机器学习与神经网络学习笔记

Author	Taosheng Shi
WeChat Contact	data-lake
Mail Contact	tshshi@126.com
Organization	NOKIA
Document category	Distributed System
Document location	https://github.com/stone-note/articles

Version	Status	Date	Author	Description of changes
0.1	Draft	12/7/2017	Taosheng Shi	Initiate
0.2	Draft	DD-MM-YYYY	YourNameHere	TypeYourCommentsHere
1.0	Approved	DD-MM-YYYY	YourNameHere	TypeYourCommentsHere

Contents

1 人工智能简史

Artificial Intelligence (AI)，是在 1956 年的达特茅斯会议上提出来的，标志着人工智能这一学科的诞生。

从 1956 年到 2016 年，刚好是 60 年。在过去的 60 年里，人工智能经历了三个阶段：

- 二十世纪五十年代到七十年代：推理期，其出发点是，数学家真聪明。让计算机具有逻辑推理能力：为什么仅有逻辑推理能力不能实现人工智能？困难在哪里？
- 二十世纪七十年代中期开始：知识期，其出发点是，知识就是力量。让计算机具有知识：由人把知识总结出来，再教给计算机——这是相当困难的。
- 二十世纪九十年代到现在：学习期，其出发点是，让系统自己学。

同时，也催生了人工智能的三大派别：

- 符号主义：主要内容是关于符号计算、演算和逻辑推理，用演算和推理的办法来证明。比如说像机器证明就是符号主义。
- 连接主义：目前非常流行的神经网络、神经元网络、深度学习，这些都是连接主义。
- 行为主义：行为主义其实是从控制论衍生出来的，目前提及较少，但不能忽略。

注：关于学派的分法，《终极算法》一书分为五类：符号学派，联结学派，进化学派，贝叶斯学派和类推学派。

人工智能的三个派别和三个阶段并没有对应和界限，三个派别是在三个阶段的交织中发展起来的。著名信息论和人工智能专家钟义信在《弘扬 Simon 的源头创新精神，开拓 AI 的新理念新路径》报告中指出三大学派的的出现是一直以来还原论把复杂的系统分而治之研究的结果。因为整体上解决智能问题在物理和数学上都存在巨大的困难，所以在模仿大脑的功能研究上，出现了符号主义；在模仿大脑结构的研究上，出现了连接主义，在模仿人类行为的研究上（什么样的环境刺激会产生什么样的行为反应），出现了行为主义。

注：看待人工智能的历史，要把人工智能的历史和神经网路的历史稍微区分一下，不能把神经网络的历史看作是人工智能的历史。所以本文不单独列举神经网络的发展历史和重大事件，留在下一篇文章中探讨。

人工智能发展的过程中，经历了三次大事件，这些大事件导致了人工智能的发展进入三次低谷，被称为"AI winter"：

- 1973 年，英国发表了 James Lighthill 报告，批评人工智能研究进展令人失望，建议取消机器人的研究。为了回应批评和国会的压力，美国和英国政府停止了人工智能研究的资助。
- 1992 年，日本智能（第五代）计算机的研制宣告失败。这次失败有一个收获，是在潘云鹤《人工智能走向 2.0》一文指出的，这次失败表明：驱动人工智能的发展主要靠创新的知识和软件，硬件的作用是支持其运行。
- 在 80 年代，也诞生了 cyc 项目，一个包含所有人类常识的数据库。该项目随着互联网搜索引擎的崛起而衰败。潘云鹤在《人工智能走向 2.0》指出：海量知识不能靠专家人工表达，要从环境中自动学习。也就是周志华指出的：由人把知识总结出来，再教给计算机——这是相当困难的。

在过去的 60 年里，人工智能领域共有 8 位科学家成为图领奖得主：

- 1969，Marvin Minsky：奖励他在创造，塑造，推动和加速人工智能这一领域的核心作用。
- 1971，John McCarthy：麦卡锡的讲座《人工智能的研究现状》概括了他在人工智能领域的成就，也概括了值得奖励的原因。
- 1975，Allen Newell and Herbert A. Simon：奖励他们在二十多年的联合科学工作中，最初与兰德公司的 JC Shaw 合作，随后与卡内基梅隆大学的众多教师和学生同事合作，对人工智能，人类认知心理学和列表处理方面做出的基础贡献。
- 1994，Edward Feigenbaum and Raj Reddy：奖励他们在开创了大规模人工智能系统的设计和建造，展示了人工智能技术的实际重要性和潜在的商业影响。
- 2010，Leslie G. Valiant：奖励他对于计算理论的变革性贡献，包括可能近似正确（PAC）学习的理论，枚举和代数计算的复杂性以及并行和分布式计算的理论。
- 2011，Judea Pearl：奖励他对人工智能的基础贡献：概率和因果推理的微积分。

上面这 8 位科学家，Marvin Minsky 是 MIT 教授，最早提出连接主义，后来发表的《Perceptrons》一书指出感知机无法处理异或问题，导致连接主义长时间陷入低谷。不过著名信息论和人工智能专家钟义信说，另一个方面来看，马文·明斯基指出这个问题以后，经过人们的研究，提出了所谓的多层感知机，我们只要增加一个顶层就可以极大地提高神经网络表达的能力，可以逼近任意的问题。所以这个事情又从它的负面走向了正面，产生了积极的效果。

John McCarthy，Allen Newell， Herbert A. Simon、Edward Feigenbaum 几位都是非常典型的符号主义代表，他们最早推动了机器证明、人工智能、通用人工智能机、知识工程的进步。

注：值得一提的是 Herbert A. Simon 是美国卡内基－梅隆大学心理学教授，1978 年诺贝尔奖金获得者（经济学）。1968-1972 年任美国总统科学顾问、行为科学和人工智能的创始人之一。西蒙教授为科学界的知名学者，在企业管理、计算机设计和决策理论方面有所创见。

Raj Reddy 主要是做语音识别的，李开复、沈向阳的老师。

Leslie G. Valiant 的贡献是机器学习理论，Judea Pearl 的贡献是概率计算和因果推理，高文院士说，他们的工作是未来人工智能的重点走向。

以上从分别从三个时期，三大学派，三次大事件以及 8 位图领奖得主的角度，总结了人工智能的简史。

本部分参考文献：

《人工智能走向 2.0》 潘云鹤

《类脑智能研究的回顾与展望》 曾毅等

《脑启发计算》苏中

《机器学习》序言 陆汝钤

《机器学习：发展与未来》周志华

《H. A. Simon 学术生平》林建祥

《Simon 的认知科学思想》傅小兰

《人工智能--螺旋上升的 60 年》高文院士

《沿 Simon 开拓下去》李衍达

《塞蒙终生学术经历简介》林建祥

《人工智能的历史》中国人工智能学会

《司马贺的创新之路》史忠植

《弘扬 Simon 学术思想 》钟义信

《探寻大师足迹，一览马文•明斯基学术风采》史忠植

《站在巨人的肩膀上，从人工智能与认知商务》苏中

《弘扬 Simon 的源头创新精神开拓“AI”的新理念新路径》钟义信

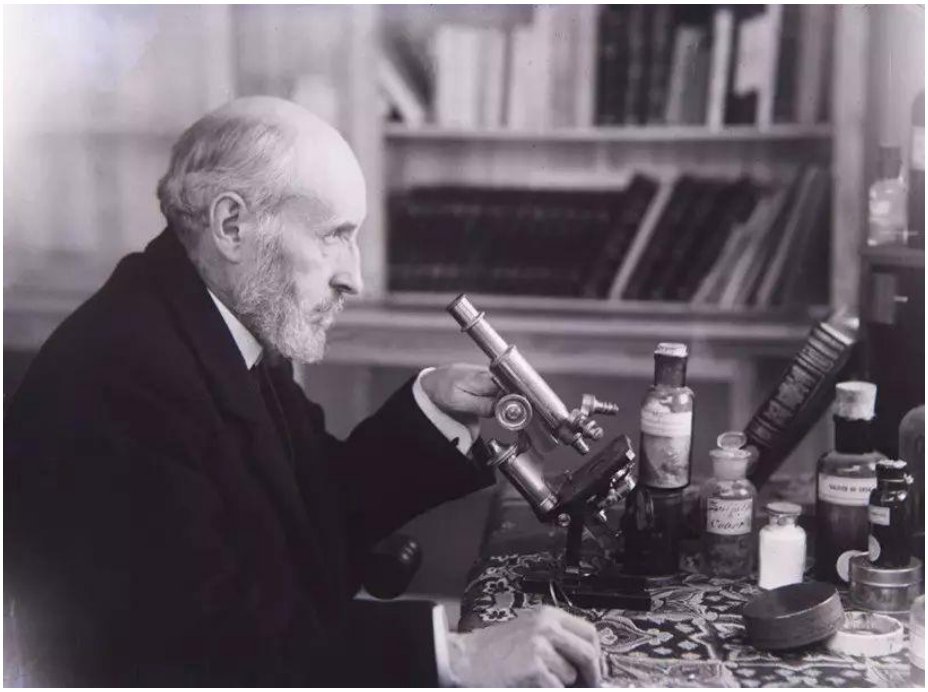
《独家 | 周志华：深度学习很有用，但过度追捧就有危险了》AI 科技大本营

2 神经网络简史

追根溯源，神经网络诞生于人类对于人脑和智能的追问。而这个追问经历了旷远蒙昧的精神至上学说，直到 19 世纪 20 年代。

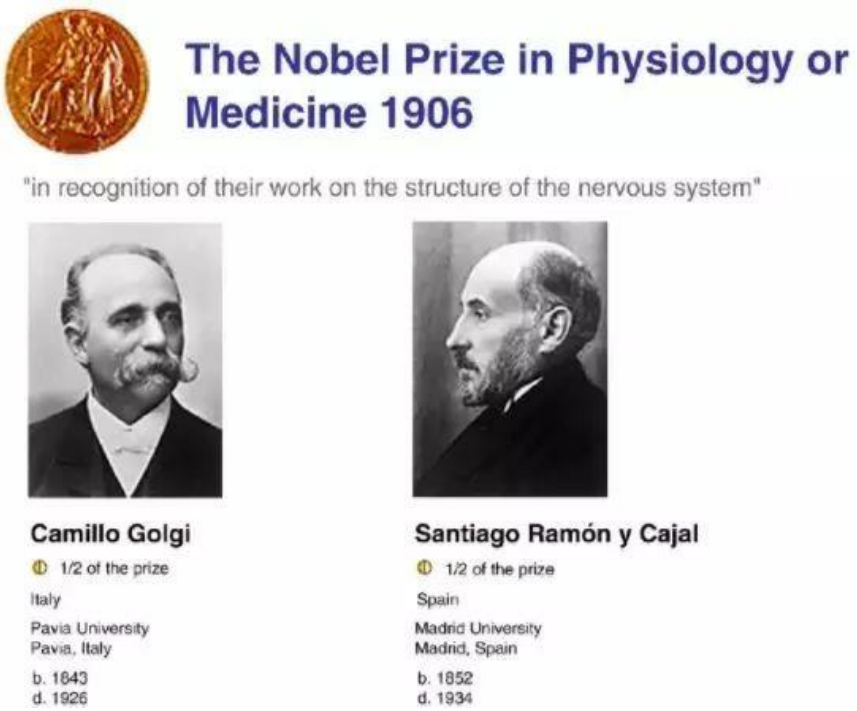
奥地利医生 Franz Joseph Gall （ 1758-1828 ） 推测人类的精神活动是由脑的功能活动而实现的，这才使人们认识到意识和精神活动具有物质基础，从而使人们对精神活动的认识从唯心主义的错误观点转到了唯物主义的正确轨道上来。

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。

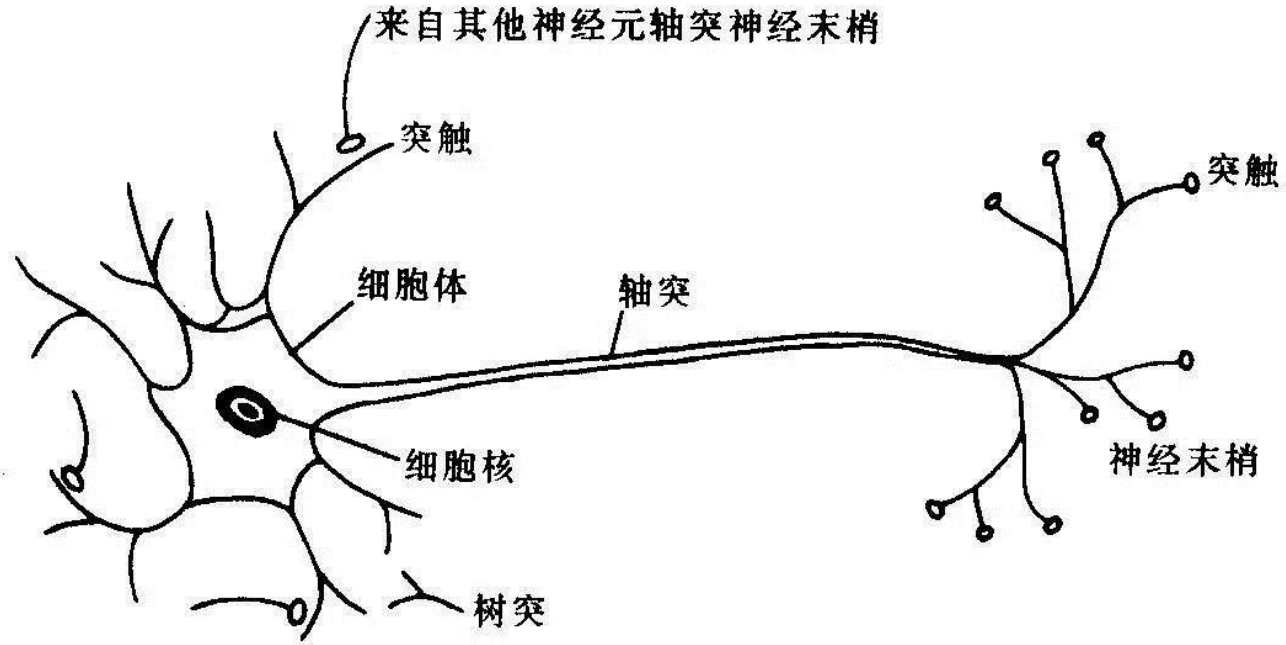


西班牙神经组织学家 Santiago Ramón y Cajal （ 1852~1934 ）在掌握了 Golgi 染色法后，又进一步改良了 Golgi 染色法，并发明了独创的银染法——还原硝酸银染色法， 此法可显示神经纤维的微细结构 。他发现神经细胞之间没有原生质的联系，因而提出神经细胞是整个神经活动最基本的单位（故称

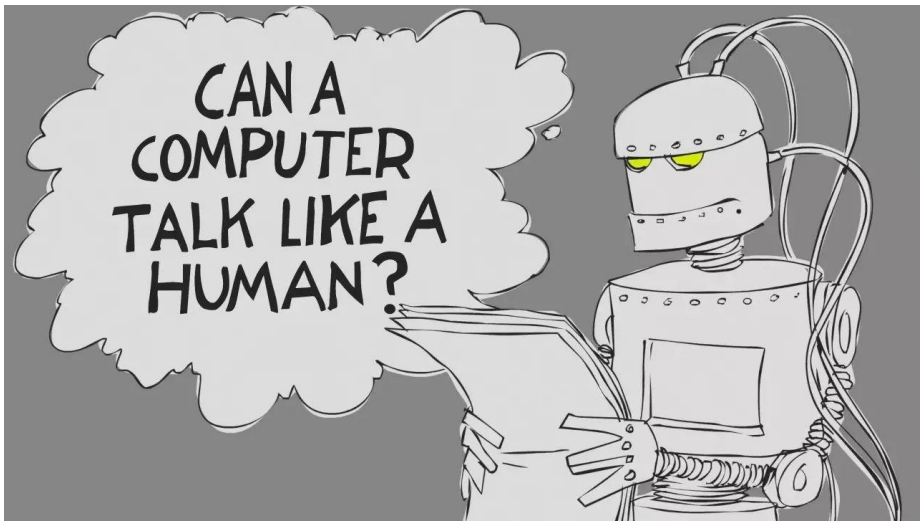
神经元），从而使复杂的神经系统有了进一步研究的切入口。他对于大脑的微观结构研究是开创性的，被许多人认为是现代神经科学之父。他绘图技能出众，他的关于脑细胞的几百个插图至今用于教学。



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

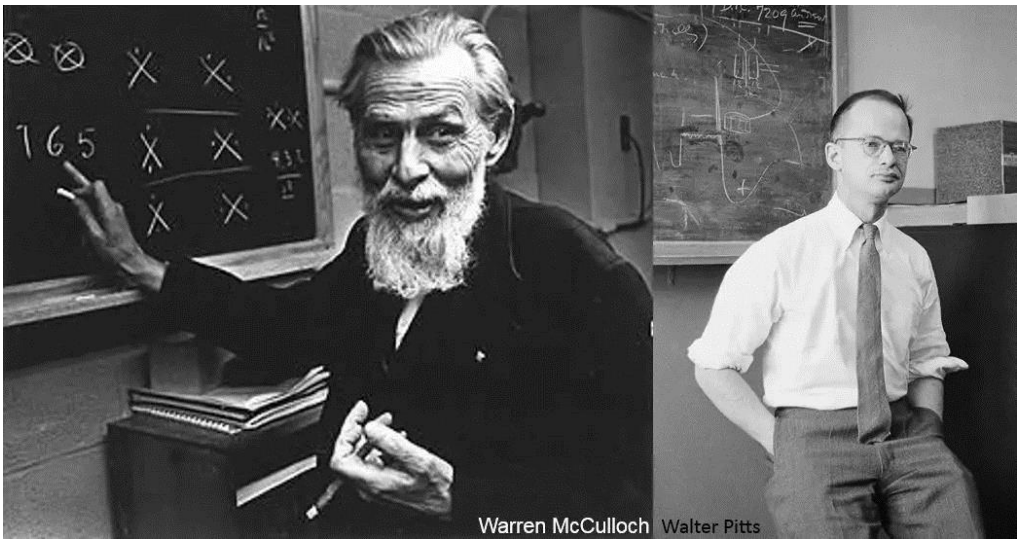


此后， Cajal 经过大量精细的实验，创立了 “ 神经元学说 ” ，该学说的创立为神经科学的进一步发展开创了新纪元。

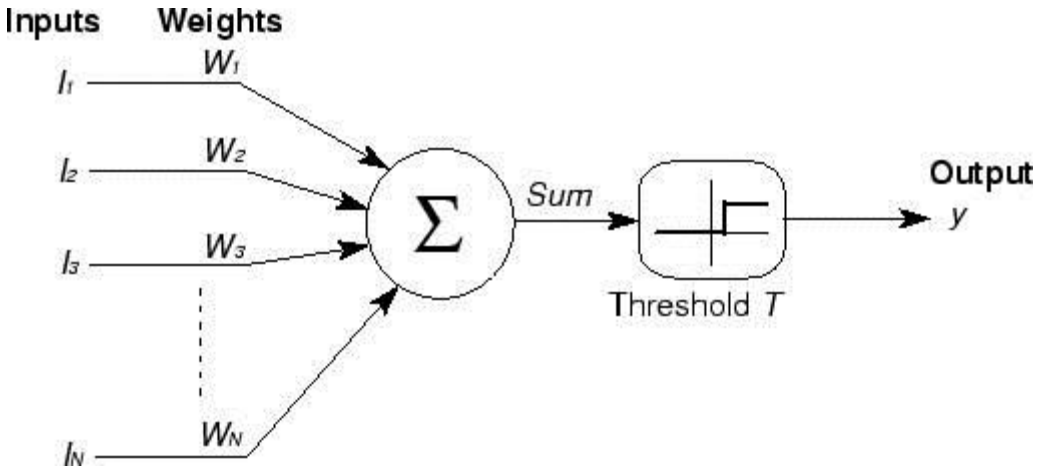


对智能机器的探索和计算机的历史一样古老。尽管中文里“电脑”一开始就拥有了“脑”的头衔，但事实上与真正的智能相去甚远。艾伦图灵在他的文章《COMPUTING MACHINERY AND INTELLIGENCE》中提出了几个标准来评估一台机器是否可以被认为是智能的，从而被称为“图灵测试”。

神经元及其连接里也许藏着智能的隐喻，沿着这条路线前进的人被称为连接主义。

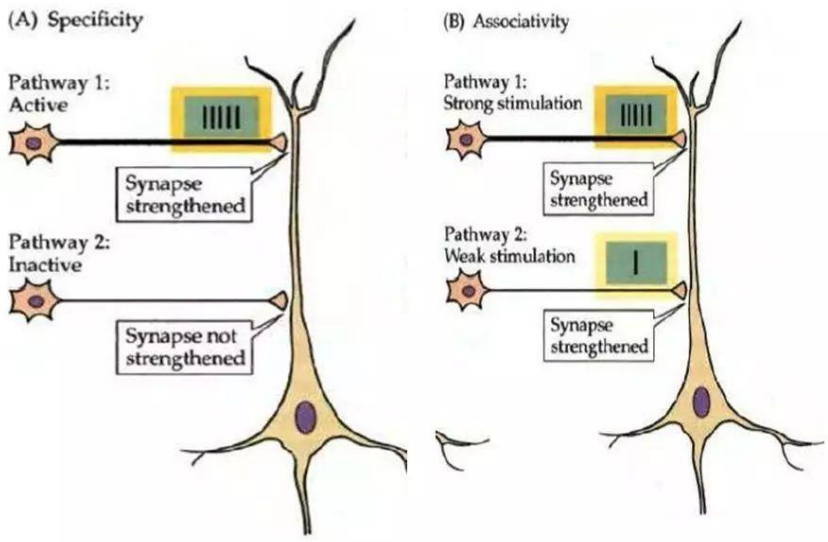


1943 年，Warren McCulloch 和 Walter Pitts 发表题为《A Logical Calculus of the Ideas Immanent in Nervous Activity》的论文，首次提出神经元的 M-P 模型。该模型借鉴了已知的神经细胞生物过程原理，是第一个神经元数学模型，是人类历史上第一次对大脑工作原理描述的尝试。



M-P 模型的工作原理是神经元的输入信号加权求和，与阈值比较再决定神经元是否输出。这是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。

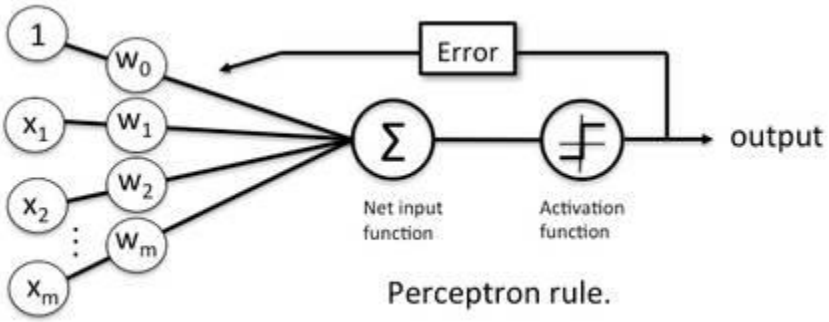
20 世纪 40 年代末，Donald Olding Hebb 在《The Organization of Behavior》中对神经元之间连接强度的变化进行了分析，首次提出来一种调整权值的方法，称为 Hebb 学习规则。



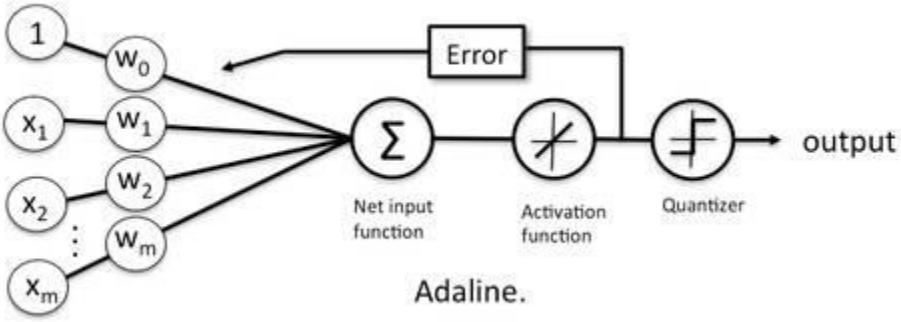
Hebb 学习规则主要假定机体的行为可以由神经元的行为来解释。Hebb 受启发于巴普罗夫的条件反射实验，认为如果两个神经元在同一时刻被激发，则它们之间的联系应该被强化。这就是 Hebb 提出的生物神经元的学习机制，在这种学习中，由对神经元的重复刺激，使得神经元之间的突触强度增加。

Hebb 学习规则隶属于无监督学习算法的范畴，其主要思想是根据两个神经元的激发状态来调整期连接关系，以此实现对简单神经活动的模拟。继 Hebb 学习规则之后，神经元的有监督 Delta 学习规则被提出，用于解决在输入输出已知的情况下神经元权值的学习问题。

1958 年，就职于 Cornell 航空实验室的 Frank Rosenblatt 发明了的一种称为感知器（Perceptron）的人工神经网络。它可以被视为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 sign(x)）。感知机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。



Perceptron rule.



Adaline.

这次成功的应用也引起了许多学者对神经网络的研究兴趣。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。同一时期，Steinbuch 等还提出了称为学习矩阵的二进制联想网络。

周志华《机器学习》第 5 章神经网络解释了 BP 算法实质是 LMS 算法（Least Mean Square）算法的推广。LMS 试图使网络的输出均方差最小化，可用于神经元激活函数可微的感知机学习；将 LMS 推广到由非线性可微神经元组成的多层前馈神经网络，就得到 BP 算法，因此 BP 算法也被称为广义 δ 规则。

1969 年，Marvin Minsky 和 Seymour Papert 发表《Perceptrons: an introduction to computational geometry》一书，从数学的角度证明了单层神经网络具有有限的功能，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。

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。

1982 年，John Hopfield 提出了连续和离散的 Hopfield 神经网络模型，并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，Hinton, G. E. 和 Sejnowski, T. J.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，这就是玻尔兹曼机。

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。

一个沉睡十年的伟大算法即将被唤醒。

1986 年，David E. Rumelhart, Geoffrey E. Hinton 和 Ronald J. Williams 发表文章《Learning representations by back-propagating errors》，重新报道这一方法，BP 神经网络学习算法才受到重视。BP 算法引入了可微分非线性神经元或者 sigmod 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。

BP 算法的性能分析文章：

G. Cybenko. 《Approximation by Superpositions of a Sigmoidal Function》

Funahashi, K-I. 《On the approximate realization of continuous mappings by neural networks》

Kur Hornik, Maxwell Stinchcombe and Halber White. 《Multilayer feedforward networks are universal approximators 》

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，并证明对于具有单隐层，传递函数为 sigmod 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。

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。Vapnik, V.N.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，随着其理论的不断发展和成熟，也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。同时, 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)，它已初步表现出很多优于已有方法的性能。

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。值得一提的是，1997 年，Sepp Hochreiter 和 Jurgen Schmidhuber 首先提出长短期记忆（LSTM）模型。

直到 2006 年，计算机处理速度和存储能力大大提高，为深度学习的提出铺平了道路。G. E. Hinton 和他的学生 R. R. Salakhutdinov 在《科学》杂志上发表题为《Reducing the Dimensionality of Data with Neural Networks》的文章，掀起了深度学习在学术界和工业界的研究热潮。文章摘要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。从此，历史这样写就：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。

总结起来，典型的浅层学习模型包括：传统隐马尔可夫模型（HMM）、条件随机场（CRFs）、最大熵模型（MaxEnt）、boosting、支持向量机（SVM）、核回归及仅含单隐层的多层感知器（MLP）等。

同年，G. E. Hinton 又提出了深度信念网络(Deep BeliefNetwork, DBN) 。深度信念网络基于受限玻尔兹曼机构建。

限制玻尔兹曼机（RBM）是一种玻尔兹曼机的变体，但限定模型必须为二分图。模型中包含对应输入参数的输入（可见）单元和对应训练结果的隐单元，图中的每条边必须连接一个可见单元和一个隐单元。与此相对，“无限制”玻尔兹曼机(BM)包含隐单元间的边，使之成为递归神经网络。BM 由 Geoffrey Hinton 和 Terry Sejnowski 在 1985 年发明，1986 年 Paul Smolensky 命名了 RBM，但直到 Geoffrey Hinton 及其合作者在 2006 年左右发明快速学习算法后，受限玻兹曼机才变得知名。

自动编码器早在 1986 年就被 Rumelhart 等人提出（也有资料说第一个自动感应器是福岛神经认知机），2006 年之后，G. E. Hinton 等人又对自动编码器进行改造，出现了深度自编码器，稀疏自编码

器等。2008 年，Pascal Vincent 和 Yoshua Bengio 等人在《Extracting and composing robust features with denoising autoencoders》中提出了去噪自编码器，2010 年又提出来层叠去噪自编码器。2011 年，Richard Socher 等人也提出了递归自编码器 (RAE)。

目前，卷积神经网络作为深度学习的一种，已经成为当前图像理解领域研究的热点。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。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》 。

GRAVERS A, MOHAMED A, HINTON G. 《Speech recognition with deep recurrent neural networks 》

XU K, BA J, KIROS R, et al. 《Show, attend and tell: neural image caption generation with visual attention 》

PINHEIRO P, COLLOBERT R. 《Recurrent convolutional neural networks for scene labeling》

HE K M, ZHANG X, REN S, et al. 《Deep residual learning for image recognition》

2013 年，Graves 证明，结合了长短时记忆(long short terms memory, LSTM) 的递归神经网络(recurrent neural network, RNN)比传统的递归神经网络在语音处理方面更有效。2014 年至今， 深度学习在很多领域都取得了突破性进展，发展出了包括注意力(attention)，RNN--CNN， 以及深度残差网络等多种模型。

最后，给出一个神经网络发展历史的回顾总结：

本部分参考文献：

《神经网络七十年：回顾与展望》焦李成等

《深度学习的昨天、今天和明天》余凯等

《卷积神经网络研究综述》周飞燕等

《卷积神经网络研究综述》李彦东等

《神经网络新理论与方法》张代远

《机器学习》周志华

《图像理解中的卷积神经网络》常亮等

《深度强化学习综述：兼论计算机围棋的发展》赵东斌等

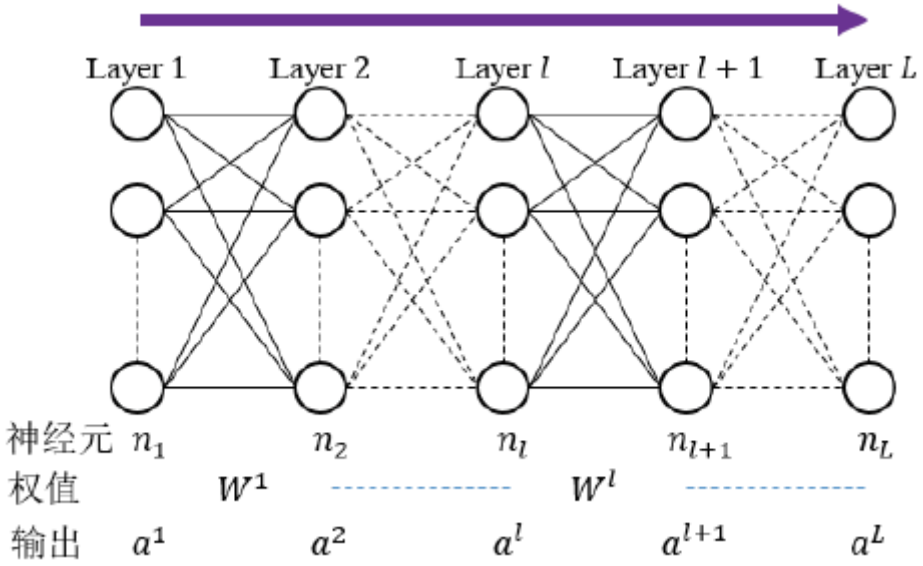
《深度学习研究综述》孙志军等

3 误差反向传播算法浅解

反向传播（英语：Backpropagation，缩写为 BP）是“误差反向传播”的简称。由于多层前馈神经网络的训练经常采用误差反向传播算法，人们也常把多层前馈神经网络称为 BP 网络。

反向传播算法发明的历史请参考我的前文《神经网络简史》。

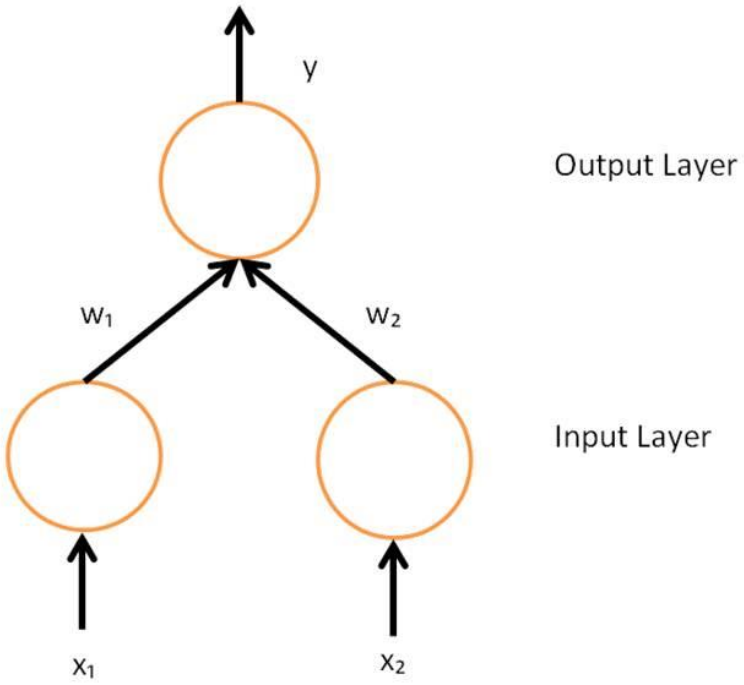
多层前馈神经网络是指通过按某种方式将神经元连接起来，就可构成相关神经网络。下图所示为一个熟知的前馈网络模型，该网络有 L 层，第 1 层为输入层，第 L 层为网络输出层。在这个网络中，前一层神经元全连接到后一层神经元，同层的神经元之间没有任何连接。



反向传播通常被认为是基于优化理论的一种监督式学习方法，虽然它也用在一些无监督网络（如自动编码器）中。

3.1 直观理解

考虑一个有两个输入单元、一个输出单元、没有隐藏单元的简单神经网络。每个神经元都使用输入的加权和作为线性输出。



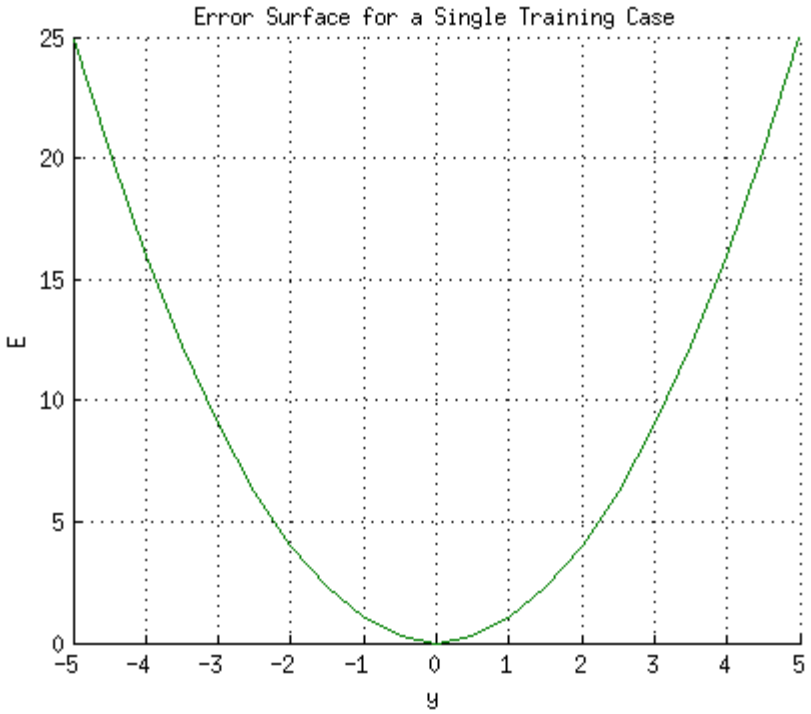
最初在训练之前，会随机分配权重。之后神经元根据训练实例进行学习，在此情况下包含元组 (x1,x2,t) 的集合，其中 x1 与 x2 是网络的输入，t 为正确输出（在给定相同的输入时网络最终应当产生的输出）。网络在给定 x1 和 x2 时，会计算一个输出 y，很可能与 t 不同（因为权重最初是随机的）。衡量期望输出 t 与实际输出 y 之间的差异的一个常见方法是采用平方误差测度：

$$E = (y - t)^2$$

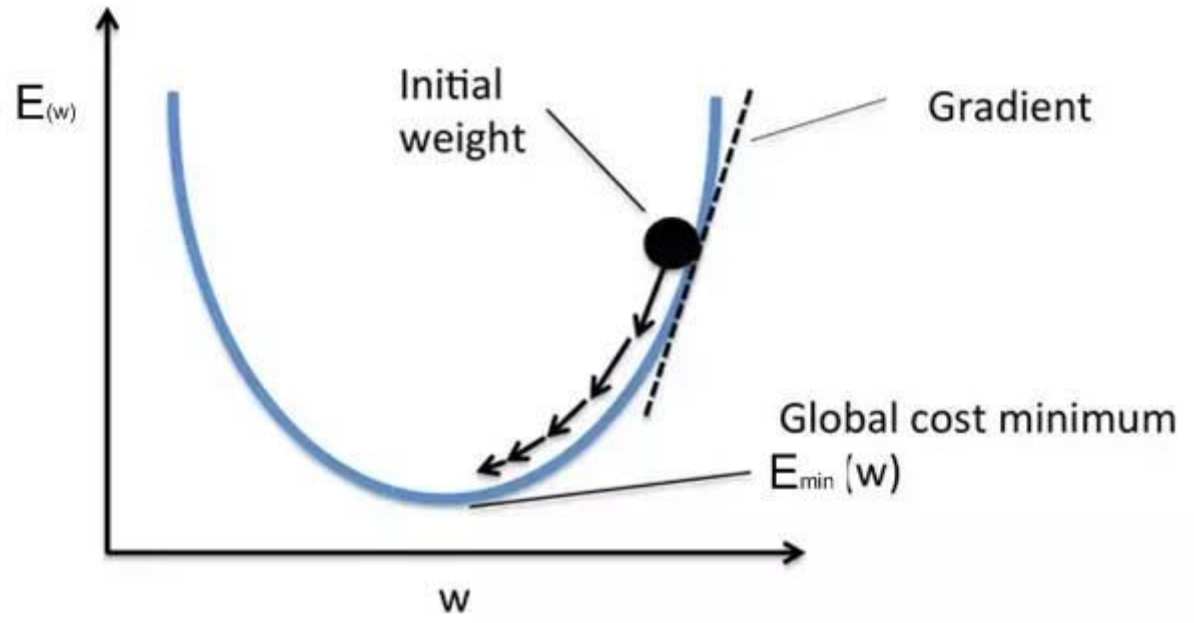
其中 E 为差异或误差。

为什么采用平方差？其数学背景是最小二乘法，也可以理解为空间两点的距离或者平方误差等。最小二乘法（又称最小平方法）是一种数学优化技术。它通过最小化误差的平方和寻找数据的最佳函数匹配。最重要的应用是在曲线拟合上。最小平方所涵义的最佳拟合，即残差（残差为：观测值与模型提供的拟合值之间的差距）平方总和的最小化。

举例来讲，考虑单一训练实例的网络：(1,1,0)，输入 x1 与 x2 均为 1，正确输出 t 为 0（网络只有一个输出）。现在若将实际输出 y 画在 x 轴，误差 E 画在 y 轴，得出的是一条抛物线。抛物线的极小值对应输出 y，最小化了误差 E。对于单一训练实例，极小值还会接触到 x 轴，这意味着误差为零，网络可以产生与期望输出 t 完全匹配的输出 y。因此，把输入映射到输出的问题就化为了一个找到一个能产生最小误差的函数的最优化问题。



单一实例的神经网络的误差函数非常容易理解，可以通过解方程，直接找到最小值。这里给一个梯度表示，如下图，便于理解多实例神经网络的梯度表示。



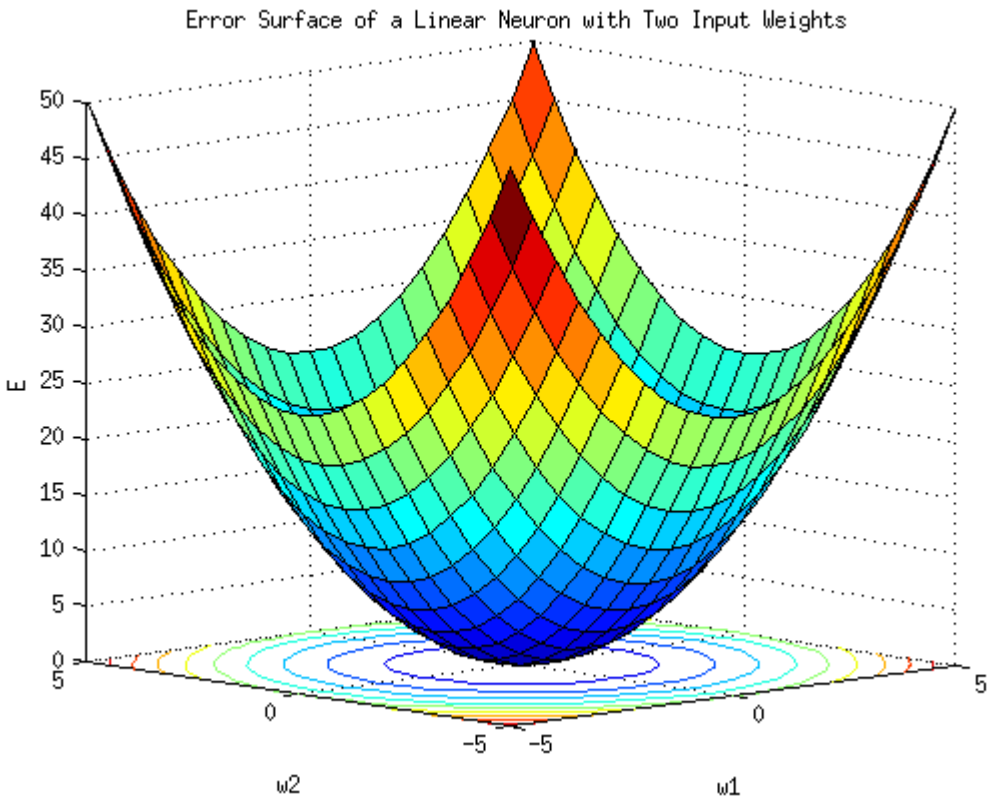
反向传播算法的目的是找到一组能最大限度地减小误差的权重。寻找抛物线或任意维度中任何函数的极大值的方法有若干种。其中一种方法是通过求解方程组，但这依赖于网络是一个线性系统，而目标也需要可以训练多层非线性网络（因为多层线性网络与单层网络等价）。

如果考虑两个实例呢（依然是单层神经网络，不考虑非线性变换）？

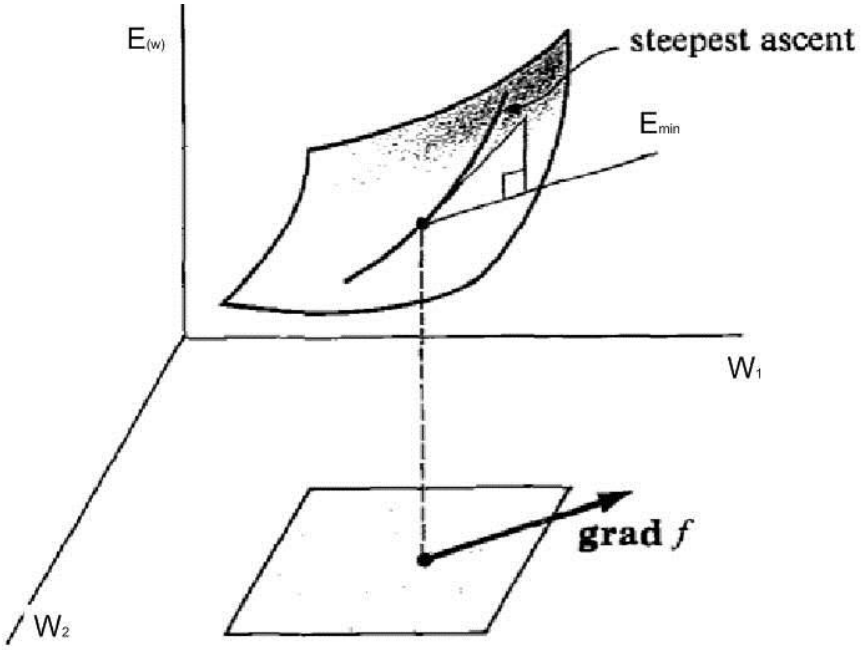
考虑一个神经元的输出取决于其所有输入的加权总和：

$$y = w_1x_1 + w_2x_2$$

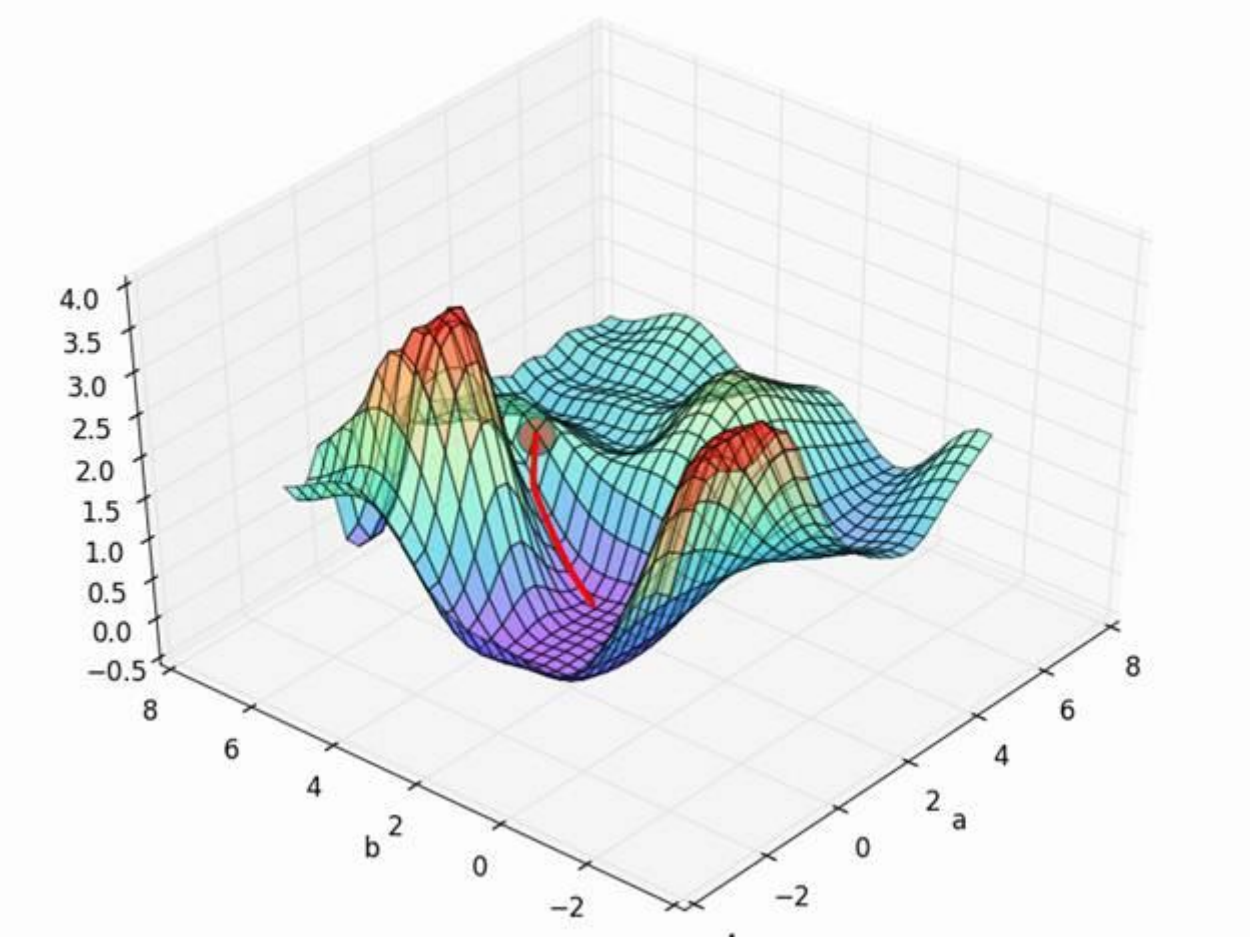
其中 w1 和 w2 是从输入单元到输出单元相连的权重。因此，误差取决于输入到该神经元的权重，也是网络要学习最终需要改变的。若每个权重都画在一个水平的轴上，而误差画在垂直轴上，得出的就是一个抛物面（若一个神经元有 k 个权重，则误差曲面的维度就会是 k+1，因而就是二维抛物线的 k+1 维等价）。



抛物面的最小值需要通过梯度下降法求得。如下图所示。



如果是多个实例呢？并且是多层神经网络的非线性变换呢？从数学角度看，已经不能通过求解方程组得到最小值，也不能简单的描绘多维权重系数构成的函数所对应的几何形状（比如抛物面）。但是运用抽象推理，大概想象成这样子：

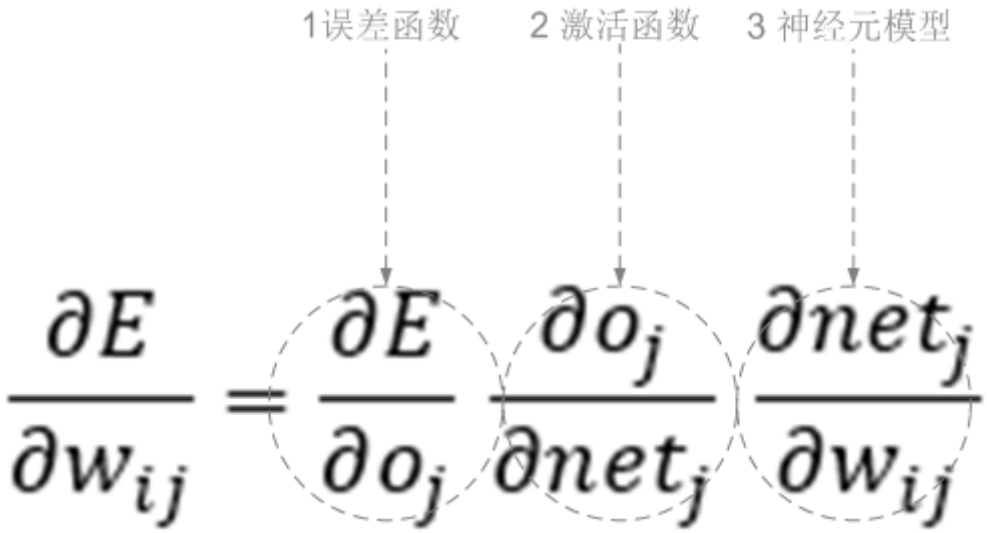


以上就是神经网络学习和误差函数原理的直观表示。

结合梯度的概念，这里先给出梯度下降的推导结果（公式所示），下面逐步解释为什么有这个推导结果。

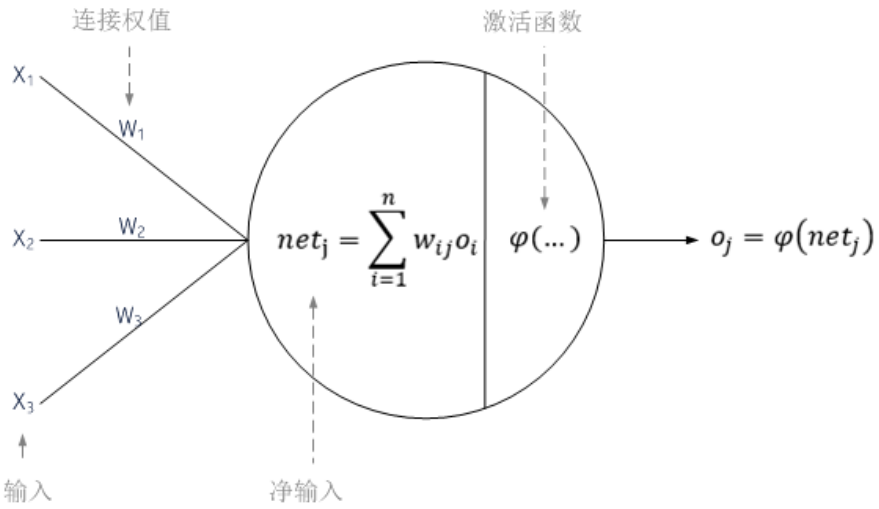
$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}}$$

这个公式推导所带出的三个导数非常重要，是影响梯度的三个重要因素。我把上面三个偏导分成三部分来讨论，分别对应于误差函数，激活函数，神经元模型。假设我们要得到的是第 i 层到第 j 层的连接权重 w_ij，用于梯度的偏导可以通俗的表达为：



下面分别从神经元模型，误差函数和激活函数的角度解释这个公式。

一个神经元的基本模型如下图所示。



通向一个神经元的输入 net_j 是之前神经元的输出 o_i 的加权和。若该神经元输出层后的第一层，输入层的输出 o_i 就是网络的输入 x_i。该神经元的输入数量是 n。变量 w_ij 表示神经元 i 与 j 之间的权重。对于每一个神经元，其对应的输出为：

$$net_j = \sum_{i=1}^{n} w_{ij} o_i$$

$$o_j = \varphi(net_j) = \varphi\left(\sum_{i=1}^{n} w_{ij} o_i\right)$$

从数学的角度来看，神经元模型定义了两层的复合函数：内层是 net_k，外层是 φ 函数。激活函数 φ 一般是非线性可微函数（反向传播要求人工神经元的激励函数可微）。常用作激活函数的是 Sigmoid 函数：

$$\varphi(z) = f(z) = \frac{1}{1 + exp(-z)}$$

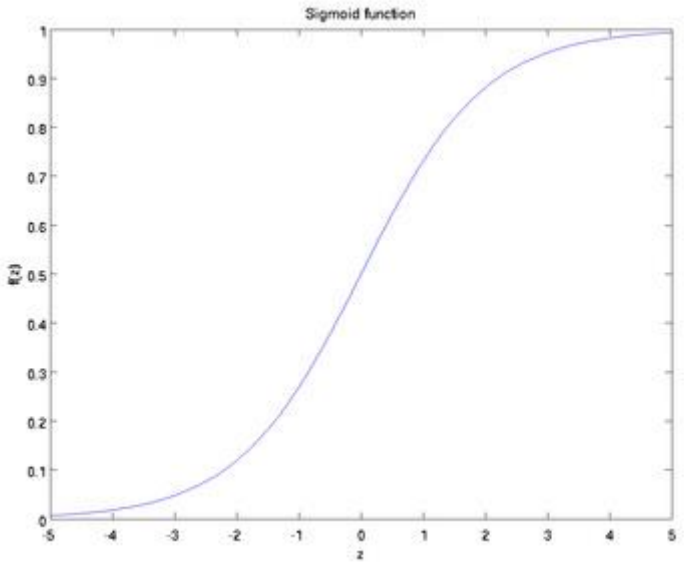
或者写成：

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

这个函数也被称为单极性 Sigmoid 函数。其导数形式很好：

$$\frac{\partial \varphi}{\partial z} = \frac{\partial f}{\partial x} = f(x)(1 - f(x)) = \varphi(1 - \varphi)$$

其图像如何下：



双极性 Sigmoid 函数要比较常用。公式如下：

$$\varphi(z) = f(z) = \frac{exp(z) - exp(-z)}{exp(z) + exp(-z)}$$

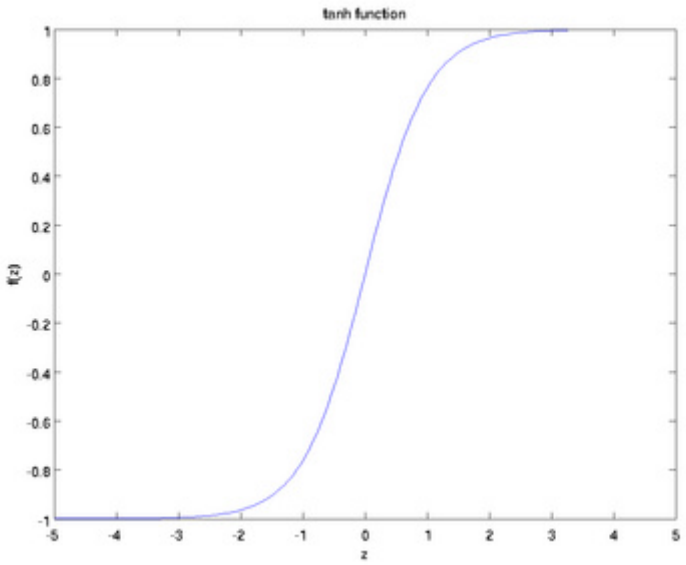
或者写成（把上式中分子分母同时除以 e^z，令 x=-2z 就得到第二个式子）：

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

其导数为：

$$\frac{\partial \varphi}{\partial z} = \frac{\partial f}{\partial x} = 1 - \big(f(x)\big)^2 = 1 - (\varphi)^2$$

其图像为：



从上图可以看出，对于 Sigmoid 函数，当 z 的取值越来越大后（饱和区），函数曲线变得越来越平缓，意味着此时的导数也越来越小。同样的，当 z 的取值越来越小时（饱和区），也有这个问题。仅仅在 z 取值为 0 附近时，导数的取值较大。在后文讲到的反向传播算法中，每一层向前递推都要乘以导数，得到梯度变化值。Sigmoid 的这个曲线意味着在大多数时候，我们的梯度变化值很小，导致我们的 W,b 更新到极值的速度较慢，也就是我们的算法收敛速度较慢。

由于反向传播使用梯度下降法，需要计算平方误差函数对网络权重的导数。假设对于一个输出神经元，平方误差函数为：

$$E = \frac{1}{2}(y - t)^2$$

其中：

- E 为平方误差，
- t 为训练样本的目标输出，
- y 为输出神经元的实际输出。
- 加入系数 1/2 是为了抵消微分出来的指数。之后，该表达式会乘以一个任意的学习速率，因此在这里乘上一个常系数是没有关系的。

由梯度的定义，与方向导数有关联的一个概念是函数的梯度。多元函数的偏导向量构成了梯度，梯度的方向是函数在这点增长最快的方向，那么函数的偏导在这点的梯度方向也达到最大值。其中，要使用梯度下降法更新 w_ij，必须选择一个学习速率 μ 。要加在原本的权重上的变化，等于学习速率与梯度的乘积，乘以-1：

$$\Delta w = -\mu \frac{\partial E}{\partial w_{ij}}$$

$$w = w + \Delta w$$

之所以要乘以-1 是因为要更新误差函数极小值而不是极大值的方向。

从数学的角度看，平方误差函数形成了对输出 o_j 的复合函数：

$$E = (y-t)^2 = (o_j - t)^2 = \left(\varphi(net_j) - t\right)^2 = (\varphi\left(\sum_{i=1}^{n} w_{ij} o_i\right) - t)^2$$

这个式子无论对于输出层的神经元，还是隐藏层的神经元，都是成立的。

综上所述，误差函数是关于权重的函数，为了在由权重系数构成的多维空间中寻找一个下降最快的梯度方向，我们需要对所有权重系数求偏导。根据复合函数的求导规则，其一般形式为：

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}}$$

3.2 反向传播算法的学习过程

学习过程由信号的正向传播与误差的反向传播两个过程组成。正向传播时，输入样本从输入层传入，经各隐层逐层处理后，传向输出层。若输出层的实际输出与期望的输出(教师信号) 不符，则转入误差的反向传播阶段。误差反传是将输出误差以某种形式通过隐层向输入层逐层反传，并将误差分摊给各层的所有单元，从而获得各层单元的误差信号，此误差信号作为修正各单元权值的依据。这种信号正向传播与误差反向传播的各层权值调整过程，是周而复始地进行的。权值不断调整的过程，也就是网络学习训练过程。此过程一直进行到网络输出的误差减少到可接受的程度，或进行到预先设定的学习次数为止。

学习过程的伪码描述如下：

```
输入：训练集和学习率
初始化网络权值（通常是小的随机值）
 do
   forEach 训练样本 ex
      prediction = neural-net-output(network, ex)  // 正向传递，得到当前样本的输出
      actual = teacher-output(ex)//从监督老师那里获得真实输出
      计算输出单元的误差 (prediction - actual)//获得残差
      计算    对于所有隐藏层到输出层的权值和阈值          // 反向传递
      计算    对于所有输入层到隐藏层的权值和阈值          // 继续反向传递
      更新网络权值和阈值 // 输入层不会被误差估计改变
 until 所有样本正确分类或满足其他停止条件
 return 权重与阈值确定的多层前馈神经网络
```

完整的误差反向传播算法包括前向计算和反向传播两部分。

3.3 反向传播算法推导

为了最小化误差 E，最终归结为优化问题。前面说过，反向传播算法的目的是找到一组能最大限度地减小误差的权重，在反向传播中使用的方法是梯度下降法。这样我们就需要计算误差函数 E 对权重的偏导。由上文，误差函数对权重 w_ij 的偏导数是三个偏导数的乘积：

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}}$$

我们的目标就是分别求得这三个偏导。

在右边的最后一项中（神经元模型），只有加权和 net_j 取决于 w_ij，因此：

$$\frac{\partial net_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \left(\sum_{i=1}^{n} w_{ij} o_i \right) = o_i$$

当对一个权重求导时，其他权重就视为常量。这里如有不确定，把加权和展开即可明白。

对于激活函数部分，神经元 j 的输出对其输入的导数就是激活函数的偏导数（这里假定使用 Sigmoid 函数）：

$$\frac{\partial o_j}{\partial net_j} = \frac{\partial}{\partial net_j} \varphi(net_j) = \varphi(net_j) \left(1 - \varphi(net_j)\right) = \varphi(1 - \varphi)$$

这就是为什么反向传播需要的激活函数是可微的。同时，前向传播时，很容易求得 net_j（各层神经元计算得到的加权和），所以该偏导也容易求得。

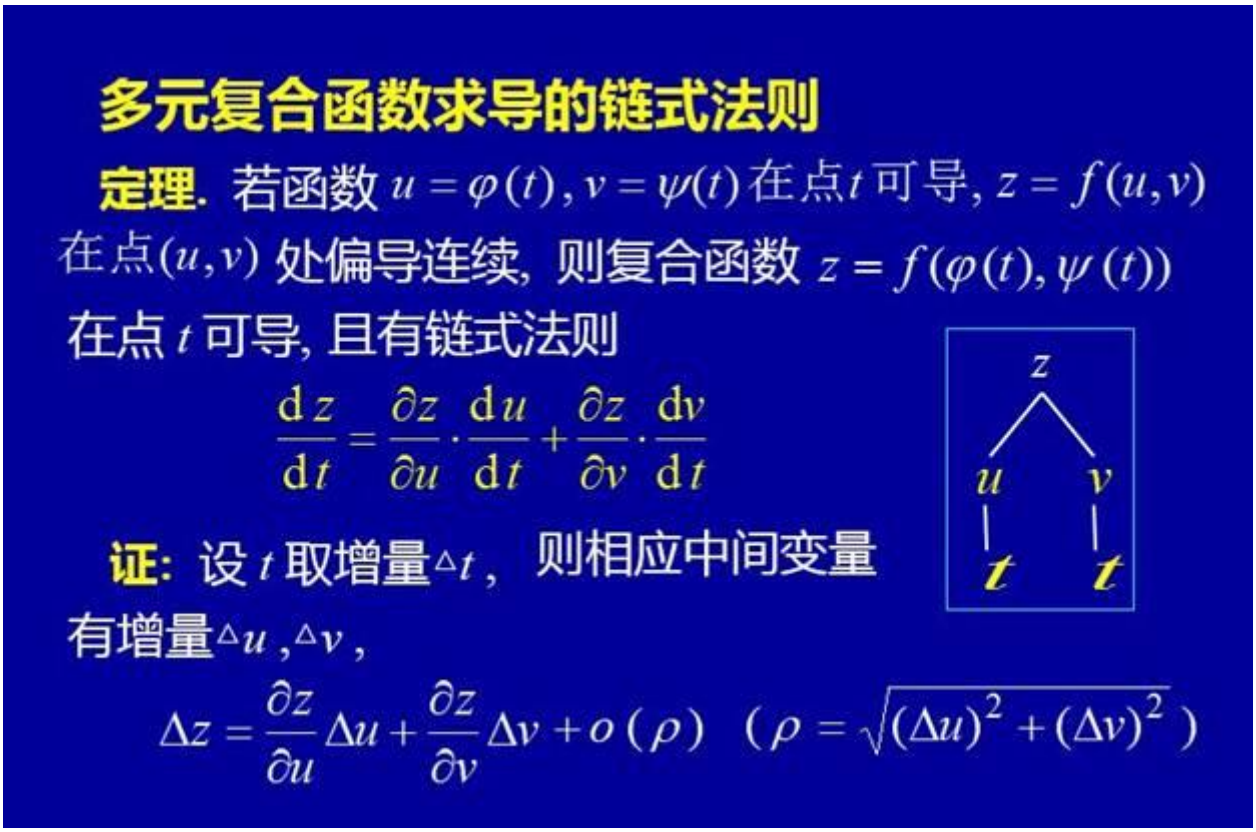
对于误差函数部分的偏导，为了方便理解，分输出层和隐藏层来讨论。

如果神经元在输出层中，因为此时 o_j=y 以及

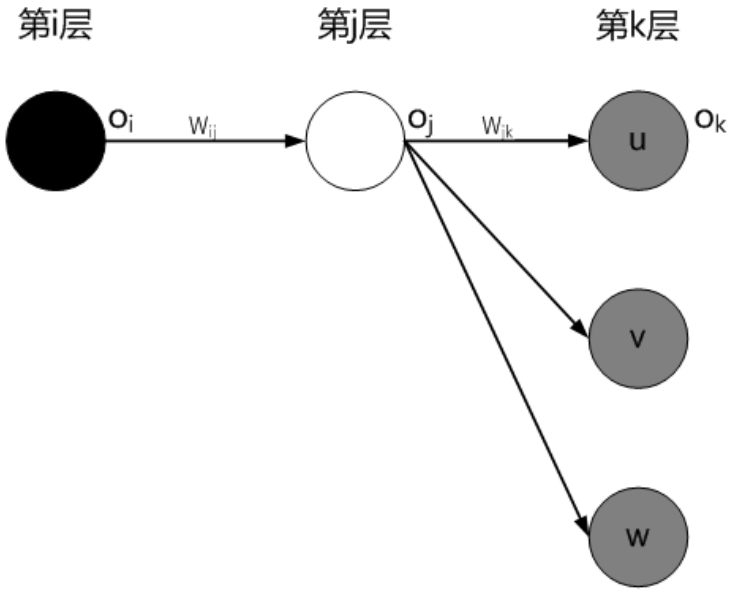
$$\frac{\partial E}{\partial o_j} = \frac{\partial E}{\partial y} = \frac{\partial}{\partial y} \left(\frac{1}{2} (y - t)^2 \right) = y - t$$

所以第一项可以直接算出。

但如果 j 是网络中任一内层（隐藏层），就需要用到链式求导法则。



下面以一个神经网络的局部来说明。神经网络中相邻的两层构成一个计算单元，因此只需要理解第 j 层和第 k(j+1) 层之间的结构和运算，便可以通晓整个网络的结构和运算（前面说过，层与层之间全连接，同层之间没有连接）。



注意到我们要求的是第 i 层到第 j 层的连接权重 w_ij，考虑第 j 层中的某个神经元的输出 o_j 是第 k 层所有神经元{u,v,⋯,w}的输入，把误差（损失）函数 E 看作是{u,v,⋯,w}的函数（注意，这里的第 k 层可能是输出层，也可能是隐藏层；作为输出层时，既可能是多值输出，也可能是单值输出）。可以得到：

$$\frac{\partial E}{\partial o_j}=\frac{\partial E(o_j)}{\partial o_j}=\frac{\partial E(net_u,net_v,\dots,net_w)}{\partial o_j}$$

这里应用链式求导法则：

$$\frac{\partial E}{\partial o_j}=\frac{\partial E(o_j)}{\partial o_j}=\frac{\partial E(net_u,net_v,\dots,net_w)}{\partial o_j}$$

$$\frac{\partial E}{\partial o_j}=\frac{\partial E(o_j)}{\partial o_j}=\frac{\partial E(net_u)}{\partial(net_u)}\frac{\partial(net_u)}{\partial o_j}+\frac{\partial E(net_v)}{\partial(net_v)}\frac{\partial(net_v)}{\partial o_j}+\cdots+\frac{\partial E(net_w)}{\partial(net_w)}\frac{\partial(net_w)}{\partial o_j}$$

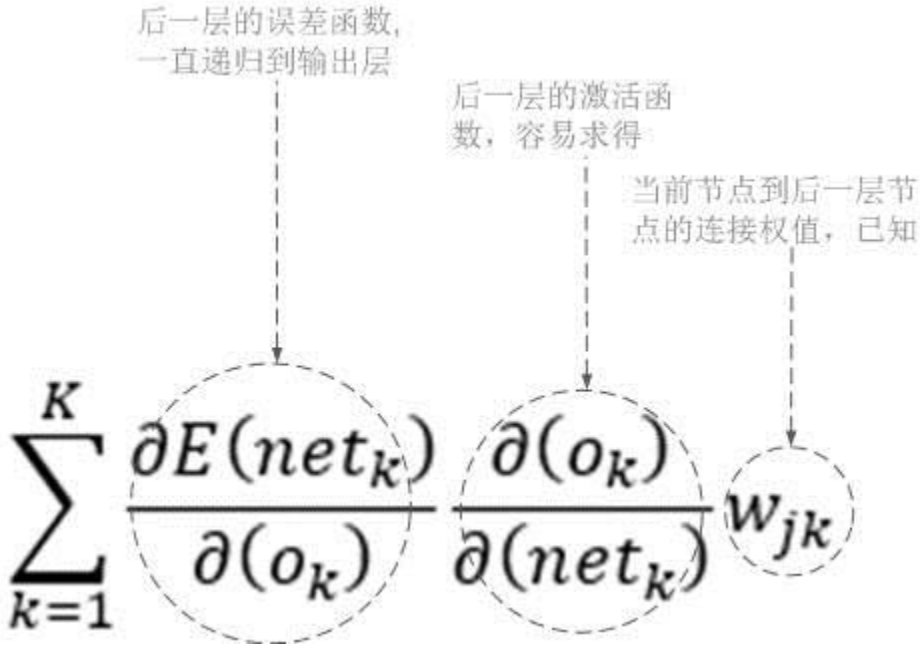
$$\frac{\partial E}{\partial o_j}=\frac{\partial E(o_j)}{\partial o_j}=\sum_{k=1}^{K}\frac{\partial E(net_k)}{\partial(net_k)}\frac{\partial(net_k)}{\partial o_j}=\sum_{k=1}^{K}\frac{\partial E(net_k)}{\partial(o_k)}\frac{\partial(o_k)}{\partial(net_k)}w_{jk}$$

最后一步做了两次替换：

$$\frac{\partial(net_k)}{\partial o_j}=w_{jk}$$

$$\frac{\partial E(net_k)}{\partial(net_k)}=\frac{\partial E(net_k)}{\partial(o_k)}\frac{\partial(o_k)}{\partial(net_k)}$$

对于上面推导结果，我们发现：



因此，若已知所有关于下一层（更接近输出神经元的一层）的输出关于 o_k 的导数，则可以计算 o_j 的导数。

现在把上述推导放在一起：

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial net_j}o_i = \delta_j o_i$$

此时：

$$\delta_j = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial net_j} = (\sum_{k=1}^{K}\frac{\partial E(net_k)}{\partial (o_k)}\frac{\partial (o_k)}{\partial (net_k)}w_{jk})\,\varphi(net_j)\left(1-\varphi(net_j)\right)\,o_i$$

则：

$$\delta_j = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial net_j} = \begin{cases} (y-t)\varphi(net_j)\left(1-\varphi(net_j)\right), & if\ j\ is\ an\ output\ neuron \\ (\sum_{k=1}^{K}\delta_k w_{jk})\varphi(net_j)\left(1-\varphi(net_j)\right), & if\ j\ is\ an\ inner\ neuron \end{cases}$$

综上，权重的调整只和以下因素有关：

- 前向神经元的输出（和所调整权重有关的一个前向神经元的输出作为输入）
- 当前神经元的激活函数
- 所有后向神经元（误差函数导数，激活函数导数，并一直递归下去）及其前向传播时的权重（这些权重用来作为误差后向传播的权重）
- 递归会终结在输出层，从而使用残差(y-t)。注意到递归的层数以及系数作用，这里有一大串连乘，如果连乘的数字小于 1，则梯度越乘越小，导致梯度消散；如果连乘的数字大于 1，则梯度越乘越大，导致梯度爆炸
- 对于偏置来说，注意到偏置是没有权重，调整的是偏置本身

3.4 总结探讨

总结起来，BP 的误差反向传播思想可以概括为：利用输出层的误差来估计出其直接前导层的误差，再借助于这个新的误差来计算更前一层的误差，按照这样的方式逐层反传下去便可以得到所有各层的误差估计。

BP 算法的提出在一定程度上解决了多层网络参数训练难的问题，但是其自身也存在如下一些问题。

首先，误差在反向传播过程中会逐渐衰减/增大，经过多层的传递后将会变得消散/爆炸，这使得 BP 在深层网络中并不可行。对于梯度爆炸，则一般可以通过调整神经网络模型中的初始化参数得以解决。对于无法完美解决的梯度消失问题，目前有很多研究，一个可能部分解决梯度消失问题的办法是使用 ReLU（Rectified Linear Unit）激活函数（ σ (z)=max(0,z)）。

其次，BP 采用最速梯度下降的优化思想，而实际问题的误差函数通常不是凸的，存在众多局部极小值点，算法很难得到最优解。极小值问题，有多种解决方案，比如从多个不同的初始点开始搜索，模拟退火，随机梯度下降，遗传算法等。但这些都是启发式，理论上尚缺乏保障。

第三，由于训练过程中依靠于导数信息来进行权值的调整，当权值调节过大时会使大部分神经元的加权和过大，致使传递函数工作于 S 型函数的饱和区，所以权值的调整会出现停顿的情况。

第四，隐层神经元的个数设置是个未解决的问题，实际应用中通常靠试错法调整。

第五，BP 神经网络的过拟合，常用的方法是早停和正则化。早停是指：将数据分成训练集和验证集，训练集用来计算梯度，更新连接权和阈值，验证集用来估计误差，如果训练集误差降低但是验证集误差升高，则停止训练，同时返回具有最小验证集误差的连接权和阈值。正则化是指：误差目标函数中增加一个用于描述网络复杂度的部分，例如连接权与阈值的平方和。

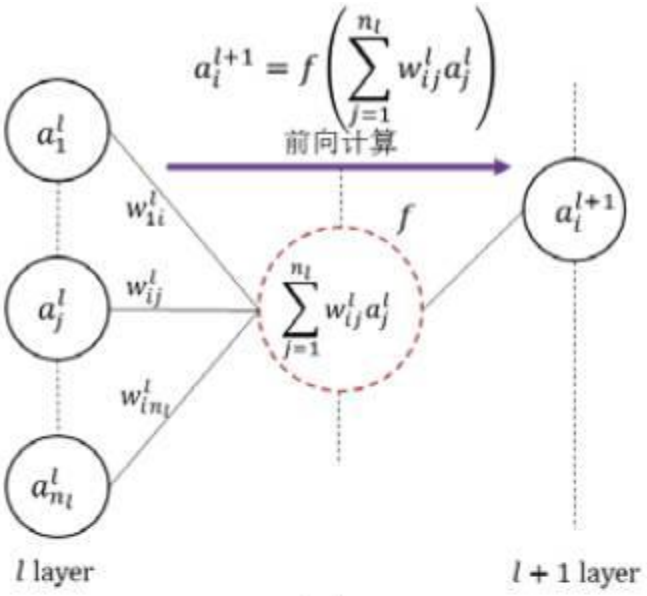
最后，对于一些复杂网络的优化问题，BP 算法受到学习速率的限制需要花费几个小时，甚至更长的时间来完成训练任务。累积 BP 算法和标准 BP 算法各有优缺点。

3.5 灵感来源

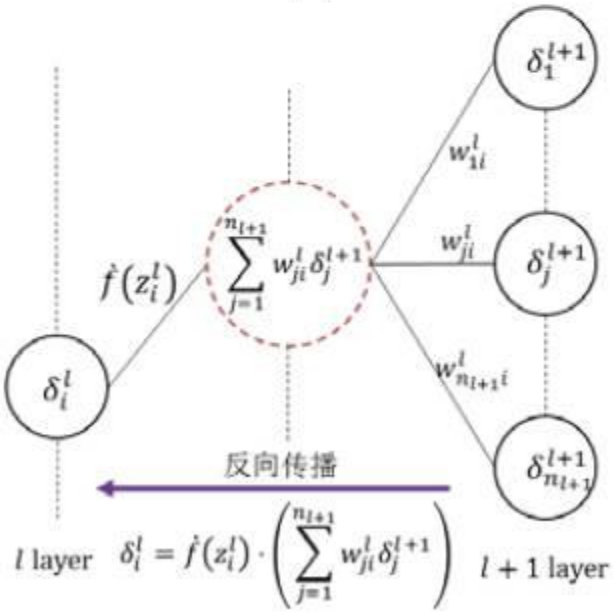
影响本部分写作的有几篇文章。

第一篇是四川大学章毅教授的《一张图看懂 BP 算法》，初读此文，不明觉厉。现在回头看一下，颇有启发。文章通过引入敏感性这个概念，提供了一个很好的角度解释反向传播。

通过构造敏感性，可以看到前向计算和反向传播呈中心对称。它显示了网络的前向计算和反向传播具有相似的计算形式。这对于理解反向传播的加权和特别有帮助。



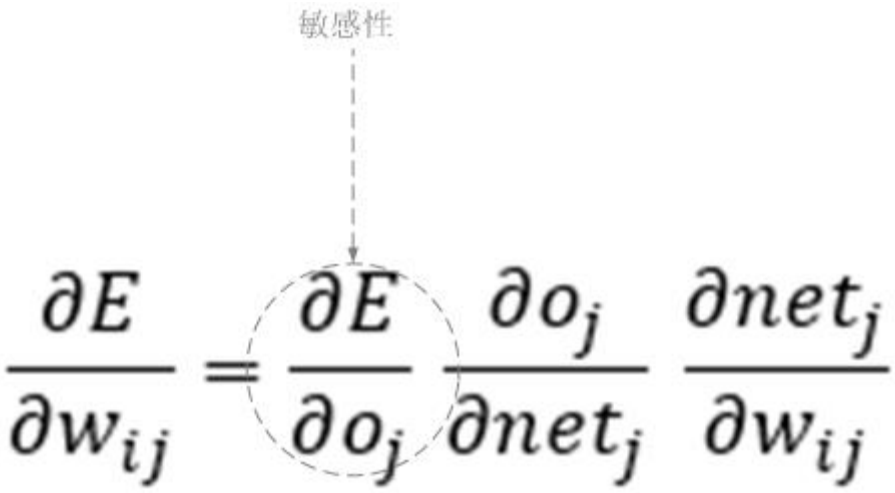
(a)



(b)

文章引入虚拟神经元的概念，来表示加权和的中间过程。在前向计算中， 网络传递的信息是神经元的输出， 向虚拟神经元施加的映射是激活函数。而在反向传播中，网络传递的信息是神经元的敏感性。两个计算均通过虚拟神经元对传入的信息求和。不同的地方仅在于对虚拟神经元施加非线性映射的方式， 向虚拟神经元施加的映射是与激活函数的导数乘积。

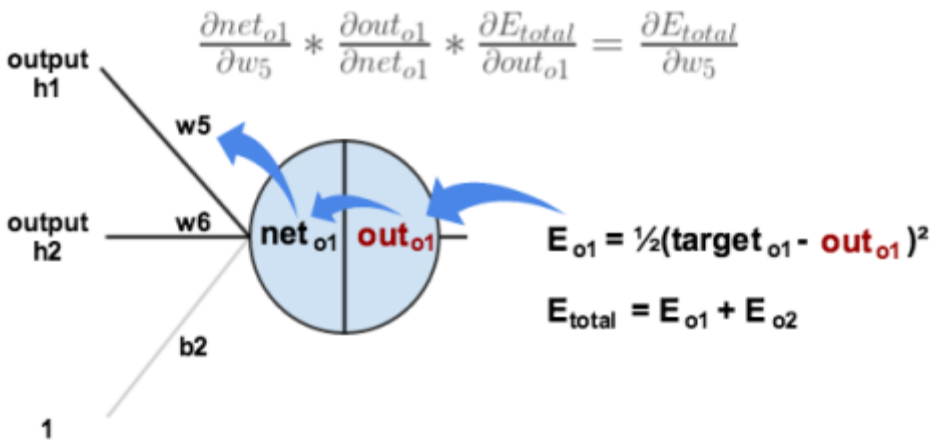
如何理解敏感性呢？其实敏感性就是前面推导梯度求导三部分的误差函数导数部分，如下图所示：



用公式表示为：

$$敏感性 = \frac{\partial E}{\partial o_j} = \begin{cases} (y-t), & if\ j\ is\ an\ output\ neuron \\ (\sum_{k=1}^{K} \delta_k w_{jk}), & if\ j\ is\ an\ inner\ neuron \end{cases}$$

第二篇文章是《A Step by Step Backpropagation Example》。文章从输出层开始，一步一步，非常清晰的推导反向传播算法。如下图所示：

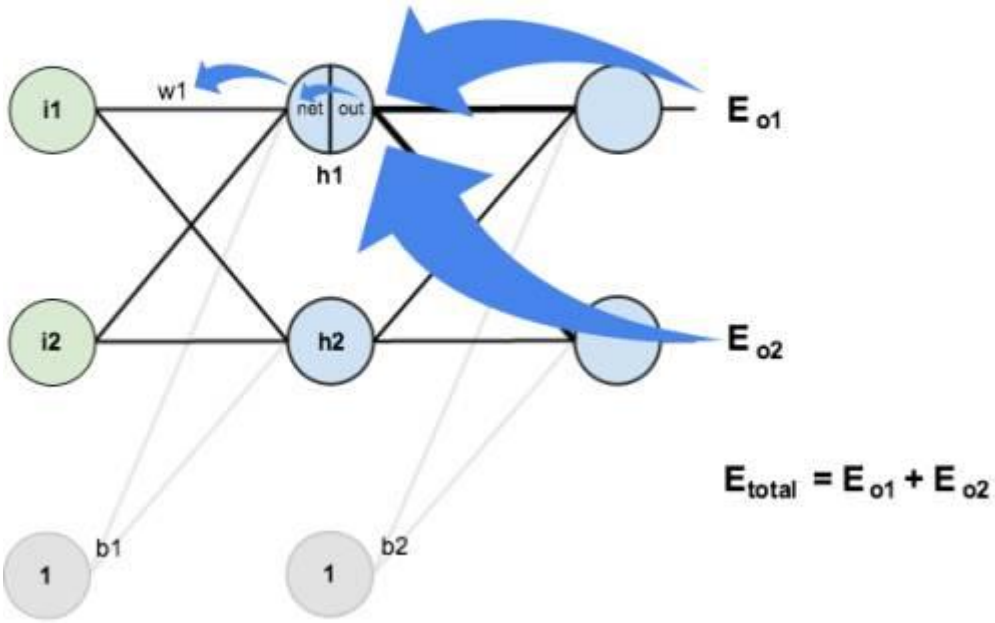


如果是第一次理解算法，还没有熟悉形式化推导，该文章很有帮助。文章明晰了一下几点：

- 误差函数可以是多个目标误差之和
- 误差函数可以对任意一层神经元求导（下图所示）
- 对特定神经元求导时候应用复合函数求导的规则
- 应用链式法则，对神经元的求导转化为后一层神经元的导数之和。

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



3.6 本部分参考文献

《反向传播神经网络极简入门》

《一张图看懂 BP 算法》四川大学 章毅

《神经网络七十年：回顾与展望》

《BP 神经网络的发展现状综述》周政

《基于导数优化的 BP 学习算法的研究综述》

《机器学习》第五章：神经网络 周志华

http://www.cnblogs.com/pinard/p/6437495.html

《Principles of training multi-layer neural network using backpropagation》

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

《A Step by Step Backpropagation Example》

英文：https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

中文： http://yongyuan.name/blog/back-propagtion.html

http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/

https://www.gitbook.com/book/tigerneil/neural-networks-and-deep-learning-zh/details

https://en.wikipedia.org/wiki/Backpropagation

https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

4 神经网络的第一性原理

学习的本质是什么？神经网络的本质是什么？生物智能的隐喻给了我们哪些启示？为什么层级结构（浅层和深层）适用于学习（自然学习）？这些问题不仅困扰着机器学习的很多入门者，也困扰着很多进阶者。本部分尝试从以下几个方面探讨神经网络的第一性原理：

- 学习的本质
- 神经网络的数学本质
- 生物学的启示
- 物理世界的层级结构

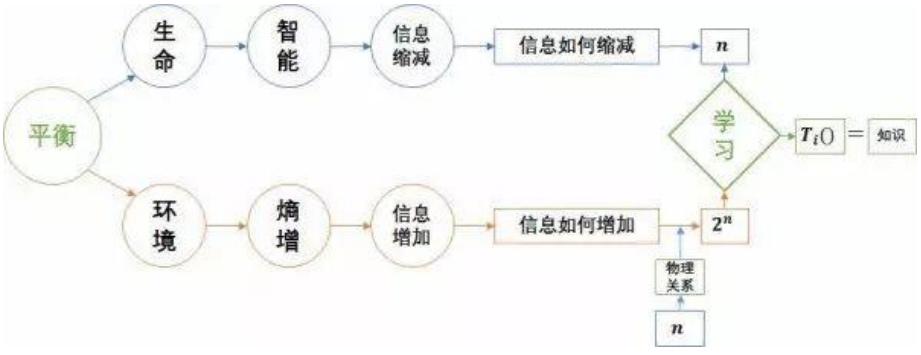
前面两点主要关注学习的功能问题，后面两点主要关注学习的结构问题。功能和结构是统一的。

在学习的本质中，借鉴熵的概念，讨论学习的本质之一；同时也给出一个统计学习的本质定义，可以直接推导出神经网络的学习原理。在神经网络的数学本质中，主要论述一个观点：神经网络本质上是一个信息的非线性变换系统，是对复杂函数的一种逼近表示。可以看出，无论是熵定义的学习本质概念，还是神经网络的数学本质，原理上是统一的。

在生物学的启示中，主要关注大脑结构本身，以及视觉和记忆的层级机制。在物理世界的层级结构中，主要关注语音，文字和图片的层级化本质和表示；同时也提到宇宙组成的两个重要属性。

4.1 学习的本质

网络上 YJango 先生的围绕减熵阐述了他的整个思考流程，我深以为然，故直接引用如下。



生物要做的是降低环境的熵，将不确定状态变为确定状态。通常机器学习是优化损失函数，并用概率来衡量模型优劣。然而概率正是由于无法确定状态才不得不用的衡量手段。生物真正想要的是没有丝毫不确定性。

深层神经网络在自然问题上更具优势，因为它和生物学习一样，是找回使熵增加的“物理关系”（知识，并非完全一样），将变体（2^n）转化回因素（n）附带物理关系的形式，从源头消除熵（假设每个因素只有两种可能状态）。

这样所有状态间的关系可以被确定，要么肯定发生，要么绝不发生，也就无需用概率来衡量。一个完美训练好的模型就是两个状态空间内所有可能取值间的关系都被确定的模型。

由此，YJango 先生得到的学习本质是：

学习目标：是确定（determine）两个状态空间内所有可能取值之间的关系，使得熵尽可能最低。

在统计学习理论中，学习的本质定义如下：

系统S为研究对象，通过一系列的观测样本来求得学习机LM，使得LM的输出$\hat{y}$能够尽量准确的预测S的输出y。

$$(x_1,y_1),\ (x_2,y_2),\ \ldots,\ (x_n,y_n)$$

学习机LM的输出$\hat{y}$与输入x之间可以看作是一个函数关系：

$$\hat{y} = f(x)$$

一般需要将函数$f(x)$限定在特定的一组函数$\{f(x,w)\}$中求取。

定义风险：$L(y, f(x,w))$

□ 均方误差：$L(y, f(x,w)) = (y - f(x,w))^2$

□ 似然函数：$L(p(x,w)) = -\ln p(x,w)$

通过定义风险，求取映射函数。这样的定义比上面熵的定义更加具体和数学化。

4.2 神经网络的数学本质

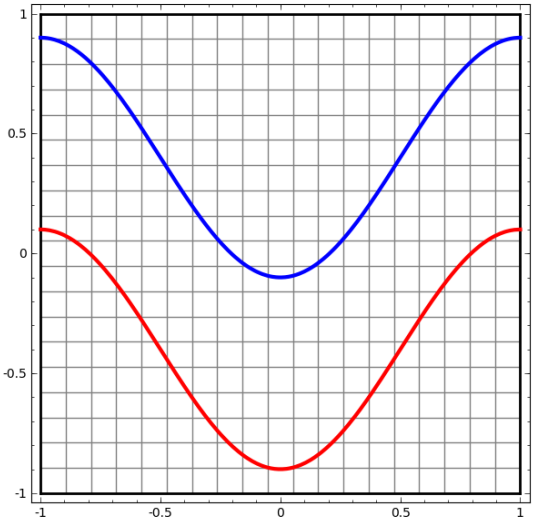
神经网络本质上是一个信息的非线性变换系统。设一个三层网的输入层，隐含层及输出层的节点数分别为 n、k、p，则 p 个输出分别为：

$$y_i = \sum_{j=1}^{k} C_{ji}\phi_j(W_j^T X + \theta_j), i = 1, 2, \cdots p$$

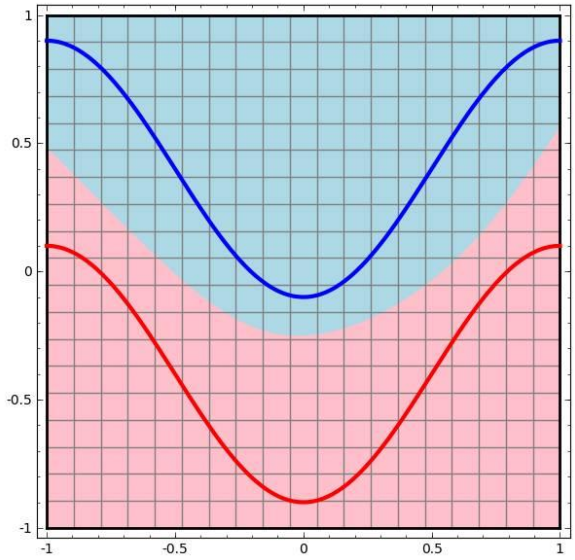
上式定义了一个函数逼近结构，并且式子中的参数都可调，调整 W 和 θ 就使得用于逼近的基函数相应调整。相比一般逼近方法，神经网络所定义的函数逼近结构的优越性就在于它不但是一个对逼近系数寻优的过程，而且是一个对逼近基函数组自适应寻优的过程。并且，随着网络层数的增加，叠加的结果使基函数寻优的自由度增加了。网络的非线性建模能力随着层数的增加而快速增长。这是一般逼近方法所不能比拟的。

1989 年，Funabashi，Arai 和 Hecht-nielsen 等人分别证明了（其证明相当复杂）三层前馈神经网络能任意逼近紧集上的连续函数和平方可积函数。也有文献提到“理论证明两层神经网络可以无限逼近任意连续函数”。面对复杂的非线性分类任务，多层神经网络（用连接点表示）可以对输入空间进行整合，使得数据线性可分。下面借用 Yann LeCun, Yoshua Bengio 和 Geoffrey Hinton 在 2015 年《自然》杂志上发表的《Deep learning》一文给出了一个例子，说明输入空间中的规则网格是如何被隐藏层转换的。

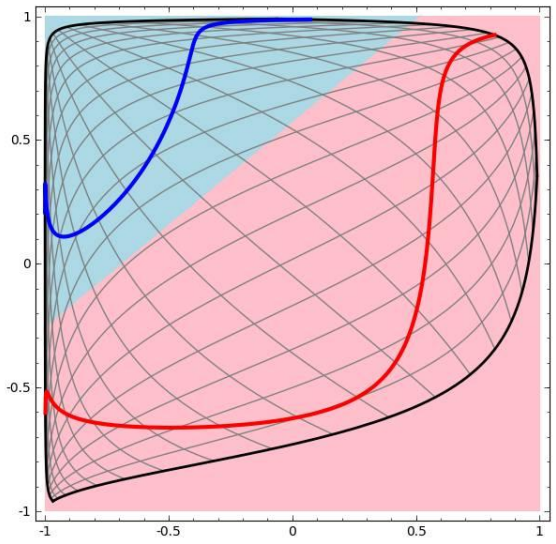
两条曲线，神经网络需要学习区分两条曲线上的点：



只有输入层和输出层神经网络的分类效果：



增加隐藏层的分类效果：



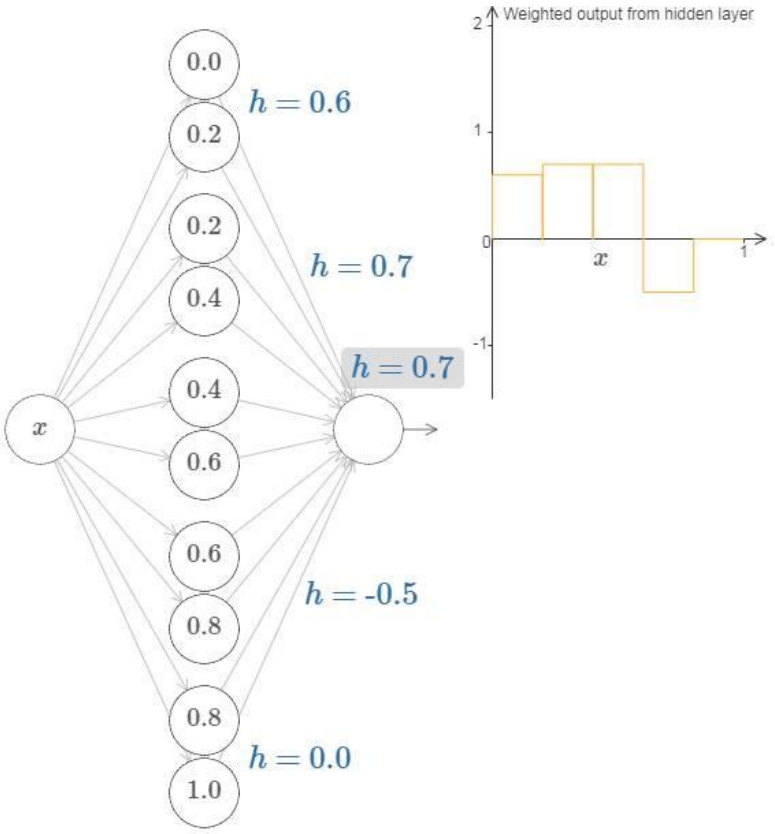
我们知道两层神经网络是线性分类问题，那么两个线性问题叠加在一起为什么就可以解决非线性分类问题了呢？上图很好的解释了我们的疑问。首先上图 6 可以看出，三层神经网络的决策分界非常平滑，而且分类的很好。而上图 7 展示了该图经过空间变换后的结果，我们可以看到输出层的决策分界仍然是直线，关键是，从输入层到隐含层时，发生了空间变换。也就是说三层神经网络可以做非线性分类的关键便是隐含层的加入，通过矩阵和向量相乘，本质做了一次线性变换，使得原先线性不可分的问题变得线性可分。所以多层神经网络本质就是复杂函数的逼近和拟合。

下面从神经网络的权重训练的角度讨论一下函数的拟合问题。我们知道训练神经网络的目的是为了调整权重参数，为什么调整参数就可以改变线性变换的结果呢？我们通过一组动画形象的看一下（动画和示例来自《Neural Networks and Deep Learning》）。

二维效果，单一输入 （动画 1）：增加权重，曲线变得更陡峭，直到最终看起来像一个阶梯函数。

动画 1

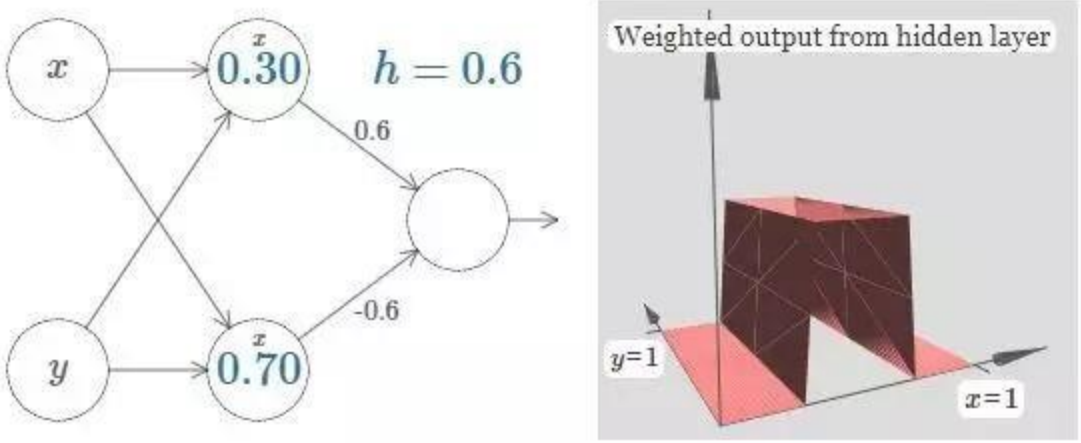
随着同层节点的增多，图像呈现为多级阶梯函数。



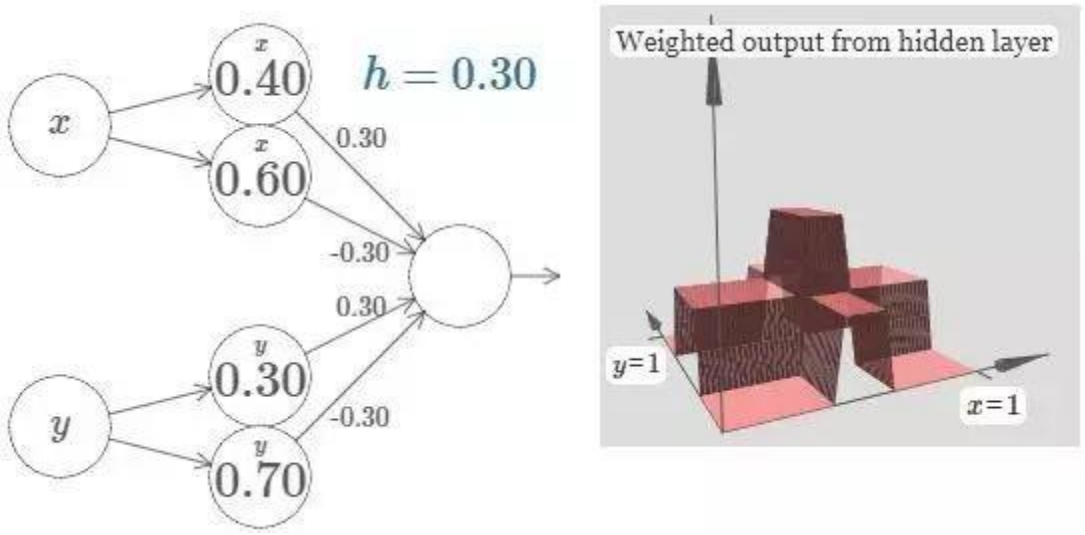
三维效果，两个输入（动画 2 ）：增加一个输入的权重，曲面变得更陡峭，直到最终看起来像一个跃阶函数。

动画 2

增加一个隐藏层，图像看起来将会是这样的：



继续增加隐藏层的节点数，可以得到一个塔形结构：



三维效果，两个输入，隐藏层含有 4 个节点（动画 3 ）：通过修改隐藏层的权重，对塔形结构进行调整：

动画 3

文献《神经网络的函数逼近理论》从理论上研究神经网络的非线性逼近能力。文章给出了以前馈网络为代表的一类网络结构所定义的映射关系究竟对哪些非线性映射具有逼近能力，逼近的阶及精度是怎样的，与经典函数逼近相比有哪些特点、优点，能否建立与经典函数逼近中 Weierstrass 第一定理、Chebyshev 定理、Borel 定理以及 Jackson 定理相应的结果。

下面附录一些逼近定理的解释：

魏尔斯特拉斯逼近定理：

闭区间上的连续函数可用多项式级数一致逼近。

闭区间上周期为 2π 的连续函数可用三角函数级数一致逼近。

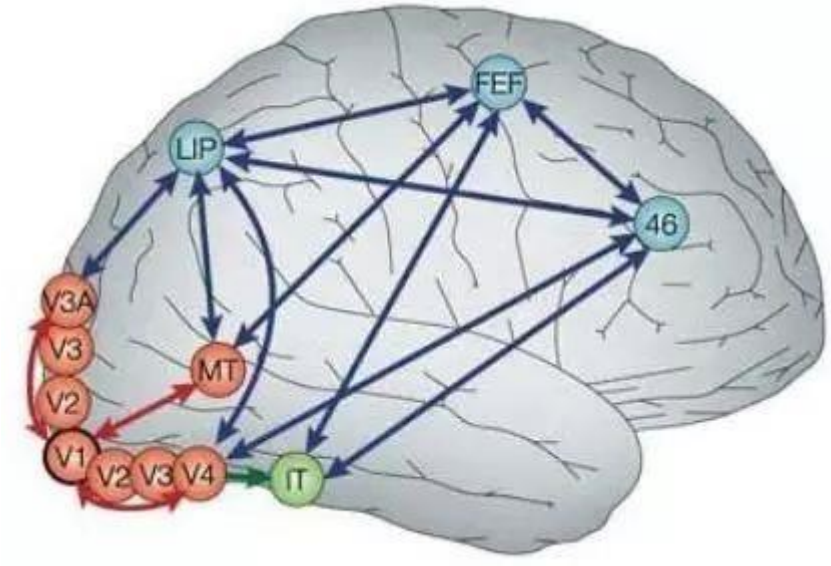
Chebyshev 定理说明随机变量 X 取值基本上集中在 EX 附近，这进一步说明了方差的意义。

Borel 定理又称有限覆盖定理。大意是精心地调整使用的开区间的位置和大小，使得为了覆盖住这个闭区间，必须使用无穷多个开区间才得以完成。其关键有两点：第一“被覆盖区间必须是闭区间”，第二“覆盖闭区间的区间区间系必须是开区间”。

在逼近论中，Jackson 定理给出函数利用多项式逼近的上界估计。

4.3 生物学的启示

生物学给出的启示是，模仿人脑。这一部分主要给出一些视觉和记忆方面的例子，这些例子都证实了人类神经系统和大脑的工作其实是不断将低级抽象传导为高级抽象的过程，高层特征是低层特征的组合，越到高层特征就越抽象。



在对哺乳类动物开展的解剖研究中发现，大脑皮质存在着层级化的系列区域；在此基础上，神经科学研究人员又通过测试视觉信号输入人脑视网膜后经大脑前额皮质层到达运动神经的时间，推断发现大脑皮质层的主要功能在于将视觉信号通过复杂的多层网络模型后加以提取观测信息，而并未直接对视觉信号进行特征处理(如上图所示)，而是使接收到的刺激信号通过一个复杂的层状网络模型，进而获取观测数据展现的规则。

也就是说，人脑并不是直接根据外部世界在视网膜上投影，而是根据经聚集和分解过程处理后的信息来识别物体。因此视皮层的功能是对感知信号进行特征提取和计算，而不仅仅是简单地重现视网膜的图像。

除了层级结构，神经网络的权值训练思想也是来自于著名的 Hebb 学习规则。前文《神经网络简史》介绍了从大脑神经元功能的发现，到感知机，再到 Hebb 学习规则等神经网络发蒙的简述历史。

人类感知系统这种明确的层级结构极大地降低了视觉系统处理的数据量，并保留了物体有用的结构信息。对于要提取具有潜在复杂结构规则的自然图像、视频、语音和音乐等结构丰富数据，深度学习能够获取其本质特征。

David Marr - as a schoolboy - 1960

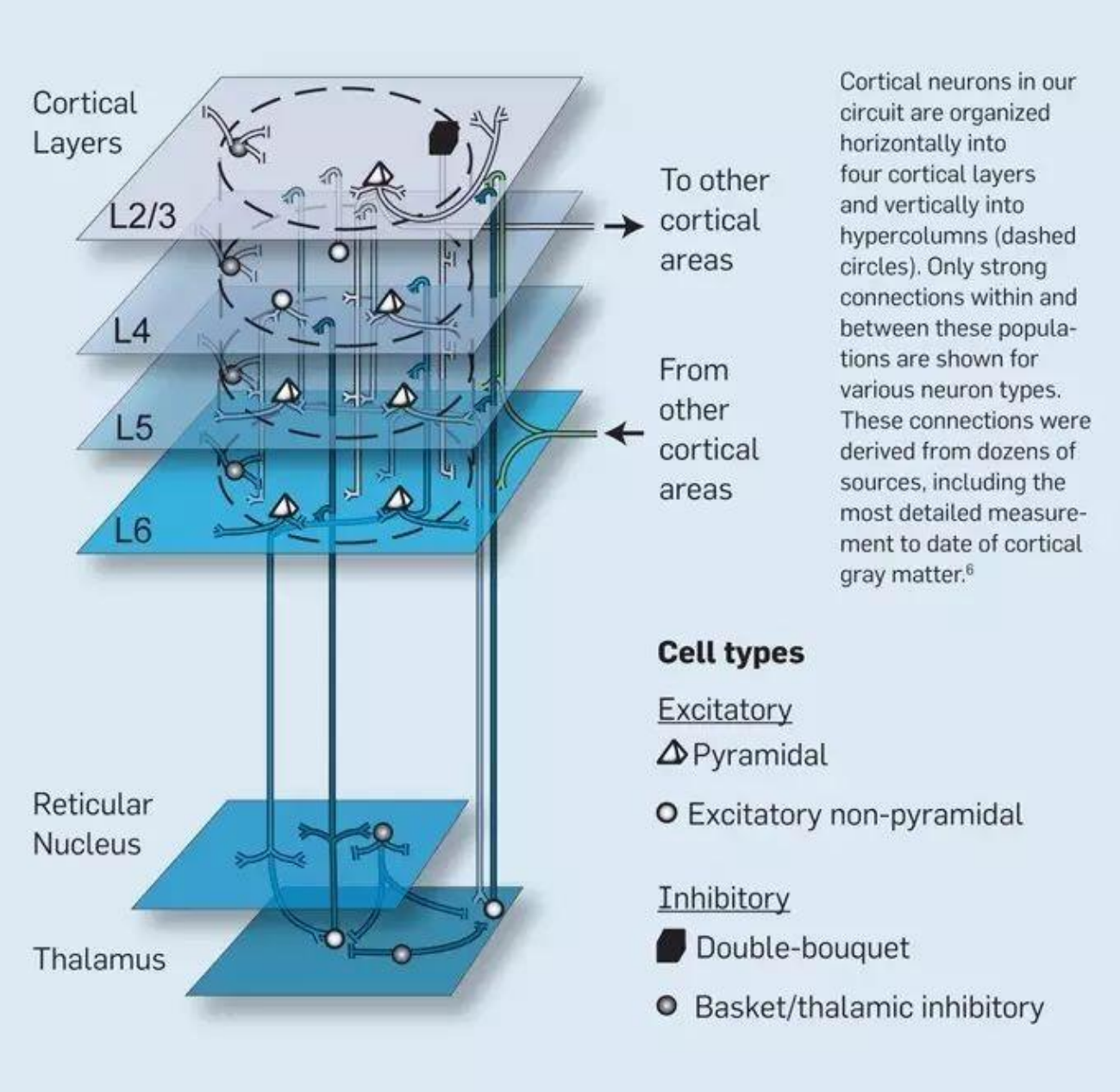
David Marr - in La Jolla, CA - 1974

David Marr 是英国心理学家和神经科学家。 他将心理学，人工智能，和神经生理学的研究成果结合起来，提出了全新的关于视觉处理的理论。他被认为是计算神经科学的创始人。

关于人脑对于视觉信息的处理，著名神经学家 David Marr 在在 1982 年出版的《视觉计算》书中提出了视觉计算框架，并在序言中指出，视觉计算的关键是如何表示(representation)。他认为人类视觉主要完成的工作就是从外部世界投射得到内部表示。对应于人类视觉，提出视觉计算框架应分为初级视觉、中级视觉和高级视觉三个层级，组成一个自底向上的流水线。初级视觉主要是从图像中提取一些基本的表示，也就是所谓特征；中级视觉是如何把这些基本的元素组合成不同部分，这涉及到分割；而高级视觉是从分割结果中得到物体的三维表示。

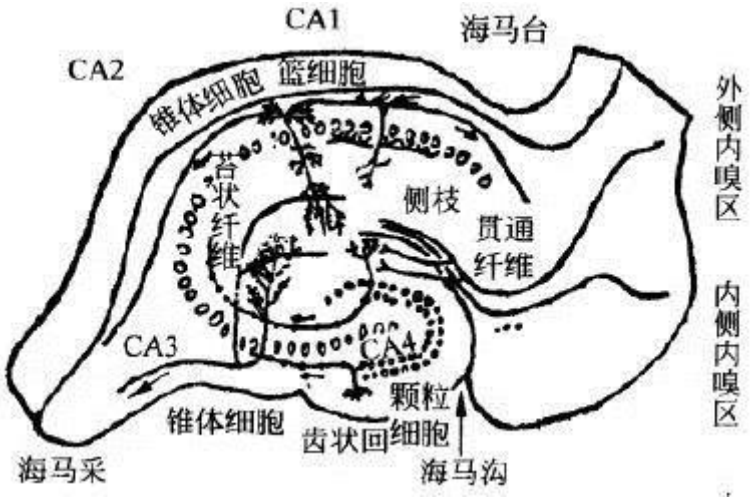
人脑对于记忆是如何处理的呢？

记忆就是对过去的经验或是经历，在脑内产生准确的内部表征，并且能够正确、高效地提取和利用它们。记忆涉及信息的获得、储存和提取等多个过程，这也就决定了记忆需要不同的脑区协同作用。在最初的记忆形成阶段，需要脑整合多个分散的特征或组合多个知识组块以形成统一的表征。从空间上讲，不同特征的记忆可能储存于不同的脑区和神经元群。在时间上，记忆分为工作记忆、短时记忆和长时记忆。



一般认为，记忆的生理基础与新皮质和海马有关。新皮质发展成为六层，如图 2 所示。第一层是皮质内部神经元投射信息交汇的地方。底下 L2/3 和 L5 层的锥体细胞投射上来轴突和顶树突，在这里交汇，这里的神经元细胞很少，其中大部分都是抑制性的。L2/3 层有各种神经元，主要是小椎体细胞，构建皮质内的局部回路，这些锥体细胞主要连接是在皮质内部，但也有连到胼胝体的。L4 主要是颗粒性细胞，胞体较小而密集，负责接收丘脑传递的感觉信号。L5 主要负责传出信号，包含了最大的椎体细胞，将轴突投射到其他不同的脑区。L6 也是主要负责传出信号，但也接收丘脑传入的反馈信号。

新皮质记忆结构化的知识，存储在新皮质神经元之间的连接中。当多层神经网络训练时，逐渐学会提取结构，通过调整连接权值，使网络输出的误差最小化，成为相对稳定的长时记忆。

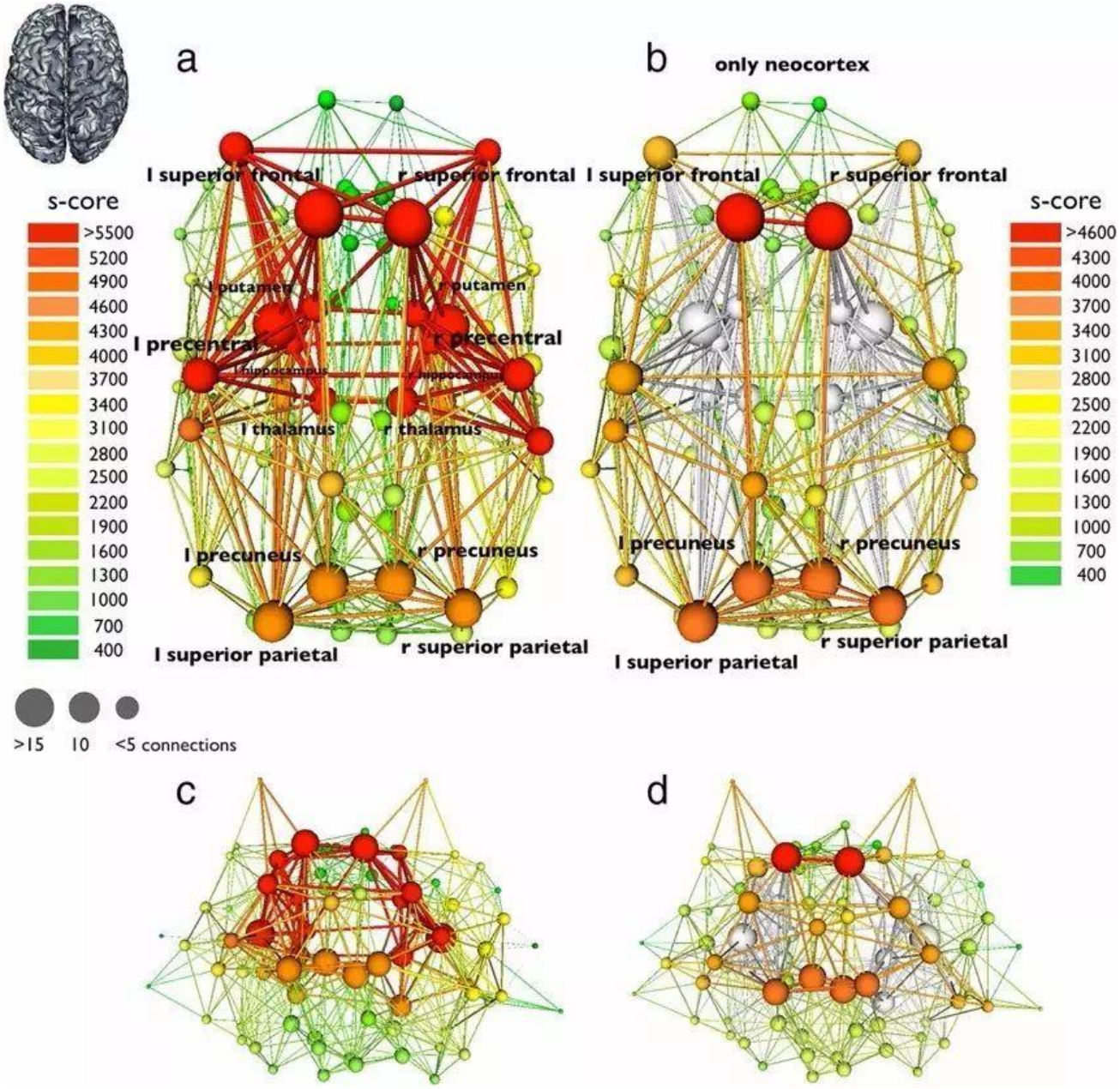


短时记忆记录在海马体中。在海马中，椎体细胞和细胞体组成层状并行的锥体细胞层，它的树突是沿海马沟的方向延伸。

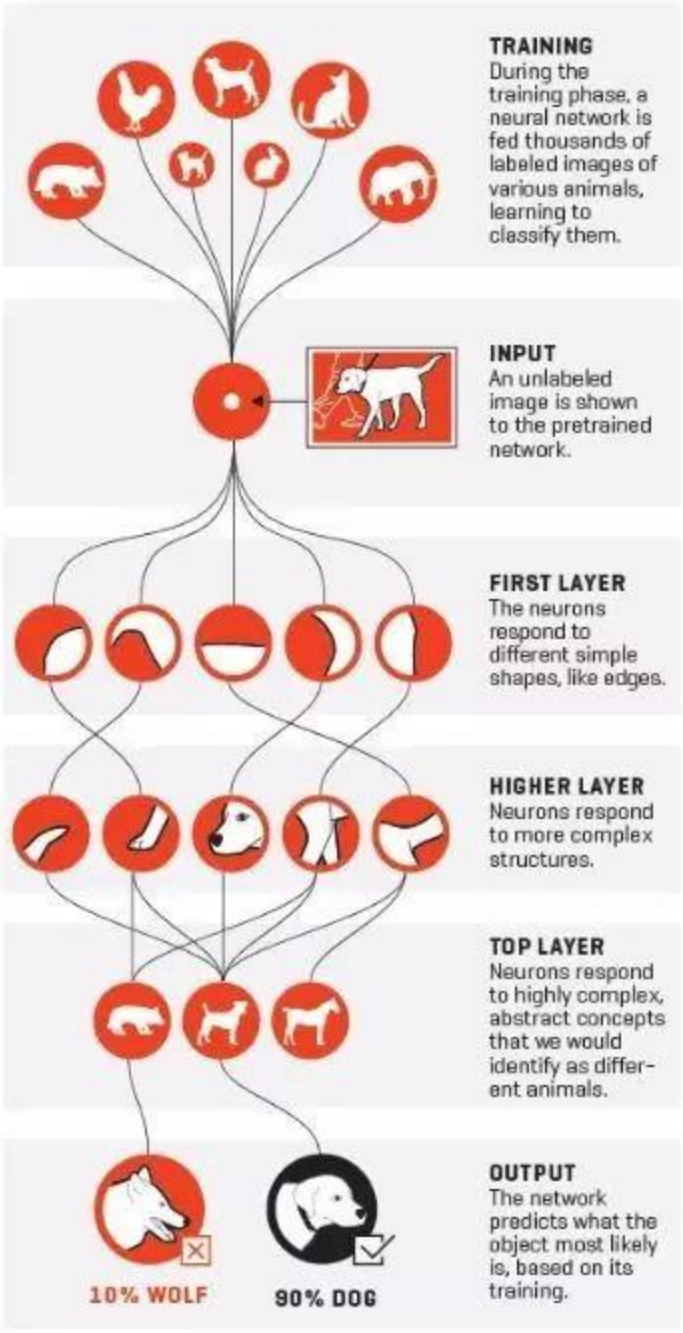
4.4 物理世界的层级结构

层级，这个词来源于希腊语的 hierarchia，意思是“大祭司的规则”，表达了一种分明的等级性。它是对事物进行分门别类的一种方法，即用高低、同级别这样的关系来对事物做出划分。用数学的语言来讲，所谓的层级，就是指在我们所讨论的事物集合上定义了一种偏序关系。

自然界存在着大量的层级关系。比如，高低就能构成一种层级。住在楼上的人就比楼下的人位于更高的层级。再比如，尺度也构成了层级。比如，我们说人体是由不同的小尺度器官组成的，而器官又是由小尺度大量的细胞组成的，而细胞是由更小尺度的分子构成的……。泛化和抽象是人类语言中的层级，比如：“动物”就是一个高高在上的抽象的层级，“鸟类”则是一个更具体的层级，“麻雀”则是更具体的概念。



大自然和人工系统中这些形形色色的层级性会反映到我们的数据中，这就迫使我们能够读懂层级性的数据。于是，深度学习技术应运而生，它通过加深神经网络层级，从而应付数据中的层级性。从对卷积神经网络的剖析来看，不同层级的神经元实际上是在不同尺度上提取特征。例如，如果我们用大量的图片训练了一个可以对动物进行分类的深度神经网络，那么该网络就会抽取数据之中的多尺度（层级）信息。



采用特征来表示待处理问题中的对象，是所有应用任务的首要工作。比如在处理文本分类时，经常用词集合特征来表示文档，之后采用不同的分类算法来实现分类。类似的，在图像处理任务中，最为普遍的就是把图像用像素集合特征加以表示。选取不同的特征对任务的最终结果影响较大。因此，在解决实际问题时，如何选取合适的特征非常重要。

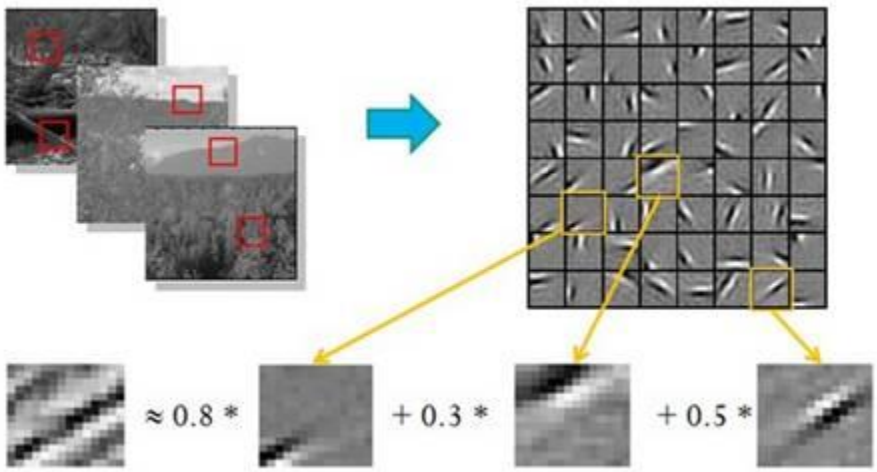
对于很多训练任务来说，特征具有天然的层级结构。在语音、图像、文本处理任务中，处理对象的层级结构如下表所示。以图像识别为例。最初的原始输入是图像的像素，之后众多相邻像素可以组成线条， 多个线条组成纹理，并进一步形成图案；局部图案又构成了整个物体。不难发现，原始输入和浅

层特征之间的联系较容易找到。那么，在此基础上，能否通过中间层特征，逐步获取原始输入与高层特征的联系呢？这是特征的层级可表示性问题。

语音、图像、文本领域的特征层级结构：

任务领域	原始输入	浅层特征		中层特征		高层特征	训练目标
语音	样本	频段	声音	音调	音素	单词	语音识别
图像	像素	线条	纹理	图案	局部	物体	图像识别
文本	字母	单词	词组	短语	句子	段落	文章	语义理解

特征的层级可表示性也得到了证实。1995 年前后，Bruno Olshausen 和 David Field]收集了很多黑白风景照，从这些照片中找到了 400 个 16×16 的基本碎片，然后从照片中再找到其他一些同样大小的碎片，希望将其他碎片表示为这 400 个基本碎片的线性组合，并使误差尽可能小，使用的碎片尽可能少。表示完成后，再固定其他碎片，选择更合适的基本碎片组合优化近似结果。反复迭代后，得到了可以表示其他碎片的最佳的基本碎片组合。他们发现，这些基本碎片组合都是不同物体不同方向的边缘线。这说明可以通过有效的特征提取，将像素抽象成更高级的特征。类似的结果也适用于语音特征。



最近的文章《Why does deep and cheap learning work so well?》和《The Extraordinary Link Between Deep Neural Networks and the Nature of the Universe》提出的论点也很有意思。神经网络利用了宇宙的两个属性。第一是宇宙是所有可能功能的一小部分，深层神经网络不需要逼近任何可能的数学函数，只需要逼近它们的一小部分。宇宙中事物的组合可能是无限的，但在物理规律中只是以多项式的形式出现。这也可以称为物理世界的局部性。第二是宇宙的层级结构，原子核形成原子，然后又形成分子，细胞，有机体，行星，太阳系，星系等。复杂结构通常通过一系列更简单的步骤形成。神经网络的层可以逼近因果序列中的每个步骤。

物理学的基本规律包括对称性，局部性，组成性和多项式对数概率等属性，现在需要探讨的是如何将这些属性转化为神经网络。

《Why does deep and cheap learning work so well?》引起了广泛的讨论。文章的作者是 Max Tegmark 和 Henry Lin，Max Tegmark 是宇宙学家，Henry Lin 是他的学生。文章的中心观点是深度和 cheap 学习的成功不仅取决于数学，而且还取决于物理学。这个论点假设所有问题数据遵循“自然法则”，某种意义上解释了深度学习在“自然学习”上成功，但是却无法解释深度学习在其他非自然领域的成功，比如识别汽车，自动驾驶，创造音乐和围棋游戏等。

另外，网络上也有资料讨论深度学习背后的统计物理和能量模型，未得其要领，暂不解读。

4.5 本部分参考文献

《Learning Deep Architectures for AI》

《Why does deep and cheap learning work so well?》Henry W. Lin 等

《Neural Networks, Manifolds, and Topology》

《The Extraordinary Link Between Deep Neural Networks and the Nature of the Universe》

《Neural Networks and Deep Learning》 Michael Nielsen

《Theoretical Motivations for Deep Learning》Yoshua Bengio

《The Holographic Principle: Why Deep Learning Works》Carlos E. Perez

《WHY DEEP LEARNING IS SUDDENLY CHANGING YOUR LIFE》 Roger Parloff

《心智模型 CAM 的学习记忆机制》史忠植

《面向自然语言处理的深度学习研究》奚雪峰等

《深度学习及并行化实现概述》

《深度学习与层级性：从 RNN 到注意力与记忆》

《神经网络的函数逼近理论》李明国等

《深度学习研究综述》孙志军等

《沿着 Marr 的道路继续前进》 王天树

《统计学习理论的本质》

《深层学习为何要“Deep”》YJango

5 神经网络的本质在于泛化能力

前文尝试讨论神经网络作为机器学习的一种有效方法的本质。后来被李博士批评了，李博士一直强调泛化能力是学习的本质，也是神经网络的本质。回来做了一些功课，简单记为笔记，算是速读。

归纳和演绎是科学推理的两大基本手段。周志华在《机器学习》的第一章绪论[1]中讨论了假设空间和归纳偏好。学习过程，特别是“从样例中学习”是一个归纳过程。但学习过程是基于有限样本训练集进行的，因此可能有多个假设与训练集一致，即存在一个假设空间。为了在庞大的假设空间中找到一个合适的模型，就必然有归纳偏好。归纳偏好对应了“什么样的模型更好”的假设，也就是什么样的模型的泛化能力更强。

什么样的模型更好?这个问题被称为 ill-posed 问题。从这个问题延伸出了“奥卡姆法则”和“没有免费午餐”NFL 定理。

几乎所有文章在讨论神经网络本质时都会讨论函数逼近理论。董聪[3]认为经典函数逼近论的研究所基于的数学空间和多层前向网络的实际逼近过程所基于的数学空间的特性是不同的，它们是两类不同性质的逼近问题，具有完全不同的逼近机制。经典函数逼近论是以公理和定理为基础的演绎体系，归纳逻辑才是通往知识发现的主要途径。而无论是联结机制还是物理符号机制，学习过程都是归纳过程，应用过程均表现为广义的演绎过程。当演绎推理的结果超出了早先用于归纳学习的原始知识的具体描述时，泛化问题便产生了。

江学军等[4]指出影响网络泛化能力的关键因素是训练网络所使用的样本，包括样本的质量、样本的数量和样本的代表性三个方面。

王凯等[5]从人工神经网络的学习过程偏差/方差分解中说明损失函数与样本复杂性和网络复杂性都有关系，并指出为了深入研究学习过程中出现的过拟合现象，必须能够有效的度量网络复杂性、样本复杂性以及两者之间的关系。

样本复杂性尚没有定量的结果，目前通常以训练样本规模表示。网络复杂性的研究主要基于 VC 维的概念，已经有了一些定量结果[7][8]。在网络复杂性方面，也有人认为网络可调权重的规模比网络结构对避免过拟合问题更加重要。

在网络复杂性与样本复杂性两者关系方面的研究大多基于 PAC 学习理论。周志华《机器学习》第 12 章计算学习理论[2]有专门介绍。

从“奥卡姆剃刀法则”出发，一般都选择保证样本的复杂性，尽可能选用具有低复杂性的简单网络；或者增大训练样本尺寸以适应网络复杂度。

魏海坤等[6]的文章详细综述了神经网络的泛化理论和泛化方法，值得一读。

张承福[9]综述了对神经网络的一些看法，阎平凡也发表了相应的看法[10]，综述了学习理论的演变[11]。这两篇都是旧文，但值得一读。

本部分参考文献：

1：《机器学习》第 1 章 绪论 周志华

2：《机器学习》第 12 章 计算学习理论 周志华

3：《人工神经网络：当前的进展与问题》 董聪

4：《前馈神经网络泛化性能力的系统分析》 江学军 等

5：《人工神经网络泛化问题研究综述》王恺 等

6：《神经网络的泛化理论和泛化方法》魏海坤 等

7：《BP 网络学习能力与泛化能力之间的定量关系式》 李祚泳 等

8：《BP 网络学习能力与泛化能力满足的不确定关系式》李祚泳 等

9：《对当前神经网络研究的几点看法》张承福

10：《对多层前向神经网络研究的几点看法》阎平凡

11：《对多层前向神经网络研究的进一步看法》阎平凡

6 一些不成熟的思考

第一，计算的本质与智能的本质。《类脑智能研究的回顾和展望》指出，现有人工智能系统通用性较差与其计算理论基础和系统设计原理有密不可分的关系。计算机的计算本质和基础架构是图灵机模型和冯诺伊曼体系结构，其共同的缺点是缺乏自适应性。图灵计算的本质是使用预定义的规则对一组输入符号进行处理，规则是限定的，输入也受限于预定义的形式。图灵机模型取决于人对物理世界的认知程度，因此人限定了机器描述问题，解决问题的程度。而冯诺伊曼体系结构是存储程序式计算，程序也是预先设定好的，无法根据外界的变化和需求的变化进行自我演化。总结来看，计算的本质可以用一个数学公式 f(x)=y 来表达，是问题求解的范畴。

那智能的本质是什么？如何表达？著名信息论和人工智能专家钟义信给了一个探讨性的定义：智能一定是在环境的作用下，人跟环境相互作用，不断的去学习，不断的去进化，在这个过程当中展开了智能的活动。反之，如果没有这种主体跟客体的相互作用，如果一切都是十全十美，如果不需要做出任

何的改进，那就不需要思考、不需要学习，也就不需要智能。所以，一定要在主体跟客体相互作用过程当中来考察智能才有意义。李衍达院士在《 沿 Simon 开拓下去》的报告中探讨了智能的功能与智能的机理问题，指出基因的层次没有鸿沟，人和所有生物的机理是相同的，区别的是进化：自动适应外界变化而优化自身结构的功能。而且人脑在进化过程里面通过 DNA 的改变，改变了神经元的连接，这个连接既记录了学习的结果，又优化了学习算法。既简化了所需要的元件，又节省了能耗，非常巧妙。

智能路径：感知反应->条件反射（存储，记忆）->决策（意志、欲望和目的）

第二，关于程序员转型。和第一个问题有关，我们都是学习图灵机模型和冯诺伊曼架构长大的，思维方式相对固定。深度学习今年非常火爆，程序员又要开始转型。关于转型，我注意到几个论调：

- 转型深度学习，数学是首要的基础；
- 转型深度学习，开始大量学习 TensorFlow 框架；
- 大二大三优秀学生学习起来很快，有经验的程序员学习来很苦；

以上我都不太认同，人类是万物之灵，遇到新问题，学习新东西，再正常不过的事情，何来转型之说？如果非要说有什么需要转变，我觉得是到思维方式的转变：

- 数学只是工具，TensorFlow 只是封装的平台，而深度学习是有理论瓶颈的，工程界一直以来轻视学术的思维定势需要改变了。国内程序员同时是科学家的太少了，科学家有点高，做个学者吧。感觉要做一个好的科学家，不只是研究技术，而是在研究哲学，研究一些物质的本质、规律，研究一些最基础的东西。
- 大多数程序员都是“程序员”思维，这是软件工业化的结果。重接口，重输入，重交付，这是一种软件外包的思维。输入是什么？输出是什么？程序如何实现？这些都造成了思维懒惰的一代程序员，从来不去问为什么程序这么做。而深度学习恰恰是讨论程序为什么这么实现的问题，其输出是模型，是算法。这是程序员需要改变的思维方式。
- 人工智能更强调创新，特别是源头创新。在这个领域，有大量的问题都是崭新的，需要采用一些数学理论，结合实际需求来探索。我们在学习机器学习理论和算法的时候，需要有意识的突破已有的认知，特别是图灵机模型和冯诺伊曼体系结构。

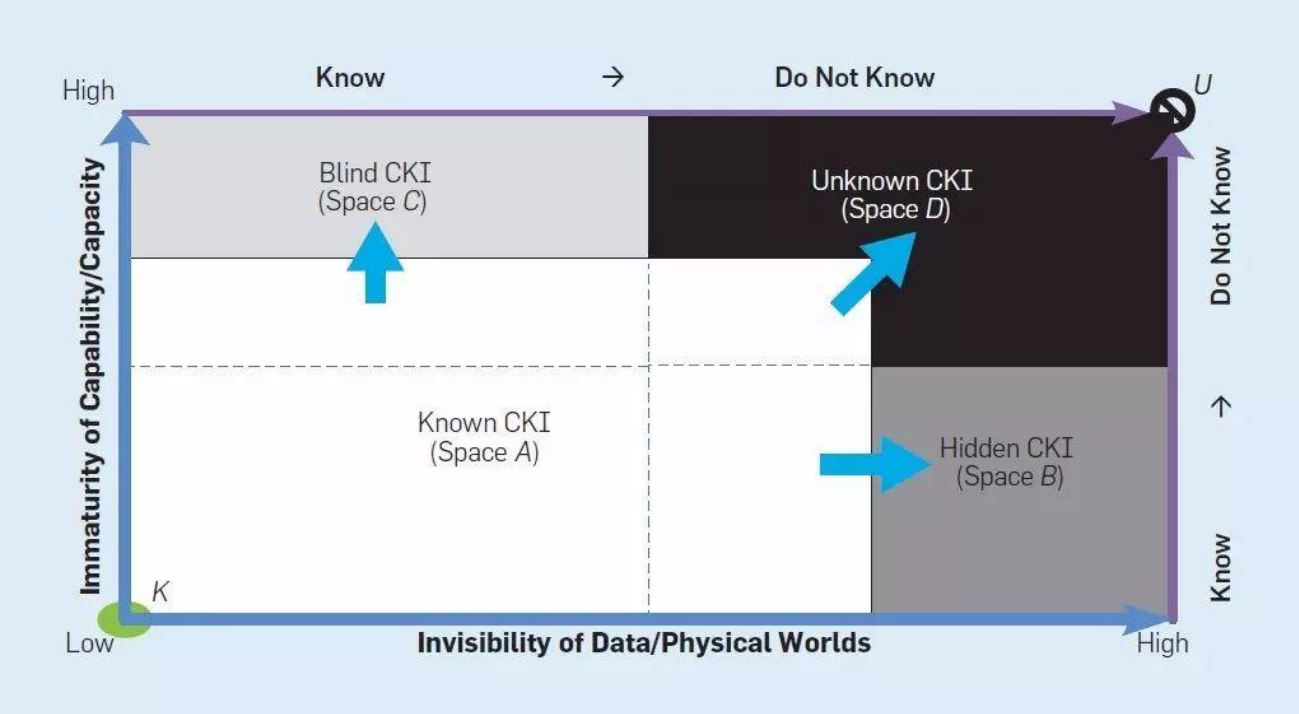
第三，脑复杂？还是环境复杂？傅小兰在《Simon 与认知科学研究》报告中提到了《分布式认知》，指出认知现象在认知主体和环境间分布的本质：认知既分布于个体内与个体间，也分布于媒介、环境、文化、社会和时间等之中（Cole & Engestrom, 1993）。Herbert A. Simon 也指出，一个人，若视作行为系统，是很简单的。他的行为随时间而表现出的表面复杂性主要是他所处环境的复杂性的反映。人——或至少人的智力要素——也许是比较简单的，人的行为的复杂性也许大半来自人的环境，

来自人对优秀设计的搜索，因此，“在相当大的程度上，要研究人类便要研究设计科学。它不仅是技术教育的专业要素，也是每个知书识字人的核心学科”。

第四，从上而下还是从下而上？人工智能从上而下研究的开创者和代表人物是 Herbert A. Simon，他当时想到，人的大脑活动是分层次的，在底层的机理没有搞清楚时，他认为也不妨碍对于高层概念、推理、问题求解层次进行研究。符号学派就是自上而下的典型代表，但至今符号学派一直受到自下而上的连接主义压制。自下而上的代表是日本的第五代计算机计划，东京大学元岗达教授提出“第五代计算机的构想”，随后日本制定了研制五代机的十年计划，总预算达 4.3 亿美元。以渊一博为所长的“新一代计算机技术研究所”苦苦奋战了近十年，他们几乎没有回过家，近乎玩命式的拼搏；然而，由于没有突破关键性技术难题，无法实现自然语言人机对话，程序自动生成等目标，最终于 1992 年宣告失败！这或许也是图灵机模型和冯诺伊曼架构的失败。然而，峰回路转，得益于分布式计算和大数据时代，深度学习成为主流的自下而上方法。近五年来，深度学习在“视”、“听”、“说”等领域取得了的巨大成功，但这还不能表明自下而上的胜利或者神经网络模型的正确。神经网络只是从下而上对大脑的粗糙模拟和抽象，是否是正确的大脑学习隐喻还不得而知。但神经网络的成功又引发了一些自下而上的尝试，据称 IBM 有一个名为“突触”的项目，研究芯片级类脑计算设备，支持低频率，低功耗，和大量链接等神经网络功能。

第五，鲁棒性？可解释性？魔术性？这几个问题是现在机器学习，特别是深度学习面临的主要问题。人类犯错：水平从九段降到八段，机器犯错：水平从九段降到业余，这就是鲁棒性。鲁棒性要求，“好的时候”要好，“坏的时候”不能太坏。在封闭静态环境中，重要因素大多是“定”的，而在开放动态环境中，一切都是变的，开放环境的鲁棒性，这也是自动驾驶面临的困难所在。关于可解释性，也被称为深度学习的黑箱模型。若学习器不能给出治疗理由，则难以说服患者接受昂贵的治疗方案。若学习器不能给出停机检测的理由，则难以判断停机检测的风险和代价。这些案例都需要机器学习的模型给出解释，否则难以应用到难以用于高风险应用。而机器学习魔术性是指即便相同数据，普通用户很难获得机器学习专家级性能。就是专家之间，是特别考验团队实力的，也有一点运气在里面。门派都一样，功力不一般。

第六，目前的研究热点和我的方向。深度学习是很火的，不过周志华说的很中肯：“深度学习中间还有很多困难而又重要的问题值得深入研究，但这些真正值得研究的问题，就我看到的情况而言，好像做的人非常少。大多数人在干什么呢？拿它做做应用，调调参数，性能刷几个点，然后发几篇文章。这样虽然容易发表文章，但恐怕很难产生有影响的成果。” 另外，周志华在引领集成学习的发展方向，CCAI17 可以看到一些方向，香港科技大学计算机系主任杨强谈到的迁移学习，日本理化学研究所杉山将谈到的弱监督机器学习等。我的计划是，从历史中观其大略；感知机，神经网络，反向传播，深度学习是一条线，已经是必备的基础了；然后向增强学习发力；在技术上打通分布式系统，大数据和机器学习；在业务和需求上结合金融场景。

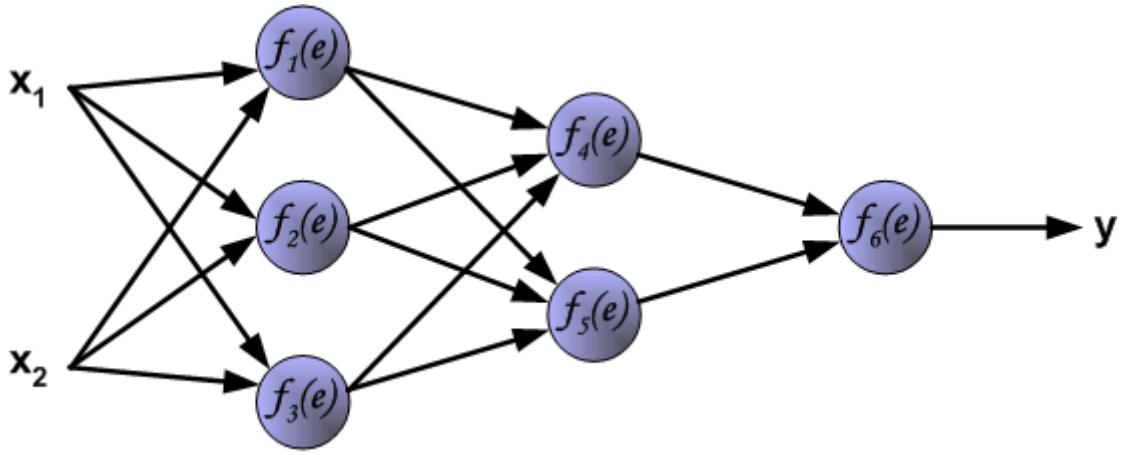


第七，已知和未知。我们参考神经生理学，研制了神经网络和深度学习，并且取得了良好的效果。有人指出，大脑的生物物理结构，机制和功能只是大脑处理信息过程中的印记，其中很少一部分可用于有意识的思想（认知）。在学习未知的过程中，我们对学习到底了解了多少？在未知的区域里，既有要学习的对象，也有学习本身。

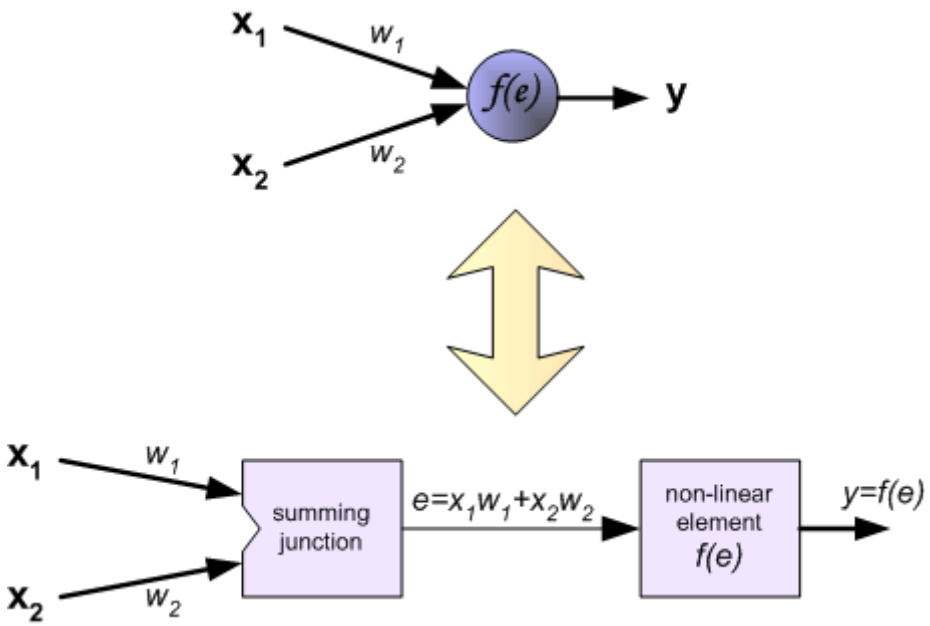
7 附录 1：使用反向传播训练多层神经网络的原理

文章《Principles of training multi-layer neural network using backpropagation 》提供了直观理解反向传播的完整例子。以下是原文翻译。

文章描述采用反向传播算法训练多层神经网络的学习过程。为了说明这个过程，使用了具有两个输入和一个输出的三层神经网络，如下图所示：

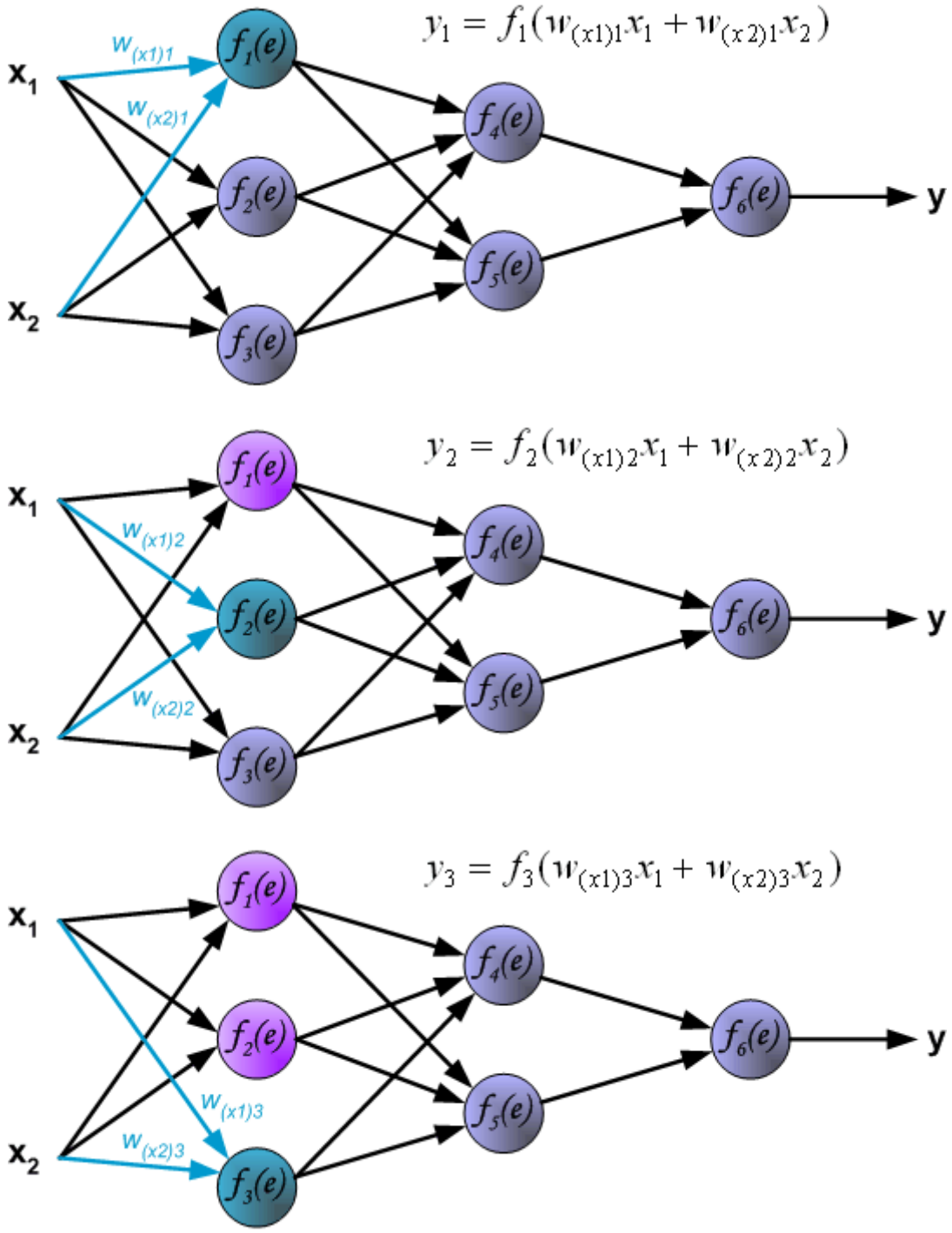


每个神经元由两部分组成。第一部分是输入信号和权重系数的加权和。第二部分是非线性函数，称为神经元激活函数。信号 e 是加权和的输出（加法器的输出）信号。y=f(e)是非线性函数（元件）的输出信号。信号 y 也是神经元的输出信号。

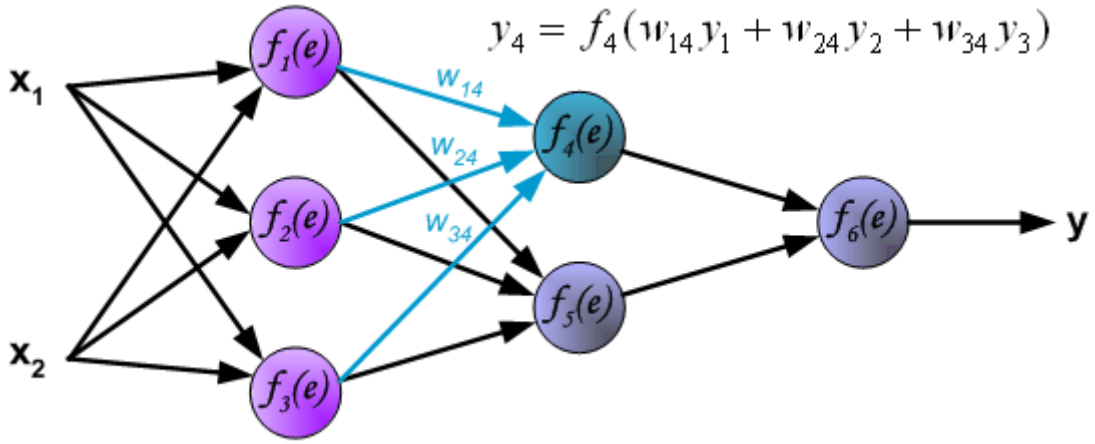


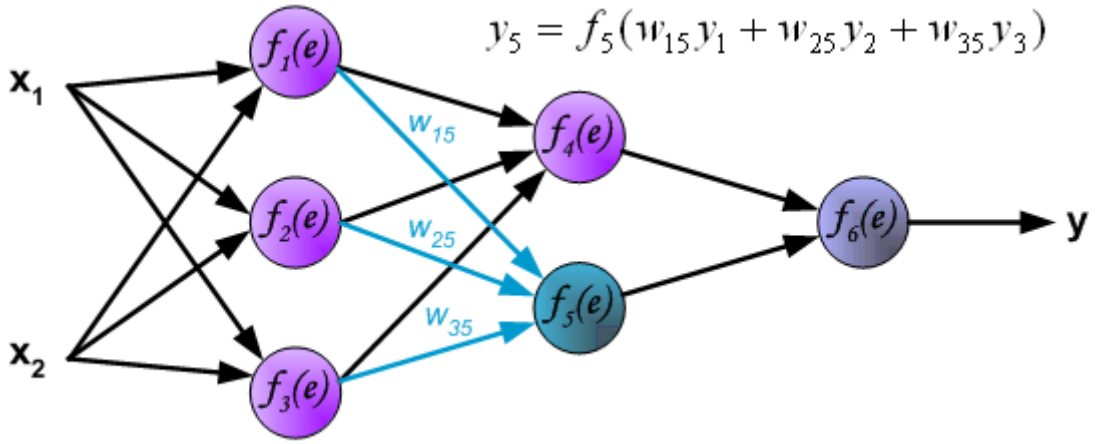
要训练神经网络，我们需要“训练数据集”。训练数据集是由对应目标 z（期望输出）的输入信号（x_1 和 x_2）组成。神经网络的训练是一个迭代过程。在每个迭代中，使用来自训练数据集的新数据修改网络节点的加权系数。整个迭代由前向计算和反向传播两个过程组成。

前向计算：每个学习步骤从来自训练集的两个输入信号开始。前向计算完成之后，我们可以确定每层网络中每个神经元的输出信号值（译者注：隐藏层神经元的误差是没有的，因为训练数据集中没有隐藏层的目标值）。下图显示了信号如何通过网络传播，符号 w(xm) 表示网络输入 x_m 和神经元 n 之间的连接权重。符号 y_n 表示神经元 n 的输出信号。

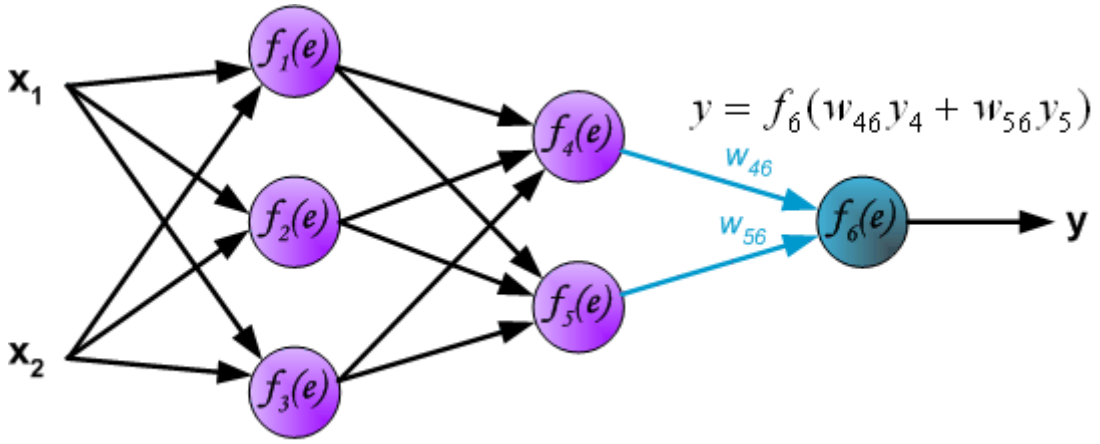


隐藏层信号传播。符号 w_mn 表示神经元 m 的输出和后一层神经元 n 的输入之间的连接权重 。

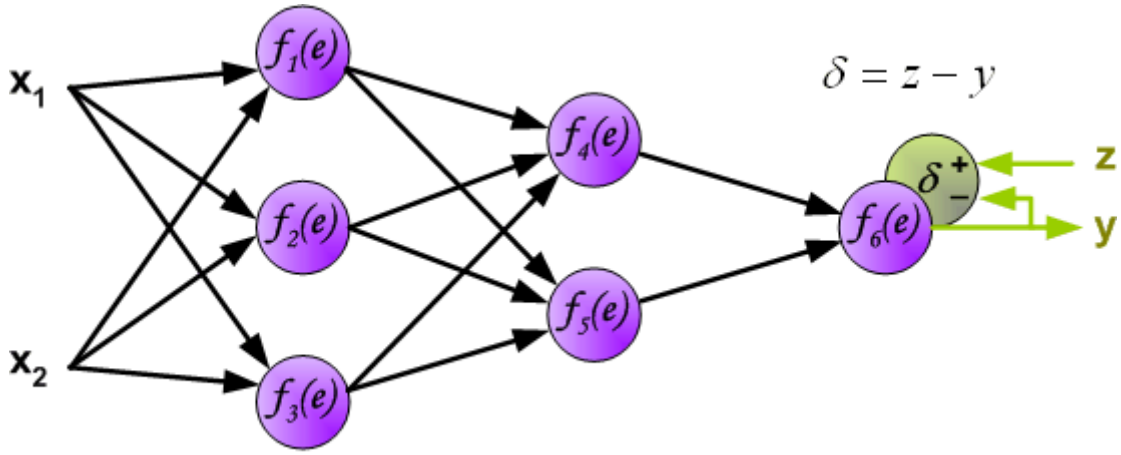




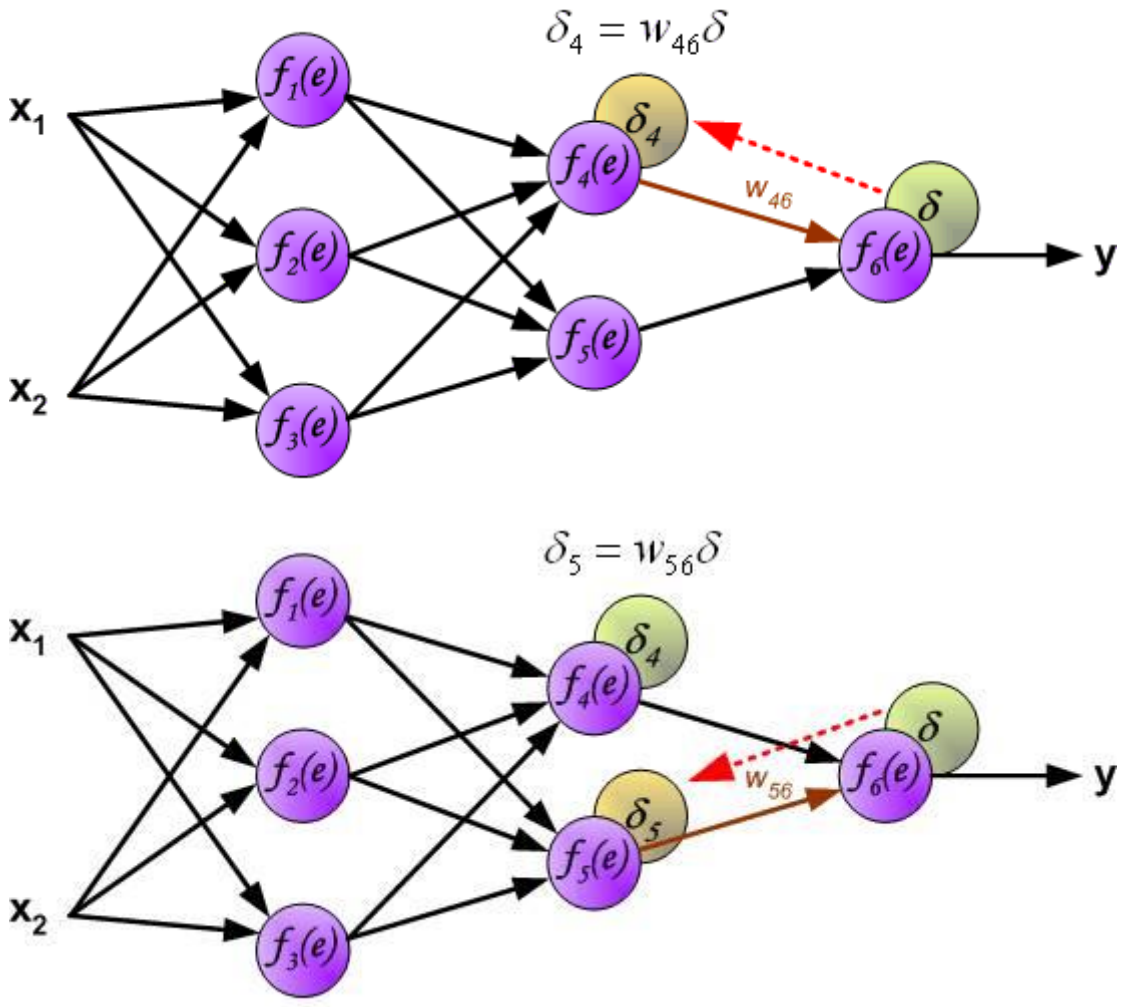
输出层信号传播：



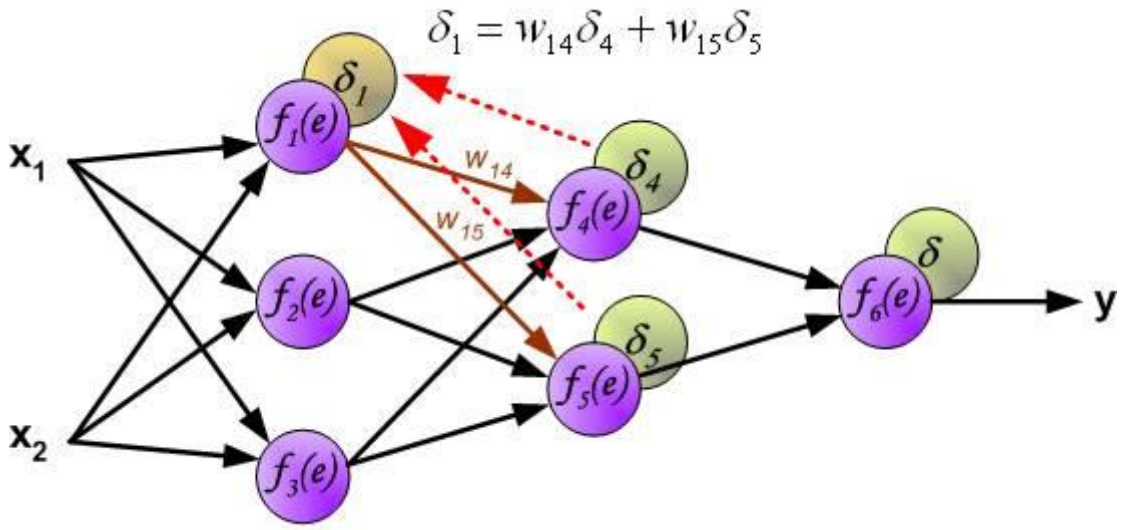
在下一个算法步骤中，将网络 y 的输出信号与训练数据集中的输出值（目标）进行比较。差异称为输出层神经元的误差信号 δ 。

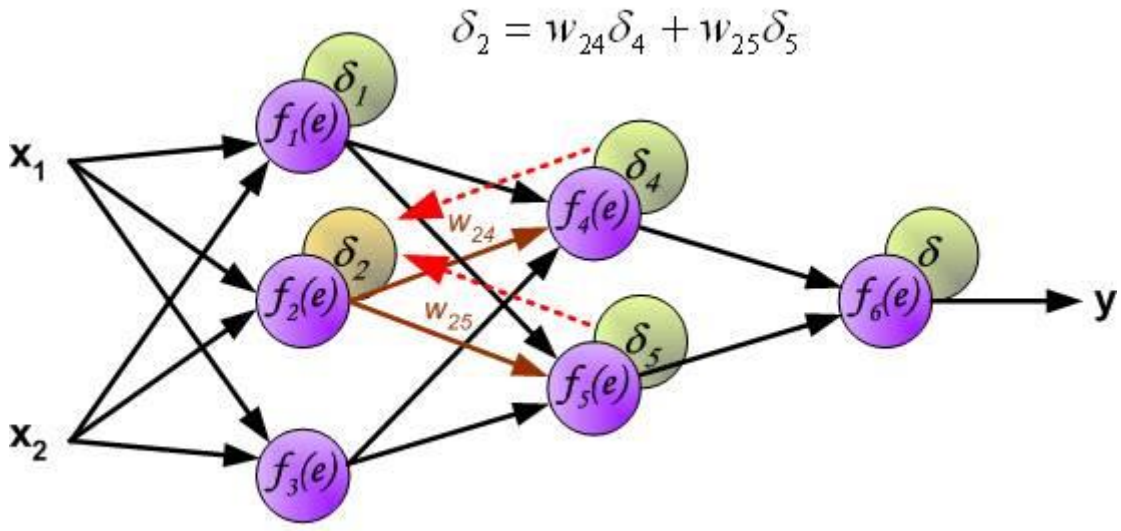


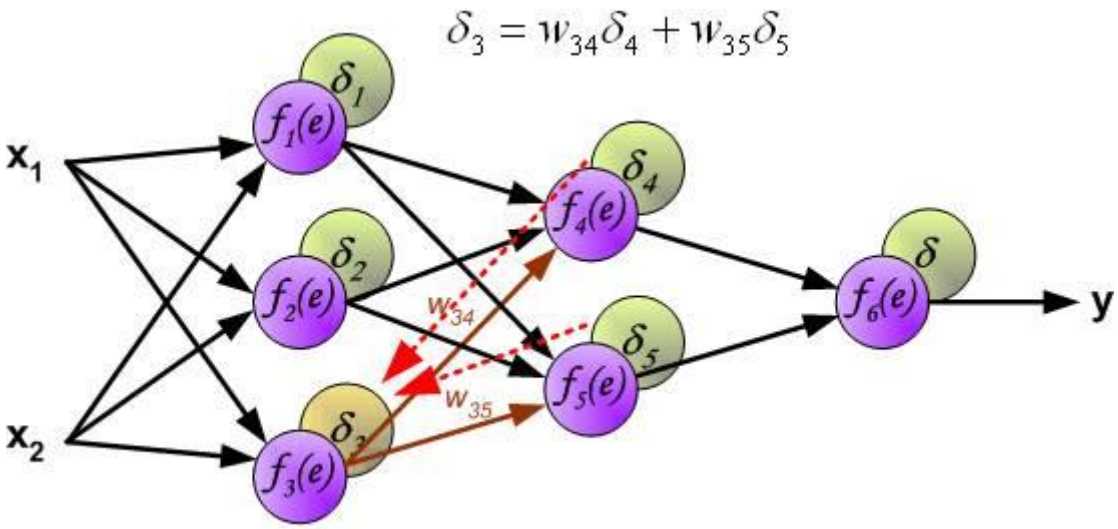
因为隐层神经元的输出值（训练集没有隐藏层的目标值）是未知的，所以不可能直接计算内部神经元的误差信号。多年来，一直没有找到训练多层神经网络的有效方法。直到在八十年代中期，反向传播算法才被制定出来。反向传播算法是将误差信号 δ（在单个训练步骤中计算）传播回所有神经元，对于神经元来说，误差信号反向传播。



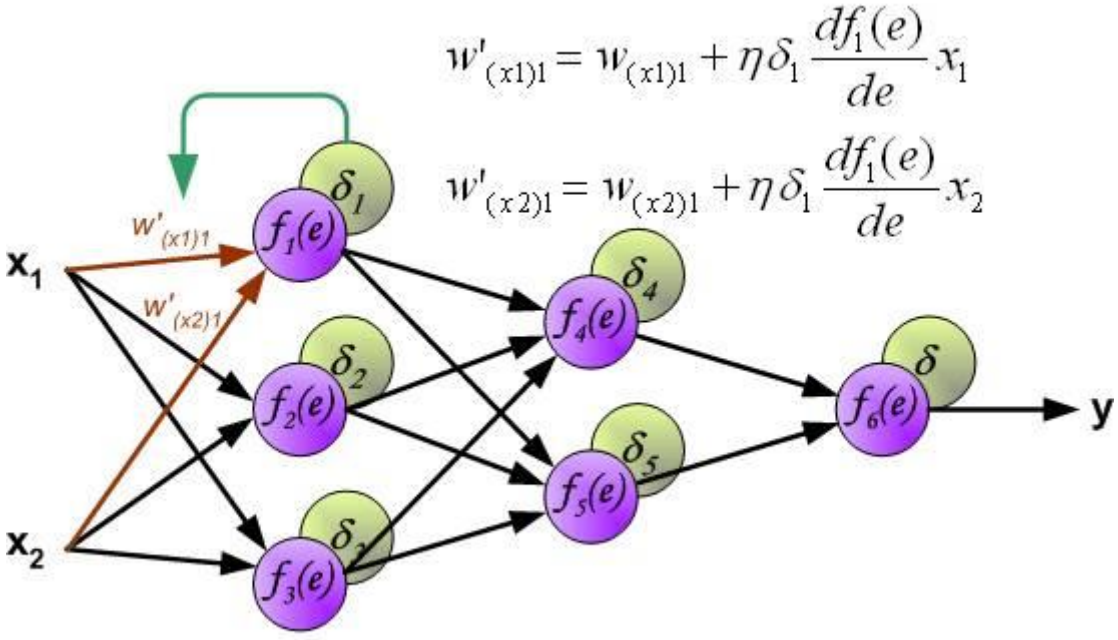
用于传播误差的权重系数 w_mn 等于前向计算使用的权重系数，只是数据流的方向改变（信号从输出到输入一个接一个地传播）。该技术用于所有网络层。如果误差是来自多个神经元，则把它们加起来(译者注：反向来看，也是加权和)。下图所示：





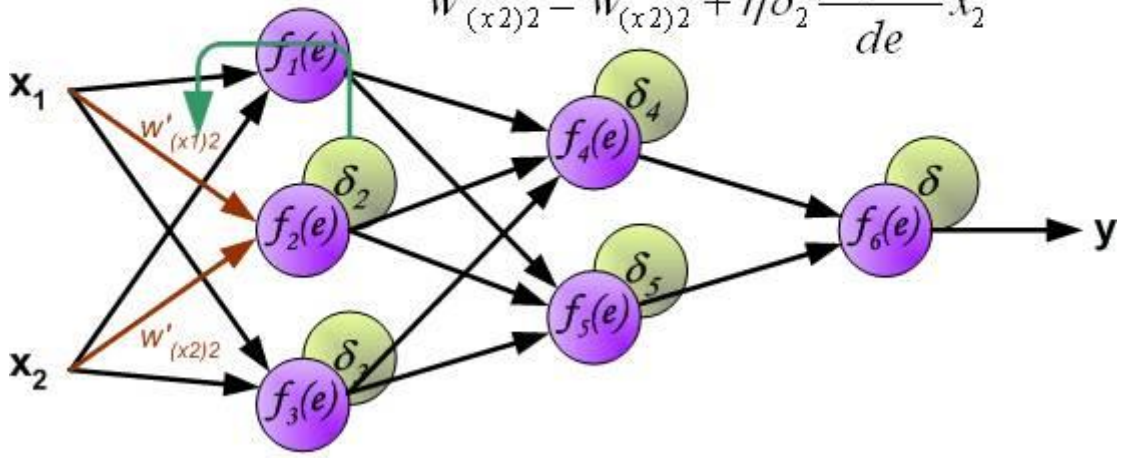


计算每个神经元的误差信号，用于修改每个神经元输入连接的加权系数。在下面的公式中， df(e)/de 表示神经元激活函数的导数。影响权重的因素除了神经元激活函数的导数之外，还有反向传播的误差信号，以及神经元输入方向连接的前一个神经元。（译者注：这里忽略了推导过程，直接给出权重的修改方法。具体的推导过程参考我的前一篇文章：《误差反向传播算法浅解》。原理是一样的，影响权重的三个因素体现在下面的公式中。）。



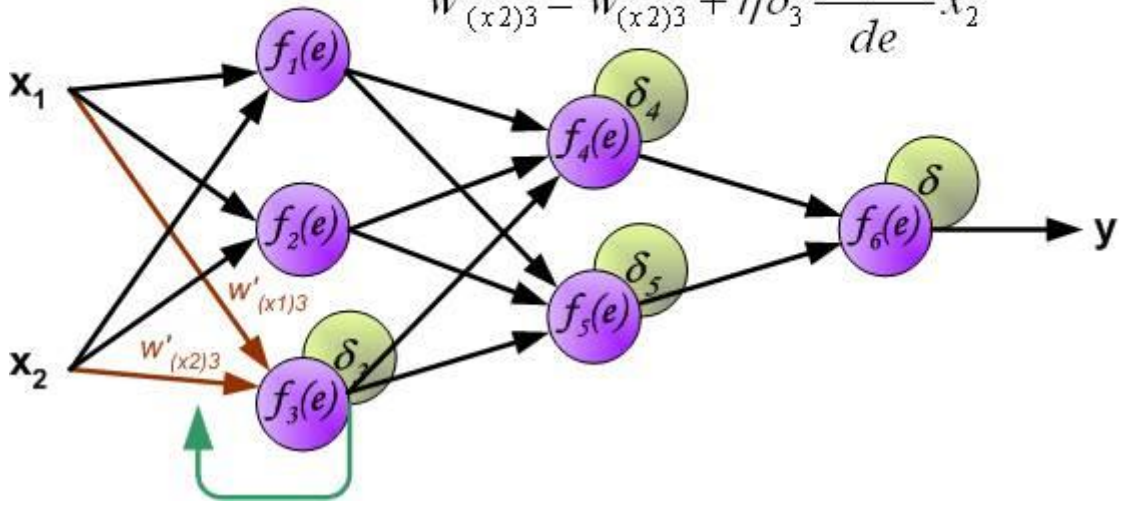
$$w'_{(x1)2} = w_{(x1)2} + \eta \delta_2 \frac{df_2(e)}{de} x_1$$

$$w'_{(x2)2} = w_{(x2)2} + \eta \delta_2 \frac{df_2(e)}{de} x_2$$



$$w'_{(x1)3} = w_{(x1)3} + \eta \delta_3 \frac{df_3(e)}{de} x_1$$

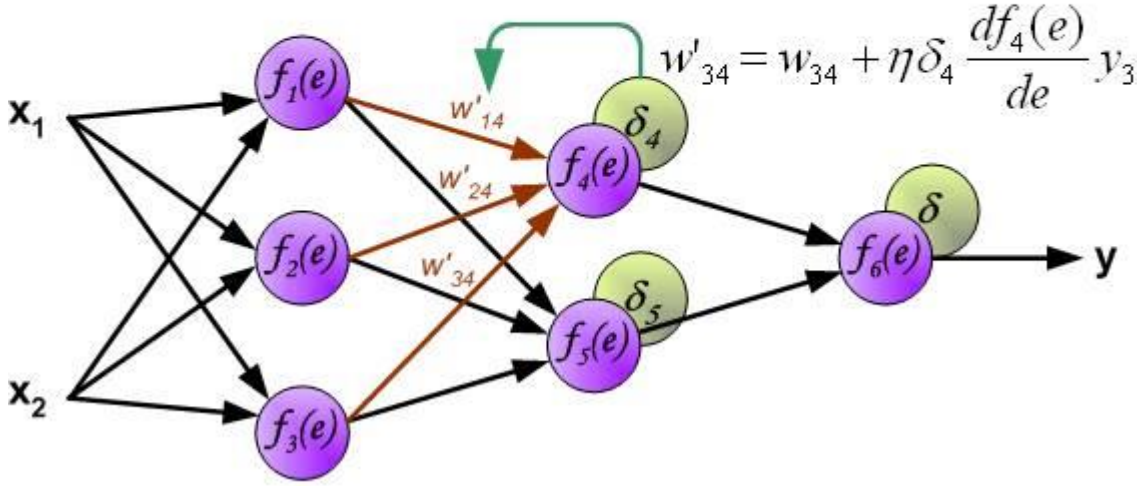
$$w'_{(x2)3} = w_{(x2)3} + \eta \delta_3 \frac{df_3(e)}{de} x_2$$



$$w'_{14} = w_{14} + \eta\delta_4 \frac{df_4(e)}{de} y_1$$

$$w'_{24} = w_{24} + \eta\delta_4 \frac{df_4(e)}{de} y_2$$

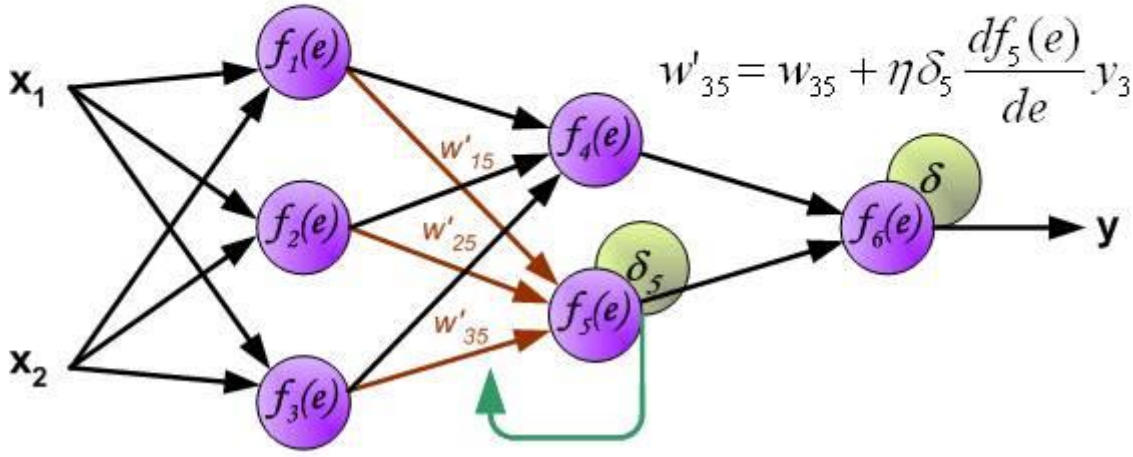
$$w'_{34} = w_{34} + \eta\delta_4 \frac{df_4(e)}{de} y_3$$

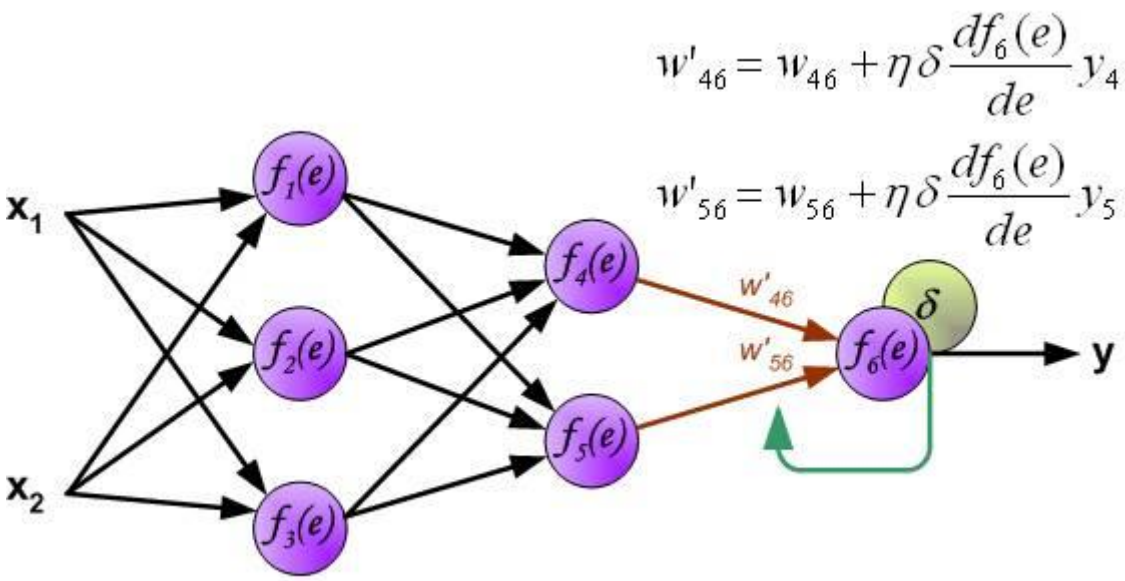


$$w'_{15} = w_{15} + \eta\delta_5 \frac{df_5(e)}{de} y_1$$

$$w'_{25} = w_{25} + \eta\delta_5 \frac{df_5(e)}{de} y_2$$

$$w'_{35} = w_{35} + \eta\delta_5 \frac{df_5(e)}{de} y_3$$





其中系数 η 影响网络训练速度（译者：训练步长）。有几种技术来选择此参数。第一种方法是开始具有较大参数值。当权重系数正在建立时，参数逐渐减少。第二个方法是开始用小参数值进行训练。在训练过程中，参数逐渐增加，然后在最后阶段再次下降。开始具有低参数值的训练过程能够确定权重系数。