbf{t}\ _2^2$
Binary classification	Sigmoid	Binary cross-entropy	$L(\mathbf{y}, \mathbf{t}) = -\sum_{i=1}^n t_i \log y_i - \sum_{i=1}^n (1 - t_i) \log(1 - y_i)$
Multi-class classification	Softmax	Categorical cross-entropy	$L(\mathbf{y}, \mathbf{t}) = -\sum_{i=1}^n t_i \log y_i$

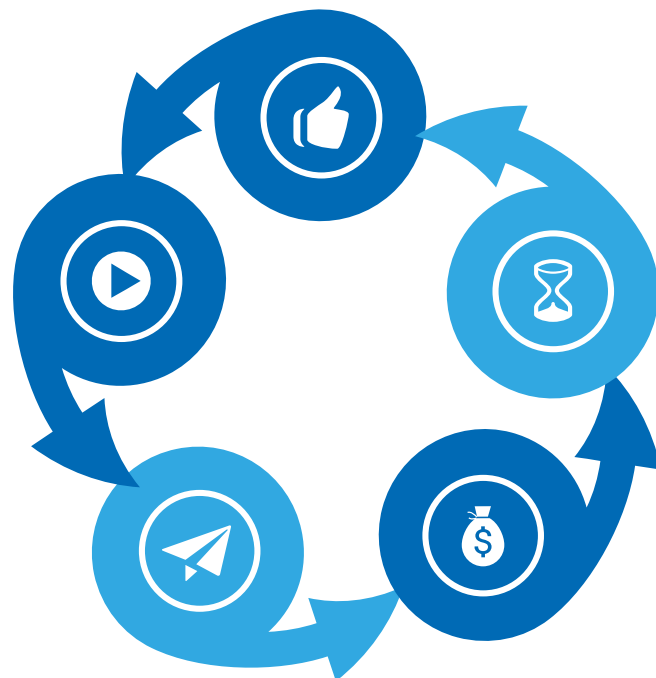
X. Glorot, X. and Y. Bengio, Understanding the Difficulty of Training Deep Feedforward Neural Networks. Journal of Machine Learning Research, 2010. 9: p. 249-256.

感知器与线性分类

多层感知器

误差反传与梯度下降

损失函数



激活函数

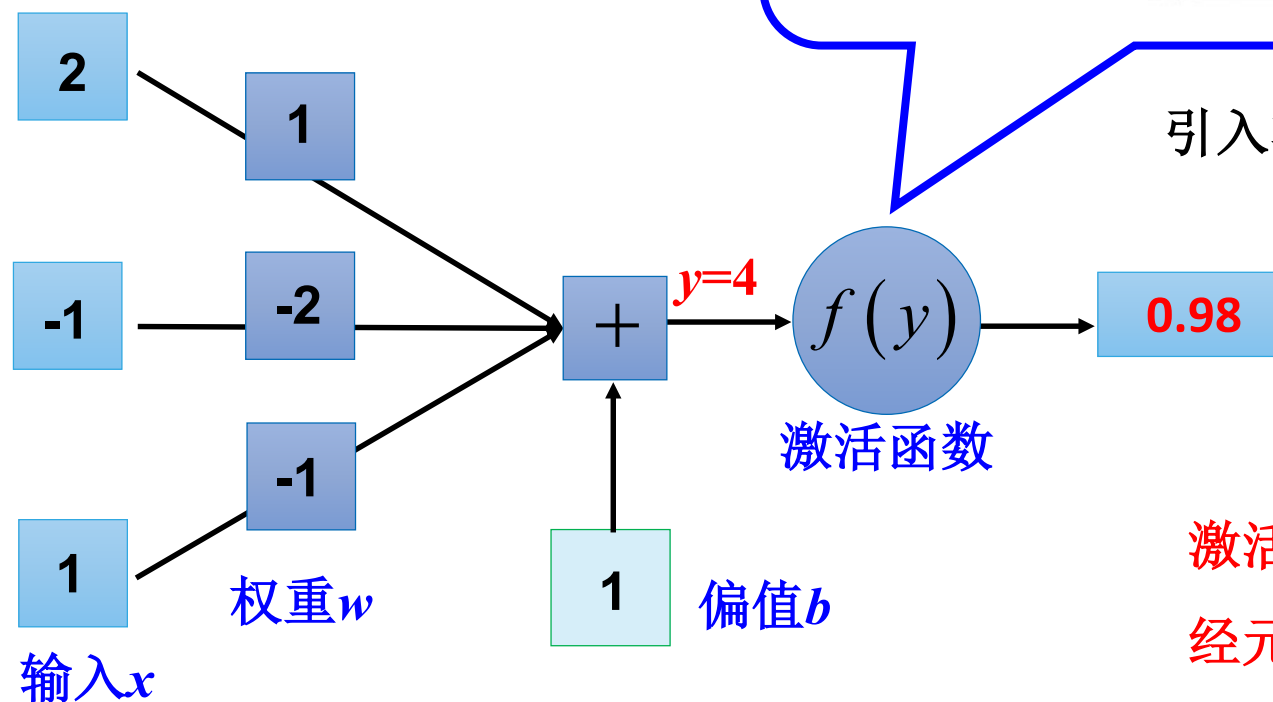
优化算法

其他

激活函数(Activation Function)

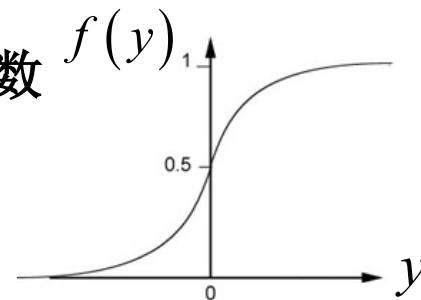
$$y = x_1 w_1 + \dots + x_k w_k + \dots + x_K w_K + b$$

例如:



Sigmoid 函数

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



引入非线性因素

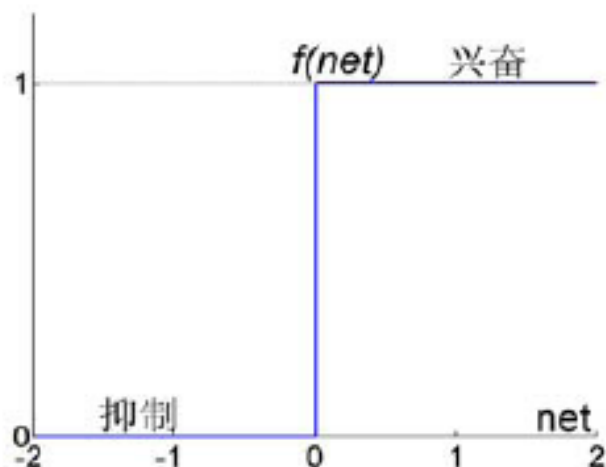
激活函数作用：引入非线性，将神经元线性模型转为广义线性模型。

早期激活函数——阈值型激活函数

早期麦克卡洛—匹茨模型的激活函数采用阈值函数：

当神经元的总输入小于0，则输出 $y_k=0$ ，神经元处于抑制状态；

当总输入大于或等于0，则输出 $y_k=1$ ，神经元处于兴奋状态。



人工神经元的激活函数

早期常采用较简单的“硬”阈值函数：

$$y_k = \begin{cases} 1 & \text{如果 } net \geq 0 \\ 0 & \text{如果 } net < 0 \end{cases}$$

从生理学角度：

0 代表抑制，1 代表兴奋。

早期激活函数——阈值型激活函数

维纳和匹茨认为：

- 阈值型激活函数提出，使得数字计算机的开关（二进制）工作方式，与神经元兴奋、抑制的性质非常类似。
- 认为“解释动物记忆的性质和变化的问题，与机器中的人工记忆的问题也是相互类似的。”

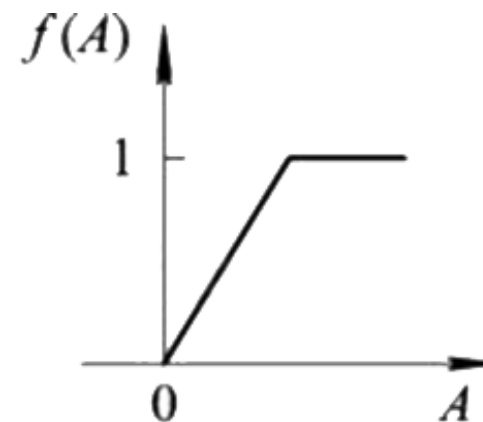
实际上，生物脑的记忆方式与计算机储存器的工作方式大相径庭：

- 前者是将信息分散（分布式）到由大量神经元组成的网络中保存，依靠记忆的内容联想方式寻找所需信息；
- 后者储存的信息具有确定地点（定位），依靠“地址码”寻找所需信息。

早期激活函数——分段线性型激活函数

神经元的输入—输出特性满足一定的区间线性关系，其特性函数表达为：

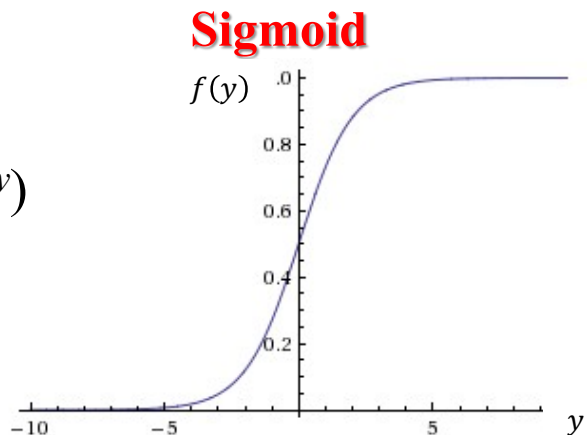
$$y = \begin{cases} 0 & A \leq 0 \\ KA & 0 < A \leq A_k \\ 1 & A_k < A \end{cases}$$



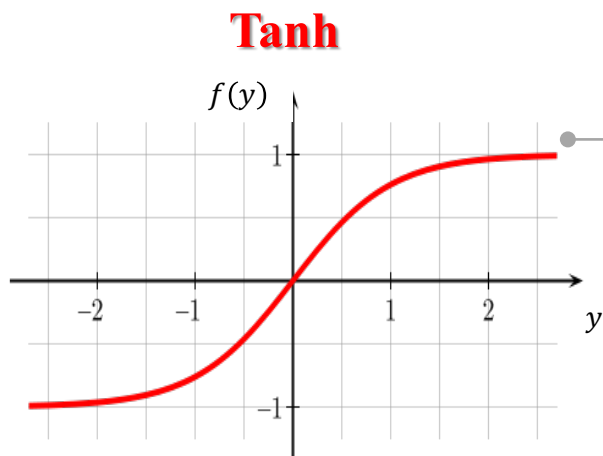
式中, K 、 A_k 均表示常量。

常用激活函数

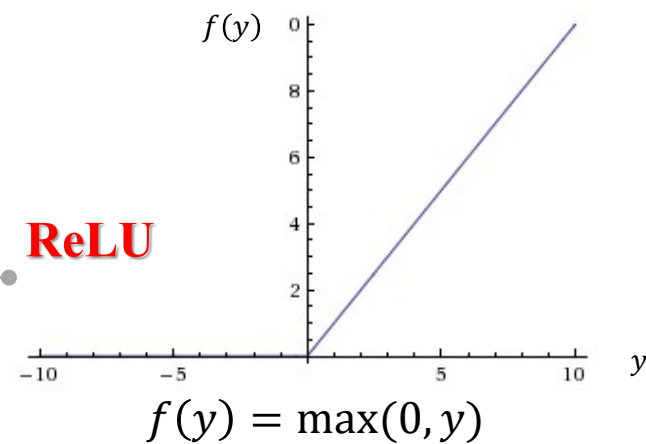
$$f(y) = 1 / (1 + e^{-y})$$



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



ReLU

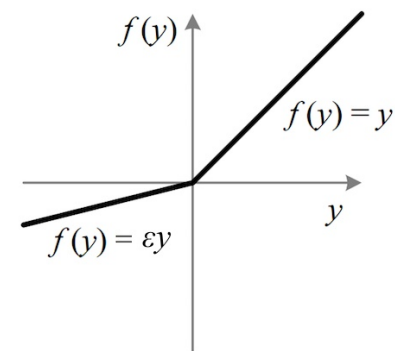


Softmax

$$f(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

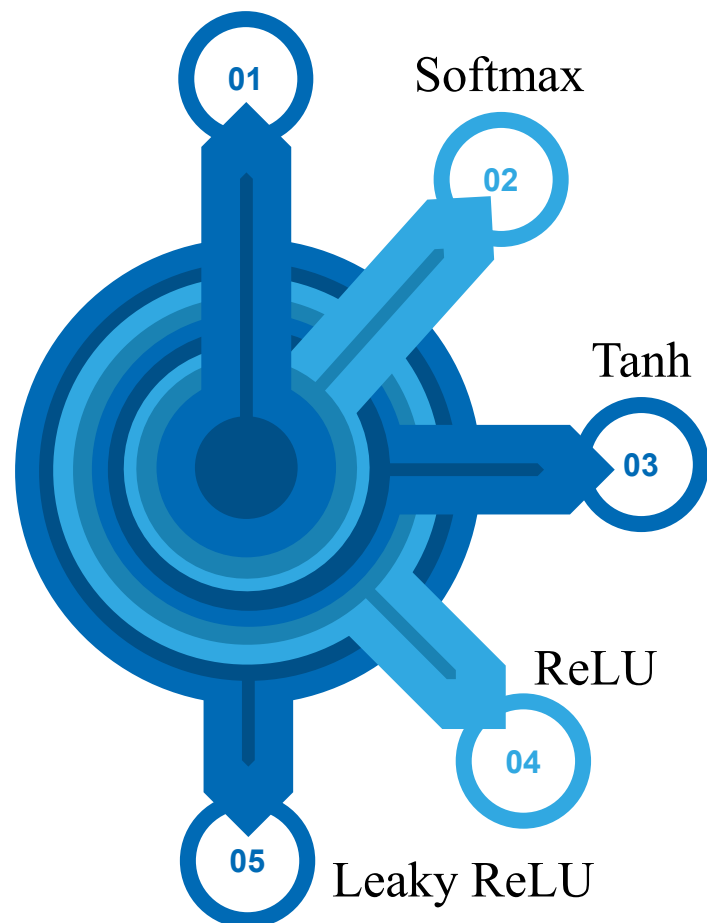
Leaky ReLU

$$f(y) = \max(\epsilon y, y)$$



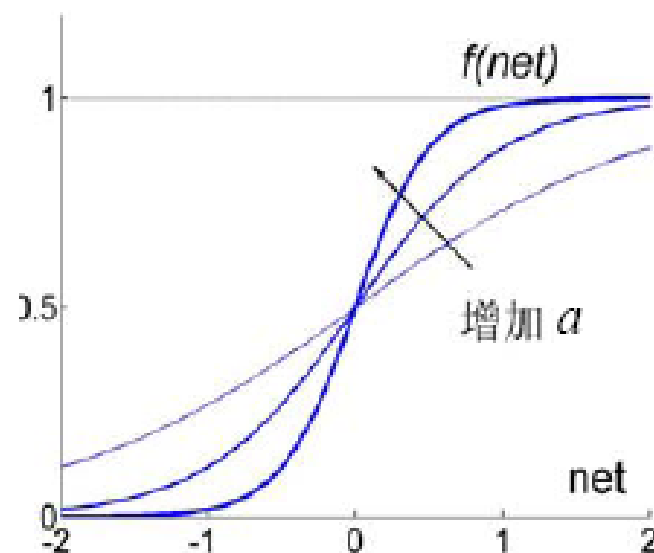
常用激活函数——Sigmoid

Sigmoid



又称S型激活函数，函数的输入—输出特性多采用指数、对数或双曲正切等S型函数表示。

S型特性函数反映了神经元的非线性输出特性。

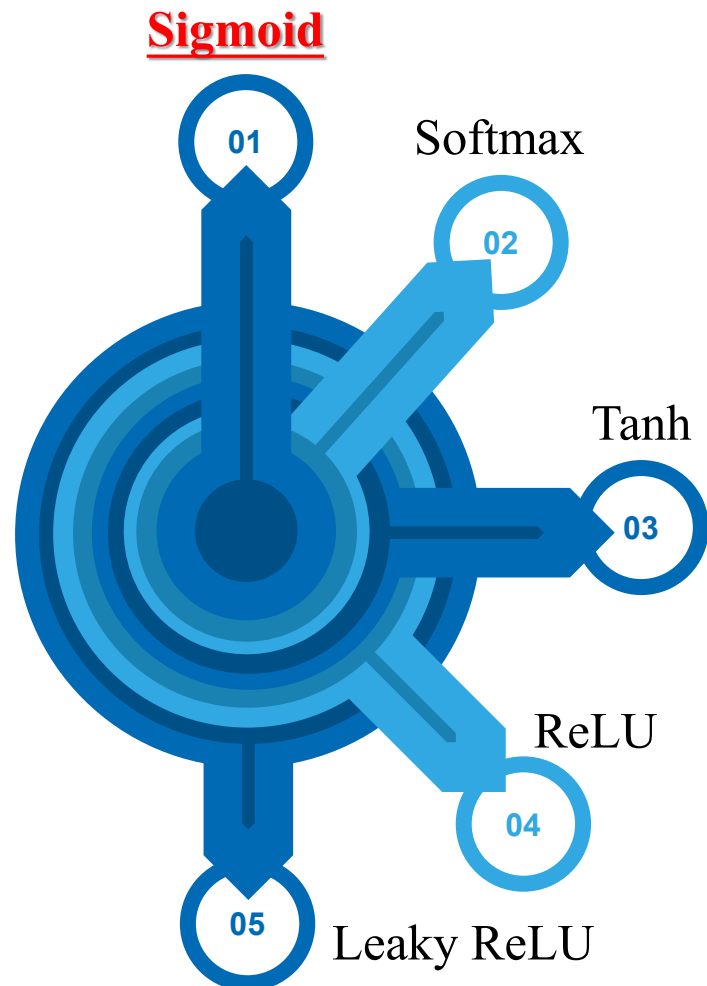


“软”S型函数：

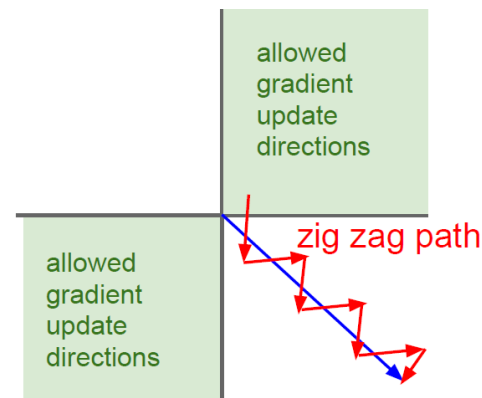
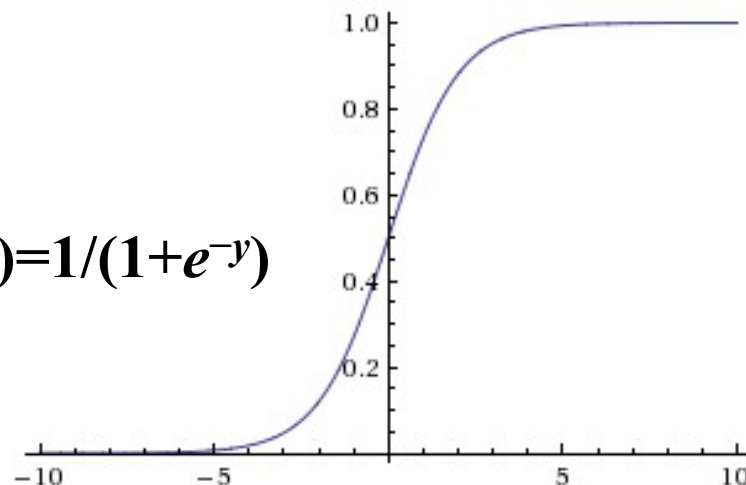
$$\text{Sig}(\text{net}, a) = \frac{1}{1 + \exp(-a * \text{net})}$$

- a 是 S 型函数的斜率
- 半线性函数：线性与非线性之间较好的平衡

常用激活函数——Sigmoid



$$f(y)=1/(1+e^{-y})$$



存在问题:

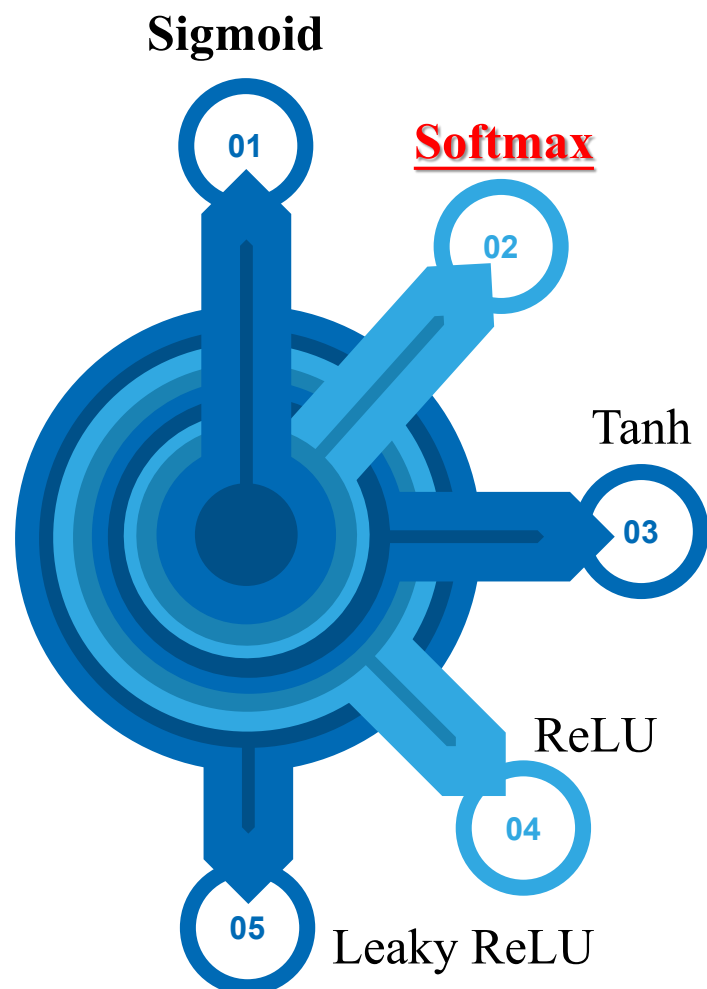
1) 梯度消失

当神经元激活在接近0或1处时会饱和，这些区域梯度几乎为0，导致梯度消失，几乎就没有信号通过神经传回上一层。

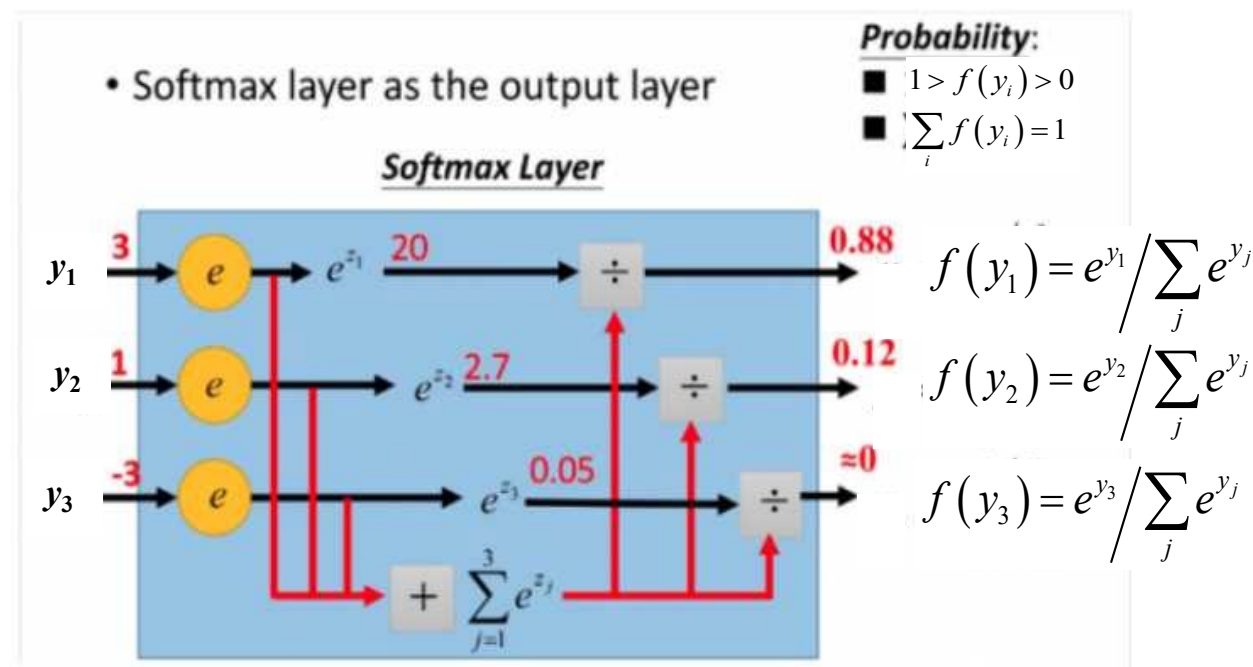
2) Sigmoid函数输出非零中心

如果输入神经元的数据总是正数，则梯度在反向传播的过程中，将会要么全部是正数，要么全部是负数，这将会导致梯度下降权重更新时出现z字型的下降。

常用激活函数——Softmax

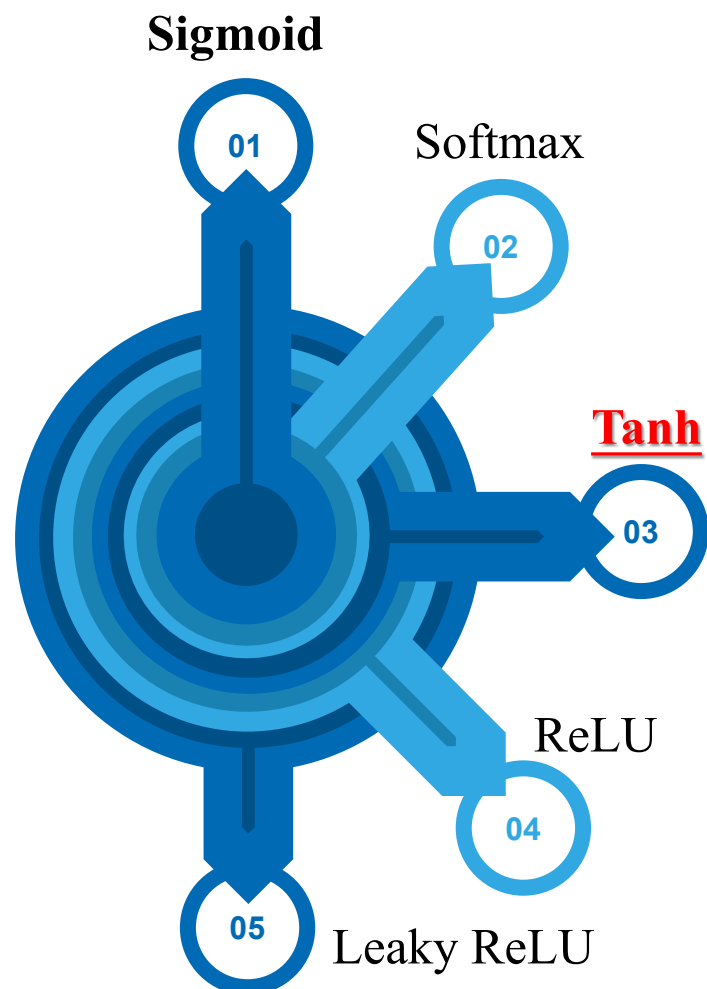


$$f(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

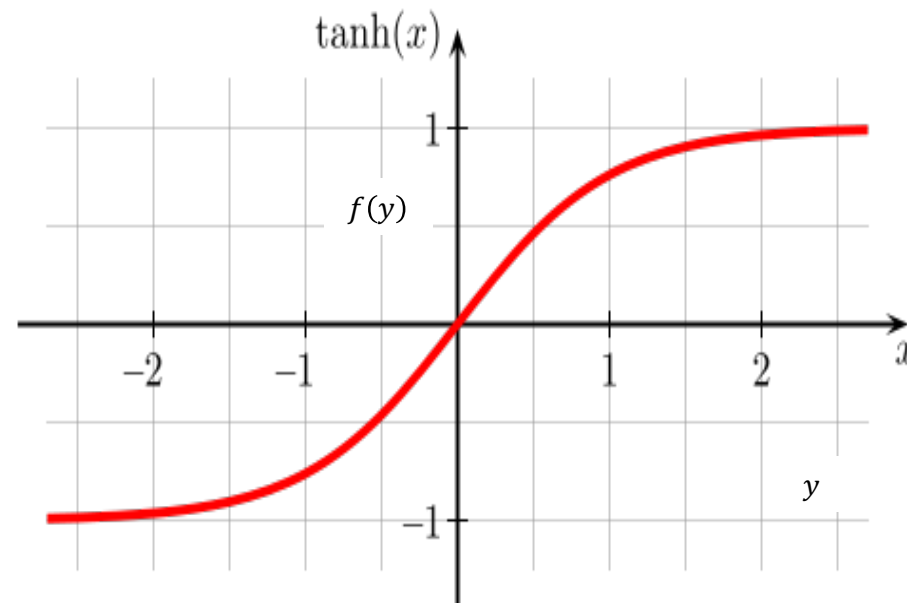


Softmax是Sigmoid的扩展，将多个神经元的输出，映射到（0,1）区间内，可以看成概率来理解，适用于多分类问题。

常用激活函数——Tanh



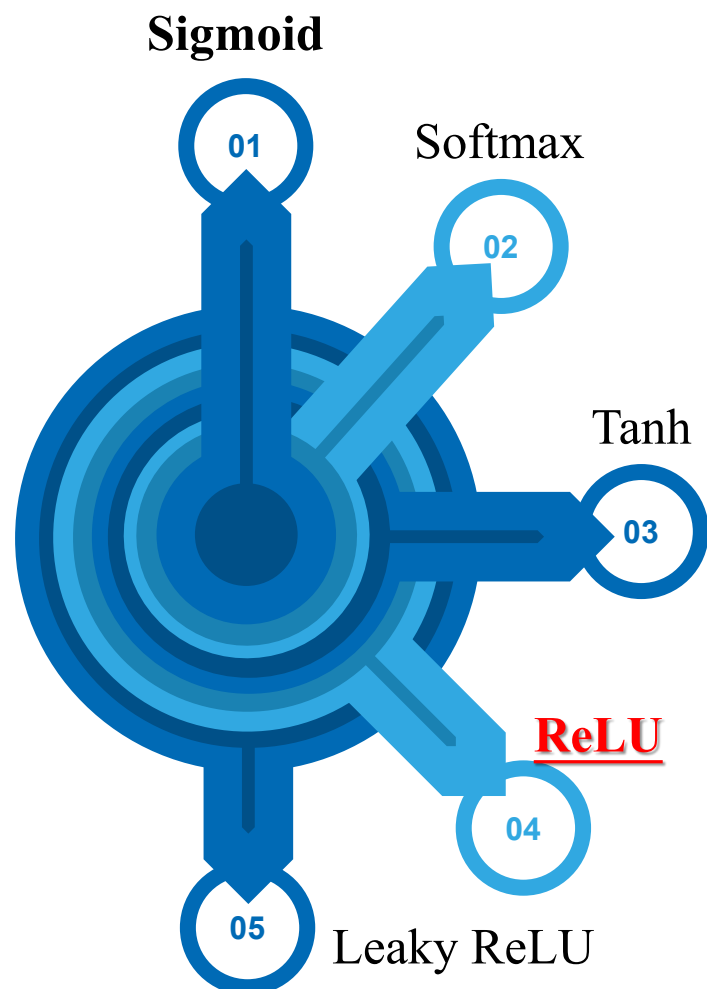
$$f(y) = \frac{e^y - e^{-y}}{e^y + e^{-y}}$$



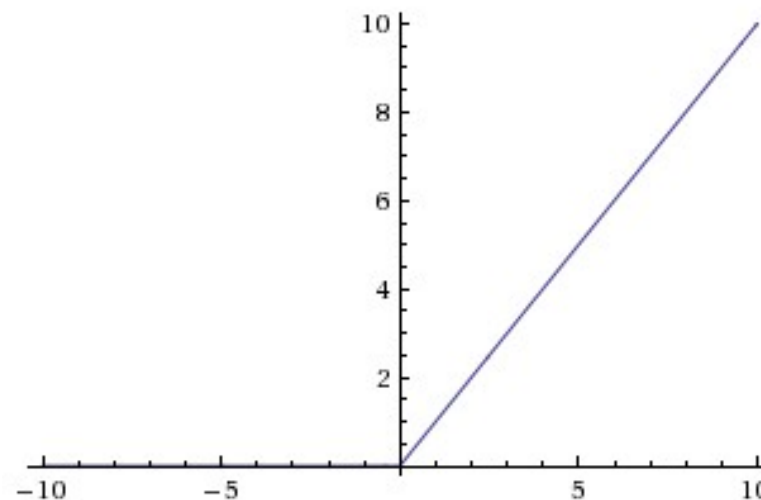
Tanh将实数值压缩到 $[-1,1]$ 之间，解决了Sigmoid的输出是不是零中心的问题，但仍然存在饱和问题。

为了防止饱和，现在主流的做法会在激活函数前多做一步 *batch normalization*，尽可能保证每一层网络的输入具有均值较小的、零中心的分布。

常用激活函数——ReLU



$$f(y) = \max(0, y)$$

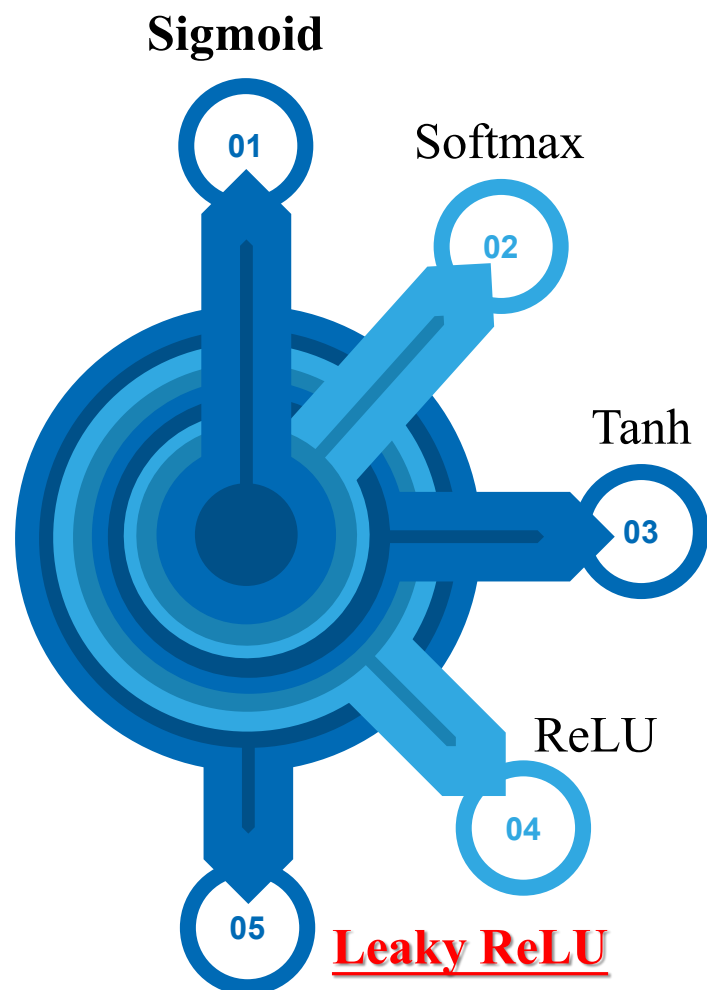


相较于Sigmoid和Tanh函数，ReLU对于随机梯度下降的收敛有巨大的加速作用；

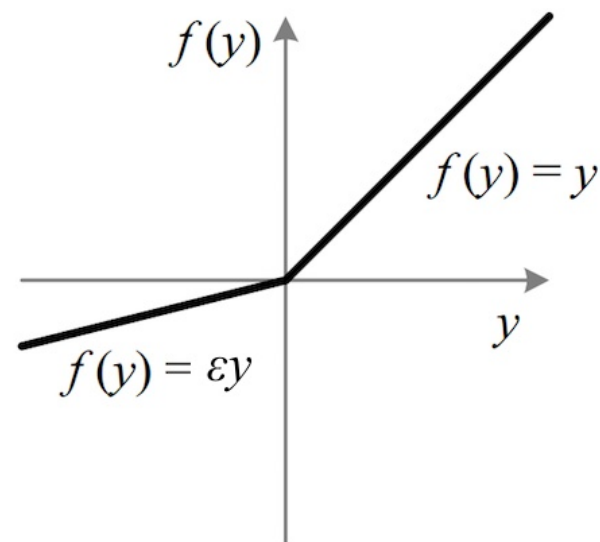
Sigmoid和Tanh在求导时含有指数运算，而ReLU求导几乎不存在任何计算量。

缺点是ReLU的单侧抑制性会丢失负轴信息。

常用激活函数——Leaky ReLU



$$f(y) = \max(\varepsilon y, y)$$



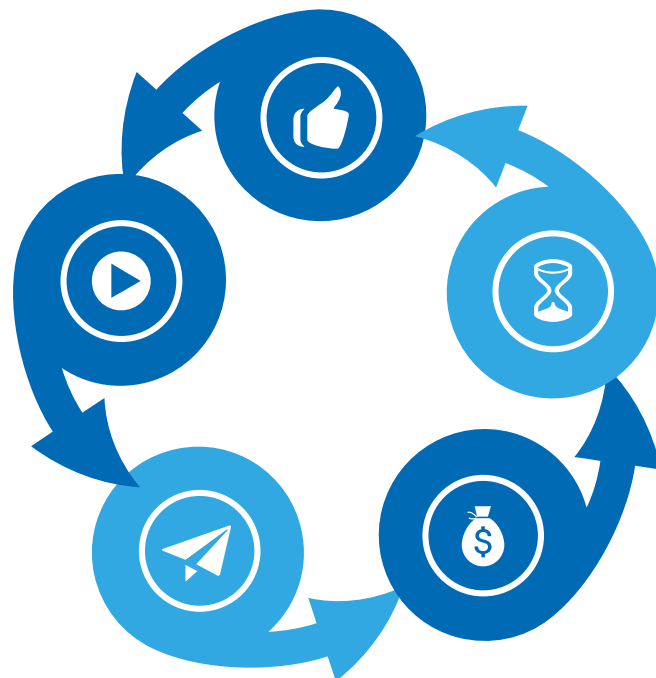
ε 提供了很小的负数梯度值，如0.01，目的是使负轴信息不会全部丢失。

感知器与线性分类

多层感知器

误差反传与梯度下降

损失函数



激活函数

优化算法 (见深度学习基础3)

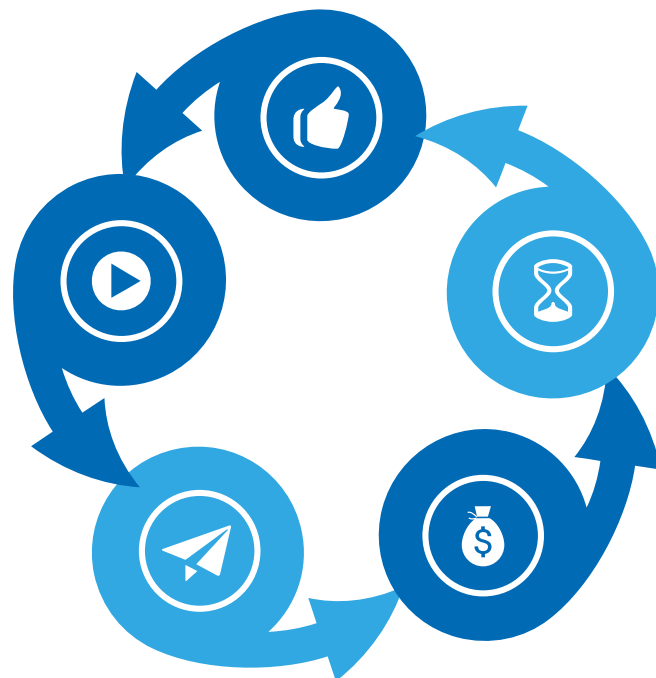
其他

感知器与线性分类

多层感知器

误差反传与梯度下降

损失函数



激活函数

优化算法

其他（见深度学习基础3）

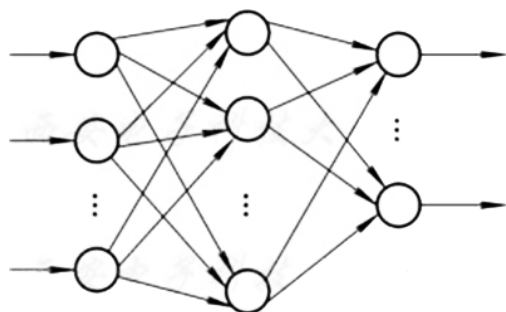


- ✓ 学习的本质
学什么？ 怎么学？ ...
- ✓ 从生物神经元到人工神经元
生物神经元、人工神经元， ...
- ✓ 神经网络发展历史
发展历史 ...
- ✓ 神经网络基本概念
激活函数、学习规则、优化算法、 ...
- ✓ **经典人工神经网络**
前向神经网络、BP神经网络， Hopfield神经网络， ...

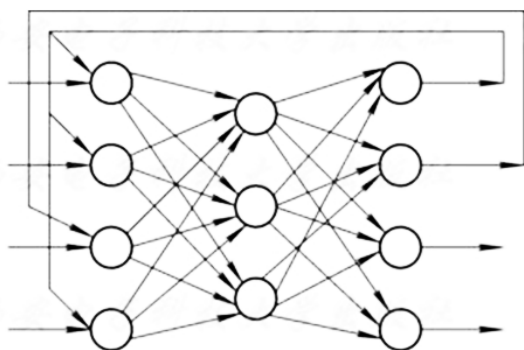
经典神经网络结构分类

(浅层) 神经网络分类:

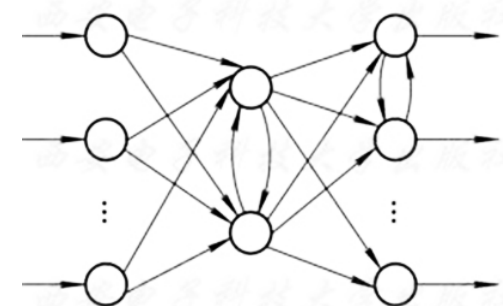
- 分层前向网络



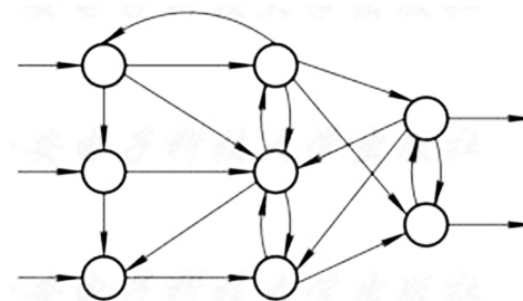
- 反馈前向网络



- 互连前向网络



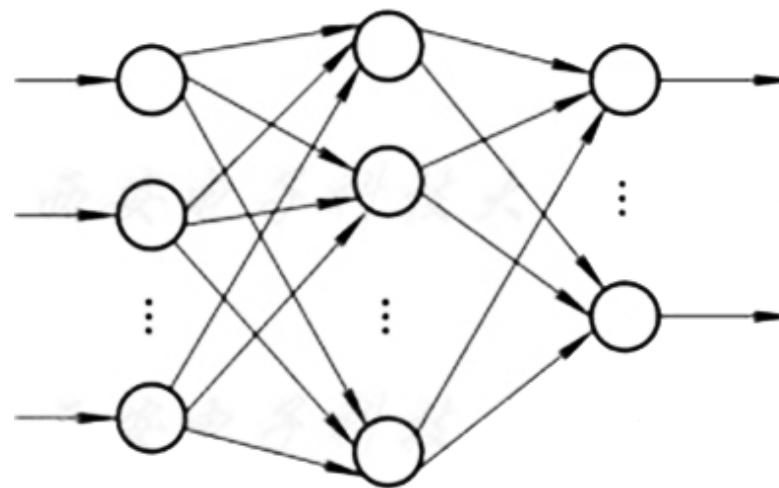
- 广泛互连网络



分层前向网络

网络的结构特征：

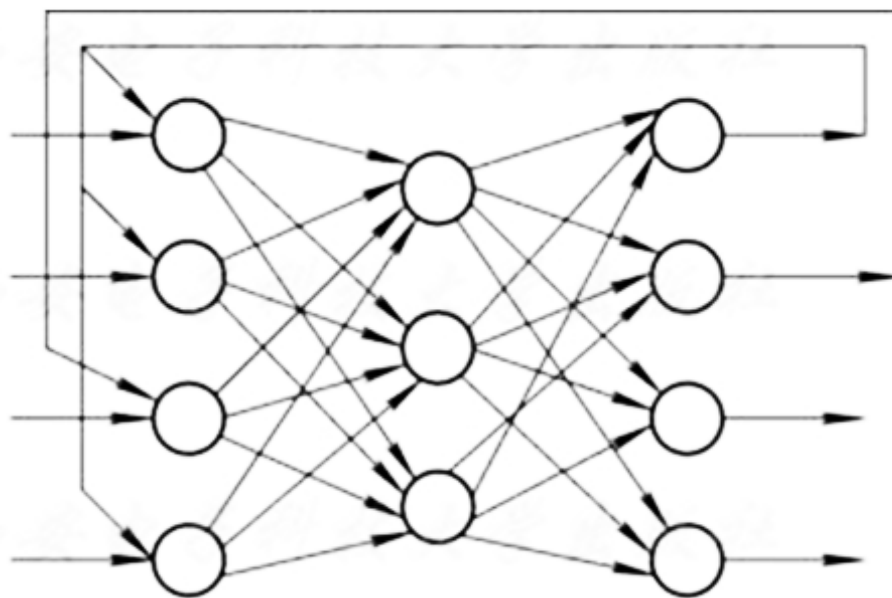
- 由若干层神经元组成
- 输入层、一层或多层中间层(隐层)和输出层。
- 各层顺序连接；且信息严格地按照从输入层进,经过中间层,从输出层出的方向流动。故称“前向”。
- 输入层是网络与外部环境的接口,它接受外部输入；
- 隐层是网络的内部处理层,神经网络具有的模式变换能力,如模式分类、模式完善、特征抽取等,主要体现在隐层神经元的处理能力上；
- 输出层是网络的输出接口,网络信息处理结果由输出层向外输出。



反馈前向网络

一种分层前向网络

输出层到输入层具有反馈连接。

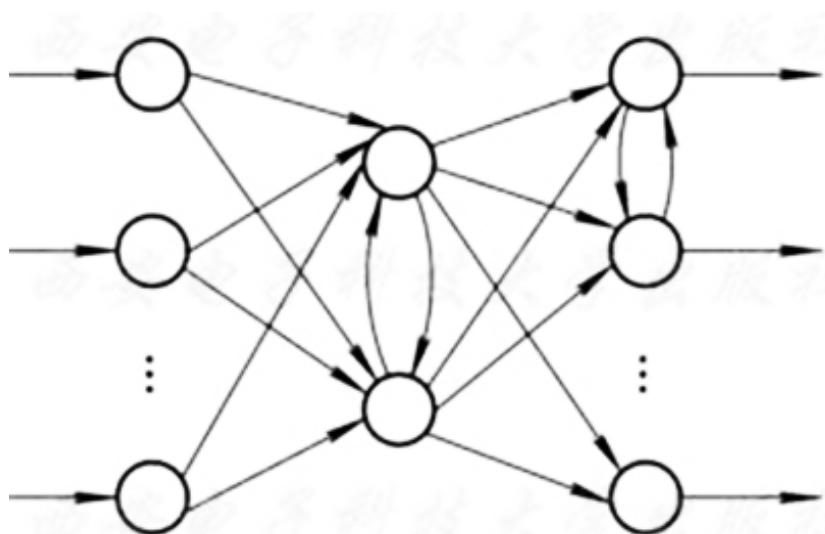


互连前向网络

一种分层前向网络

同层神经元之间有相互连接。

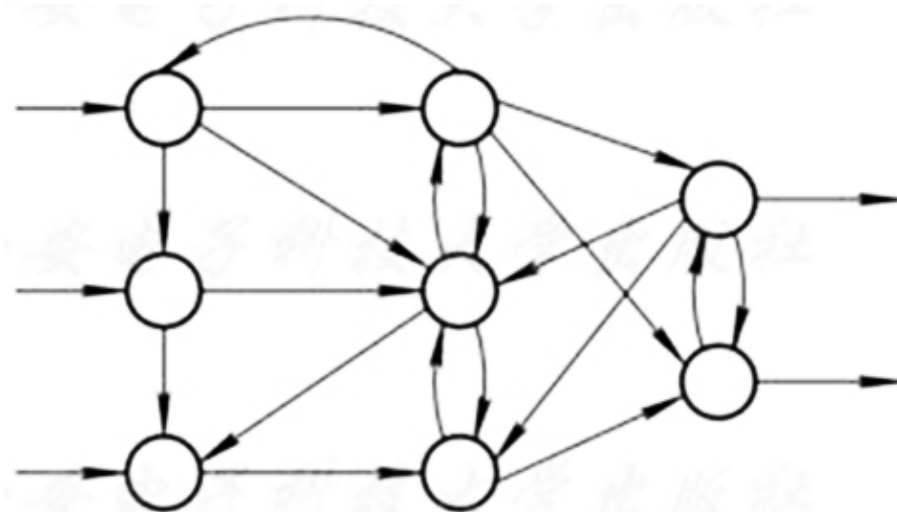
同一层内单元的相互连接使它们之间有彼此牵制作用。



广泛互连网络

任意两个神经元之间都可以或可能是可达的,即存在连接路径。

如Hopfield网络、波尔茨曼机模型结构



神经网络模型

神经网络模型

- 是神经网络的综合描述和整体概念。
- 包括
 - 网络的拓扑结构
 - 输入输出信号类型
 - 信息传递方式
 - 神经元特性函数
 - 学习方式、学习算法
 - 等等。

神经网络模型分类(按学习方式)

神经网络的学习方式:

- 有导师学习、强化学习和无导师学习。

神经网络模型按学习方式分类:

- 有导师学习网络
- 强化学习网络
- 无导师学习网络。

神经网络模型分类(按网络结构)

神经网络的连接结构分为两大类

- 分层结构:有明显的层次,信息的流向由输入层到输出层.
- 互连结构:没有明显的层次,任意两处理单元之间都是可达的,具有输出单元到隐单元(或输入单元)的反馈连接.

神经网络模型按网络结构分类:

- 对于分层结构网络:前向网络。
- 对于互连型结构网络:反馈网络。

神经网络模型分类(按网络状态)

神经网络模型中,处理单元(即神经元)的状态有两种形式:

- 连续时间变化状态: 所有处理单元状态能在某一区间连续取值。
- 离散时间变化状态: 所有处理单元状态只能取离散的二进制值0或1(或-1、+1)。

神经网络模型按网络状态分类:

- 连续型网络: 如连续型Hopfield网络
- 离散型网络: 如离散型Hopfield网络

神经网络模型分类(按活动方式)

神经网络处理单元的状态取值有两种活动方式:

- 确定性输入经确定性作用函数, 产生确定性的输出状态;
- 由随机输入或随机性作用函数, 产生遵从一定概率分布的随机输出状态。

神经网络模型按网络的活动方式分类:

- 确定性网络: 大部分神经网络模型均属此类。
- 随机性网络: 波尔茨曼机、柯西机和高斯机等。

一些著名神经网络模型

名 称	学习方式	拓扑结构	典 型 应 用
感知机(Perceptron)	有导师	前向	线性分类
误差反向传播网(BP)	有导师	反馈前向	模式分类、映射、特征抽取
自适应线性元件(Adaline)	有导师	前向	控制、预测、分类
自适应共振理论(ART)	无导师	反馈	模式识别、分类
双向联想记忆(BAM)	不学习	反馈	图像识别
波尔茨曼机(BM)	有导师	互连	模式识别、组合优化
柯西机(CM)	有导师	反馈	模式识别、组合优化
盒中脑(BSB)	有导师	反馈	数据库知识提取
反传网络(CPN)	有导师	前向	联想记忆、图像压缩、统计分析
Hopfield 网络 (DHNN) (CHNN)	无导师 不学习	互 连	联想记忆 组合优化
多层自适应线性元件(Madaline)	有导师	前向	自适应控制
新认知机 Neocognitron	有导师	前向	字符识别
自组织映射(SOM)	无导师	互连	聚类、特征抽取
细胞神经网络(CNN)	不学习	反馈	图像处理、图形辨识

浅层人工神经网络所能达到的功能

数学上的映射逼近

- 通过一组映射样本 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, 网络以自组织方式寻找输入与输出之间的映射关系: $y_i = f(x_i)$ 。
- 这种映射逼近能力可用于系统建模、模式识别与分类等。
- 典型网络如BP网络等。

浅层神经网络所能达到的功能 2

数据聚类、压缩

- 通过自组织方式对所选输入模式聚类。
- 同一聚类可对应于多个输入模式；
- 另一种是编码形式。
- 典型的网络如ART模型，其应用如语音识别中用来减小输入的维数，减小存储数据的位数等。

浅层人工神经网络所能达到的功能 3

联想记忆

- 实现模式完善、恢复,相关模式的相互回忆等。
- 典型的如Hopfield网络、CPN网络等。

优化计算和组合优化问题求解

- 利用神经网络的渐进稳定态,特别是反馈网络的稳定平衡态,进行优化计算或求解组合优化问题的近似最优解。
- 如Hopfield网络、波尔茨曼机等。

浅层人工神经网络所能达到的功能 4

模式分类

- 大多数神经网络模型都有这种分类能力。
- 首先对样本模式能够进行分类, 即要离线学习, 像BP网、CPN网、Hopfield网等。
- 训练得到各连接权值。
- 在线运行时, 无需训练。

浅层人工神经网络所能达到的功能 5

概率密度函数的估计

- 根据给定的概率密度函数, 通过自组织网络来响应在空间 $\mathbf{R}^n$ 中服从这一概率分布的一组向量样本 $X_1, X_2, \dots, X_k$ 。
- 如波尔茨曼机模型、CPN 网、SOM网。

经典神经网络 <https://www.asimovinstitute.org/neural-network-zoo/>

A mostly complete chart of Neural Networks

©2019 Fjodor van Veen & Stefan Leijnen asimovinstitute.org

- Input Cell
- Backfed Input Cell
- Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- Spiking Hidden Cell
- Capsule Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- Gated Memory Cell
- Kernel
- Convolution or Pool

Perceptron (P)



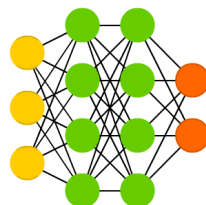
Feed Forward (FF)



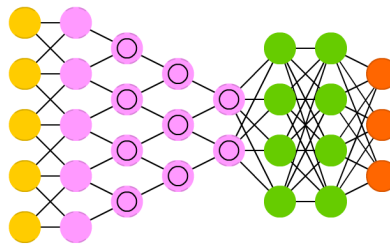
Radial Basis Network (RBF)



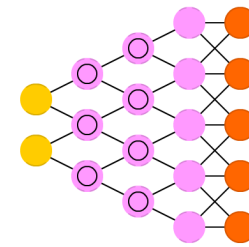
Deep Feed Forward (DFF)



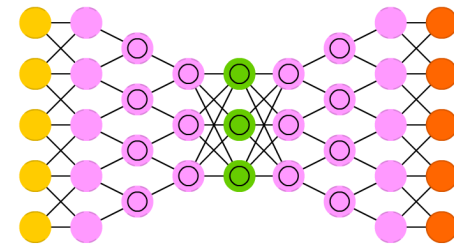
Deep Convolutional Network (DCN)



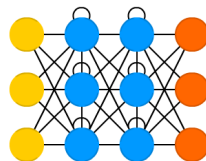
Deconvolutional Network (DN)



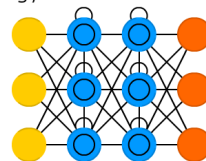
Deep Convolutional Inverse Graphics Network (DCIGN)



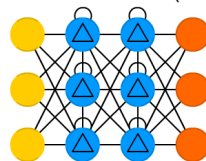
Recurrent Neural Network (RNN)



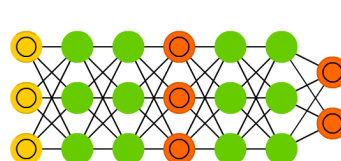
Long / Short Term Memory (LSTM)



Gated Recurrent Unit (GRU)



Generative Adversarial Network (GAN)



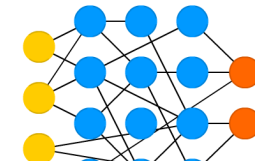
Liquid State Machine (LSM)



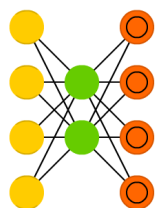
Extreme Learning Machine (ELM)



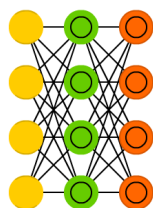
Echo State Network (ESN)



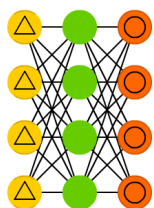
Auto Encoder (AE)



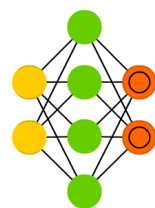
Variational AE (VAE)



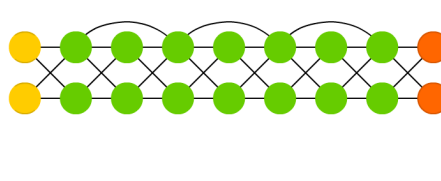
Denoising AE (DAE)



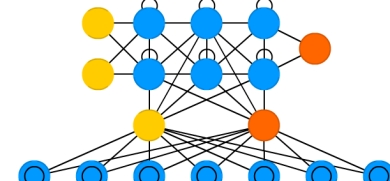
Sparse AE (SAE)



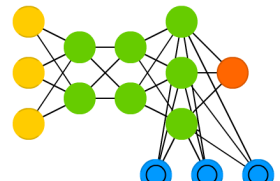
Deep Residual Network (DRN)



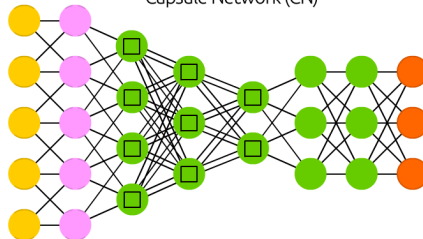
Differentiable Neural Computer (DNC)



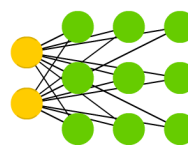
Neural Turing Machine (NTM)



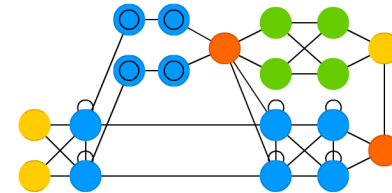
Capsule Network (CN)



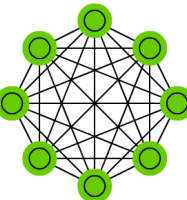
Kohonen Network (KN)



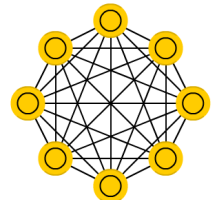
Attention Network (AN)



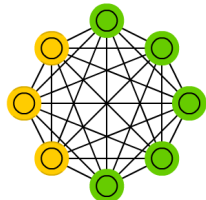
Markov Chain (MC)



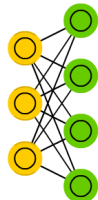
Hopfield Network (HN)



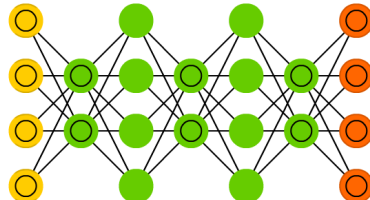
Boltzmann Machine (BM)



Restricted BM (RBM)



Deep Belief Network (DBN)



经典神经网络——前馈神经网络（FF 或 FFNN）和感知器（P）



Feed forward neural networks (FF or FFNN) and Perceptrons (P)

最简单实用的网络有两个输入单元和一个输出单元，可用于对逻辑门进行建模。通常通过反向传播训练 FFNN。

Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.”
Psychological review 65.6 (1958): 386.

经典神经网络——径向基函数（RBF）网络 [Broomhead and Lowe, 1988]

Radial basis function (RBF) 网络是一种单隐层前馈神经网络, 它使用径向基函数作为隐层神经元激活函数, 而输出层则是隐层神经元输出的线性组合.

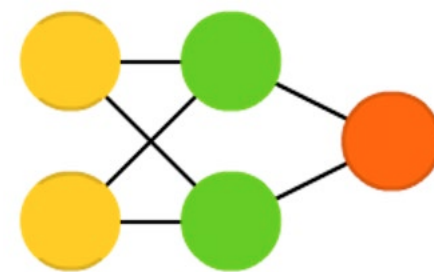
RBF网络模型

假定输入为 d 维的向量 $\mathbf{x}$, 输出为实值, 则RBF 网络可以表示为

$$\varphi(\mathbf{x}) = \sum_{i=1}^q w_i \rho(\mathbf{x}, \mathbf{c}_i)$$

其中 q 为隐层神经元的个数, $\mathbf{c}_i$ 和 w_i 分别是第 i 个神经元的中心和权重; $\rho(\mathbf{x}, \mathbf{c}_i)$ 是径向基函数。常用的高斯径向基函数形如

$$\rho(\mathbf{x}, \mathbf{c}_i) = e^{-\beta_i \|\mathbf{x} - \mathbf{c}_i\|^2}$$



Broomhead, David S., and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. No. RSRE-MEMO-4148. ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (UNITED KINGDOM), 1988.

经典神经网络—— RBF 网络 [Broomhead and Lowe, 1988]

□ RBF网络性质

具有足够多隐层神经元RBF 神经网络能以任意精度逼近任意连续函数. [Park and Sandberg, 1991]

□ RBF网络训练

- Step1: 确定神经元中心，常用的方式包括随机采样、聚类等
- Step2: 利用BP算法等确定参数

经典神经网络——Hopfield神经网络

感知器之后，人工神经网络陷入低谷。

1982年美国科学院院士霍普菲尔德教授提出了具有联想记忆能力的Hopfield神经网络。

在其著名论文“具有集体计算能力的神经网络和实际系统”中，将网络（节点集合）视为原子集合处理，用统计力学方法分析网络的存储和优化特性。

该神经网络可巧妙地求解“旅行商路径问题”，

经典神经网络——Hopfield神经网络

霍普菲尔德教授文中描述了：

- 采用运算放大器电路实现神经元；
- 采用运算放大器反馈电路实现神经网络；
- 采用克希霍夫电流定律、微分方程与能量函数概念分析Hopfield神经网络的动力学工作原理；
- Hopfield神经网络分为离散型和连续型两种网络模型，分别记为DHNN（Discrete Hopfield Neural Network）和CHNN（Continues Hopfield Neural Network）。，离散型适用于联想记忆，连续性适合处理优化问题。

Hopfield神经网络

网络特点:

- 单层全反馈网络
- 每个神经元接收其他所有神经元输出;

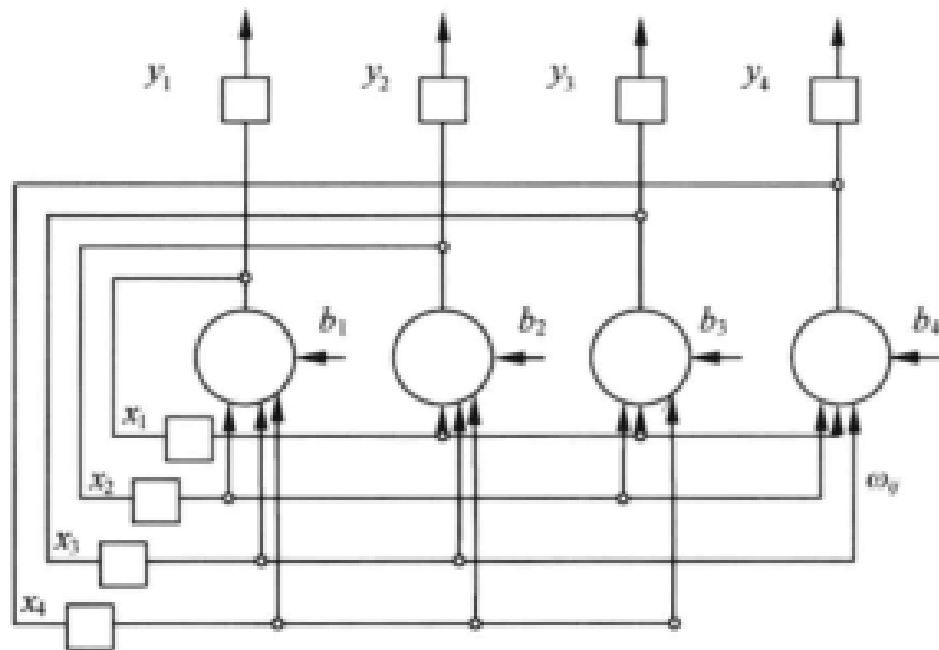


图 9-1 离散 Hopfield 网络

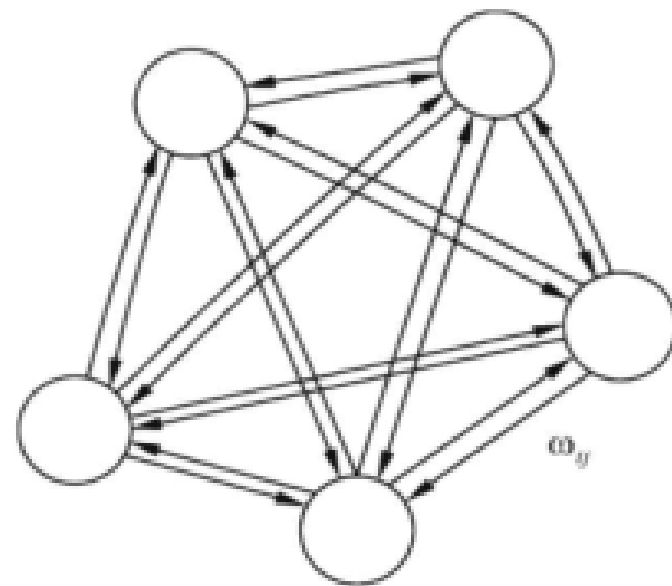


图 9-2 Hopfield 网络的网状结构

任一神经元均受到其他所有神经元输出的控制，从而使各神经元的输出相互制约。

离散Hopfield网络中，每个神经元只取二元的离散值0、1或-1、1。

离散型Hopfield神经网络适合处理联想记忆问题。

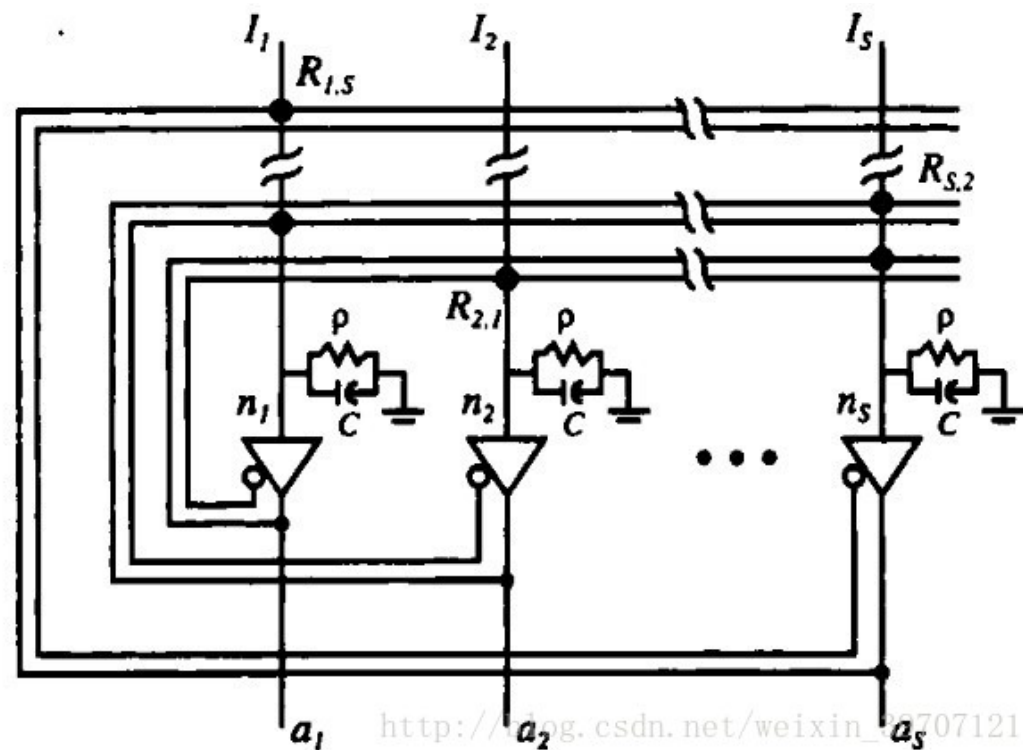
连续型Hopfield神经网络

神经元包括细胞膜输入电容、跨膜电阻和运算放大器。

神经元输出是0~1之间的连续值。

由电阻电容微分方程决定状态的更新速率。

采用电阻、电容和放大器



连续型Hopfield神经网络适合解决组合优化问题。

其关键是把问题映射为一神经网络动力系统，并写出相应的能量函数表达式和动力学方程。根据微分方程来更新神经元确保了能量函数和优化问题可以同时被最小化。神经元的模拟性质和更新过程的硬件实现可以结合起来，创建一个快速而强大的解决方案。

例如：连续型Hopfield神经网络可用于解决TSP问题。

Hopfield神经网络特点

(1) 反馈性神经网络可以用离散变量也可以用连续变量，考虑输出与输入在时间上的延迟，需要用动态方程（差分方程或微分方程）来描述神经元和系统的数学模型。

(2) 前馈神经网络采用误差修正法，计算一般比较慢，收敛速度也慢。反馈型神经网络Hopfield神经网络主要采用Hebb规则，一般情况下收敛速度很快，它与电子电路存在对应关系，使得该网络易于理解和易于硬件实现。

(3) Hopfield神经网络也具有类似于前馈神经网络的应用，而在优化计算方面的应用更加显示出Hopfield神经网络的特点。

由于联想记忆和优化计算是对偶的，当用于联想记忆时，通过样本模式的输入来给定网络的稳定状态，经过学习求得突触权重值；当用于优化计算时，以目标函数和约束条件建立系统的能量函数，确定突触权重值，网络演化到稳定状态，既是优化问题的解。

经典神经网络——自组织映射 SOM 网络 [Kohonen, 1982]

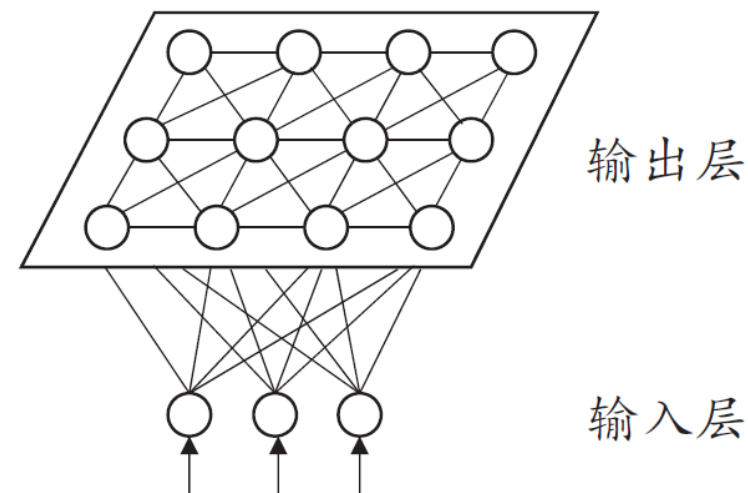
自组织映射 (Self-Organizing Map, SOM)

- 自组织映射现象，普遍存在于生物的神经系统中。如，大脑中将不同特征的感觉输入（运动、触觉、视觉、听觉、味觉等）映射到大脑皮层相应区域。
- 通过研究大脑自组织映射过程中神经元间相互激励、竞争或抑制的机理所提出的。
- 自组织映射网络已成功地应用于语音识别、数据挖掘等领域。

经典神经网络——自组织映射 SOM 网络 [Kohonen, 1982]

自组织映射 (Self-Organizing Map, SOM)

- 一种竞争型的无监督神经网络。
- 它能将高维数据映射到低维空间（通常为2维），同时保持输入数据在高维空间的拓扑结构，即将高维空间中相似的样本点映射到网络输出层中邻近神经元。



SOM 网络输出层神经元以矩阵方式排列在二维空间中，每个神经元都拥有一个权值向量，网络接收输入向量后，会确定输出层获胜神经元，它决定了该输入向量在低维空间中的位置。

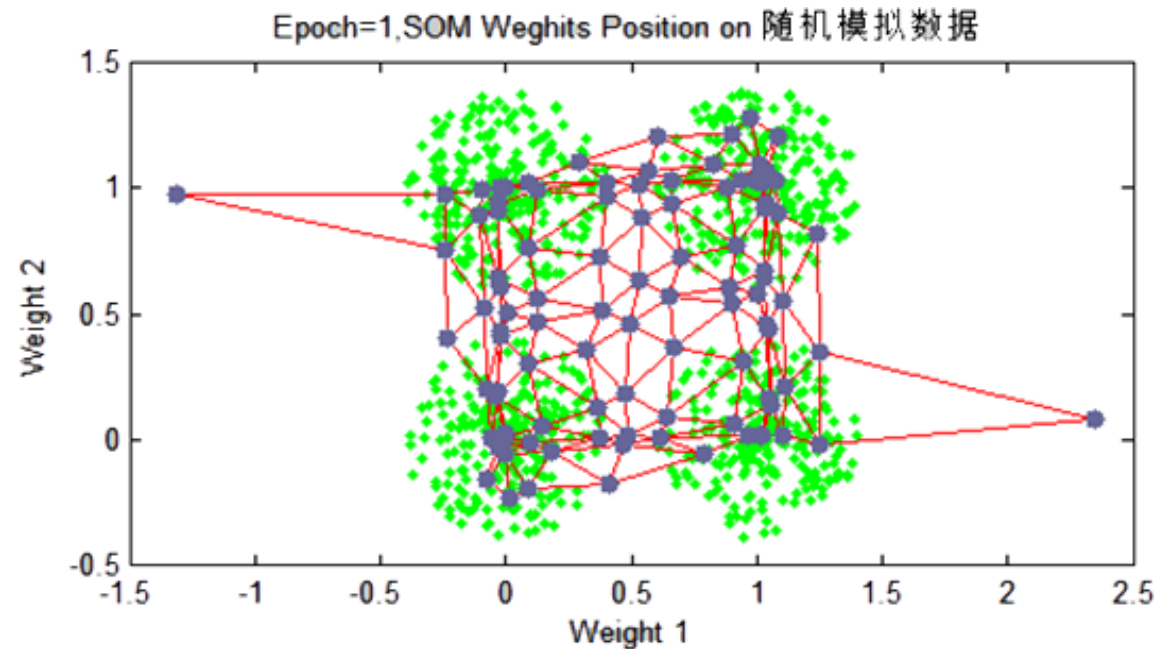
经典神经网络——自组织映射

SOM 网络 [Kohonen, 1982]

□ SOM 网络训练

Step1:接受到一个训练样本后,每个输出层神经元计算该样本与自身携带的权向量之间的距离,距离最近的神元成为竞争获胜者。

Step2:最佳匹配单元及其近邻神经元的权值将被调整,使得这些权向量与当前输入样本的距离缩小。

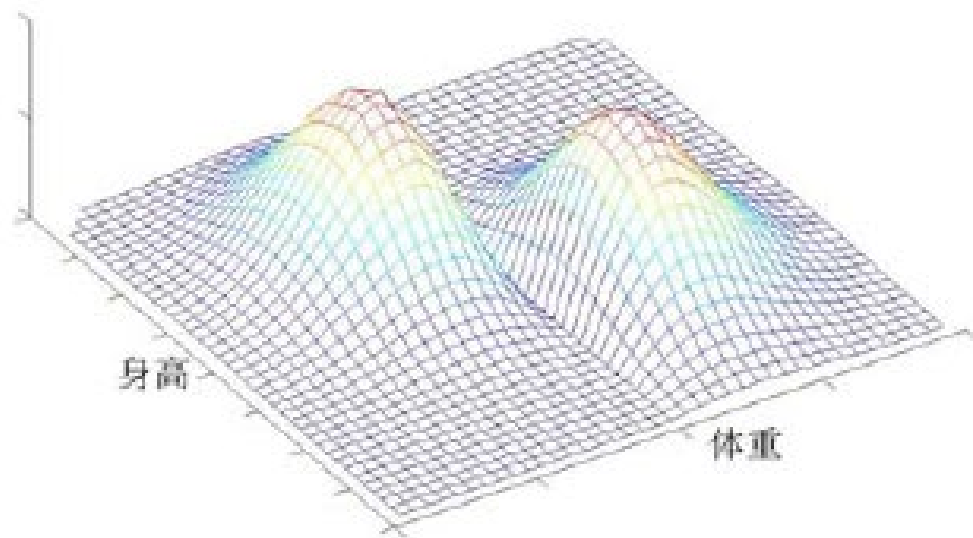


经典神经网络——自组织映射 例

假设为一群运动员检测身高与体重，把数据输入到一个自组织网络中。

网络的输出层显示平面上出现两个明显的高峰。

含义？



含义：

- 这些运动员可分为两类；
- 一类体重较重而个子较矮，另一类体重较轻而个子较高。

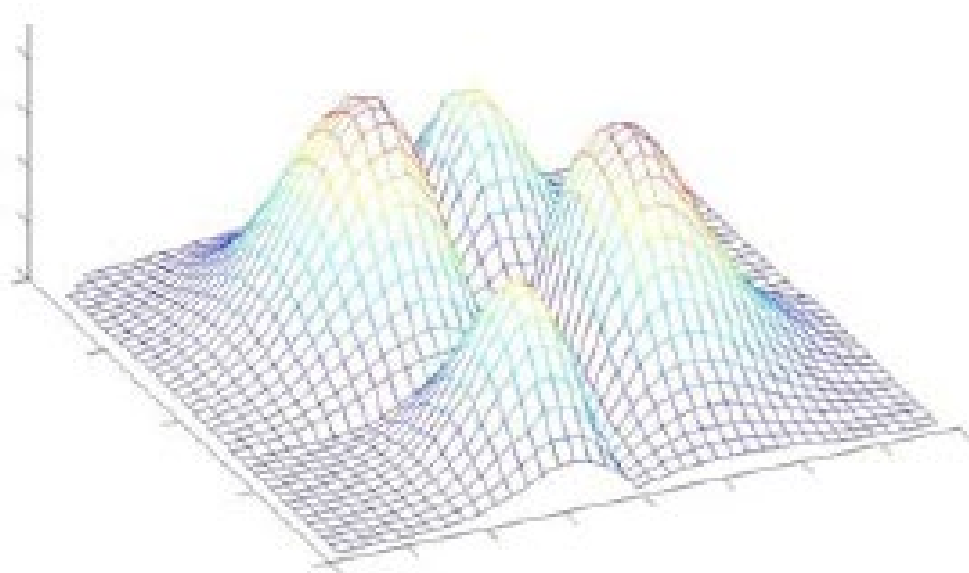
结合现有知识，结论：

- 刚才来的是跳高和举重运动员。

经典神经网络——自组织映射 例2

新疆某山区矿藏分析

- 将相关数据输入以后，自组织映射网络输出层显示出4个类的标志。
- 表明这些数据属于4种矿物；根据其他地质知识就可以确定是哪4种矿物。



自组织映射网络比传统聚类方法
更精确、更快捷。

经典神经网络——反传学习（BP）算法

反传学习（BP, Back- Propagation）算法

- 由美国数学家和经济学家保罗·维勃斯（Paul Werbos）提出。
- BP网络的特点在于误差反传。
- 由多层神经元构成：包括输入层、隐含层、输出层。
- 需要有指导性的训练。

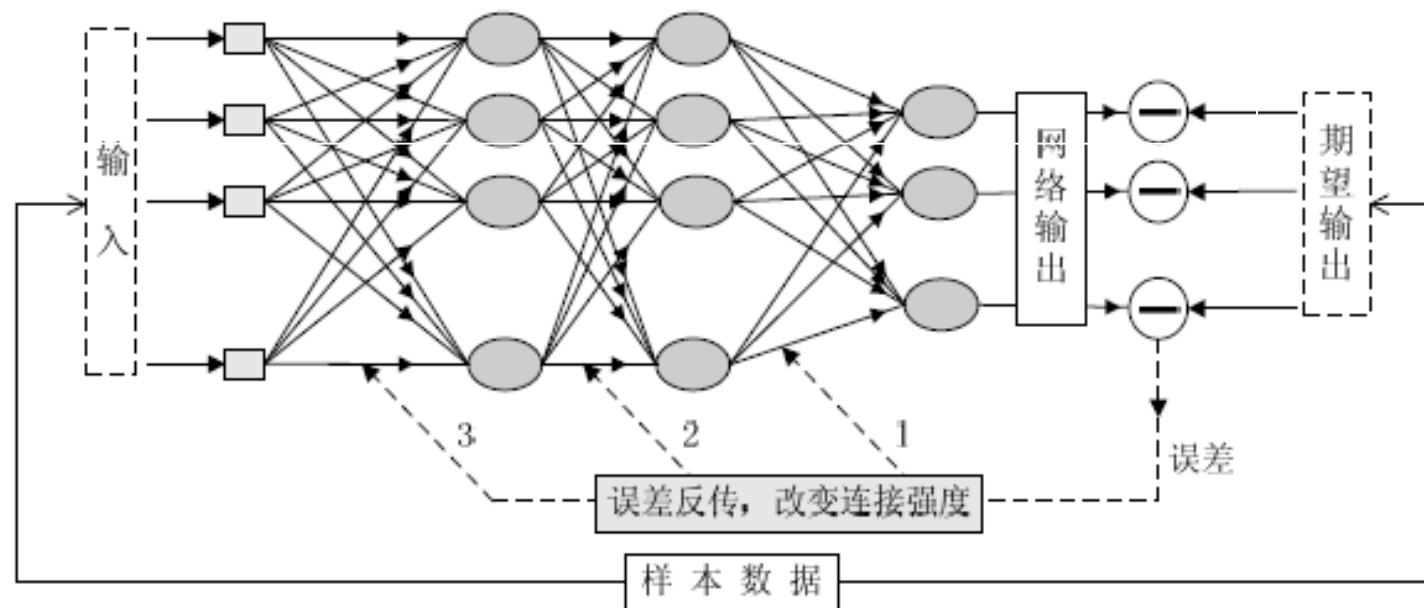
经典神经网络——反传学习（BP）算法

BP 神经网络：输入层、隐层、输出层构成。

训练样本：样本数据对应期望输出。

学习算法：网络实际输出与期望的误差，由输出层反向传递（BP），修改神经元之间的突触连接强度，以减小输出误差。

反复循环，直到误差减小到符合要求为止。



经典神经网络——BP神经网络原理

输入层

- 负责把要处理的外部信号或数据送入网络。

隐层（隐含层）

- 可以有任意多个，但通常只采用一两层。
- 是网络内部的神经元层，不与外部信号直接连接。
- 在完成神经计算之后，传递给输出层。

输出层

- 输出层神经元将计算结果输出。

经典神经网络——BP神经网络原理 2

BP网络中信号（数据）从输入层→隐层→输出层，层层传递的方式完成计算过程。

在训练之前，连接强度（权值）是任意选定的随机值，所以，网络输出的结果与期望值通常相差甚远。

换句话说，神经网络输出不对，是由于神经元之间突触连接状态不对。

训练（学习）就是要改变这众多的突触，使网络输出正确。

经典神经网络——BP网络特点

网络拓扑结构为分层前向网络。

神经元特性函数为Sigmoid型(S型)函数, 一般取为

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

网络输入为连续信号量(实数)。

学习方式为有导师学习。

学习算法为推广的 δ 学习规则, 称为误差反向传播算法, 简称BP学习算法。

BP神经网络原理——学习过程

BP神经网络的学习过程

- 需要“教师指导”，即需要用一些样本（经验数据）来教它：
- 通过给定样本的“期望输出值”训练网络。认为网络的实际输出结果和期望输出值间的误差与网络全部的连接强度有关。
- 将误差反传到网络内，从输出层到输入层，逐层、逐个调整神经元之间的连接权值，以减小输出误差，直到符合要求为止。

BP网络算法及执行步骤

步1 初始化网络权值、 阈值及有关参数(如学习因子 η 等)。

步2 计算总误差

$$E = \frac{1}{2P} \sum_k E_k$$

$$E_k = \frac{1}{2} \sum_j (y_{kj} - y_{kj'})^2$$

其中 P 为样本的个数, y_{kj} 为输出层节点 j 对第 k 个样本的输入对应的输出(称为期望输出), $y_{kj'}$ 为节点 j 的实际输出。

BP网络算法及执行步骤 2

步3 前向计算过程

- 输入一样本数据，向前计算各层节点 (记为 j) 的输出：

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

$$a_j = \sum_{i=0}^n \omega_{ij} O_i$$

- a_j 节点 j 的输入加权和； i 为 j 的信号源方向的相邻层节点, O_i 为节点 i 的输出, 节点 j 的输入。

BP网络算法及执行步骤 3

步4 误差反传，反向修正过程

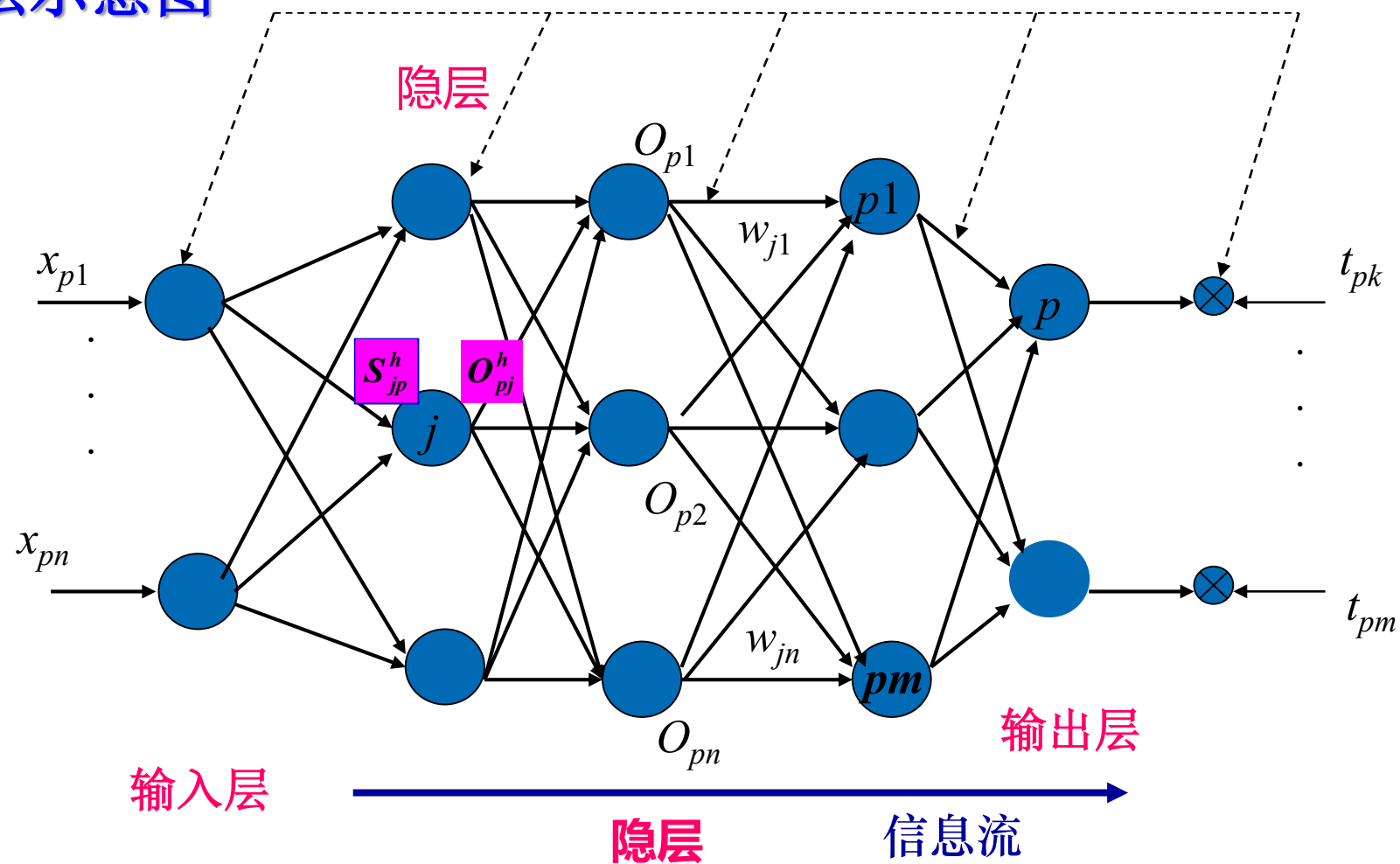
- 从输出层节点到输入层节点以反向顺序，对各连接权值 w_{ij} 进行修正：

$$W_{ij}(t+1) = W(t) + \eta \delta_j O_i$$

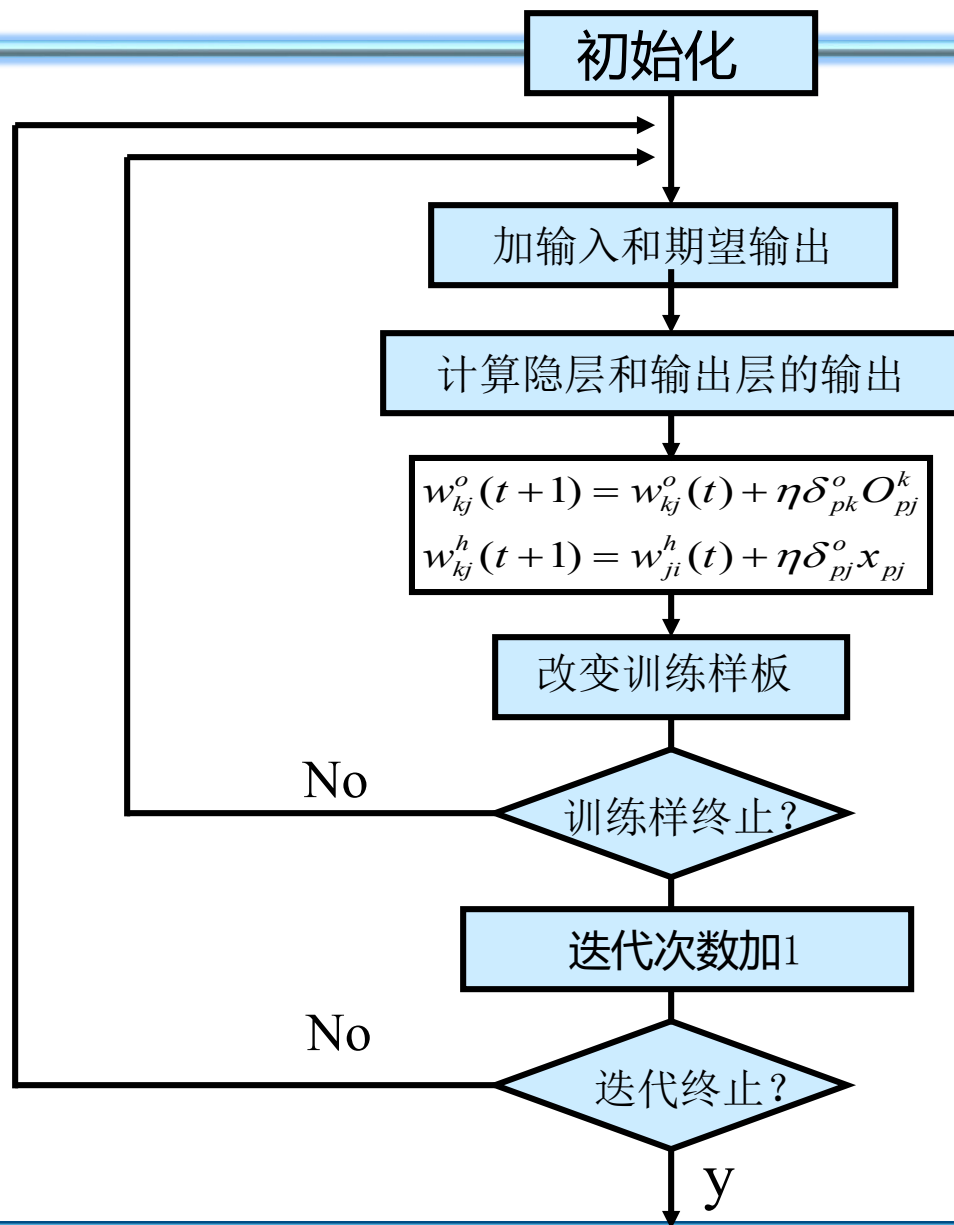
$$\delta_j = \begin{cases} a_j(1-a_j)(y_j - y'_j) & \text{对于输出节点} \\ a_j(1-a_j) \sum_i \delta_l \omega_{lj} & \text{对于中间节点} \end{cases}$$

l 为与节点 j 在输出侧有连接的节点个数。

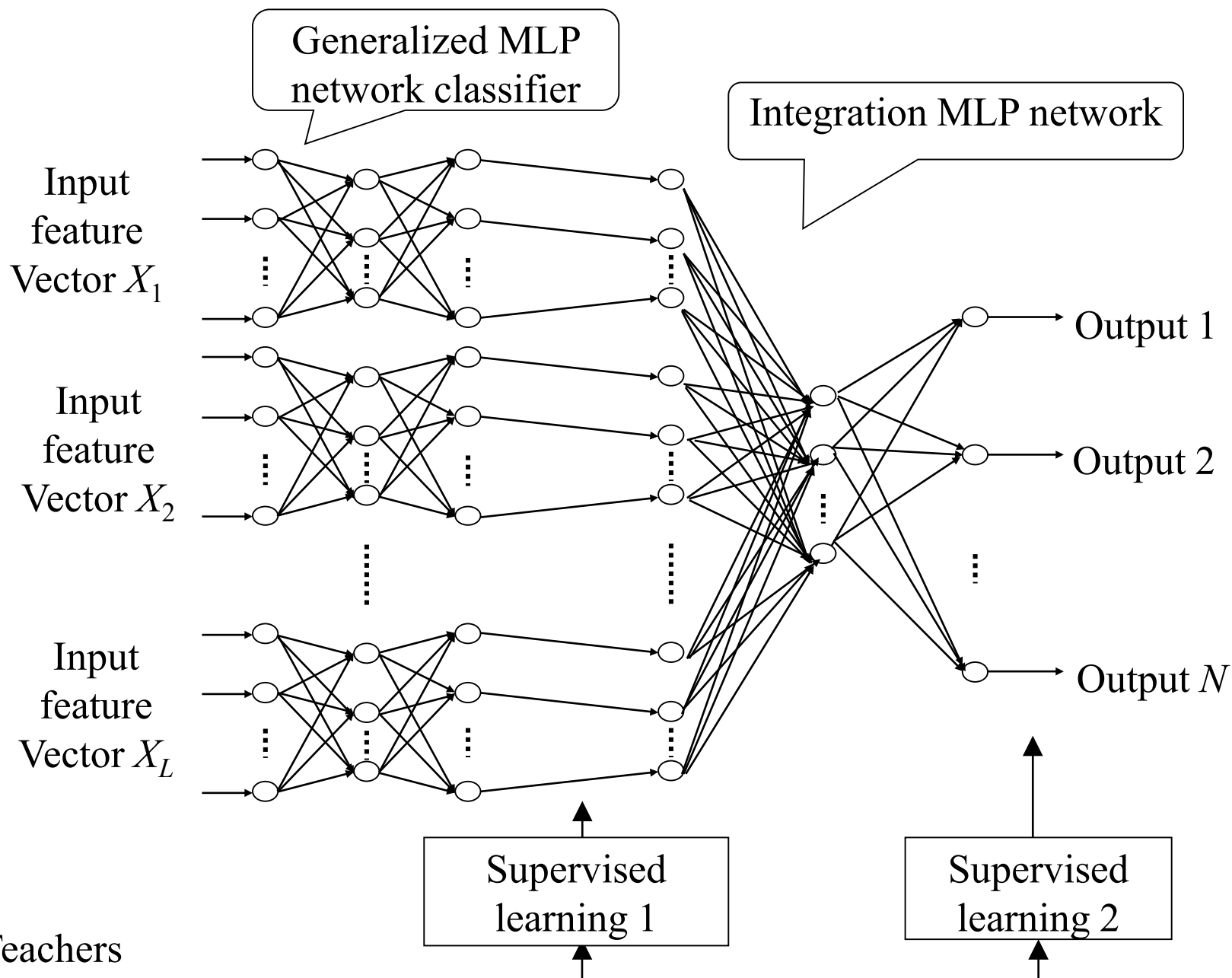
反传（BP）算法示意图



BP算法的基本流程



多反馈神经网络系统结构图



经典神经网络——ART 网络 [Carpenter and Grossberg, 1987]

□ 竞争学习

竞争学习是神经网络中一种常用的无监督学习策略, 在使用该策略时, 网络的输出神经元相互竞争, 每一时刻仅有一个神经元被激活, 其他神经元的状态被抑制.

□ ART 网络 [Carpenter and Grossberg, 1987]

- ART 网络是竞争学习的重要代表
- ART 网络由比较层、识别层、识别阈值和重置模块构成
- 比较层负责接收输入样本, 并将其传送给识别层神经元
- 识别层每个神经元对应一个模式类, 神经元数目可在训练中动态增长以增加新的模式类。

经典神经网络——ART 网络 [Carpenter and Grossberg, 1987]

□ ART 网络性能依赖于识别阈值

- 识别阈值高时, 输入样本将会分成比较多、得到较精细分类
- 识别阈值低时, 输入样本将会分成比较少、产生较粗略分类

□ ART 网络的优势

- ART较好的解决了竞争学习中的“可塑性-稳定性窘境”, 可塑性是指神经网络要有学习新知识的能力; 稳定性是指神经网络在学习新知识时要保持对旧知识的记忆.
- ART 网络可以增量学习或在线学习

□ ART 网络的发展

ART2 网络、FuzzyART 网络、ARTMAP 网络

写在下一讲之前的内容...

写在下一讲之前的内容

