深度学习中的无监督学习方法综述①

殷瑞刚[1], 魏 帅[2], 李 晗[2], 于 洪[2]

[1](解放军第 181 医院, 桂林 541002)
[2](国家数字交换系统工程技术研究中心, 郑州 450002)

摘 要: 从 2006 年开始, 深度神经网络在图像/语音识别、自动驾驶等大数据处理和人工智能领域中都取得了巨大成功, 其中无监督学习方法作为深度神经网络中的预训练方法为深度神经网络的成功起到了非常重要的作用. 为此, 对深度学习中的无监督学习方法进行了介绍和分析, 主要总结了两类常用的无监督学习方法, 即确定型的自编码方法和基于概率型受限玻尔兹曼机的对比散度等学习方法, 并介绍了这两类方法在深度学习系统中的应用, 最后对无监督学习面临的问题和挑战进行了总结和展望.

关键词: 自编码; 受限玻尔兹曼机; 无监督学习; 深度学习; 神经网络

Introduction of Unsupervised Learning Methods in Deep Learning

YIN Rui-Gang[1], WEI Shuai[2], LI Han[2], YU Hong[2]

[1](Chinese People's Liberation Army 181th Hospital, Guilin 541002, China)
[2](National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

Abstract: Since 2006, Deep Neural Network has achieved huge access in the area of Big Data Processing and Artificial Intelligence, such as image/video discriminations and autopilot. And unsupervised learning methods as the methods getting success in the depth neural network pre training play an important role in deep learning. So, this paper attempts to make a brief introduction and analysis of unsupervised learning methods in deep learning, mainly includs two types, Auto-Encoders based on determination theory and Contrastive Divergence for Restrict Boltzmann Machine based on probability theory. Secondly, the applications of the two methods in Deep Learning are introduced. At last a brief summary and prospect of the challenges faced by unsupervised learning methods in Deep Neural Networks are made.

Key words: auto-encoders; restrict boltzmann machine; unsupervised learning; deep learning; neural network

20 世纪八十年代 Hopfield 将能量函数引入到神经网络中, 用非线性动力学解释循环反馈网络的运行[1], 同一时期, 反向传播算法(Back Propagation 算法, 简称 BP 算法)[2]给多层神经网络提出了一种可靠的学习方法, 引起了神经网络的一次高潮. 但是在实际应用中, 如果在神经网络初始参数完全随机的情况下, BP 算法被证明在浅层神经网络中是有效的, 而对于更深层网络效果不佳, 这是因为在 BP 算法中误差是反向传播的, 对底层参数的影响较小, 也就是梯度弥散问题. 而复杂问题往往需要更多的抽象, 自然会对应到深层网络, 比如对自行车的图像识别来说, 通常需要首先从像素中抽象出线条, 再抽象圆形, 再抽象出轮子, 最后抽象出自行车, 事实上生物的脑神经系统的确具有丰富的层次结构, 比如曾获得过诺贝尔医学和生理学奖的 Hubel－Wiesel 模型. 与之相比, 浅层网络往往难以达到理想效果, 这也导致单靠 BP 算法难以在实际应用取得理想效果. 2006 年 Hinton 等人[3]提出了用自学习初始化参数, 然后再逐步调优的方法来解决深层网络的学习问题, 深层神经网络才开始受到重视. 自此之后, 无监督的初始化学习方法和深度学习的框架受到很多机器学习和人工智能领域的研究者广泛学习和研究.

① 基金项目:国家重点基础研究发展计划(973)(2012CB315901)
收稿时间:2015-12-08;收到修改稿时间:2016-01-11 [doi:10.15888/j.cnki.csa.005283]

最近的研究结果证明了深度学习方法在多个机器学习和人工智能领域的有效性, 特别是图像和语音处理领域. 2012 年 Hinton 和 Krizhevsky 等人[4]利用 GPU 实现了一个深度神经网络, 在 ImageNet[5]的比赛中取得了创纪录的结果, 他们训练了一个参数规模非常大的深度神经网络, 并通过 dropout 方法来抑制模型的过拟合, 在大规模图像分类任务上 Top 5 分类精度达到了 84.7%, 比第二名使用的 Fisher 向量编码算法[6]要高大约 10 个百分点. 2011 年以来, 微软研究院和谷歌的语音识别研究人员先后采用 DNN 技术降低语音识别错误率 20%～30%, 是语音识别领域 10 多年来最大的突破性进展[7]. 虚拟人脑是谷歌 2012 年研发出来的基于深度学习的具有自动学习能力的人工智能项目. 它采用 1000 台计算机共 16000 个计算节点, 利用 YouTube 网站上的视频作为训练集, 花费 3 天的时间训练出 9 层的深度自编码器网络. 其训练出来的深度神经网络已经可以模拟一些人脑的功能. 在完全没有标签的情况下, 该网络能够自动地从训练集中学习到某些概念, 例如, 当输入是“猫”的图像时某些节点的响应会很强烈, 而当输入的是“人脸”时另外一些节点会响应强烈, 而且这些节点对于输入图像的旋转, 平移等变化具有一定的不变性[8].

如今深度学习方法在许多应用中都取得了比较好的效果, 但是却缺乏坚实的理论基础, 针对特定问题需要多少层的神经网络, 每层神经网络需要多少个神经单元, 都只有经验公式, 没有理论支撑. 深层网络架构本身可以比浅层网络更简洁地表达复杂关系, 但是深层架构的不可或缺性和无监督学习算法的必要性还有待进一步论证. 不可否认的是, 作为构建深度学习的重要方法, 无监督学习技术所起的作用是不容忽视的. 无监督学习技术可以在没有标签的情况下自主学习数据的抽象形式, 不仅拓展了学习的范围, 也为神经网络提供了一个较优的初始化参数, 因此, 理解和分析无监督学习技术的机理和方法, 对于理解和拓展深度学习具有非常重要的意义.

论文的其余部分是这样组织的, 第 1 节介绍了无监督学习方法及其在深度学习中的应用, 第 2 节介绍确定型的无监督学习方法-自编码及降噪自编码技术, 第 3 节介绍概率型的无监督学习方法—基于受限玻尔兹曼机的对比散度学习方法, 第 4 节介绍了基于无监督学习方法的深度学习系统并简要介绍了两类无监督学习方法的区别, 最后, 对全文进行了总结和展望.

1 无监督学习方法及其在深度学习中的应用

机器学习算法可以分为有监督学习和无监督学习两种, 有监督学习是指训练的样本带有标签, 而无监督学习在训练过程中样本没有标签. 在现实世界中, 大部分样本是不带标签的, 所以无监督学习要比监督学习应用更广泛. 常用的无监督学习算法主要有主成分分析方法 PCA[9]等, 等距映射方法[10]、局部线性嵌入方法[11]、拉普拉斯特征映射方法[12]、黑塞局部线性嵌入方法[13]和局部切空间排列方法[14]等.

深度学习是由多层神经网络组成, 需要一层一层地抽取主要特征, 忽略次要细节, 所以深度学习中采用的无监督学习方法需要满足三个条件:

① 可以从多维空间中抽取主要特征映射映射至低维空间;

② 具有递归性;

③ 算法不能太过复杂, 否则深层架构的计算量太大.

从原理上来说 PCA 等数据降维算法同样适用于深度学习, 但是这些数据降维方法复杂度较高, 并且其算法的目标太明确, 使得抽象后的低维数据中没有次要信息, 而这些次要信息可能在更高层看来是区分数据的主要因素. 所以现在深度学习中采用的无监督学习方法通常采用较为简单的算法和直观的评价标准. 目前深度学习中的无监督学习主要分为两类, 一类是确定型的自编码方法及其改进算法, 其目标主要是能够从抽象后的数据中尽量无损地恢复原有数据, 一类是概率型的受限波尔兹曼机及其改进算法, 其目标主要是使受限玻尔兹曼机达到稳定状态时原数据出现的概率最大.

2 确定型无监督学习

2.1 自编码及稀疏自编码

自编码可以看作是一个特殊的 3 层 BP 神经网络, 特殊性体现在需要使得自编码网络的输入输出尽可能近似, 即尽可能使得编码无损(能够从编码中还原出原来的信息).

一个典型的自编码例子[15]如图 1 所示, 从可见层到第一个隐含层的转换相当于是一个编码过程(encoder), 从第一个隐含层到输出层相当于一个解码过程(decoder). 自编码过程使用的一般都是无标签数据, 输入数据(input data)经过第一层变换(encoder), 就

会被进行一定程度的抽象，得到一个更深层的编码(code)，然后可以通过第二层变换(decoder)得到一个近似于输入数据(input data)的输出数据(output data). 如果输入数据和输出数据相等，则表明该编码是无损的(这只是一个理想情况). 编码和解码过程可由式(1)至式(4)表示，其中 $a^{(2)}$为编码结果，$a^{(3)}$为解码结果，均为向量形式:

$$z^{(2)} = W^{(1)}x + b^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

$$z^{(3)} = W^{(2)}x + b^{(2)} \quad (3)$$

$$a^{(3)} = f(z^{(3)}) \quad (4)$$

代价函数如下所示，

$$\frac{1}{2}\times\left\|a^{(3)} - x\right\|^2 \quad (5)$$

但是在实际应用过程中，一般要考虑系数正则化，因此需要在代价函数里面加入这一项，则代价函数就如式(6)所示.

$$\frac{1}{2}\times\left\|a^{(3)} - x\right\|^2 + \frac{\alpha}{2}\times\left(\left\|W^{(1)}\right\|^2 + \left\|W^{(2)}\right\|^2\right) \quad (6)$$

容易看出在自编码网络中输入输出数目相同，当隐含单元数目和输入输出单元数目一样时，则恒等函数可以实现无损编码，但是这显然不是自编码所期望的. 一般隐含单元数码要小于输入输出单元数目，比如自编码的输入输出为一个 28*28 像素的图片，则输入输出单元为 784，隐含层单元为 50，则自编码学习到的编码方式就相当于是原始图像的一种压缩，当输入是完全随机变化时，这种压缩的意义也不大，值得庆幸的是事实上输入往往是全集中的一个很小的子集.

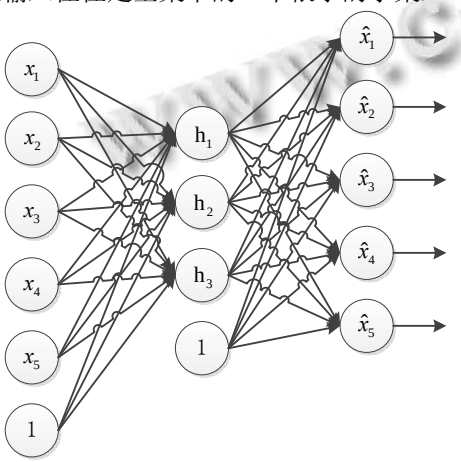

图 1 自编码三层网络示意图

此外，自编码中还常常采用了稀疏性的要求，这是从生物医学中得到的启发，在生物神经网络中，大部分神经在同一时刻都是处于抑制状态，只有少量神经才被激发，在人工神经网络中，假定激活函数为 Sigmoid 函数，则如果一个神经元的输出接近于 1，我们认为这个神经元被激活，输出接近于 0，我们认为它被抑制，那么稀疏性限制指的是使神经元在大部分时间内被抑制. 隐层神经元的评价兴奋程度可用公式(7)来衡量，即隐层神经元 j 在整个训练集上的平均激活状态.

$$\tilde{\rho}_j = \frac{1}{m}\times\sum_{i=1}^{m}(a_j^{(2)}(x^{(i)})) \quad (7)$$

稀疏性要求引发的代价函数一般用所有神经元的平均 KL(Kullback-Leibler)距离来衡量，所以，稀疏自编码的代价函数可由式(8)表示，有了代价函数，便可以按照 BP 算法求解自编码中的各个参数.

$$\frac{1}{2}\times\left\|a^{(3)} - x\right\|^2 + \frac{\alpha}{2}\times\left(\left\|W^{(1)}\right\|^2 + \left\|W^{(2)}\right\|^2\right) + \sum_{j=1}^{n}\left(\rho\times\log\frac{\rho}{\tilde{\rho}_j} + (1-\rho)\times\log\frac{1-\rho}{1-\tilde{\rho}_j}\right) \quad (8)$$

2.2 降噪自编码

虽然稀疏自编码可以学习一个相等函数，使得可见层数据和经过编码解码后的数据尽可能相等，但是其鲁棒性仍然较差，尤其是当测试样本和训练样本概率分布相差较大时，效果较差. 为此，Vincent 等[16]在稀疏自编码的基础上提出了降噪自编码，其基本思想是，以一定概率使输入层某些节点的值为 0，此时输入到可视层的数据变为 $\hat{x}$，隐含层输出为 y，然后由 y 重构 x 的输出 z，使得 z 和 x 的差值尽可能的小.

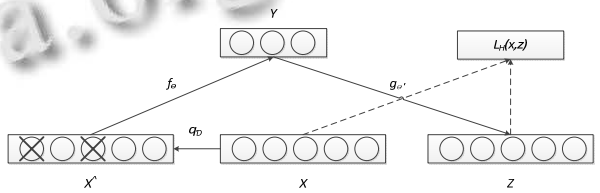

图 2 降噪自编码重构示意图[16]

降噪自编码可以从仿生学的角度进行解释，人体对外界的输入会进行综合判断，缺失某些少量信息对整体判断的影响不大，比如人眼判断某个手写数字图片时，图片中某些地方污损了，但仍然能够被识别.

3 概率型无监督学习

概率型无监督学习的典型代表就是限制玻尔兹曼机，限制玻尔兹曼机是玻尔兹曼机的一个简化版本，可以方便地从可见层数据推算出隐含层的激活状态.

3.1 玻尔兹曼机和受限玻尔兹曼机

作为单层反馈网络时玻尔兹曼机可以看作为 Hopfield 网络的一个扩展，网络中的每个节点与其他节点均相连，假若系统中共有 N 个节点，每个节点的状态可能为 ON 或者 OFF(可用 0 和 1 代替)，不妨将其标识为(s_1、s_2、…、s_N).

作为一个概率模型，玻尔兹曼机中每个节点的状态是随机的. 概率大小取决于网络的能量变化，假定神经元 i 和 j 之间的连接权值为 W_{ij}，神经元 i 的阀值为 Q_i，网络的能量函数定义为

$$E=-\sum_{i=1}^{N}\sum_{j=1}^{N}W_{ij}s_is_j+\sum_{i=1}^{N}Q_is_i \tag{9}$$

当神经元 s_i 的状态发生变化，假定其状态从 0 变为 1，则产生的能量差值为

$$\Delta E_i=E_{i=0}-E_{i=1}=\sum_{i=1}^{N}W_{ij}-Q_i \tag{10}$$

玻尔兹曼机的变化过程是一个马尔科夫过程，假定某个时刻的温度为 T，则在其他神经元不变的情况下，神经元 s_i 为 1 的概率为

$$p_{k(s_k=1)}=\frac{1}{1+e^{-\frac{\Delta E_i}{T}}} \tag{11}$$

容易看出，当 ΔE_i >0 时，$p_{k(s_k=1)}$ >1/2，即系统总是以较大的概率向能量低的状态变化，这与 Hopfield 网络相似，不同的是 Hopfield 网络只会固定地向低能量状态移动，而玻尔兹曼机则以概率方式进行迁移，其向低能量状态移动的概率较大，但是也有向高能量状态迁移的可能，只是这种可能性随温度的减小而减小. 玻尔兹曼机可以比 Hopfield 网络更好地寻找最优解，一般是从高温开始，高温时，网络忽略微小的能量差，可以粗略地寻找全局状态的结构，寻找到大致的最小解，然后随着温度的降低，快速收敛到最小解.

容易验证，假定网络最终达到热平衡状态，此时两个时刻玻尔兹曼网络的状态分别为 α 和 β，则 α 和 β 出现的概率满足

$$\frac{p_\alpha}{p_\beta}=e^{-\frac{(E_\alpha-E_\beta)}{T}} \tag{12}$$

即满足玻尔兹曼分布，其具有良好的数学性质，与信息论有着密切的联系. 受限玻尔兹曼机(Restrict Boltzmann Machine, RBM)模型[17]是两层神经网络，分为可见层和隐含层，如果直接将玻尔兹曼机中的一部分节点作为可见层,一部分作为隐含层，则可见层之间，隐含层之间，可见层和隐含层之间都有连接，使得训练和学习时的计算比较复杂. 受限玻尔兹曼机可见层节点之间和隐含层节点之间都没有连接，只有可见层和隐含层之间有连接，使其便于训练和学习.

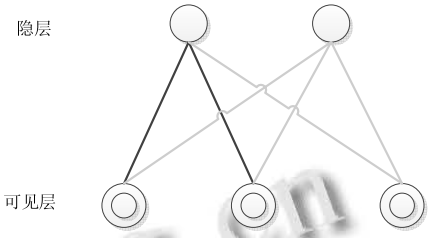

图 3　受限玻尔兹曼机示意图

3.2 基于受限玻尔兹曼机的无监督学习

只有层间有连接的结构决定了受限玻尔兹曼机在给定可见层的情况下，隐层各个节点的分布是相互独立的，可由式(13)进行计算

$$p_{(h_j=1|v,\theta)}=\frac{1}{1+e^{-(b_j+\sum_{i=1}^{N}v_iW_{ij})}} \tag{13}$$

同样地，给定隐层各个节点的状态，可见层各个节点的分布也是相互独立的，可由式(14)进行计算

$$p_{(v_i=1|h,\theta)}=\frac{1}{1+e^{-(a_i+\sum_{j=1}^{M}h_jW_{ij})}} \tag{14}$$

整个限制玻尔兹曼机的状态转移是一个马尔科夫过程，状态之间的转移概率容易计算，但是整体达到热平衡状态的分布概率难以计算. 与自编码网络追寻代价函数的最小值不同，限制玻尔兹曼追求的是训练样本的对数似然概率最大化，即寻找参数 θ 使得对数似然概率最大，对数似然概率如下所示:

$$\mathrm{L}(\theta)=\sum_{i=1}^{T}\log p\left(v^{(t)}\mid\theta\right)=\sum_{i=1}^{T}\log\frac{\sum_h e^{-E(v^{(t)},h|\theta)}}{\sum_v\sum_h e^{-E(v,h|\theta)}} \tag{15}$$

对其进行求导，便可以得到公式如式(16)至式(18)所示. 其中 $<X>_{data}$ 表示当可见层 V 确定时变量 X 的数学期望，$<X>_{model}$ 表示当热平衡状态时变量 X 的数学期望. 但是热平衡状态时各个状态的概率难以计算，所以需要采用 Gibbs 采样[18]的方法进行近似，Hinton 等证明了仅需一步采样便能得到较好的结果，这便是基于对比散度(Contrastive Divergence, CD)的快速学习算法[19].

$$\frac{\partial \mathrm{L}}{\partial W_{ij}}=\varepsilon(<v_ih_j>_{data}-<v_ih_j>_{model}) \tag{16}$$

$$\frac{\partial \mathrm{L}}{\partial a_i} = \varepsilon(< v_i >_{data} - < v_i >_{model}) \quad (17)$$

$$\frac{\partial \mathrm{L}}{\partial b_j} = \varepsilon(< h_j >_{data} - < h_j >_{model}) \quad (18)$$

虽然CD算法是RBM进行无监督学习的有效方法, 但是梯度近似可能会带来误差, 为此, Cho 等[20]采用了基于MCMC方法的PT算法, 可以更有效地训练RBM, 这点被 Desjardins[21]所证明, 但是 PT 算法采用的方法在时间复杂度和空间复杂度上都要远大于 CD 算法, Cho 等人后来又采用增强梯度估计和自适应学习率的方法[22], 提高了 PT 算法的性能, 并在 MNIST 手写数字数据集上证明了算法的有效性.

4 基于无监督学习的深度学习系统

基于无监督学习的深度学习系统主要包括栈式自编码神经网络和深度信任网络, 其基本单元分别是第三节中介绍的自编码器和受限玻尔兹曼机, 采用栈式结构搭建深层神经网络.

4.1 栈式自编码神经网络

栈式自编码神经网络[23]的每一层都是一个自编码器, 这种自编码器既可以是稀疏自编码器, 也可以是降噪自编码器. 底层自编码器的输出作为高层自编码器的输入, 依次迭代至所需的层数. 如果用 $W^{(k,1)}$, $W^{(k,2)}$, $b^{(k,1)}$, $b^{(k,2)}$代表第 k 层自编码器的参数 $W^{(1)}$, $W^{(2)}$, $b^{(1)}$, $b^{(2)}$, 则第k层自编码器的编码方式可用式(19)和式(20)表示, 其中 $a^{(l)}$是一个向量, 代表第 l 层神经元的激活状态. 如果是分类问题, 最顶层可以设置一个诸如 sofmax 的分类器.

$$z^{(l+1)} = W^{(l,1)}a^{(l)} + b^{(l,1)} \quad (19)$$

$$a^{(l+1)} = f(z^{(l+1)}) \quad (20)$$

相应地, 第 k 层自编码器的解码方式可用下式表示, 其中 n 代表神经网络的总层数:

$$z^{(n-l)} = W^{(n-l,2)}a^{(n-l+1)} + b^{(n-l,2)} \quad (21)$$

$$a^{(n-l)} = f(z^{(n-l)}) \quad (22)$$

栈式自编码神经网络一种比较好的初始化方法就是采用贪婪的逐层初始化, 首先从最低层开始, 因为最底层的输入即为可见层, 是已知的, 所以可以按照第 2 节中介绍的方法初始化参数 $W^{(1,1)}$, $W^{(1,2)}$, $b^{(1,1)}$, $b^{(1,2)}$, 得到第一层隐含层神经单元的激活状态 $a^{(1)}$. 接着以 $a^{(1)}$为第二层自编码器的输入, 就可以初始化第二层网络的参数 $W^{(2,1)}$, $W^{(2,2)}$, $b^{(2,1)}$, $b^{(2,2)}$, 得到第二层隐含层神经单元的激活状态 $a^{(2)}$, 重复这个步骤直至达到所需的层数. 如果顶层采用的是 sofmax 分类器, 则用最顶层的激活状态作为输入, 初始化 sofmax 分类器所需的参数.

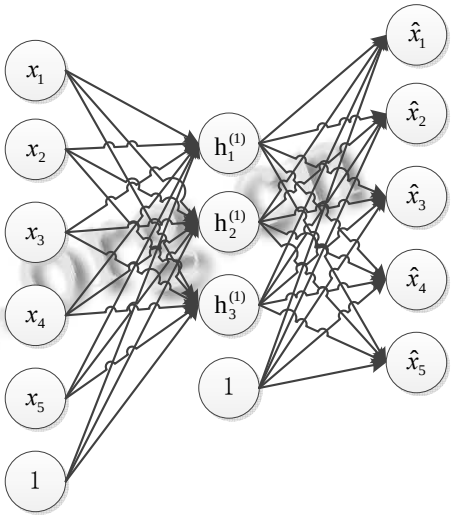

图 4 栈式自编码第一层示意图

虽然采用贪婪的无监督学习算法为深层网络找到了一个较优的初始化参数, 但是这还不够好, 采用 BP 算法可以使网络得到更优的结果. 如果只是关心分类问题, 可以忽略解码参数, 只用 BP 参数调整编码参数.

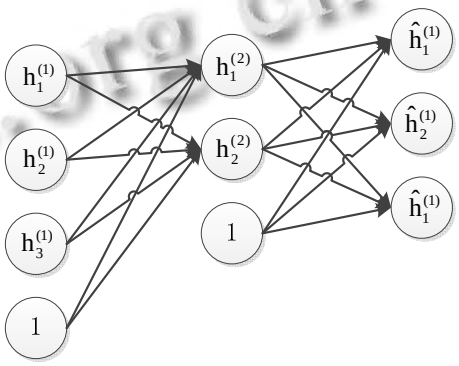

图 5 栈式自编码第二层示意图

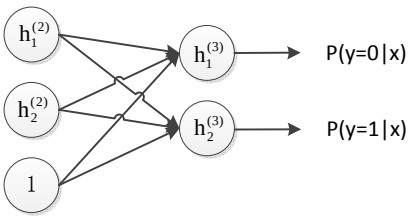

图 6 栈式自编码顶层示意图

4.2 深度信念网络

深度信念网络 (Deep Belief Network, DBN)[24]由 Geoffrey Hinton 在 2006 年提出. 基本思想是以受限玻尔兹曼机为基本单元搭建的信任网络, 采用了逐层初始化和整体反馈的方法, 成功克服了深层网络难以训练的弊端, 开启了深度学习的热潮. 它不仅可以用来判别, 还可以进行生成.

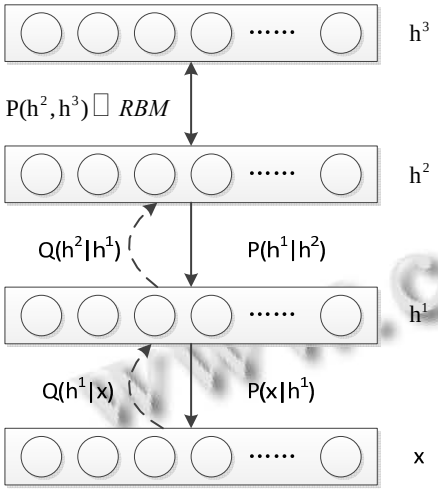

图 7 深度信念网络架构图[17]

深度信念网络具体结构如图 7 所示, 最上面的两层之间采用无向连接, 组成联合内存 (associative memory). 其余的层间均采用有向连接, 其中向上的权重称为认知权重, 用来自下而上产生认知, 向下的权重为生成权重, 用于自上而下生成数据. 最底层是可见层, 由训练数据决定, 其中的每个神经元代表可见层向量中的一维. DBN 的预训练过程是一层一层地进行的, 在每一层中, 用可见层来推断隐层, 再把这一隐层当作下一层 (高一层) 的可见层.

DBN 的训练过程如下:

① 以训练数据为输入训练底层的 RBM;

② 以上一步中产生的隐含层状态为输入训练该层 RBM;

③ 重复步骤 2,直到产生出模型所需要的隐层数(除了顶部两层);

④ 顶层的两层 RBM 训练方式需要考虑数据有标签的情况, 如果训练集中的数据有标签, 那么在顶层的 RBM 训练时, 需要有代表分类标签的神经元, 一起进行训练. 假设顶层 RBM 的顶层层有 10 个神经元, 训练数据的分类一共分成了 2 类, 那么顶层 RBM 的神经元数目可以设为 12, 对每一训练数据, 如果属于相应的分类, 则相应的标签神经元置为 1, 而其他的置为 0.

DBN 的调优过程与栈式自编码相似, 可以以交叉熵为代价函数, 优化参数使其达到最小值, Geoffrey Hinton 将其分为两个阶段, 分别是 Wake 阶段和 Sleep 阶段.

① Wake 阶段: 认知过程, 通过外界的特征和向上的权重 (认知权重) 产生每一层的抽象表示 (结点状态), 并且使用梯度下降修改层间的下行权重 (生成权重) . 也就是“如果现实跟我想象的不一样, 改变我的权重使得我想象的东西就是这样的”.

② Sleep 阶段: 生成过程, 通过顶层表示 (醒时学得的概念) 和向下权重, 生成底层的状态, 同时修改层间向上的权重. 也就是“如果梦中的景象不是我脑中的相应概念, 改变我的认知权重使得这种景象在我看来就是这个概念”.

4.3 两类学习算法的比较

总体来说, 采用 RBM 构建的深度学习系统如 DBN 效果要比自编码构建的栈式自编码好, 这主要是因为自编码只能通过重构平均误差的方法来进行逼近, 而 RBM 则可以通过极大化似然估计来逼近真实的联合概率分布. 但是如果用降噪自编码构成栈式降噪自编码深度学习系统, 则其学习能力与 DBN 相当, 这主要是由于降噪自编码可以随机产生一些输入, 增强了自编码的抗噪性, 提升了泛化能力. 此外, 稀疏自编码在图像的初步特征提取上显示了类似人脑的功能, 将 2.1 中所述的算法用于图像特征提取, 可以发现对隐层神经刺激最大的是类似边、角的图像碎片, 与人脑相似, 这是 RBM 所不具备的.

在学习算法和效率方面, 自编码支持各种自定义的代价函数, 只要这些函数相对于参数是连续的; 而 RBM 的训练算法则相对有限, 基本上都是通过最大化似然估计来求解[25-27]. 算法效率与具体的学习算法相关, 如果采用简单的 CD 训练算法, 则 DBN 预训练算法的效率比栈式自编码预训练算法的效率高.

5 结语

利用自编码和 RBM 等无监督学习方法对深层神经网络架构进行初始化, 然后再逐步调优的方法使得深度神经网络在许多领域都得到了巨大成功. 但是, 深度学习架构的许多方法都来自于经验或者受仿生学

的启发，缺乏坚实的数学基础. 目前，深度网络还要许多问题需要解决，主要有

① 如何更好地得到好的数据抽象，比如可以在自编码中可以加入稀疏性及抗噪性要求来得到更好的表达，但是还有没有更好的方式，以及这些方式对最终效果的定量分析.

② 现在的神经网络每层都采用相同的函数(sigmoid 函数)，这样使得容易进行计算，但是在生物大脑中，每层神经元的工作方式都不同，有没有更好的方法来体现这种差别，也是一个值得探讨的问题.

③ 针对具体问题采用多少层网络结构，多少神经单元可以达到较好的效果，还是网络层次越深，神经单元越多越好，在大神经网络里面如何避免欠拟合问题，在小神经网络中如何避免过拟合问题.

④ 虽然当前的无监督训练算法已经比较简便了，但是在大型的深度神经网络中计算量仍然非常巨大，寻找更加简便有效的训练算法也是深度学习需要解决的一个重要课题.

参考文献

1 Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. Proc. of the National Academy of Sciences of the United States of America, 1982, 79(8): 2554–2558.

2 Rumelhart D, Hinton GE, William R. Learning representations by back-propagating errors. Nature, 1986, 323(6088): 533–536.

3 Hinton GE, Salakhutdinov R. Reducing the dimensionality of data with neural networks. Science, 2006, 313(7): 504–507.

4 Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, eds. Proc. of the Advances in Neural Information Processing Systems. Lake Tahoe: Neural Information Processing Systems Foundation. 2012. 1106–1114.

5 Deng J, Berg A, Satheesh S. Large scale visual recognition challenge. http://www.image-net.org/challenges/LSVRC/2012/index, 2013.

6 Perronnin F, Snchez J, Mensink T. Improving the fisher kernel for large-scale image classification. Proc. of the European Conference on Computer Vision. Crete: Springer Berlin Heidelberg. 2010. 6314. 143–156.

7 余凯,贾磊,陈雨强,徐伟.深度学习的昨天、今天和明天.计算机研究与发展,2013,50(9):1799–1804.

8 郑胤,陈权崎,章毓晋.深度学习及其在目标和行为识别中的新进展.中国图象图形学报,2014,19(2):175–184.

9 Jose CA. Fast On-line alogrithm for PCA and its convergence characteristic. IEEE Trans. on Neural Network, 2000, 4(2): 299–307.

10 Tenenbaum JB, de Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. Science, 2000, 290(5500): 2319–2323.

11 Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. Science, 2000, 290(5500): 2323–2326.

12 Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Vomputation, 2002, 15: 1373–1396.

13 Donoho DL, Grimes C. Hessian eigenmaps: Locally linear embedding techniques for high dimensional data. PNAS, 2003, 100(10): 5591–5596.

14 Zhang Z, Zha HY. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM Journal on Scientific Computing, 2004, 26(1): 313–338.

15 Andrew N. UFLDL_Tutorial. http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial, 2015.

16 Vincent P, Larochelle H, Bengio Y. Extracting and composing robust features with denoising autoencoders. Proc. of the 25th International Conference on Machine Learning. New York: ACM. 2008. 1096–1103.

17 刘建伟,刘媛,罗雄麟.玻尔兹曼机研究进展.计算机研究与发展,2014,51(1):1–16.

18 Liu JS. Monte Carlo strategies in scientific computing. 1th ed., New York: Springer-Verlag, 2001: 129–151.

19 Hinton GE. Training products of experts by minimizing contrastive divergence. Neural Computation, 2002, 14(8): 1771–1800.

20 Cho K, Raiko T, Ilin A. Parallel tempering is efficient for learning restricted Boltzmann machines. Proc. of 2010 International Joint Conferrence on Neural Networks. New York. ACM. 2010. 1–8.

21 Desjardins G, Courville A, Bengio Y. Tempered Markov chain Monte Carlo for training of restricted Boltzmann machines. Journal of Machine Learning Research-Processing Track, 2010, 9(1): 145–152.

22 Cho K, Raiko T, Ilin A. Enhanced gradient and adaptive learning rate for training restricted Boltzmann machines. Proc. of 28th International Conferrence on Machine Learning. New York. ACM. 2011. 105–112.

23 Bengio Y. Learning deep architectures for AI. Foundations and Trends in Machine Learning, 2009, 2(1): 1–127.

24 Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural Computation, 2006, 18(7): 1527–1554.

25 刘建伟,刘媛,罗雄麟.深度学习研究进展.计算机应用与研究,2014,31(7):1921–1942.

26 尹宝才,王文通,王立春.深度学习研究综述.北京工业大学学报,2015,41(1):48–59.

27 王蕾,张宝昌.深度学习最新研究进展综述.中国科技论文在线,2015,8(6):510–517.