基于多隐层 Gibbs 采样的深度信念网络训练方法

史 科 [1] 陆 阳 [1,2] 刘广亮 [1] 毕 翔 [1,2] 王 辉 [1]

摘 要 深度信念网络 (Deep belief network, DBN) 作为一类非常重要的概率生成模型, 在多个领域都有着广泛的用途. 现有深度信念网的训练分为两个阶段, 首先是对受限玻尔兹曼机 (Restricted Boltzmann machine, RBM) 层自底向上逐层进行的贪婪预训练, 使得每层的重构误差最小, 这个阶段是无监督的; 随后再对整体的权值使用有监督的反向传播方法进行精调. 本文提出了一种新的 DBN 训练方法, 通过多隐层的 Gibbs 采样, 将局部 RBM 层组合, 并在原有的逐层预训练和整体精调之间进行额外的预训练, 有效地提高了 DBN 的精度. 本文同时比较了多种隐层的组合方式, 在 MNIST 和 ShapeSet 以及 Cifar10 数据集上的实验表明, 使用两两嵌套组合方式比传统的方法错误率更低. 新的训练方法可以在更少的神经元上获得比以往的训练方法更好的准确度, 有着更高的算法效率.

关键词 深度信念网络, 受限玻尔兹曼机, Gibbs 采样, 对比散度

引用格式 史科, 陆阳, 刘广亮, 毕翔, 王辉. 基于多隐层 Gibbs 采样的深度信念网络训练方法. 自动化学报, 2019, **45**(5): 975−984

DOI 10.16383/j.aas.c170669

A Deep Belief Networks Training Strategy Based on Multi-hidden Layer Gibbs Sampling

SHI Ke[1] LU Yang[1,2] LIU Guang-Liang[1] BI Xiang[1,2] WANG Hui[1]

Abstract Deep belief network (DBN) is a very important probabilistic generative model that can be used in many areas. The current training approach of DBN involves two phases. The first is a fully unsupervised pre-training process, which is a down-top and layer-by-layer one to train the restricted Boltzmann machine (RBM) layers, making the reconstruction error of each layer minimal. The second is a supervised stage which uses the back propagation to fine-tune the entire parameters of the model. In this paper, a new training strategy for DBN is proposed. Between the current two training phases, this paper introduces another training strategy to combine multiple local RBMs into an overall probability model for multi hidden layer Gibbs sampling, which effectively improves the accuracy of DBN. This paper has compared a variety of combinations of RBM layers, experiments on the MNIST, ShapeSet and Cifar10 dataset show that our method outperforms the existing training algorithms for DBN. The new algorithm can achieve better accuracy with fewer neurons, also achieves higher algorithm efficiency.

Key words Deep belief network (DBN), restricted Boltzmann machine (RBM), Gibbs sampling, contrastive divergence (CD)

Citation Shi Ke, Lu Yang, Liu Guang-Liang, Bi Xiang, Wang Hui. A deep belief networks training strategy based on multi-hidden layer Gibbs sampling. *Acta Automatica Sinica*, 2019, **45**(5): 975−984

在机器学习领域里, 最重要也是最困难的莫过于特征的提取, 抓住事物区分度强的特征也就抓住了事物的本质. 在此基础上, 分类器的性能会得到极大的提高. 但长期以来如何进行特征提取一直是个棘手的问题, 不同领域的数据涉及到不同的提取方法, 需要大量的领域知识作为支撑. 另一方面, 一直以来各种深度神经网络模型都困扰在如何找到有效的训练方法. 传统的反向传播算法在多隐层神经网络上存在着梯度消失的问题, 使得深度网络的性能甚至还不如浅层网络[1]. 这两个关键问题在 2006 年 Hinton 提出的文献 [2] 中得到了很大程度上的解决. 在文献 [2] 中提出的多层限制玻尔兹曼机 (Restrict Boltzmann machine, RBM) 堆叠降维的方法, 在无监督的情况下实现了自动化的特征学习, 实验表明

收稿日期 2017-11-22 录用日期 2018-03-24
Manuscript received November 22, 2017; accepted March 24, 2018
国家重点研发计划专项 (2016YFC0801804, 2016YFC0801405), 国家自然科学基金 (61572167) 资助
Supported by National Key Research and Development Program of China (2016YFC0801804, 2016YFC0801405), National Natural Science Foundation of China (61572167)
本文责任编委 王占山
Recommended by Associate Editor WANG Zhan-Shan
1. 合肥工业大学计算机与信息学院 合肥 230009 2. 安全关键工业测控技术教育部工程研究中心 合肥 230009
1. School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009 2. Engineering Research Center of Safety Critical Industry Measure and Control Technology, Ministry of Education, Hefei 230009

效果比传统的 PCA 方法要好得多. 在此基础上增加分类器就构成了深度信念网络模型 (Deep belief network, DBN). 作为一种生成模型, DBN 有着重要的研究价值. 相对于判别式模型, 生成模型可以反向生成研究对象的实例, 可以直观地观察出生成对象的各种特征, 为进一步的研究提供可能. 在随后的大量研究中, DBN 被广泛应用到了图像识别[3−4]、语音识别[5]、自然语言处理[6]、控制[7] 等多个领域, 并取得了很好的效果.

针对 DBN 训练方法的研究一直是一个热点[8−9]. Goh 等[10] 提出了一种有监督的预训练方法, 提高了 DBN 的精度. 李飞等[11] 从 Gibbs 采样的次数入手, 提出了动态的采样方法, 乔俊飞等[12] 将自适应学习率引入到对比散度 (Contrastive divergence, CD) 算法中, 提高了算法收敛速度. 典型的 DBN 的训练分为 2 个阶段[13], 分别是逐层预训练和整体精调. 在逐层预训练阶段, 从网络最底层的 RBM 开始, 自底向上逐层使用无监督的贪婪方法来使得每层 RBM 的损失误差最小. 然后在整体精调阶段使用有监督的学习方法, 针对有标签的数据使用梯度下降进行整体权值修正. 实验表明此种方法是有效的, 很好地解决了一直以来深度网络无法有效训练的难题. 逐层预训练将网络的权重调整到一个 “合适 ”的初始位置, 如果不进行逐层预训练而直接进行整体精调, 则网络很难收敛, 在逐层预训练的基础上进行整体精调可以确保网络能够收敛到很好的位置上. 在此基础上, 网络权重的初始位置有没有进一步改进的可能, 从而获得更好的网络性能呢? DBN 的逐层预训练是在堆叠着的每个 RBM 内进行多步 Gibbs 采样来逼近数据的真实分布的, 采样在 RBM 的可视层和隐藏层之间迭代进行. 本文在此基础上, 提出了一种两阶段的无监督预训练方法, 在已有预训练的基础上引入多隐层 Gibbs 采样预训练方法, 将多个 RBM 组合成一个整体概率模型进行预训练, 使得 Gibbs 采样在多个 RBM 中进行, 从而获得更 “合适 ”的网络权值初始位置. 在 MNIST、ShapeSet 和 Cifar10 数据集上的实验表明, 此种方法比传统的深度信念网络训练方法可以获得更好的分类效果, 在包含 (1 300, 1 300, 1 300, 1 300) 四层隐层的 DBN 上使用固定学习率的实验, 相对于传统方法的可以将 MNIST 的错误率从 1.25 % 降低到 1.09 %.

本文先介绍了受限玻尔兹曼机和深度信念网络模型, 然后提出了改进后的算法, 最后在 MNIST、ShapeSet 和 Cifar10 数据集上验证并讨论了实验结果.

1 受限玻尔兹曼机模型

DBN 的预训练是通过受限玻尔兹曼机的训练进行的, 所以我们先描述 RBM 模型. RBM 是一个无向图模型, 它可以被看做是一个二部图 (Bipartite graph), 两个部分分别是可视层 $\boldsymbol{v}$ 和隐层 $\boldsymbol{h}$, 层间结点全连接, 层内结点不连接, 如图 1 所示. 可视层接收数据输入, 两层间的连接权值用 W 表示, $W \in \mathbf{R}^{n\times m}$. 可视层的偏置用 $\boldsymbol{a}$ 表示, $\boldsymbol{a} \in \mathbf{R}^n$, 隐层的偏置用 $\boldsymbol{b}$ 表示, $\boldsymbol{b} \in \mathbf{R}^m$. RBM 的隐层可以理解为模型中尚未被观测到的部分, 可视层可以理解为可以观测到的部分, 它们的节点状态一般是二进制的, 取值 1 或 0.

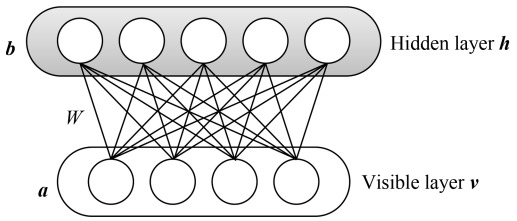


图 1 RBM 模型

Fig. 1 Restricted Boltzmann machine

RBM 是能量模型系统, 它通过能量来表示系统当前的状态, 能量定义为[2]:

$$E(\boldsymbol{v},\boldsymbol{h}|\theta) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i W_{ij} h_j \tag{1}$$

其中, n 表示可视层的节点数目, m 表示隐藏层节点数目, 就表示可视层 i 节点到隐层 j 节点的权值大小. 使用 $\theta = \{W, \boldsymbol{a}, \boldsymbol{b}\}$ 表示系统所有参数的集合.

给定了能量定义, 就可以在此基础上定义系统整体的概率分布[2]:

$$P(\boldsymbol{v},\boldsymbol{h}|\theta) = \frac{\mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}|\theta)}}{\sum\limits_{\boldsymbol{v},\boldsymbol{h}} \mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}|\theta)}} \tag{2}$$

其中, 分母部分称之为归一化因子或配分函数 (Partition function), 使得系统概率取值在 [0, 1] 范围内, 一般用 $Z(\theta) = \sum_{\boldsymbol{v},\boldsymbol{h}} \mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}|\theta)}$ 表示.

RBM 的结构决定了隐层和可见层是相互条件独立的, 于是可以得到条件概率分布[14]:

$$P(\boldsymbol{h}|\boldsymbol{v},\theta) = \prod_{j=0}^{m} \frac{1}{\left(1+\exp\left(-W_j^{\mathrm{T}}\boldsymbol{v} - b_j\right)\right)} \tag{3}$$

以及

$$P(\boldsymbol{v}|\boldsymbol{h},\theta) = \prod_{i=0}^{n} \frac{1}{\left(1+\exp\left(-W_i\boldsymbol{h} - a_i\right)\right)} \tag{4}$$

RBM 的训练就是依据训练样本来估计模型的参数, 使得此模型的推断数据尽可能地接近真实数据. 由式 (2) 可得边缘概率分布, 通过使用最大似然的方法来估计, 对其对数求导数有[1]:

$$\frac{\partial \log P(\boldsymbol{v}|\theta)}{\partial \theta} = \frac{\partial \log \sum_{\boldsymbol{h}} \mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}|\theta)}}{\partial \theta} - \frac{\partial \log \sum_{\boldsymbol{v},\boldsymbol{h}} \mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}|\theta)}}{\partial \theta} = -\sum_{\boldsymbol{h}} P(\boldsymbol{h}|\boldsymbol{v},\theta)\frac{\partial E(\boldsymbol{v},\boldsymbol{h}|\theta)}{\partial \theta} + \sum_{\boldsymbol{v},\boldsymbol{h}} P(\boldsymbol{v},\boldsymbol{h}|\theta)\frac{\partial E(\boldsymbol{v},\boldsymbol{h}|\theta)}{\partial \theta} \tag{5}$$

对于训练样本, 使用 "data"表示分布 $P(\boldsymbol{h}|\boldsymbol{v},\theta)$, 使用 "model"表示分布 $P(\boldsymbol{v},\boldsymbol{h}|\theta)$. 其中, 使用 $\langle\cdot\rangle_p$ 表示关于分布 p 的数学期望. 因为 $\theta = \{W, \boldsymbol{a}, \boldsymbol{b}\}$, 联合式 (1) 的导数, 式 (5) 可表示为[1]:

$$\frac{\partial \log P(\boldsymbol{v}|\theta)}{\partial W_{ij}} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}} \tag{6}$$

$$\frac{\partial \log P(\boldsymbol{v}|\theta)}{\partial a_i} = \langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{model}} \tag{7}$$

$$\frac{\partial \log P(\boldsymbol{v}|\theta)}{\partial b_j} = \langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{model}} \tag{8}$$

期望 $\langle\cdot\rangle_{\text{model}}$ 没有有效的解析解计算方法, 目前主要的做法是通过对于上述模型使用式 (3) 和式 (4) 进行交替 Gibbs 采样[15], 来近似逼近对数似然概率. K 次交替 Gibbs 采样, 具体来说就是先将用样本赋值给 $\boldsymbol{v}^0$, 然后使用式 (3) 和式 (4) 交替进行采样, $\boldsymbol{v}^0 \sim \hat{P}(\boldsymbol{v})$, $\boldsymbol{h}^0 \sim P(\boldsymbol{h}|\boldsymbol{v}^0,\theta)$, $\boldsymbol{v}^1 \sim P(\boldsymbol{v}|\boldsymbol{h}^0,\theta)$, $\boldsymbol{h}^1 \sim P(\boldsymbol{h}|\boldsymbol{v}^1,\theta)$, $\cdots$, $\boldsymbol{v}^{k+1} \sim P(\boldsymbol{v}|\boldsymbol{h}^k,\theta)$, 上标表示采样顺序, $\hat{P}(\boldsymbol{v})$ 表示训练集的分布. 通过式 (6) ~ (8) 获得 RBM 参数的梯度来更新 W, $\boldsymbol{a}$, $\boldsymbol{b}$.

$$\begin{cases} \Delta W_{ij} = \varepsilon\left(v_i^0 h_j^0 - v_i^k h_j^k\right) \\ \Delta a_i = \varepsilon\left(v_i^0 - v_i^k\right) \\ \Delta b_j = \varepsilon\left(h_j^0 - h_j^k\right) \end{cases} \tag{9}$$

其中, ε 为学习率.

2　深度信念网络

多个玻尔兹曼机堆叠后, 就形成了深度信念网络[13]. 通常可以在最上层再增加一层逻辑回归 (Logistic regression) 层来作为有监督学习分类器. DBN 模型示意图见图 2.

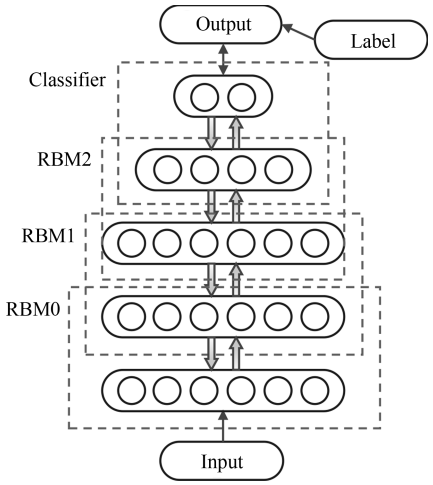


图 2　DBN 模型

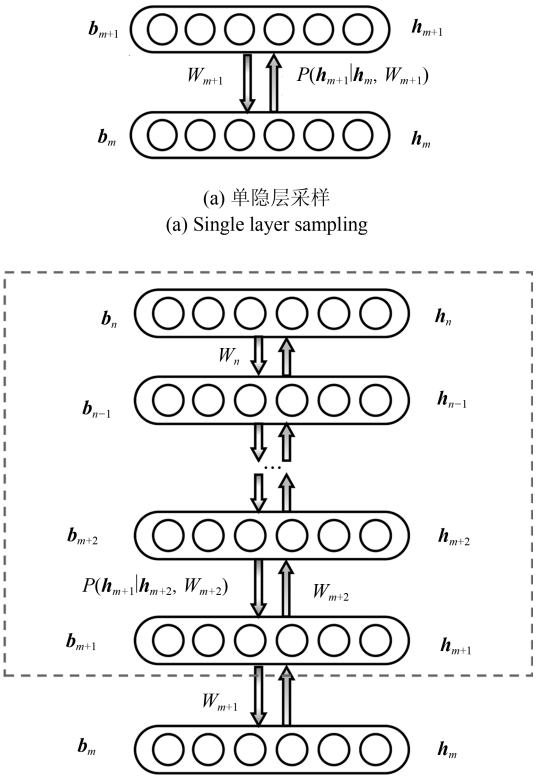
Fig. 2　Deep belief networks

Hinton 在文献 [13] 中提出了深度信念网络的训练方法, 分为逐层预训练和整体精调两个阶段. 在逐层预训练阶段, 从网络最底层的 RBM 开始, 自底向上逐层使用无监督的贪婪方法来使得每层 RBM 的损失误差最小. 在此过程中相邻的 RBM 两两连接, 下层 RBM 的输出传递到上一层 RBM 作为输入, 最底层的输入为训练数据, 最顶层的输出传递给分类器. 在整体精调阶段使用有监督的学习方法, 将所有隐层的权值看做一个整体, 使用梯度下降的方法针对有标签的数据进行权值修正. 第一阶段的学习过程提高了在构造模型下训练数据的似然概率的变分下限, 是无监督的, 不需要标签信息. 如果不进行第一阶段的逐层预训练, 直接使用随机初始化的参数直接进行梯度下降法则很容易导致训练失败, 模型容易陷入局部极值点[1]. 通过 RBM 的逐层训练, 深度网络每层的参数都已处于一个比较好的位置上, 在此前提下进行全局性的梯度下降可以精调整个模型的精度, 获得更好的结果.

3　多隐层 Gibbs 采样预训练

作为生成模型的 DBN, 在向下的生成方向上, 不仅是最底层的可视层, 整个网络的每一层都是为了使得重构数据的分布和真实数据的分布尽可能地接近. 如图 3 所示, $\boldsymbol{h}_m, \cdots, \boldsymbol{h}_n$ 表示 DBN 的隐层. 传统的逐层训练算法, 针对 $\boldsymbol{h}_{m+1}, \boldsymbol{h}_m$ 组合的 RBM, $\boldsymbol{h}_m$ 是其可视层, $\boldsymbol{h}_{m+1}$ 是其隐层, Gibbs 采样是在这两层之间迭代的, 使得此 RBM 参数 W_{m+1} 收敛, 以更好地接近 $\boldsymbol{h}_m$ 层输入的分布. 采样先从 $\boldsymbol{h}_m$ 层的输入开始向上构建 $\boldsymbol{h}_{m+1}$ 层, 概率为 $P(\boldsymbol{h}_{m+1}|\boldsymbol{h}_m, W_{m+1})$, 再在此基础上反向重构出 $\boldsymbol{h}_m$ 层的数据, 概率为 $P(\boldsymbol{h}_m|\boldsymbol{h}_{m+1}, W_{m+1})$. Salakhutdinov 等在文献 [16] 中指出, 在 DBN 中

如果每层 RBM 都被正确地初始化 (可以通过逐层预训练保证), 则反向的 $P(\boldsymbol{h}_{m+1}|\boldsymbol{h}_{m+2}, W_{m+2})$ 是比 $P(\mathbf{h}_{m+1}|\boldsymbol{h}_m, W_{m+1})$ 更好的 $\boldsymbol{h}_{m+1}$ 上的后验分布. 反向的概率 $P(\boldsymbol{h}_{m+1}|\boldsymbol{h}_{m+2}, W_{m+2})$ 是由高层多个隐层计算得到, 包含了比低层更抽象更丰富的信息. 在此基础上我们使用 $P(\boldsymbol{h}_{m+1}|\boldsymbol{h}_{m+2}, W_{m+2})$ 来代替 $P(\boldsymbol{h}_{m+1}|\boldsymbol{h}_m, W_{m+1})$ 以获得更好的近似.



(a) 单隐层采样
(a) Single layer sampling

(b) 多隐层采样
(b) Multi-layer sampling

图 3 针对 $\boldsymbol{h}_{m+1}$ 的采样
Fig. 3 Sampling for $\boldsymbol{h}_{m+1}$

于是针对 $\boldsymbol{h}_{m+1}, \boldsymbol{h}_m$ 组合的 RBM, $\boldsymbol{h}_m$ 层的对数似然梯度为

$$\frac{\partial \log P(\boldsymbol{h}_m|\theta)}{\partial \theta} = -\sum_{\boldsymbol{h}_{m+1}} P(\boldsymbol{h}_{m+1}|\boldsymbol{h}_m, \theta)\frac{\partial E(\boldsymbol{h}_m, \boldsymbol{h}_{m+1}|\theta)}{\partial \theta} + \sum_{\boldsymbol{h}_m, \boldsymbol{h}_{m+1}} P(\boldsymbol{h}_m, \boldsymbol{h}_{m+1}|\theta)\frac{\partial E(\boldsymbol{h}_m, \boldsymbol{h}_{m+1}|\theta)}{\partial \theta} \tag{10}$$

考虑到

$$P(\boldsymbol{h}_m, \boldsymbol{h}_{m+1}|W_{m+1}) = P(\boldsymbol{h}_m|W_{m+1})\, P(\boldsymbol{h}_{m+1}|\boldsymbol{h}_m, W_{m+1}) \approx P(\boldsymbol{h}_m|W_{m+1})\, P(\boldsymbol{h}_{m+1}|\boldsymbol{h}_{m+2}, W_{m+2}) \tag{11}$$

以及能量对于 W_{m+1} 的梯度, 所以式 (10) 的第二部分为

$$\begin{aligned}&\sum_{\boldsymbol{h}_m, \boldsymbol{h}_{m+1}} P(\boldsymbol{h}_m, \boldsymbol{h}_{m+1}|W_{m+1}) \times \frac{\partial E(\boldsymbol{h}_m, \boldsymbol{h}_{m+1}|W_{m+1})}{\partial W_{m+1,i,j}} \approx \\ &-\sum_{\boldsymbol{h}_m, \boldsymbol{h}_{m+1}} P(\boldsymbol{h}_m|W_{m+1})\, P(\boldsymbol{h}_{m+1}|\boldsymbol{h}_{m+2}, W_{m+2})\, h_{m,i}h_{m+1,j} = \\ &-\sum_{\boldsymbol{h}_m} P(\boldsymbol{h}_m|W_{m+1})\, P(h_{m+1,j}=1|\boldsymbol{h}_{m+2}, W_{m+2})\, h_{m,i} = \\ &-\langle \boldsymbol{h}_m|W_{m+1}\rangle\, P(h_{m+1,j}=1|\boldsymbol{h}_{m+2}, W_{m+2})\end{aligned} \tag{12}$$

同理, 式 (10) 的第一部分为

$$\begin{aligned}&-\sum_{\boldsymbol{h}_{m+1}} P(\boldsymbol{h}_{m+1}|\boldsymbol{h}_m, W_{m+1}) \times \frac{\partial E(\boldsymbol{h}_m, \boldsymbol{h}_{m+1}|W_{m+1})}{\partial W_{m+1,i,j}} = \\ &\sum_{\mathbf{h}_{m+1}} P(\boldsymbol{h}_{m+1}|\boldsymbol{h}_m, W_{m+1}) h_{m+1,j}h_{m,i} = \\ &P(h_{m+1,j}=1|\boldsymbol{h}_m, W_{m+1})\, h_{m,i}\end{aligned} \tag{13}$$

最终可得对于 $\boldsymbol{W}_{m+1}$ 的对数似然梯度

$$\begin{aligned}&\frac{\partial \log P(\boldsymbol{h}_m|W_{m+1})}{\partial W_{m+1,i,j}} = \\ &P(h_{m+1,j}=1|\boldsymbol{h}_m, W_{m+1})\, h_{m,i} - \\ &\langle \boldsymbol{h}_m|W_{m+1}\rangle\, P(h_{m+1,j}=1|\boldsymbol{h}_{m+2}, W_{m+2})\end{aligned} \tag{14}$$

条件概率为

$$P(\boldsymbol{h}_{m+1}|\boldsymbol{h}_m, W_{m+1}) = \prod_{j=0}^{|\boldsymbol{h}_{m+1}|} P(h_{m+1,j}|\boldsymbol{h}_m, W_{m+1}) = \prod_{j=0}^{|\boldsymbol{h}_{m+1}|} \frac{1}{\left(1+\exp\left(-W_{m+1,j}^{\mathrm{T}}\boldsymbol{h}_m - b_{m+1,j}\right)\right)} \tag{15}$$

$$P\left(\boldsymbol{h}_{m+1}|\boldsymbol{h}_{m+2},W_{m+2}\right)=\prod_{i=0}^{|\boldsymbol{h}_{m+1}|}P\left(h_{m+1,i}|\boldsymbol{h}_{m+2},W_{m+2}\right)=\prod_{i=0}^{|\boldsymbol{h}_{m+1}|}\frac{1}{\left(1+\exp\left(-\boldsymbol{h}_{m+2}W_{m+2,i}-b_{m+1,i}\right)\right)} \tag{16}$$

对于 b_m 和 b_{m+1} 的梯度可以通过类似的方法推导, 使用之前的记号, 可以得到类似式 (6) ∼ (8) 的结论.

$$\frac{\partial\log P\left(\boldsymbol{h}_m|\theta\right)}{\partial W_{m+1,i,j}}=\left\langle h_{m,i}h_{m+1,j}\right\rangle_{\text{data}}-\left\langle h_{m,i}\tilde{h}_{m+1,j}\right\rangle_{\text{model}} \tag{17}$$

$$\frac{\partial\log P\left(\boldsymbol{h}_m|\theta\right)}{\partial b_{m,i}}=\left\langle h_{m,i}\right\rangle_{\text{data}}-\left\langle\tilde{h}_{m,i}\right\rangle_{\text{model}} \tag{18}$$

$$\frac{\partial\log P\left(\boldsymbol{h}_m|\theta\right)}{\partial b_{m+1,j}}=\left\langle h_{m+1,j}\right\rangle_{\text{data}}-\left\langle\tilde{h}_{m+1,j}\right\rangle_{\text{model}} \tag{19}$$

其中, $\tilde{h}_{m,j}$ 和 $\tilde{h}_{m+1,j}$ 表示从高层反向生成的 $\boldsymbol{h}_m$ 和 $\boldsymbol{h}_{m+1}$ 层单元的估计值, 且比传统的自下而上生成的 $h_{m,j}$ 和 $h_{m+1,j}$ 要更精确. $\tilde{h}_{m,j}$ 和 $\tilde{h}_{m+1,j}$ 可以通过在高层多个隐层中使用的 Gibbs 采样来进行估计.

Gibbs 采样是一种马尔科夫蒙特卡罗 (Markov chain Monte Carlo, MCMC) 方法, 可以利用已有数据来推断丢失的数据. 对于从 $\boldsymbol{h}_m$ 到 $\boldsymbol{h}_n$ 的多隐层 Gibbs 采样是在 $\boldsymbol{h}_m,\boldsymbol{h}_{m+1},\cdots,\boldsymbol{h}_{n-1},\boldsymbol{h}_n$ 这 $n-m+1$ 个隐层上进行的. 信号先从 $\boldsymbol{h}_m$ 层开始向上使用式 (15) 传播, 到达最上层 $\boldsymbol{h}_n$ 层后开始使用 (16) 反向传播, 等到信号回退到 $\boldsymbol{h}_{m+1}$ 层后使用采样值来估计 $\tilde{\boldsymbol{h}}_{m+1}$, 随后向下采样估计 $\tilde{\boldsymbol{h}}_m$, 再使用式 (17) ∼ (19) 来更新网络参数. 注意到我们仅更新 $\boldsymbol{h}_{m+1},\boldsymbol{h}_m$ 层相关权重, 而将高层权重固定, 以保持模型的稳定, 所以本质上还是一种逐层训练方法. 在迭代时要注意从底向上逐层进行. 如前两节所述, 现有的 DBN 预训练是针对每层的 RBM 使用 Gibbs 采样来逼近模型的真实分布, 使得每层 RBM 和其真实分布的差异减小, 但 DBN 的推断过程是将所有的 RBM 层看做一个整体进行的, 通过引入多隐层 Gibbs 采样可以在逐层逼近的基础上进一步在局部模型上逼近真实分布.

多隐层的选择和组合方式有多种可能, 本文通过实验讨论了以下 4 种类型的组合方式: 两两不嵌套组合 (Non-nested)、两两嵌套组合 (Nested)、增量不嵌套组合 (Incremental non-nested) 和增量嵌套组合 (Incremental nested). 以 4 隐层的 DBN 举例, 假设 4 个隐层分别为 $\boldsymbol{h}_0$, $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, $\boldsymbol{h}_3$, 那么两两不嵌套组合的 RBM 序列为 ($\boldsymbol{h}_0$,$\boldsymbol{h}_1$), ($\boldsymbol{h}_2$,$\boldsymbol{h}_3$); 两两嵌套组合的 RBM 为 ($\boldsymbol{h}_0$,$\boldsymbol{h}_1$), ($\boldsymbol{h}_1$,$\boldsymbol{h}_2$), ($\boldsymbol{h}_2$,$\boldsymbol{h}_3$); 不嵌套增量组合的序列为 ($\boldsymbol{h}_0$, $\boldsymbol{h}_1$), ($\boldsymbol{h}_2$, $\boldsymbol{h}_3$), ($\boldsymbol{h}_0$, $\boldsymbol{h}_1$, $\boldsymbol{h}_2$), ($\boldsymbol{h}_0$, $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, $\boldsymbol{h}_3$); 嵌套增量组合的序列为 ($\boldsymbol{h}_0$,$\boldsymbol{h}_1$), ($\boldsymbol{h}_1$,$\boldsymbol{h}_2$), ($\boldsymbol{h}_2$,$\boldsymbol{h}_3$), ($\boldsymbol{h}_0$,$\boldsymbol{h}_1$,$\boldsymbol{h}_2$), ($\boldsymbol{h}_1$,$\boldsymbol{h}_2$,$\boldsymbol{h}_3$), ($\boldsymbol{h}_0$, $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, $\boldsymbol{h}_3$).

综上所述, 基于多隐层 Gibbs 采样的 DBN 模型算法整体描述如下:

步骤 1. 无监督的逐层预训练. 对于 DBN 中的 RBM 层进行逐层贪婪预训练. 令 $\boldsymbol{X}$ 为最底层 RBM 的输入. 自底向上, 对于第 i 层 RBM, 计算隐层节点概率并交替采样, 具体如下.

步骤 1.1. 进行 K 次 Gibbs 采样. 使用式 (3) 计算概率分布, 然后从分布中抽取 $h_{i+1,j}^k\in\{0,1\}$, 再使用式 (4) 计算并从中抽取 $h_{i,j}^{k+1}\in\{0,1\}$.

步骤 1.2. 以下式和 (9) 来更新参数

$$\begin{cases}W_{ij}=W_{ij}-\Delta W_{ij}\\a_i=a_i-\Delta a_i\\b_j=b_j-\Delta b_j\end{cases} \tag{20}$$

步骤 2. 无监督的多隐层预训练. 自底向上组合多隐层, 对于每个 RBM 组合, 执行以下步骤进行多隐层预训练.

步骤 2.1. 依据式 (15) 向上计算每层概率分布并抽取出 $h_{i,j}\in\{0,1\}$, 直到顶层.

步骤 2.2. 依据式 (16) 计算反向概率, 每计算一层同时抽取 $\tilde{h}_{m,j}$, 直到底层.

步骤 2.3. 使用式 (17) ∼ (19) 计算梯度, 并更新权重.

步骤 3. 有监督的全局精调. 步骤如下.

步骤 3.1. 对于所有的 RBM 层, 自底向上传递信号. 第 i 层的输出作为第 $i+1$ 层的输入.

步骤 3.2. 将最上层 RBM 的输出和样本标签 $\boldsymbol{Y}$ 传递给分类器, 使用梯度递减更新所有的参数.

4 实验与分析

实验部分使用 MNIST 手写数字数据集、合成的数据集 ShapeSet 以及真实物体图像数据集 Cifar10 来测试本文的模型. MNIST 数据集包含 70 000 张人类手写数字的图片, 每张图片包含一个 0 ∼ 9 的手写数字, 被分割成 28 × 28 的黑白两色点阵. 数据集分为两部分, 一部分是包含 60 000 张图片的测试用数据, 一部分是剩下的 10 000 张用于测试. 每张图片都有对应的标签数据, 表明正确的数字是什么. MNIST 数据集是一个广泛使用的评估机器学习算法的数据库, 其中包含的手写数字信息

来自于不同的书写方式, 且数据集没有经过任何拉伸转换等几何上的处理. 在本文实验中, 也没有进行任何额外的预处理, 相当于没有任何领域知识的介入. ShapeSet 是一个人工生成的数据集, 每个样本可以包含任意多个平行四边形、三角形或圆形的图像, 图像之间可以互相叠加遮挡, 且有任意的前景和背景色. 在本文的实验中, 设置每个样本的大小为 32×32, 限制样本中出现的图形数为 1 或 2, 两个图形之间的遮挡率为不超过 50 %. Cifar10 数据集包含 60 000 张 32×32 大小, 有 RGB 三原色信息的彩色图片, 共有 10 类物体, 每个类别 6 000 张.

在 DBN 的最上层, 增加了一层逻辑回归层来预测类别, 使用 Softmax 激活函数, 用预测值和真实类别值之间的负对数似然函数来计算损失. 通常情况下动态的学习率会取得更好的结果, 学习率一般随着训练次数的增加而逐渐减小, 以防止模型错过最小值. 在本文的实验中, 目的是验证新的算法相对传统算法的有效性, 没有去讨论模型在实验数据集上所能达到的最优结果, 所以使用了常数的学习率. 无论是 DBN 还是改进后的算法, 在训练的第一阶段, 也就是逐层训练时使用的学习率都是 0.01, 在最后一个阶段整体精调时使用的学习率是 0.1, 改进后的算法的第二阶段使用 0.01 的学习率. 在所有实验的整体精调阶段和本文提出的算法的第二阶段, 都使用了 "早停" (Early stop) 的技术, 来防止模型过拟合. 为了加速算法, 本文使用了小批量 (Mini-batch) 的方法来把数据批量提交给 GPU 计算. 文献 [17] 中, Vincent 等给出了 Mini-batch 的数量设置建议, 通常情况下每个小 Batch 包含的样例数目应等于类别的数量, 在本文的实验中设置为 10. 逐层训练阶段循环 Epoch 数设置为 100, 整体精调阶段设置为 1 000, 改进的算法的第二阶段设置为 100.

本文使用 Python ver. 3.5.2 语言在 Theano ver. 0.8 库的基础上实现了基本的 DBN 算法以及提出的改进算法. 在一台 Xeno E3-1230V3, 8 GB 内存, Ubuntu16.10 64 位的系统上, 通过 GeForce GTX1070 GPU 加速来运行实验程序.

4.1 4 隐层, 不同 RBM 嵌套组合方式

在本组实验中, 使用了 $784 \times N \times N \times N \times N \times 10$ 的网络结构. 包含了 4 层相同节点数的隐层. N 的取值从 100 递增到 3 000. 针对不同的方法, 在 MNIST 数据集上做了 5 组实验, 实验结果如图 4.

可以看到对于 4 隐层 $\boldsymbol{h}_0$, $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, $\boldsymbol{h}_3$ 的深度信念网络, 使用两两嵌套组合, ($\boldsymbol{h}_0$, $\boldsymbol{h}_1$), ($\boldsymbol{h}_1$, $\boldsymbol{h}_2$), ($\boldsymbol{h}_2$, $\boldsymbol{h}_3$) 方式训练的错误率最低, 无论是全局最低值还是整体平均值. 在 $1\,300 \times 1\,300 \times 1\,300 \times 1\,300$ 隐层的结构下, 达到最好的错误率 1.09 %, 比传统的 DBN 在同样结构时的 1.25 % 要降低 0.16 %. 在整组实验中使用两两嵌套组合方式的错误率普遍要好于其他方式. 当隐层节点数逐渐增加到大于 200 以后时, 两两嵌套组合的方法要普遍好于传统方法的 DBN. 两两嵌套组合方式最小错误率 1.09 % 比传统方法的最好结果 1.15 % (1 500 隐层结点时) 要低 0.06 %, 且在更少的隐层节点下取得, 这表明两两嵌套组合方式能够比传统方法更早更好地找到数据的特征. 同时因为是深层层间全连接网络, 1 300 隐层节点的网络要比 1 500 节点的网络少大约 1/4 的层间参数, 相应的计算量要少得多, 分类的速度会更快.

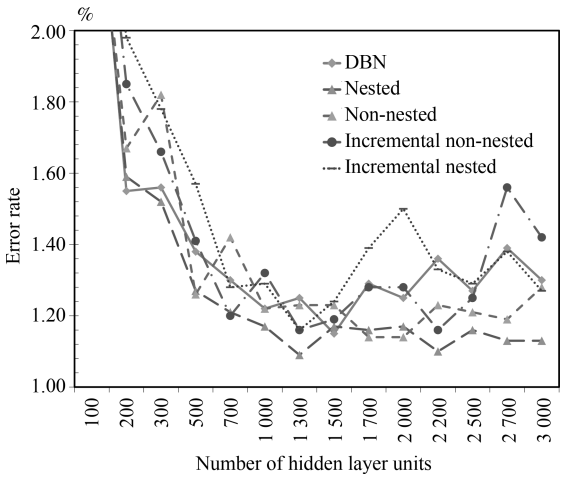


图 4 MNIST 数据集上 4 隐层模型错误率对比

Fig. 4 The error rate of 4 hidden layers model on MNIST

不嵌套组合的方法在隐层节点低于 1 500 时和传统的 DBN 接近, 大于 1 500 时比 DBN 要好, 但普遍比两两嵌套组合方式要差.

增量不嵌套和增量嵌套的组合方式表现出了较大的波动性, 错误率围绕传统 DBN 上下摆动. 相对于不递增的组合方式, 它们对数据进行了更多轮的学习, 也消耗了更多的运算时间, 出现这样现象的原因可能是因为组合了超过 3 层的隐层, 从而导致出现了梯度消失或激增的情况, 导致了网络性能的不稳定[18].

4.2 3 隐层, 2RBM 组合交叉实验

为了进一步验证模型有效性, 在本组实验中, 改变了网络的深度, 使用了 $784 \times N \times N \times N \times 10$ 的网络结构, 隐层为三层. 同样的, 为了方便考察算法性能, 设置了同样的节点数, 都为 N. N 的取值设定为从 100 到 4 000. 如果对于 3 隐层从底向上编号为 $\boldsymbol{h}_0$, $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, 增量嵌套的训练序列是 ($\boldsymbol{h}_0, \boldsymbol{h}_1$), ($\boldsymbol{h}_1, \boldsymbol{h}_2$), ($\boldsymbol{h}_0, \boldsymbol{h}_1, \boldsymbol{h}_2$), 嵌套组合的训练序列是 ($\boldsymbol{h}_0, \boldsymbol{h}_1$), ($\boldsymbol{h}_1, \boldsymbol{h}_2$). 最终的 MNIST 数据集上实验

结果如图 5.

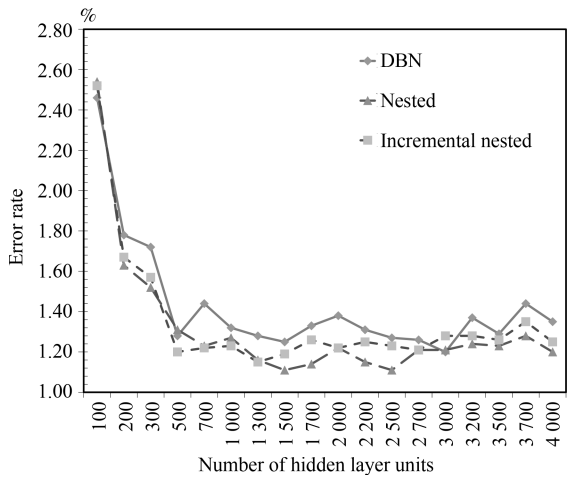


图 5　MNIST 数据集上 3 隐层模型错误率对比

Fig. 5　The error rate of 3 hidden layers model on MNIST

最好的错误率同时出现在两两嵌套组合算法隐层为 1 500 节点和 2 500 节点时, 都为 1.11 %, 对应的传统的 DBN 算法错误率为 1.25 % 和 1.27 %, 分别降低了 0.14 % 和 0.16 %. 在其他节点数的情况下, 从 200 开始改进后的算法错误率都要普遍优于传统算法. 相对于 4 隐层的结果, 3 隐层下增量嵌套组合的稳定性要更好, 虽然不如嵌套组合的效果, 但也普遍优于传统算法.

4.3　ShapeSet 数据集和 Cifar10 数据集

为了进一步验证算法的有效性, 在 ShapeSet 和 Cifar10 数据集上针对两两嵌套组合算法和传统的 DBN 算法再次做了比较. 实验结果如图 6 和图 7. ShapeSet 数据集上错误率普遍比传统方法低 2 个百分点. Cifar10 数据集上从 1 000 结点规模后普遍比传统方法要低 3 个百分点. 类似的结论再次验证本文提出的算法相对于传统 DBN 算法的有效性, 两两嵌套组合算法在各种隐层节点数量的模型上普遍获得了更低的错误率.

4.4　其他方法的比较

上述实验是在传统的 DBN 方法的基础上增加一轮无监督的组合训练得到的. 传统的 DBN 在逐层预训练阶段使用的是基于对比散度 (CD) 的采样方法, 实验表明 CD-1, 也就是 Gibbs 链迭代 1 次后的采样效果就已经很好了. Tieleman 等[19−20] 在传统的 DBN 上提出了一种改进的对比散度方法, 称之为 Persistent contrastive divergence (PCD) 算法. 实验表明 PCD 要优于传统的基于 CD-1 采样的 DBN

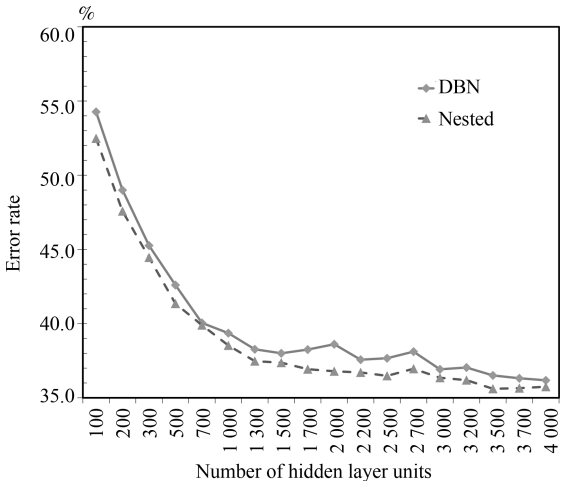


图 6　ShapeSet 数据集上 3 隐层模型错误率对比

Fig. 6　The error rate of 3 hidden layers model on ShapeSet

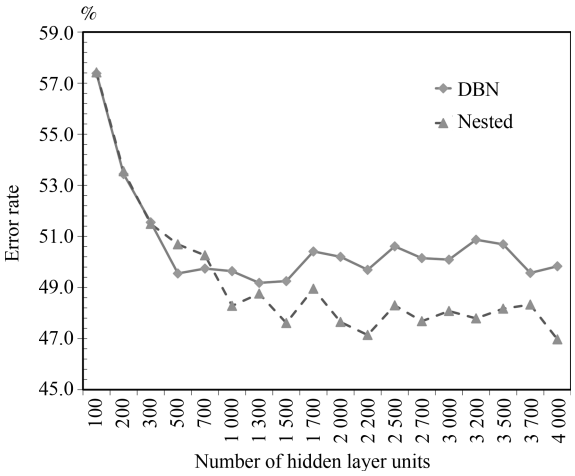


图 7　Cifar10 数据集上 3 隐层模型错误率对比

Fig. 7　The error rate of 3 hidden layers model on Cifar10

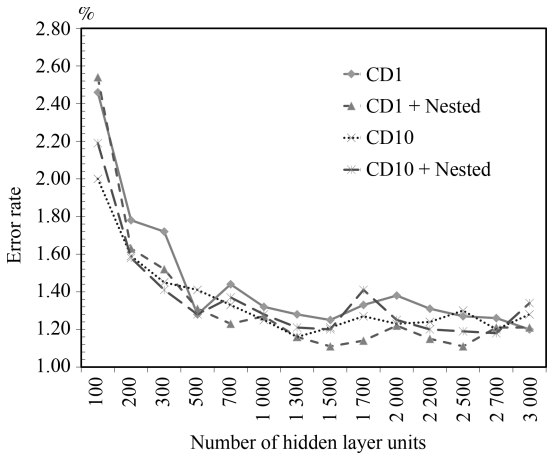


图 8　3 隐层模型 CD1、CD10 错误率对比

Fig. 8　The error rate comparison with CD1 and CD10 on 3 hidden layers model

算法, 和 10 次交替采样的 CD-10 接近. 在本文之前实验中的对比算法就是使用 CD-1 的 DBN 算法为基准. 在 CD-1 的基础上增加组合训练可以改进模型的精度, 那么在使用 PCD 或 CD-10 来逐层预训练的基础上能否进一步的改进模型精度呢? 在 MNIST 数据集 3 隐层的模型上 CD-1 和 CD-10 的对比实验结果见图 8, 4 隐层上的 CD-1 和 CD-10 以及 PCD 的对比结果见图 9 和图 10.

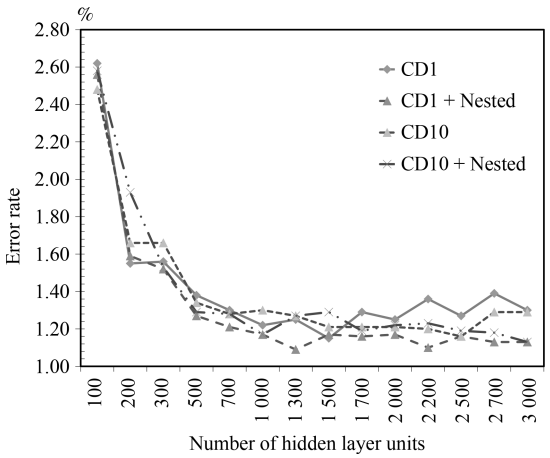


图 9 4 隐层模型 CD1、CD10 错误率对比

Fig. 9 The error rate comparison with CD1 and CD10 on 4 hidden layers model

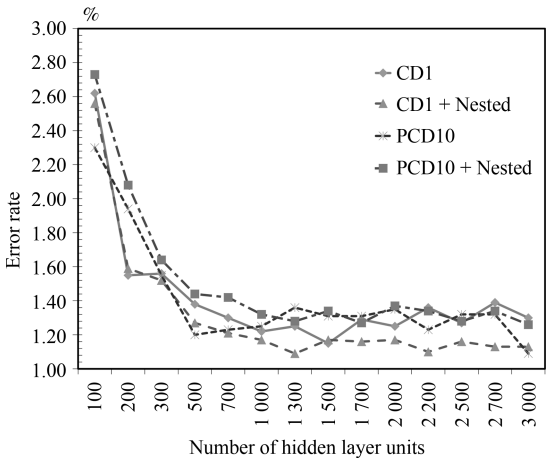


图 10 4 隐层模型 CD1、PCD 错误率对比

Fig. 10 The error rate comparison with CD1 and PCD on 4 hidden layers model

可以看到, 在 3 隐层的网络上, CD-10 的基础上再次进行 RBM 嵌套组合预训练并不能显著提高模型精度. 最好的结果仍然是在 CD-1+ 嵌套组合预训练的情况下.

在 4 隐层的网络上的结论类似, 最好的结果还是在 CD-1 的基础上进行嵌套组合预训练. CD-10 和 PCD-10 的情况下, 模型错误率围绕 CD-1 波动, 在 CD-10 或 PCD 的基础上增加一轮组合预训练并不能显著地提高系统的精度.

4.5 时间消耗和算法效率

本文在新增的多隐层 Gibbs 采样预训练中使用了 “早停 ”机制, 在训练中一旦检测到模型的代价值增加就会提前终止训练. 实际中在大部分的情况下只需额外训练很少的轮数就会满足终止条件, 实际消耗的时间非常少. 几种算法的实际训练时间对比见图 11. 可以看到本文的算法相对于 CD-1 时间略微增加, 远少于 CD-10 和 PCD 算法.

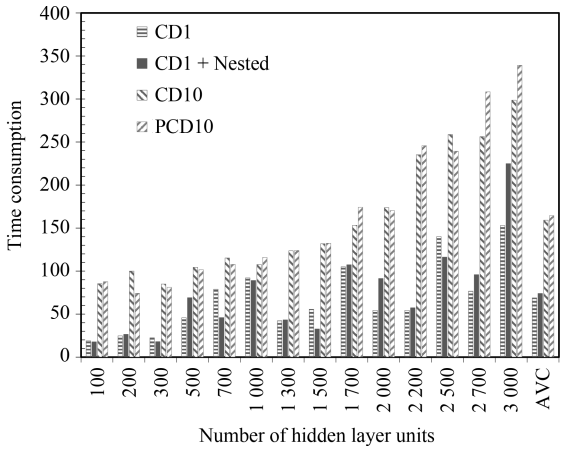


图 11 4 隐层模型上各种算法训练耗时对比

Fig. 11 The training time consumption comparison on 4 hidden layers model

上述实验表明, 本文方法能够在更小模型规模上实现比传统 DBN 更好的分类效果. 为了量化比较, 使用算法效率 (Algorithm efficiency, AE) 来度量识别速度、错误率和模型规模之间的关系. AE 定义为负的算法识别时间和错误率的乘积:

$$AE = -time \times error \tag{21}$$

在 4 隐层上的算法效率对比见图 12. 可以看出本文方法相比传统的方法有着更高的算法效率.

5 总结和展望

理论分析和实验表明在传统的 DBN 训练方法的基础上, 增加一轮基于多隐层的 Gibbs 采样无监督预训练, 对于提高深度信念网络的精度是有效的, 可以为进一步的有监督全局精调提供更好的初始化. 对比多种隐层的组合方式, 本文发现两两嵌套组合相邻的 RBM 进行训练的效果最好. 此种训练方法在原有无监督逐层训练的基础上进一步地提高了模型训练数据似然概率的变分下限, 相对于传统的使用 CD 或 PCD 的两阶段训练方法可以将错误率进一步降低, 同时也有着更高的算法效率.

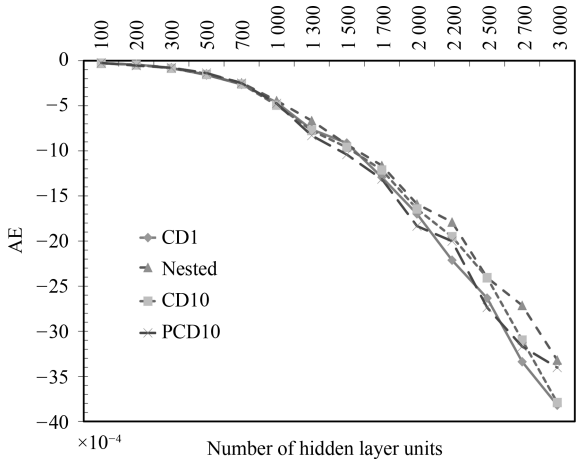


图 12　4 隐层模型上各种算法效率对比

Fig. 12　AE comparison on 4 hidden layers model

无监督的预训练不需要样本标签, 堆叠基本组件逐层预训练也是众多深度学习模型[17, 21−22] 的一种通用的学习框架. 现有的深度网络还有以其他组件为基本元素组合而成的, 如深度降噪自编码网络[23], 其使用自动编码器来代替限制玻尔兹曼机, 组合基本组件混合训练的思想在理论上也可以推广到这些结构上, 是否有效也还有待进一步的实验证明.

References

1 Bengio Y. Learning deep architectures for AI. *Foundations & Trends in Machine Learning*, 2009, **2**(1): 1−127

2 Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks. *Science*, 2006, **313**(5786): 504−507

3 Lee H, Grosse R, Ranganath R, Ng A Y. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, 2011, **54**(10): 95−103

4 Goh H, Thome N, Cord M, Lim J H. Learning deep hierarchical visual feature coding. *IEEE Transactions on Neural Networks and Learning Systems*, 2014, **25**(12): 2212−2225

5 Mohamed A R, Dahl G E, Hinton G. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 2012, **20**(1): 14−22

6 Sarikaya R, Hinton G E, Deoras A. Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2014, **22**(4): 778−784

7 Duan Yan-Jie, Lv Yi-Sheng, Zhang Jie, Zhao Xue-Liang, Wang Fei-Yue. Deep learning for control: the state of the art and prospects. *Acta Automatica Sinica*, 2016, 42(5): 643−654
(段艳杰, 吕宜生, 张杰, 赵学亮, 王飞跃. 深度学习在控制领域的研究现状与展望. 自动化学报, 2016, **42**(5): 643−654)

8 Wu F, Wang Z H, Lu W M, Li X, Yang Y, Luo J B, et al. Regularized deep belief network for image attribute detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017, **27**(7): 1464−1477

9 Wang B Y, Klabjan D. Regularization for unsupervised deep neural nets. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence. San Francisco, CA, USA: AAAI, 2017. 2681−2687

10 Goh H, Thome N, Cord M, Lim J H. Top-down regularization of deep belief networks. In: Proceedings of the 26th International Conference on Neural Information Processing Systems. Lake Tahoe, Nevada, USA: ACM, 2013. 1878−1886

11 Li Fei, Gao Xiao-Guang, Wan Kai-Fang. Research on RBM training algorithm with dynamic Gibbs sampling. *Acta Automatica Sinica*, 2016, **42**(6): 931−942
(李飞, 高晓光, 万开方. 基于动态 Gibbs 采样的 RBM 训练算法研究. 自动化学报, 2016, **42**(6): 931−942)

12 Qiao Jun-Fei, Wang Gong-Ming, Li Xiao-Li, Han Hong-Gui, Chai Wei. Design and application of deep belief network with adaptive learning rate. *Acta Automatica Sinica*, 2017, **43**(8): 1339−1349
(乔俊飞, 王功明, 李晓理, 韩红桂, 柴伟. 基于自适应学习率的深度信念网设计与应用. 自动化学报, 2017, **43**(8): 1339−1349)

13 Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets. *Neural Computation*, 2006, **18**(7): 1527−1554

14 Salakhutdinov R, Murray I. On the quantitative analysis of deep belief networks. In: Proceedings of the 25th International Conference on Machine Learning. Helsinki, Finland: ACM, 2008. 872−879

15 Hinton G E. A practical guide to training restricted Boltzmann machines. *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012. 599−619

16 Salakhutdinov R, Hinton G. Deep Boltzmann machines. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics. Florida, USA: PMLR, 2009. 1967−2006

17 Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P A. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 2010, **11**: 3371−3408

18 Nielsen M. *Neural Networks and Deep Learning*. Determination Press [Online], available: http://neuralnet worksanddeeplearning.com, February 9, 2018.

19 Tieleman T. Training restricted Boltzmann machines using approximations to the likelihood gradient. In: Proceedings of the 25th International Conference on Machine Learning. Helsinki, Finland: ACM, 2008. 1064−1071

20 Tieleman T, Hinton G. Using fast weights to improve persistent contrastive divergence. In: Proceedings of the 26th Annual International Conference on Machine Learning. Montreal, Quebec, Canada: ACM, 2009. 1033−1040

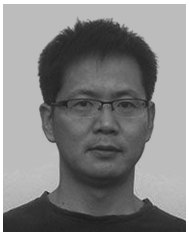
21 Abdel-Hamid O, Deng L, Yu D, Jiang H. Deep segmental neural networks for speech recognition. In: Proceedings of the 14th Annual Conference of the International Speech Communication Association. Lyon, France: International Speech and Communication Association, 2013. 1849−1853

22 Bengio Y, Thibodeau-Laufer É, Alain G, Yosinski J. Deep generative stochastic networks trainable by backprop. In: Proceedings of the 31st International Conference on Machine Learning. Beijing, China: JMLR, 2014. 226−234

23 Wang X S, Ma Y T, Cheng Y H. Domain adaptation network based on hypergraph regularized denoising autoencoder. *Artificial Intelligence Review*, DOI: 10.1007/s10462-017-9576-0

史 科 合肥工业大学计算机与信息学院博士研究生. 主要研究方向为自然语言处理, 信息检索, 机器学习.
E-mail: shike@mail.hfut.edu.cn
(**SHI Ke** Ph. D. candidate at the School of Computer and Information, Hefei University of Technology. His research interest covers natural language processing, information retrieval, and machine learning.)

陆 阳 合肥工业大学计算机与信息学院教授, 主要研究方向为人工智能, 计算机控制, 传感器网络. 本文通信作者.
E-mail: luyang.hf@126.com
(**LU Yang** Professor at the School of Computer and Information, Hefei University of Technology. His research interest covers artificial intelligence, computer control, and sensor network. Corresponding author of this paper.)



刘广亮 合肥工业大学计算机与信息学院博士研究生. 主要研究方向为数据挖掘和机器学习.
E-mail: homecs@126.com
(**LIU Guang-Liang** Ph. D. candidate at the School of Computer and Information, Hefei University of Technology. His research interest covers mining software repositories and machine learning.)

毕 翔 合肥工业大学计算机与信息学院讲师. 主要研究方向为模糊离散事件系统的建模和控制, 复杂软件可靠性.
E-mail: bixiang@hfut.edu.cn
(**BI Xiang** Lecturer at the School of Computer and Information, Hefei University of Technology. His research interest covers modeling and control of fuzzy discrete event systems, and reliability of complex software.)

王 辉 合肥工业大学高级工程师. 主要研究方向为复杂网络和神经网络.
E-mail: wanghui@hfut.edu.cn
(**WANG Hui** Senior engineer at Hefei University of Technology. His research interest covers complex networks and neural networks.)