1 关于深度学习的一点思考

周志华

（南京大学计算机软件新技术国家重点实验室，南京 210023）

1 引言

深度学习已被广泛应用到涉及图像、视频、语音等的诸多任务中并取得巨大成功。如果我们问“深度学习是什么？”很可能会得到这样的回答：“深度学习就是深度神经网络”。至少在目前，当“深度学习”作为一个术语时几乎就是“深度神经网络”的同义词，而当它指向一个技术领域时则如 *SIAM News* 头版文章所称[1]，是“机器学习中使用深度神经网络的子领域”。

关于深度学习有很多问题还不清楚。例如深度神经网络为什么要“深”？它成功背后的关键因素是什么？深度学习只能是深度神经网络吗？本文将分享一些我们关于深度学习的粗浅思考。

2 深度神经网络

神经网络并不是“新生事物”，它已经被研究了半个多世纪[2]。传统神经网络通常包含一个或两个隐层，其中每个“神经元”是非常简单的计算单元。如图 1 所示，神经元接收来自其他神经元的输入信号，这些信号通过连接权放大，到达神经元之后如果其总量超过某个阈值，则当前神经元就被“激活”并向外传递其输出信号。实际上每个神经元就是图 1 中非常简单的计算式，而所谓神经网络就是很多这样的计算式通过嵌套迭代得到的一个数学系统。

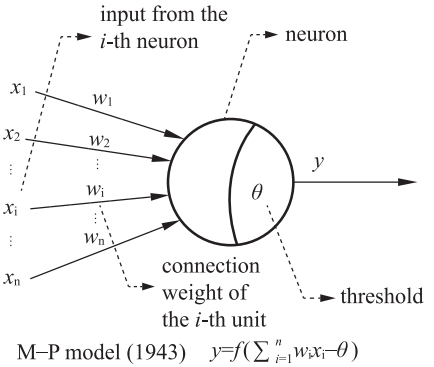

图 1 神经元模型

今天的“深度神经网络”是指什么？简单来说，就是有很多隐层的神经网络。例如 2012 年在计算机视觉领域著名的 ImageNet 竞赛夺冠的网络用了 8 层、2015 年是 152 层、2016 年是 1207 层……这样的网络明显是非常庞大的计算系统，包含了大量参数需要通过训练来确定。但有一个好消息：神经网络的基本计算单元是连续可微的。例如以往常用图 2 左边的 Sigmoid 函数作为神经元模型的激活函数，它是连续可微的；现在深度神经网络里常用图 2 右边这样的 ReLU 激活函数，它也是连续可微的。于是，在整个系统中可以相对容易地计算出“梯度”，进而就能使用著名的 BP 算法通过梯度下降优化来对神经网络进行训练。

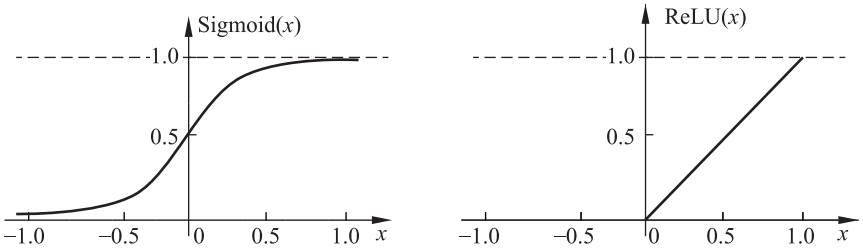

图 2 常用的神经元激活函数

有人以为深度神经网络的成功主要是因为“算力”有了巨大发展，因为神经网络早就有了，现在只不过是由于算力强了导致能算得更好了。这是一个误解。没有强大的算力当然难以训练出很深的网络，但更重要的是，现在人们懂得如何训练这样的模型。事实上，在 Hinton 等的工作[3]之前，人们一直不知道如何训练出超过五层的神经网络；这并不是由于算力不足，而是由于神经网络在训练中会遭遇“梯度消失”现象：在 BP

算法将神经网络输出层误差通过链式反传到远离输出层的部分时，可能会导出“零”调整量，导致网络远离输出层的部分无法根据输出误差进行调整，从而使得训练失败。这是从传统神经网络发展到深层神经网络所遇到的巨大技术障碍。Hinton 等通过“逐层训练后联合微调”来缓解梯度消失，使人们看到训练深层神经网络是可能的，由此激发了后来的研究，使得深度神经网络得以蓬勃发展。事实上，深度神经网络研究的主要内容之一就是设计有效措施来避免/减缓梯度消失。例如该领域一个重要技术进步就是用图 2 右边的 ReLU 函数来代替以往常用的 Sigmoid 函数，由于前者在零值附近的导数比后者更“平缓”，使得梯度不会因下降得太快而导致梯度消失。

显然，基本计算单元的“可微性”（differentiability）对深度神经网络模型至关重要，因为它是梯度计算的基础，而梯度计算是使用 BP 算法训练神经网络的基础。最近有一些研究尝试在深度神经网络中使用不可微激活函数，但其求解过程是在松弛变换后通过可微函数逼近，实质上仍依赖于基本计算单元的可微性。

3 为何“深”

虽然深度神经网络取得了巨大成功，但是“为什么必须使用很深的网络”一直没有清楚的答案。关于这个问题，几年前我们曾经尝试从模型复杂度的角度进行解释。

一般来说，机器学习模型复杂度与其“容量”（capacity）有关，而容量对模型的学习能力有重大影响，因此，模型的学习能力与其复杂度有关。机器学习界早就知道，如果能增强一个学习模型的复杂度，那它的学习能力往往能得到提升。怎样提高复杂度呢？对神经网络模型来说，很明显有两个办法：把模型加“深”，或把模型加“宽”。从提升模型复杂度的角度看，“加深”会更有效，因为简单来说，“加宽”仅是增加了计算单元，从而增加了基函数的数目；而在“加深”时不仅增加了基函数的数目，还增加了函数嵌套的层数，于是泛函表达能力会更强。所以，为提升复杂度，应该把网络“加深”。

有人可能会问，既然机器学习界早就知道能通过把神经网络模型加深来提升学习能力，为什么以往不这样做呢？

除了前面提到的“梯度消失”这个技术障碍，这还涉及另外一个问题：因为存在“过拟合”（overfitting），在机器学习中把模型的学习能力变强未必一定是件好事。过拟合是机器学习的大敌。简单来说，给定一个数据集，机器学习希望把数据集里所包含的“一般规律”学出来用于今后的数据对象，但有时候可能会把当前数据集本身的一些“特性”学出来却错误地当作一般规律去使用了，这就会犯错误，这就是过拟合。产生过拟合的

重要因素之一，就是模型的学习能力太强了，把不该学的东西也学到了。所以，以往在机器学习中都是尽量避免使用太复杂的模型。

现在为什么能使用深度神经网络这样的复杂模型了呢？有好几个重要因素：首先，现在有大数据了。机器学习中有很多缓解过拟合的策略，例如决策树剪枝、支持向量机正则化、神经网络提早终止训练等，但最简单有效的就是使用更多的数据。比方说，数据集中只有三千个样本，从它里面学出来的“特性”不太可能是一般规律，但如果有三千万，甚至三千万万个样本，那从它里面学出来的“特性”或许就已经是一般规律了。所以，现在有了大数据，我们不必再像以往那样对复杂模型“敬而远之”。第二，今天有 GPU、CPU 集群等强力计算设备，使我们有足够的算力来训练复杂模型。第三，经过机器学习界的努力，现在已经有很多有效训练深度神经网络这种复杂模型的技巧（trick），例如很多缓解神经网络梯度消失的办法。

小结一下，这套对“为什么深”的“复杂度解释”主要强调三点：第一，今天有大数据；第二，有强力的计算设备；第三，有很多有效的训练技巧。这三点导致现在能够使用高复杂度模型，而深度神经网络恰是一种便于实现的高复杂度模型。

上面这套解释有一定意义，例如它启发我们从复杂度的角度来研究深度学习中的一些机制如 dropout 等[4]。但这套解释有个重要问题没解决：为什么扁平的（宽的）网络不如深度神经网络？因为把网络“加宽”也能增加复杂度，虽然效率不如“加深”高。想象一下，如果增加无限个隐层神经元，那么即便仅使用一个隐层，网络的复杂度也可以提升非常高，甚至超过很多深度神经网络。然而在实践中人们发现，“宽”的浅层网络性能比不上相对“窄”的深层网络，这用复杂度难以解释。因此，我们需要更深入一点的思考。

我们问一个问题：深度神经网络最重要的功用是什么？

对此，机器学习界目前有一个基本共识，那就是“表示学习”（representation learning）。简单来说，如图 3 所示，以往我们拿到一个数据对象，比方说一幅图像，先要用很多特征比如说颜色、纹理等把它描述出来，这个步骤称为“特征工程”（feature engineering），然后我们再进行分类器学习。设计特征是许多应用研究领域的重要内容，例如计算机视觉与模式识别领域的研究中有相当一部分内容是关于设计视觉特征如 SIFT、HOG 等，而这个部分是机器学习研究所不关心的，后者主要关注相对通用、不依赖于具体应用域的技术，以往主要是针对表示为“特征向量”的数据去做分析建模。现在有了深度学习，只需把数据从一端扔进去，从另外一端就能得到模型，中间用到的特征描述可以通过深度学习自己来解决，这就是所谓的“特征学习”或者表示学习。从某种角度看，这是机器学习研究的疆域扩展到了一些应用研究领域的传统范围。与以往的

机器学习技术相比，在应用上来说这是一个很大的进步，因为不再需要完全依赖人类专家设计特征了，特征本身也可以跟学习器一起进行联合优化。

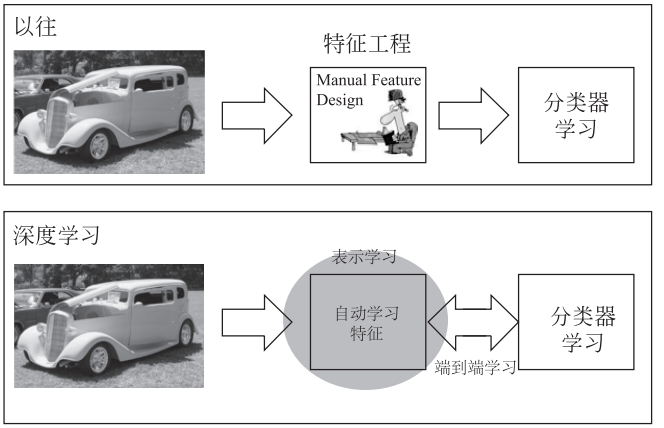

图 3　深度神经网络的重要功用：表示学习

进一步我们再问：对表示学习来说最关键的是什么？

我们的答案是：逐层加工处理。如图 4 所示，比方说在输入一幅图像时，在神经网络最底层看到是一些像素，而一层层往上会逐步出现边缘、轮廓等抽象级别越来越高的描述。虽然在真实的神经网络中未必有这么清晰的分层，但总体上确有自底向上不断抽象的趋势。

事实上浅层神经网络几乎能做到深层神经网络所做的别的任何事（例如提升复杂度），唯有深度的逐层抽象这件事，它由于层数浅而做不了。我们认为，“逐层加工处理”正是表示学习的关键，也是深度学习成功的关键因素之一。

但是在机器学习领域，逐层加工处理并不新鲜，以前已经有很多技术是在进行逐层加工处理。例如决策树、Boosting 都是“逐层加工处理”模型，但是与深度神经网络相比，它们有两个弱点：一是模型复杂度不够。例如决策树，对给定数据集来说其模型深度是受限的，假设仅考虑离散特征，则树的深度不会超过特征的个数，不像深度神经网络那样可以任意提升复杂度；二是在学习过程中缺乏特征变换，学习过程始终在同一个特征空间中进行。我们认为这两个因素对深度神经网络的成功也至关重要。

当我们同时考虑“逐层加工处理”和“内置特征变换”时就会发现，深度模型是非常自然的选择，因为基于深度模型可以容易地同时做到上面这两点。

在选用深度模型后，由于模型复杂度高、容易过拟合，所以我们要用大数据；它很难训练，所以我们要有训练技巧；计算开销大，所以我们要使用强力计算设备 …… 我

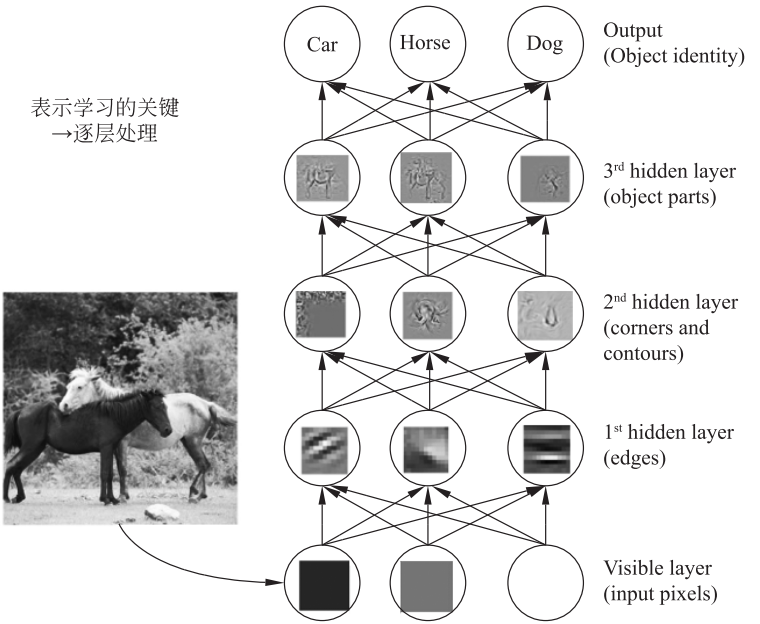

图 4　神经网络中自底向上逐层加工处理的示意图

们发现，这些是我们选择深度模型之后的结果，而不是选用深度模型的原因！ 这跟以前的认识不太一样。以前认为因为具备了这些条件而导致我们能使用深度模型，现在看来因果关系恰是反过来的。事实上，大训练数据、训练技巧，乃至强力计算设备都不仅限服务于深度模型，同样可以服务于浅层模型，因此，具备了这些条件并不必然导致深度模型优于浅层模型。

还有一点值得一提：拥有很大的训练数据时，需要使用复杂度高的模型，因为低复杂度模型无法对大数据进行充分利用。比方说仅使用一个简单的线性模型，那么有两千万样本还是两亿样本恐怕没有多少区别，因为模型已经“学不进去”了。而要模型有足够的复杂度，这又给使用深度模型加了一分，因为深度模型可以容易地通过加深层数来提升复杂度。

小结一下，我们的讨论分析导出的结论是，有三个关键因素：

- 逐层加工处理
- 内置特征变换
- 模型复杂度够

这是我们认为深度神经网络能够成功的关键原因，或者说是我们关于深度神经网络成功原因的猜想。有意思的是，这三个因素并没有“要求”我们必须使用神经网络模型。

只要能同时做到这三点，别的模型应该也能做深度学习。

4 为何有必要探讨 DNN 之外的深度模型

没有任何模型是完美的，深度神经网络模型也不例外。

首先，凡是用过深度神经网络的人都知道，需花费大量的精力来调参。这会带来很多问题。第一，调参经验很难共享，例如在图像任务上调参的经验很难在做语音任务时借鉴。第二，今天无论是科学界还是工程技术界都非常关注研究结果的可重复性，而深度学习恐怕是整个机器学习领域中可重复性问题最严重的子领域。常有这样的情况：一组研究人员发文章报告的结果，很难被其他研究人员重现，因为即便使用相同的数据、相同的方法，超参数设置稍有不同就可能使结果有巨大差别。

其次，神经网络的模型结构需要在训练前预设。但是在任务完成前，怎么能知道模型复杂度应该是多大呢？事实上，我们通常是在使用超过必需复杂度的网络。深度神经网络的一些最新研究进展，例如网络剪枝、权重二值化、模型压缩等，实质上都是试图在训练过程中适当减小网络复杂度。显然，使用过高复杂度的模型必然导致不必要地消耗了更多计算开销、导致对训练样本量不必要的高需求。有没有可能先用一个简单模型，然后在学习过程中自适应地增加模型复杂度呢？遗憾的是这对神经网络很困难，因为若网络结构未定，梯度求导对象在变化，那 BP 算法可就麻烦了。

深度神经网络的其他缺陷例如小数据上难以使用、黑箱模型、理论分析困难等就不赘述了。

或许有人会说，学术创新研究可能要考虑上述问题，而对应用实践来说只要性能好就行，有深度神经网络就足够了……其实即便从应用角度来看，探讨神经网络之外的深度学习模型也很有必要，因为虽然深度神经网络现在很流行，但在许多任务上（例如 Kaggle 的很多数据分析竞赛中）获胜的并非深度神经网络，而是随机森林、XGBoost 这些相对比较传统的机器学习模型。事实上，目前深度神经网络做得好的几乎都是涉及图像、视频、语音等的任务，都是典型的数值建模任务，而在其他涉及符号建模、离散建模、混合建模的任务上，深度神经网络的性能并没有那么好。

机器学习领域有一个著名的“没有免费的午餐”定理[2]，它告诉我们，没有任何一个模型在所有任务上都优于其他模型。实际上，不同模型各有自己的适用任务范畴，深度神经网络也不例外。因此，有充分的理由去探讨深度神经网络之外的深度学习模型，因为这样的模型或许能让我们在图像、视频、语音之外的更多任务上获得深度学习的性能红利。

小结一下，今天我们谈到的深度模型都是深度神经网络，用技术术语来说，它是多层可参数化可微分的非线性构件组成的模型，可以用 BP 算法来训练。这里有两个问题：一是现实世界中的问题多种多样，其所涉性质并不都是可微的，或能用可微构件最优建模的；二是机器学习领域几十年的积累，有许多构件能作为复杂模型的基础，其中相当一部分是不可微的。

能否基于不可微构件来构建新型深度学习模型？这是一个基础性挑战问题。一旦得到答案，就同时回答了其他一些问题，例如深度模型是否只能是深度神经网络？是否能不用 BP 算法训练？有没有可能让深度学习在图像、视频、语音之外的更多数据分析任务上发挥作用？……

我们最近在这方面进行了一些初步探索，提出了“深度森林”这种非神经网络的新型深度学习模型[5,6]。深度森林的基础构件是不可微的决策树，其训练过程不基于 BP 算法，甚至不依赖于梯度计算。它初步验证了上一节中关于深度学习奏效原因的猜想，即只要能做到逐层加工处理、内置特征变换、模型复杂度够，就能构建出有效的深度学习模型，并非必须使用神经网络。这种技术已经在大规模图像任务（我们认为此类任务的首选技术是深度神经网络）之外的许多任务中显示出优秀性能，包括互联网支付非法套现检测等大规模数据分析任务。在一定程度上验证了，在数值建模之外的任务上，有可能研制出新型深度学习模型来获得更好的性能。

需要注意的是，任何一种新技术要取得广泛成功都需经过长期探索。以深度神经网络中最著名的卷积神经网络为例，经过了三十来年、成千上万研究者和工程师探索和改进，才取得今天的成功。深度森林还在“婴儿期”，虽然在某些问题上已得以应用，但是不能期待它在广泛任务上都能够立即发挥作用。

实际上，我们以为深度森林探索的主要价值并不在于立即产生一种应用性能优越的新算法，而是为深度学习的探索提供一个新思路。以往我们以为深度学习就是深度神经网络，只能基于可微构件搭建，现在我们知道了这里有更多的可能性。好比说深度学习是一间黑屋子，里面有什么呢？以前我们都知道有深度神经网络，并以为仅有深度神经网络。现在深度森林把这个屋子打开了一扇门，今后可能会涌现更多的东西。这或许是这个探索在学科领域发展上更重要的意义。

参考文献

[1] J. Sirignano. Deep learning models in finance. SIAM News, 2017, 50(5): 1.

[2] 周志华. 机器学习. 北京: 清华大学出版社, 2016.

[3] G. E. Hinton, S. Osindero, and Y.-W. Simon. A fast learning algorithm for deep belief nets. Neural Computation, 2006, 18(7): 1527-1554.

[4] W. Gao and Z.-H. Zhou. Dropout Rademacher complexity of deep neural networks. Science China Information Sciences, 2016, 59(7): 072104: 1-072104: 12.

[5] Z.-H. Zhou and J. Feng. Deep forest: Towards an alternative to deep neural networks. In: IJCAI, 2017: 3553-3559.

[6] Z.-H. Zhou and J. Feng. Deep forest. National Science Review, 2019.

2 随机梯度下降郎之万动力学的泛化分析

王立威[1] 牟文龙[1] 翟曦雨[2] 郑 凯[1] 陈骁宇[1]

（[1]北京大学，北京 100871；[2]中国科学技术大学，合肥 230026）

1 介绍

现代统计学习理论的主要目标之一是为学习算法和模型给出依赖于算法和数据的泛化上界。学习算法可以使用很大或者很复杂的假设空间，但是其随机化探索空间的方式控制了实际表达能力，这一过程与数据相关。因此，依赖于算法的泛化界通常比例如 VC 维度和 Rademacher 复杂性等较经典的基于模型表达容量的分析更为深入。特别是对于随机梯度方法，迭代次数和步长可以用作隐式正则化并限制模型容量的增长。在凸学习问题中，随机梯度法方面已经有很多深入的研究，这些研究提出了与算法相关的泛化边界，但对于非凸情况知之甚少。尽管如此，人们往往认为凸问题下的相关理论在非凸情形依然成立。随机梯度方法的普遍成功不仅归因于计算速度，还归因于其学习理论上的优点，这被称为“收敛更快，泛化更好”。

依赖于算法的泛化界最重要的应用领域可能是深度学习。通过实验证明，当与算法无关的模型容量太大时，无法保证有意义的泛化性能。以自然图像作为输入，这一工作表明标准神经网络可以在训练集中拟合完全随机的标签信息。显然，如果神经网络本身的容量是控制泛化性能的唯一因素，这样的网络根本没有泛化能力，那么实际使用中的深度神经网络模型将面临同样的风险。不过幸运的是，随机标签和真实标签的训练过程之间的关键区别在于运行时间：随机标签将花费显著更多的时间以达到最佳点。因此，随机梯度方法的依赖于算法的界有可能可以保证具有真实标签的良好泛化性能，而随机标签训练由于运行时间太长而不能得到合理的保证。从这个意义上说，经典的依赖于算法的泛化界限对于理解深度学习的泛化性能可能有其应用的价值，具有非凸目标函数的