

# 深度学习中的数学原理

鲁国锐

学号: 17020021031

上课序号: 5

2020 年 3 月 18 日

## 摘要

自 2016 年 AlphaGo 打败围棋世界冠军开始,深度学习迅速走进人们视野并为大众所津津乐道。然而,深度学习究竟是如何工作并取得举世瞩目的成就的,大部分人却对此茫然无知。故而本着科普的目的,本文将主要从直观的角度来简单介绍深度学习中的数学原理,给出必要的推导过程,之后还会列出深度学习中正面临的一些问题,同时给出简要分析。最后我们将对深度学习的现状做出总结并展望其今后的发展。

## 1 深度学习的发展

机器学习是人工智能领域的一个重要学科,自其产生以来在诸多方面都取得了巨大的成功。深度学习则是机器学习的一个重要分支,也是目前实现人工智能的一条重要途径。

深度学习是神经网络发展到一定时期的产物[8]。尽管它近年来才逐渐为人们所熟知,但其起源可追溯到 1943 年 McCulloch 等人[7]提出的 McCulloch-Pitts 计算结构,它通过模拟人体神经元的工作原理来解决问题,但需要手动设置权重,十分不便。之后虽然由 Rosenblatt 教授[2]提出了可以自动设置权重的感知机模型,但随后又被 Minsky 教授和 Paper 教授于 1969 年证明感知机模型只能解决线性可分问题,并且否定了多层神经网络训练的可能性[1],致使此后很长时间内该领域的研究基本处于停滞状态。

直到 20 世纪 80 年代, Rumelhart 团队提出反向传播算法(Back Propagation, BP)[6],开启了机器学习的“浅层学习”时代[9],神经网络才再次焕发生机。而 2006 年机器学习泰斗 Hinton 及其团队在 Science 上发表论文并首次提出“深度学习”概念[4],则又一次将关于神经网络的研究推向高潮。

随着“互联网+”模式的发展以及计算机性能的不断提高,人工智能开始逐渐与各个领域融合,而深度学习由于其可以利用大数据来填补专业知识的特性而一跃成为其中的佼佼者。各大公司争相成立

深度学习的研究院并发起与其相关的开发项目,高校之中也纷纷建立人工智能相关的博士、硕士点甚至本科专业。这一切都表明,深度学习正飞速地发展着,并不断壮大。

回顾深度学习的发展历史,其起源虽然久远,但它真正的兴起应当是在反向传播算法提出之后,所以也有文献在总结深度学习历史时直接从 20 世纪 80 年代开始[9]。有鉴于此,本文将重点介绍反向传播相关的概念及其数学原理。

## 2 前向传播

在讲反向传播之前首先要提一下前向传播,如图1所示。

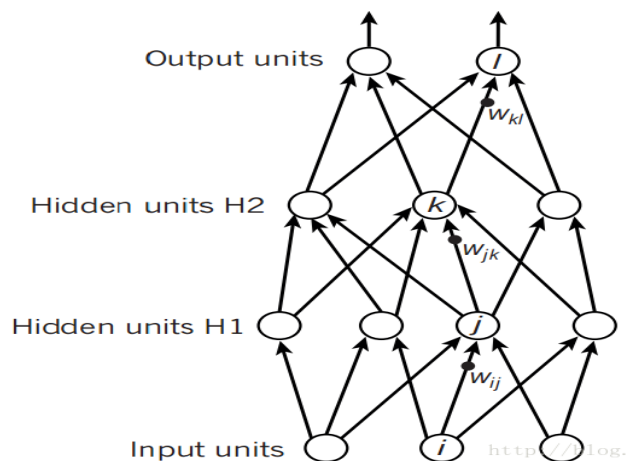


图 1: 神经网络示意图

所谓前向传播,其实就是计算一个十分复杂的复合函数。图1中每一个节点都代表着两次计算:

$$z^{[l]} = w^{[l]} \cdot a^{[l-1]} + b^{[l]} \quad (1)$$

$$a^{[l]} = \sigma^{[l]}(z^{[l]}) \quad (2)$$

其中  $\sigma(x) = \frac{1}{1+e^{-x}}$ , 我们称其为“激活函数”。

从这里我们可以看出,从输入节点开始,沿任意路径到一个输出节点,就构成了一个复合函数,且每一层函数都有着相同的形式。同时我们也可以看出,这若干个复合函数在一起就构成了一个神经网络。

### 3 反向传播

我们不难想见，若不加任何处理，随机设置公式2中的参数  $w$  和  $b$  的值，得到的结果肯定也是随机的，我们无法用其来进行任何决策或预估。所以我们需要减少误差，而要减小误差我们必须要有真值才行，于是这里又涉及到梯度下降法与有监督学习的问题，本文将分别在3.2节和3.1节讲述。

#### 3.1 数据集与有监督学习

深度学习得以进行的一大前提条件就是数据集。我们通过给模型大量的数据进行训练使其能够完成特定的任务。

数据集通常会分为训练集和测试集，训练集和测试集中又会被分为样本和真值两个部分。而对样本或真值的处理就分出了有监督和无监督学习。

有监督学习，顾名思义，就是需要人为的“监控”。这里所说的“监控”其实就是指为每一个训练样本和测试样本标注真值，或者是对输入样本进行一些手工处理。

相应的，如果不对数据做任何处理，则称为无监督学习。

#### 3.2 梯度下降法

关于梯度下降法的作用我们可以形象地理解为“找谷底”，如图??所示。

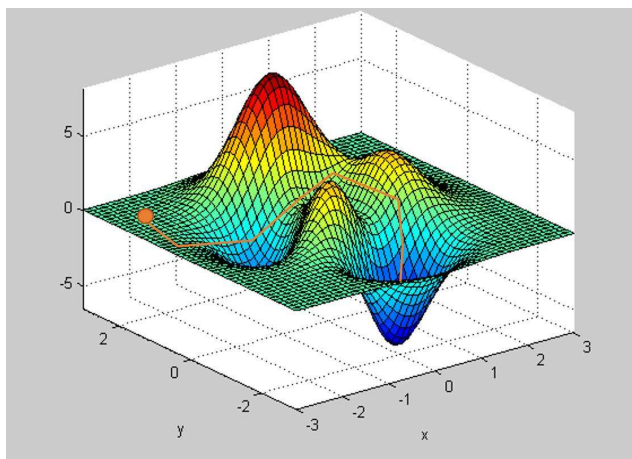


图 2: 梯度下降示例

假设我们现在正处在一个山谷的某处，需要走到谷底，但却不知道路线，怎么办？这种情况下一种理论上可行的办法是朝着当前位置最陡峭的方向向下迈一步，之后再向新位置最陡峭的方向向下迈一步，如此反复，不出意外的话，我们肯定可以到达谷底。

上述过程其实就是梯度下降法的原理。现在我们有了一个很大很深的神经网络，可以通过前向传播算出一个结果，同时对于每一个输入都有一个真值与之对应，所以接下来要做的就是如何利用这些条件来找出输入与输出之间的对应关系。

在第2节中曾说过，一个神经网络其实就是若干个复合函数并列放在一起，并且只要这些复合函数足够“复杂”，我们就可以用这个神经网络来拟合数据集中输入与输出的关系。但如果仅仅去随机初始化网络中的参数，很难得到一个令人满意的结果，所以在初始化之后我们还需要去优化这些参数，使得神经网络输出的结果尽可能接近真值。所用的方法就是梯度下降。

对于一个多元函数，我们要求其极小值，常用的做法是求它的导函数，令其等于 0。然而这种做法难以编程实现，因而可以借鉴上述“找山谷”的思路：先求出当前点的梯度，然后令坐标向梯度的负方向移动一小步，之后再求新位置的梯度，并重复之前的步骤，这样就可以逐渐移至极小值点附近。而在深度学习中，我们就是用这样的方式优化每一个节点中的参数值，最终使得整个网络可以较好地拟合输入输出之间的关系。

值得一提的是，我们不太关心神经网络拟合出的关系具体是怎样的。虽然每一个节点的函数形式都是固定的，我们也完全可以把它们的参数全部输出出来，但几乎没有人会这么做。因为一方面，一个神经网络的参数非常多，有时可能达到上十亿个，想要把若干个拥有如此多参数的复合函数写出来基本是不可能的；另一方面，在实际应用中也无须知道这些复合函数究竟是什么样的，我们只用将训练好的神经网络的结构与参数保存下来，在使用时将其导入即可。

#### 3.3 反向传播的具体实现

有了梯度下降的概念后，我们就不难想出其具体的实现方法，也就是反向传播了。

由于每一个节点的函数形式都是固定的，那么它的导函数形式也是固定的，根据式1和2我们可以很容易写出：

$$dz^{[l]} = da^{[l]} * \sigma'^{[l]}(z^{[l]}) \quad (3)$$

$$dw^{[l]} = dz^{[l]} \cdot a^{[l-1]} \quad (4)$$

$$db^{[l]} = dz^{[l]} \quad (5)$$

$$da^{[l-1]} = w^{[l]} \cdot dz^{[l]} \quad (6)$$

在优化时，我们令：

$$w^{[l]} = w^{[l]} - \alpha \cdot dw^{[l]} \quad (7)$$

$$b^{[l]} = b^{[l]} - \alpha \cdot db^{[l]} \quad (8)$$

这里  $\alpha$  叫学习率，相当于我们在??节中说的所跨一步的步长，是一个需要人根据经验来调节的超参数。

## 4 深度学习中的一些问题

第2节和第3节中我们介绍了深度学习最基本的数学原理及其简单的实现方式，它是整个算法最核心的内容，赋予了深度学习强大的力量。然而，这里面仍有着一些天然的缺陷。我们将在这一节中讨论其中的部分问题。

### 4.1 梯度消失与梯度爆炸

我们在第2节中曾提到过激活函数  $\sigma(x)$ ，同时第3节中不难看出在优化网络参数时无法回避对激活函数求导，那么我们现在来考察一下  $\sigma(x)$  导函数的性质：

$$\begin{aligned} \sigma'(x) &= \frac{e^{-x}}{(1+e^{-x})^2} \\ &= \frac{1}{1+e^{-x}} - \frac{1}{(1+e^{-x})^2} \\ &= \sigma(x) - [\sigma(x)]^2 \\ &\leq \frac{1}{4} \end{aligned} \quad (9)$$

由于我们在进行梯度反传时实际上是在用链式法则求导数，所以随着层数的加深，导数的数值会以指数形式下降，这就导致了反向传播越往前，效果越微弱，致使前后层训练的步调不一致，尤其是靠近输入层的参数，几乎无法得到优化。

相应的，如果激活函数的导数大于1，则在反向传播中导数值会以指数形式上升，造成梯度爆炸。不过这种情况很少见。

这个问题到现在应该说还没有彻底解决，但经过前人的努力，也有了许多不错的方案。其中一个效果比较好的是用其它激活函数来替代  $\sigma(x)$ ，而在所有替代品中，*relu* 函数脱颖而出，有效地缓解了梯度消失，并得到广泛认可。它的表达式如下：

$$\text{relu}(x) = \max(0, x) \quad (10)$$

### 4.2 局部极值点

从图??中我们可以看出，一个函数很可能有多个极值点，但实际需要的只有一个全局的最值点。然而，梯度下降法只能让我们找到某一个极值点，却无法判断这个点是否为全局最优。

针对这个问题也有许多解决方式被提出，不过很难说哪种方式可以彻底解决它。这里我也只介绍一种常见做法，就是用不同的权重来初始化整个网络，尤其是用别人已经在某些相近数据集上训练好的参数。这样做可以使得我们有很大的概率将整个网络参数初始化在最优点附近，从而避免落入其它局部极值点。

## 5 深度学习现状及发展

读过本文后，相信大家肯定还有很多疑惑。一方面受笔者水平、时间所限，很难在这短短数千字内将深度学习的原理彻底讲清楚；另一方面，关于深度学习基础的数学研究，实际上还处在非常初级的阶段，很多研究成果都是依据研究者的经验以及大量实验做出来的，但为何这么做，很多时候其实研究者自己也讲不清楚。

以 MUNIT [5] 为例，其整篇论文都是基于作者的若干个猜想建立起来的，再配以大量试验结果来证明自己方法的优越性，却始终给不出严格的推导证明。事实上，能够像 GAN [3] 那样在提出一个崭新的算法后又能给出理论上严格证明的还是少数。

在我看来，现在科研人员们对深度学习的推广就如同18世纪数学家们对微积分的推广。很多时候我们其实并不清楚到底发生了什么，可效果就是变好了，就连在人工智能领域赫赫有名的吴恩达也不得不承认这一点。而由于缺乏理论的支持，很多参数的调整以及网络结构的优化等许多任务都是依靠人的经验和直觉才做成的，且这些经验很难在各个领域之间移植。

尽管这样的状况大概还会持续相当长一段时间，但令人欣喜的是，对于深度学习的理论研究也有人在不断推进。我们有理由相信，深度学习和人工智能仍会飞速发展，并为人类、为社会带来越来越多的惊喜。

## 参考文献

- [1] M. Casper, M. Mengel, C. Fuhrmann, E. Her-  
rmann, M. Reichert, B. Appenrodt, F. Grün-  
hage, and F. Lammert. Perceptrons: An intro-  
duction to computational geometry. 75(3):3356–  
62, 1988. [1](#)
- [2] Rosenblatt F. Two theorems of statistical sepa-  
rability in the perceptron. 1958. [1](#)
- [3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi  
Mirza, Bing Xu, David Warde-Farley, Sherjil  
Ozair, Aaron Courville, and Yoshua Bengio.  
Generative adversarial nets. In *International  
Conference on Neural Information Processing  
Systems*, 2014. [3](#)
- [4] G. E. Hinton and R. R. Salakhutdinov. Reducing  
the dimensionality of data with neural networks.  
*Science*, 313(5786):504–507, 2006. [1](#)
- [5] Xun Huang, Ming-Yu Liu, Serge Belongie, and  
Jan Kautz. Multimodal unsupervised image-to-  
image translation. In *ECCV*, 2018. [3](#)
- [6] David E. Rumelhart, Geoffrey E. Hinton, and  
Ronald J. Williams. Learning representations  
by back-propagating errors. 323(6088):399–421,  
1986. [1](#)
- [7] Walter Pitts Warren S. McCulloch. A logical cal-  
culus of the ideas immanent in nervous activity.  
*The Bulletin of Mathematical Biophysics*, 1943.  
[1](#)
- [8] 付文博, 孙涛, 梁藉, 闫宝伟, and 范福新. 深度学  
习原理及应用综述. 计算机科学, (s1), 2018. [1](#)
- [9] 余凯, 贾磊, 陈雨强, and 徐伟. 深度学习的昨  
天、今天和明天. 计算机研究与发展, 50(9):1799–  
1804, 2013. [1](#)