基于自适应学习率的深度信念网设计与应用

乔俊飞 [1,2] 王功明 [1,2] 李晓理 [1] 韩红桂 [1,2] 柴 伟 [1,2]

摘 要 针对深度信念网 (Deep belief network, DBN) 预训练耗时长的问题, 提出了一种基于自适应学习率的 DBN (Adaptive learning rate DBN, ALRDBN). ALRDBN 将自适应学习率引入到对比差度 (Contrastive divergence, CD) 算法中, 通过自动调整学习步长来提高 CD 算法的收敛速度. 然后设计基于自适应学习率的权值训练方法, 通过网络性能分析给出学习率变化系数的范围. 最后, 通过一系列的实验对所设计的 ALRDBN 进行测试, 仿真实验结果表明, ALRDBN 的收敛速度得到了提高且预测精度也有所改善.

关键词 深度信念网, 自适应学习率, 对比差度, 收敛速度, 性能分析

引用格式 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

DOI 10.16383/j.aas.2017.c160389

Design and Application of Deep Belief Network with Adaptive Learning Rate

QIAO Jun-Fei[1,2] WANG Gong-Ming[1,2] LI Xiao-Li[1] HAN Hong-Gui[1,2] CHAI Wei[1,2]

Abstract A deep belief network with adaptive learning rate (ALRDBN) is proposed to solve the time-consuming problem in the pre-training period of DBN. The ALRDBN introduces the idea of adaptive learning rate into contrastive divergence (CD) algorithm and accelerates its convergence by a self-adjusting learning rate. The training method of weights in this case is designed, in which the adjusting scope of the coefficient in learning rate is determined by performance analysis. Finally, a series of experiments are carried out to test the performance of ALRDBN, and the corresponding results show that the convergence rate is accelerated significantly and the accuracy of prediction is improved as well.

Key words Deep belief network, adaptive learning rate, contrastive divergence, convergence rate, performance analysis

Citation 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

深度学习网络是人工神经网络的延伸, 在某种意义上等同于含有多个隐层的多层感知器 (Multilayer perceptron, MLP), 近年来在语音识别、计算机视觉以及大数据处理等方面取得了较大的进展. 深度学习通过提取底层特征信息来获取更加抽象的高层表示, 以揭示数据的分布式特征表示[1], 目前应用较为广泛的深度学习网络主要是深度信念网 (Deep belief network, DBN), 已经在多个领域得到了成功的推广和应用[2−8]. DBN 是由多个受限玻尔兹曼机 (Restricted Boltzmann machine, RBM) 顺序叠加构成, 采用无监督贪心算法对所有 RBM 逐一进行预训练, 然后再利用有监督的误差反传方法对整个网络权值进行微调[9]. DBN 的无监督预训练相当于传统神经网络中有监督训练之前的权值随机初始化, 这样可以将初始权值确定在一个较好的范围, 有利于克服由随机初始化权值导致陷入局部最优的问题[10].

尽管 DBN 已经在多个领域实现较好的应用, 但是在理论和学习算法方面仍存在许多难以解决的问题, 其中面临的最大挑战就是其预训练阶段耗时长的问题[11]. Lopes 等通过合理地选取学习参数提高了 RBM 的收敛速度[12], BN 整体学习速度方面效果不佳; 经过近几年的研究, 一种基于图像处理单元 (Graphic processing unit, GPU) 的硬件加速器被应用到 DBN 算法运算中, 并取得了显著的加速收敛效果[13−15], 该方法的主要问题是硬件设备成本和维护费用太高, 不经济并且也没有从算法的角度提高收敛速度. 随着大数据时代的到来, 处理数据的信息量会呈指数级增长, 传统 DBN 无法快速收敛甚至会难以完成学习任务, 因此如何既快速又经济地完成对大量数据的充分学习是 DBN 今后发展的一

收稿日期 2016-05-10 录用日期 2016-10-09
Manuscript received May 10, 2016; accepted October 9, 2016
国家自然科学基金 (61533002, 61473034), 国家杰出青年科学基金 (61225016), 内涵发展—引进人才科研启动费资助
Supported by National Natural Science Foundation of China (61533002, 61473034), National Natural Science Fund for Distinguished Young Scholars (61225016), Connotation Development — Scientific Research Start-up Funds of Talent Introduction
本文责任编委 王占山
Recommended by Associate Editor WANG Zhan-Shan
1. 北京工业大学信息学部 北京 100124 2. 计算智能与智能系统北京市重点实验室 北京 100124
1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124 2. Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124

个方向[16].

针对传统 DBN 预训练耗时长的问题, 文中将一种基于自适应学习率[13] 的思想引入到深度信念网 (Adaptive learning rate DBN, ALRDBN). 根据对比差度 (Contrastive divergence, CD)[17] 算法中参数每次迭代方向的异同来动态调整学习率, 从而成功避开了固定学习率所带来的欠学习和陷入局部最优的问题[12], 并通过网络性能分析, 给出自适应学习率的动态调整系数取值范围的选取方法. 同时将所设计的 ALRDBN 应用于 MNIST 数据库手写数字识别、大气二氧化碳浓度变化预测以及洛伦兹混沌时间序列预测等实验, 结果表明 ALRDBN 在学习过程中能够以更快的速度实现收敛, 同时预测精度也略有提高.

文中第 1 节介绍 ALRDBN 的结构及学习过程; 第 2 节进行网络性能分析, 并给出自适应学习率动态调整系数取值范围的选取方法; 第 3 节对网络的性能进行仿真实验, 并对结果进行分析; 第 4 节给出总结与展望.

1 自适应学习率深度信念网

1.1 ALRDBN 结构

与传统的 DBN 结构类似, ALRDBN 由若干个 RBM 顺序叠加构成, 上一个 RBM 的输出作为下一个 RBM 的输入. 考虑到网络的过拟合和泛化能力等因素, 在处理中等规模的数据时 DBN 隐含层层数一般选用 2 至 3 层, 为了方便起见, 故 ALRDBN 隐含层选用 2 层, 网络结构如图 1 所示. 双向箭头表示相邻两层之间为双向全连接, 同一层神经元之间没有连接.

从图 1 可以看出, DBN 学习过程相对复杂, 分为两个阶段: 预训练阶段和微调阶段, 在预训练阶段, 首先使用 CD 算法对构成 DBN 的所有 RBM 逐一进行无监督训练; 在微调阶段, 将整个网络展开成一个前向型网络, 再使用误差反传方法对整个网络权值进行有监督的微调.

1.2 ALRDBN 预训练过程

RBM 是一种能量生成模型, 由可视层 (输入层) 和隐含层 (输出层) 组成, 两层之间采用双向全连接, 同一层之间没有连接, 如图 2 所示.

用向量 $\boldsymbol{v}$ 和 $\boldsymbol{h}$ 分别表示可见层和隐含层的状态, 其中 v_i 表示第 i 个可见单元的状态, h_j 表示第 j 个隐单元的状态, 所有神经元的输出只有两种状态 (开启和关闭), 一般用 1 和 0 来分别表示开启和关闭. 图 2 中的偏置是具有与 v_i 和 h_j 相同量纲的标量, 那么基于给定的一组可视层和隐含层状态 $(\boldsymbol{v},\boldsymbol{h})$ 的能量函数[16] 可定义为

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

其中, $\boldsymbol{\theta} \in \{W_{ij}, a_i, b_j\}$ 是 RBM 的参数, 均为实数. $\boldsymbol{a} \in \mathbf{R}^m$ 是可视层节点的偏置, $\boldsymbol{b} \in \mathbf{R}^n$ 是隐含层节点的偏置, $W \in \mathbf{R}^{n\times m}$ 是可视层和隐含层之间的连接权值矩阵.

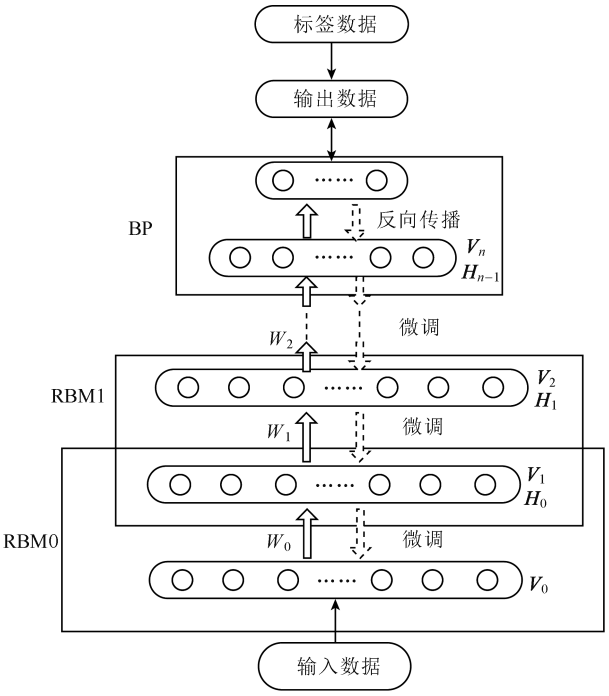


图 1 ALRDBN 结构

Fig. 1 The structure of ALRDBN

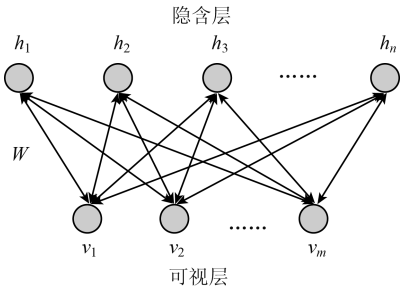


图 2 受限玻尔兹曼机

Fig. 2 Restricted Boltzmann machine

当参数确定时, 对于可视层和隐含层所出现的每一种状态 $(\boldsymbol{v},\boldsymbol{h})$, 基于能量函数, 其联合概率分布

为

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

其中, Z 是配分函数, 可以看作为可视层和隐含层所有状态下的能量函数之和, 如式 (3) 所示.

$$Z=\sum_{\boldsymbol{v},\boldsymbol{h}}\mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})} \tag{3}$$

同时, 关于可视层 $\boldsymbol{v}$ 的分布 $P(\boldsymbol{v}/\boldsymbol{\theta})$, 即联合概率分布 $P(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})$ 的边缘分布 (也称为似然函数), 如式 (4) 所示.

$$P(\boldsymbol{v}/\boldsymbol{\theta})=\sum_{\boldsymbol{h}}P(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})=\frac{\sum\limits_{\boldsymbol{h}}\mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}}{Z} \tag{4}$$

由 RBM 的结构可以看出, 从可视层到隐含层的映射以及对可视层的重构实质就是对神经元的激活, 因此神经元的状态用 0 或 1 表示, 0 表示神经元处于关闭状态, 1 表示神经元处于开启状态. 无论是可视层还是隐含层, 同一层的节点之间是相互独立的, 那么在给定可视层状态 $\boldsymbol{v}$ 的情况下, 隐含层第 j 个单元 h_j 被开启的概率可表示为

$$P(h_j=1/\boldsymbol{v},\boldsymbol{\theta})=\sigma(b_j+\sum_{i=1}^{m}v_iW_{ij}) \tag{5}$$

根据 RBM 的对称结构可知, 当给定隐含层状态 $\boldsymbol{h}$ 时可视层第 i 个单元 v_i 被开启的概率可表示为

$$P(v_i=1/\boldsymbol{h},\boldsymbol{\theta})=\sigma(a_i+\sum_{j=1}^{n}W_{ij}h_j) \tag{6}$$

其中

$$\sigma(x)=\frac{\mathrm{e}^{x}}{\mathrm{e}^{x}+1}=\frac{1}{1+\mathrm{e}^{-x}} \tag{7}$$

判断激活和开始的标准常通过设定一个阈值[17]来实现, 如式 (8) 所示.

$$h_j=\begin{cases}1, & 若\quad p(h_j=1/v)>\xi\\ 0, & 若\quad p(h_j=1/v)<\xi\end{cases} \tag{8}$$

其中, ξ 为一个介于 0.5 到 1 的常数.

重复式 (5) 和式 (8) 即为一个 Gibbs 采样过程, 具有 k 个连续 Gibbs 采样的 CD 算法如图 3 所示.

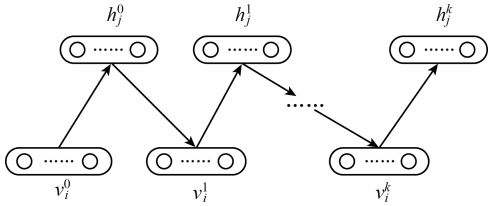


图 3　CD-k 算法

Fig. 3　The algorithm of CD-k

通过上述分析可知, $P(\boldsymbol{v}/\boldsymbol{\theta})$ 越大说明可视层被开启的概率越大, 即隐含层充分学习到了可视层数据的特征. 根据 CD 算法和梯度上升的原理[12], 可以通过极大似然法调整连接权值 $\boldsymbol{\theta}$ 来得到 $P(\boldsymbol{v}/\boldsymbol{\theta})$ 的最大值. 对 $P(\boldsymbol{v}/\boldsymbol{\theta})$ 求对数导数可得

$$\frac{\partial\log P(\boldsymbol{v}/\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}=-\frac{\sum\limits_{\boldsymbol{h}}\mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}\frac{\partial E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}}{\sum\limits_{\boldsymbol{h}}\mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}}+\frac{\sum\limits_{\boldsymbol{v},\boldsymbol{h}}\mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}\frac{\partial E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}}{\sum\limits_{\boldsymbol{v},\boldsymbol{h}}\mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}} \tag{9}$$

由贝叶斯定理可知

$$P(\boldsymbol{h}/\boldsymbol{v},\boldsymbol{\theta})=\frac{P(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}{P(\boldsymbol{v}/\boldsymbol{\theta})}=\frac{\mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}}{\sum\limits_{\boldsymbol{h}}\mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}} \tag{10}$$

故, 式 (9) 可表示为

$$\begin{aligned}&\frac{\partial\log P(\boldsymbol{v}/\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}=\\&-\sum_{\boldsymbol{h}}P(\boldsymbol{h}/\boldsymbol{v},\boldsymbol{\theta})\frac{\partial E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}+\\&\sum_{\boldsymbol{v},\boldsymbol{h}}P(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})\frac{\partial E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}=\\&\left\langle\frac{\partial E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right\rangle_{P(\boldsymbol{h}/\boldsymbol{v},\boldsymbol{\theta})}-\\&\left\langle\frac{\partial E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right\rangle_{P(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})}\end{aligned} \tag{11}$$

其中, $\langle\cdot\rangle_P$ 表示求关于分布 P 的数学期望.

对于每一个训练样本, 可以分别用 "data" 和 "model" 来表示 $P(\boldsymbol{h}/\boldsymbol{v},\boldsymbol{\theta})$ 和 $P(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})$ 这两个概率分布, 则对数似然函数对连接权值 W_{ij}、可视层单元的偏置 a_i 和隐含层单元的偏置 b_j 的偏导数分别为

$$\frac{\partial\log P(\boldsymbol{v}/\boldsymbol{\theta})}{\partial W_{ij}}=\langle v_ih_j\rangle_{\mathrm{data}}-\langle v_ih_j\rangle_{\mathrm{model}} \tag{12}$$

$$\frac{\partial\log P(\boldsymbol{v}/\boldsymbol{\theta})}{\partial a_i}=\langle v_i\rangle_{\mathrm{data}}-\langle v_i\rangle_{\mathrm{model}} \tag{13}$$

$$\frac{\partial\log P(\boldsymbol{v}/\boldsymbol{\theta})}{\partial b_j}=\langle h_j\rangle_{\mathrm{data}}-\langle h_j\rangle_{\mathrm{model}} \tag{14}$$

由于 $\langle\cdot\rangle_{\mathrm{model}}$ 的计算既费时且工作量又大, 因此将 Gibbs 的采样个数缩至 k 个 (CD-k 准则), 根据 CD-k 准则, 参数集 $\boldsymbol{\theta}\in\{W_{ij},a_i,b_j\}$ 的更新公式为

$$W_{ij}=W_{ij}+\eta\left(\langle v_ih_j\rangle_0-\langle v_ih_j\rangle_k\right) \tag{15}$$

$$a_i = a_i + \eta\left(\langle v_i\rangle_0 - \langle v_i\rangle_k\right) \tag{16}$$

$$b_j = b_j + \eta\left(\langle h_j\rangle_0 - \langle h_j\rangle_k\right) \tag{17}$$

其中, η 为学习率, 大量研究工作证明, 将 k 的值取 1 即能达到可视层和隐含层的平稳分布[12].

1.3 学习率自适应调整方法

在 CD-k 算法中, 由于每个 RBM 均需要多次迭代, 且每次迭代后的参数更新方向不尽相同, 所以固定的学习率会导致算法出现“早熟”现象或难以收敛, 因此如何做到使算法根据合适的梯度来自适应地控制学习速度成为关键. 根据 RBM 训练过程连续两次迭代后的参数更新方向的异同设计 ALRDBN 算法, 自适应学习率更新机制[15] 为

$$\eta=\begin{cases}\alpha\eta, & \Delta>0\\ \beta\eta, & \Delta<0\end{cases} \tag{18}$$

其中, $\Delta = (\langle v_i h_j\rangle_0 - \langle v_i h_j\rangle_k)\cdot(\langle v_i h_j\rangle'_0 - \langle v_i h_j\rangle'_k)$, α 和 β 分别表示学习率增大系数和减小系数, $0<\beta<\alpha$, Δ 表示 RBM 连续两次迭代的参数变化量的乘积. 当连续两次迭代后参数更新方向相反时, 学习率会减小; 当连续两次迭代后参数更新方向相同时, 学习率会加大. 针对自适应学习率算法在加速收敛方面上的有效性问题, 给出以下算法显著性差异分析.

1) 收敛速度在很大程度上是指受限玻尔兹曼机 (RBM) 无监督学习速度, 而 RBM 无监督学习是指从可视层接收的原始数据到隐含层所表示的抽象数据的过程, 这一过程被称为 Encoder, 然后再经过一个 Decoder 过程来解决人们所期望的回归或者分类问题. 自适应学习率以变步长的方式自动调节学习因子加速了单个 RBM 的 Encoder 过程, 而 ALRDBN 是由多个 RBM 顺序叠加组成的, 上一个 RBM 的 Encoder 过程的输出作为下一个 RBM 的 Encoder 过程的输入, 以此类推. Hinton 教授在论文《Reducing the dimensionality of data with neural networks》中指出分层降维能够达到高维数据维数呈现指数下降的效果. 同理, 由于 ALRDBN 在功能上也是由多个 RBM 进行的分层表述, 所以在单个 RBM 运用自适应学习率加速收敛的情况下, 多个 RBM 分层表述能够产生收敛速度指数提高的效果. 分层表述过程中 Encoder 和 Decoder 的网络结构如图 4 所示.

2) 受限玻尔兹曼机 (RBM) 的权值学习过程是一种无监督的网络权值学习过程, 它有别于以往的 BP 网络和 RBF 网络等的有监督学习过程. 在以往的 BP 网络权值学习过程中学习率也有采取自适应形式的, 也就是说学习率为非定值. 但是 BP 网络通常是根据误差的大小来动态调整学习率的变化 (如当误差较大时常常加大学习率来减小误差, 当误差较小时常常减小学习率来防止震荡或者减小模型的误差). 由于 RBM 是无监督学习网络, 它的权值训练过程中学习率也采用自适应学习的方式调整, 所以很难找到如 BP 算法类似结论, 因为它是无监督的. 无监督训练阶段, 在误差曲面寻找最优点的过程中, 这种学习率自适应调整方法能够既准确又迅速的找到满足目标函数的最优解. 具体来说, 自适应学习率算法能够根据误差曲面的凹凸性来自适应地增大或减小学习率, 实现在预训练阶段加速收敛的效果. 另外, 在有监督微调阶段, 自适应学习率算法能够有效地克服误差校正信号越来越微弱的缺点, 避开了算法在寻优过程中处于循环波动和陷入局部最优的情况.

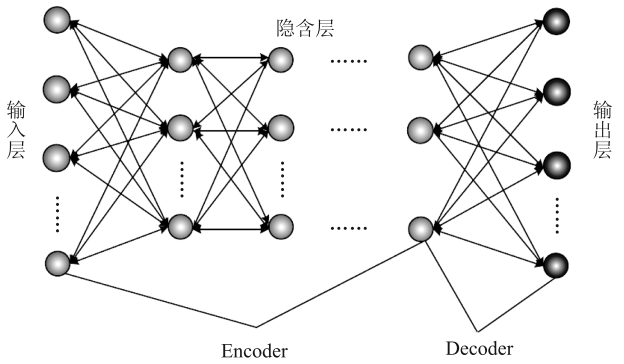


图 4 ALRDBN 分层表述结构

Fig. 4 Hierarchical representation structure of ALRDBN

3) 在有监督训练阶段, 图 5 是 ALRDBN 网络中来自输出层的误差反传过程. 其中, y_{id}^{l-1} 表示第 d 组训练样本作用下最后一个隐含层中第 i 个神经元的输入状态, $\boldsymbol{W}$ 表示输出层和最后一个隐含层之间的连接权值, S 表示输出层的输入, D 表示训练样本个数 y_{jd}^{l} 和 $\hat{y}_{jd}^{l}$ 分别表示第 d 组训练样本作用下输出层第 j 个神经元的实际输出和期望输出, 那么对应的误差可表示为 $e_d = \hat{y}_{jd}^{l} - y_{jd}^{l}$, 损失函数可定义为

$$L = \frac{1}{2}\sum_{d=1}^{D} e_d^2 \tag{19}$$

那么损失函数对权值的导数为

$$\frac{\partial L}{\partial W_{ij}} = \frac{\partial L}{\partial e_{jd}}\cdot\frac{\partial e_{jd}}{\partial y_{jd}^{l}}\cdot\frac{\partial y_{jd}^{l}}{\partial s_{ij}}\cdot\frac{\partial s_{ij}}{\partial W_{ij}} =$$

$$e_{jd}\cdot(-1)\cdot\lambda\left(s_j\right)\cdot y_{jd}^{l-1} \tag{20}$$

其中

$$\lambda(x) = \frac{1}{1-\mathrm{e}^{-x}}\left(1-\frac{1}{1-\mathrm{e}^{-x}}\right) \tag{21}$$

由式 (20) 可知, 权值更新过程中误差校正信号为

$$\eta\frac{\partial L}{\partial W_{ij}}=-\eta e_{jd}\cdot\lambda\left(s_j\right)\cdot y_{jd}^{l-1} \tag{22}$$

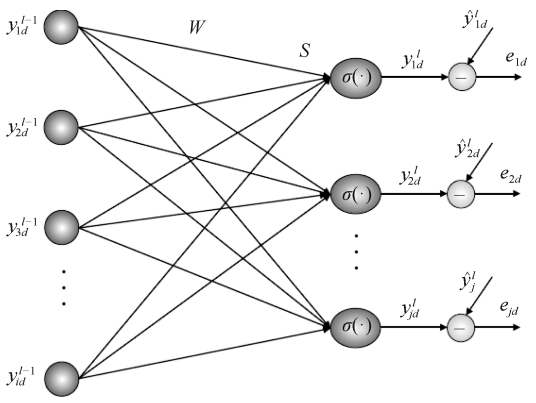


图 5　ALRDBN 顶层的反传误差

Fig. 5　Error back-propagated from top layer of ALRDBN

传统固定学习率算法的一个缺点就是, 权值更新过程中的每一步调节并不总是朝着损失函数减小的方向进行, 致使收敛速度大大降低. 但是当学习率 η 以式 (18) 所示的规律自适应变化时, 由式 (22) 可知, 有监督学习阶段误差校正信号会根据连续两次更新方向的异同自适应地变化, 能够克服传统固定学习率算法的这一缺点, 从而加速收敛速度. 同理可知, 偏置项 a_i 和 b_j 寻优过程也会以相同的自适应方法加速收敛.

综上分析可知, 基于自适应学习率算法下的权值更新思想是, 如果连续两步的学习同时降低 $\langle v_ih_j\rangle_0$ 与 $\langle v_ih_j\rangle_k$ 之间的差值, 则学习率增大; 相反地, 如果增大了 $\langle v_ih_j\rangle_0$ 与 $\langle v_ih_j\rangle_k$ 之间的差值则学习率减小[18−19]. 这种确定学习率变化系数的 ALRDBN 不仅能够提高网络的鲁棒性, 还能够加速收敛过程.

2　网络性能分析

ALRDBN 预训练过程是利用变学习率对每个 RBM 进行无监督学习, 以期望提取更多的输入信息特征. 不失一般性, 将式 (7) 中 Sigmoid 函数的上下渐近线推广到 A_L 和 A_H, f_i^0 和 f_i^1 分别表示可视层的输入信息和重构状态, f_i^0 和 f_i^1 分别表示隐含层状态, 此时一次 Gibbs 采样过程中可视层和隐含层的状态表示如下

$$f_i^0\in[A_L,A_H] \tag{23}$$

$$f_j^0=A_L+(A_H-A_L)\sigma\left(b_j+\sum_{i=1}^{m}f_i^0W_{ij}\right) \tag{24}$$

$$f_i^1=A_L+(A_H-A_L)\sigma\left(a_i+\sum_{j=1}^{n}W_{ij}f_j^0\right) \tag{25}$$

$$f_j^1=A_L+(A_H-A_L)\sigma\left(b_j+\sum_{i=1}^{m}f_i^1W_{ij}\right) \tag{26}$$

ALRDBN 算法是一种对极大似然的近似, 构成 ALRDBN 的所有 RBM 被逐一训练, 上一个 RBM 的输出作为下一个 RBM 的输入, 在每个 RBM 的一次 Gibbs 采样过程中, 网络输出与采样过程的中间状态有关. 同时, 自适应学习率对误差曲面最小点搜索的准确性起决定性作用, α 过大或过小都会影响到算法的收敛速度和精度, 甚至使网络不稳定. 故通过以上论述, 给出如下性能分析:

对于由 3 个隐含层构成的 ALRDBN, 若整个网络稳定, 顶层 RBM 输出状态范围为, $f_j^1\in[A_L,\ A_H]$ 必满足 $f_j^0,f_i^1\in[A_L,\ A_H]$ 且 α 与 ξ 正相关.

一方面, 由于 3 个隐含层是顺序堆叠的, 故当 $f_j^0,f_i^1\in[A_L,\ A_H]$ 时, 根据式 (23)∼(26) 可知, 下一个 RBM 的输出范围必定也为 $[A_L,\ A_H]$, 即 $f_j^1\in[A_L,\ A_H]$, 所以满足整个网络输入输出的有界性, 即网络稳定.

另一方面, 若网络稳定, 则每个 RBM 的可视层和隐含层状态均满足有界性, 即当第 1 个 RBM 的可视层输入为 $f_i^0\in[A_L,\ A_H]$ 时, 顶层 RBM 的输出范围为 $f_j^1\in[A_L,\ A_H]$. 又因为 Sigmoid 函数是单调递增的, 且随着被开启的神经元个数不断增加, 可得

$$f_j^1>f_i^1 \tag{27}$$

$$f_i^1>f_j^0 \tag{28}$$

即中间状态 f_j^0 和 f_i^1 范围为

$$f_j^0,f_i^1\in[A_L,A_H] \tag{29}$$

从式 (8) 可知, 当 ξ 增大时隐含层神经元被开启的概率减小, Gibbs 采样过程中所得到的可视层和隐含层神经元状态采样值的稀疏性[20−22] 越明显, 那么连续两次 Gibbs 迭代之后权值更新方向一致的概率增大, 即

$$\begin{aligned}P\big(&\left(\langle f_if_j\rangle_0-\langle f_if_j\rangle_k\right)\times\\&\left(\langle f_if_j\rangle_0'-\langle f_if_j\rangle_k'\right)>0\big)\propto\xi\end{aligned} \tag{30}$$

根据式 (15)∼(17) 可知, 当权值调整处于误差曲面比较平坦的区域时, 加大学习率可以在不影响精度的情况下加快收敛速度, 则有

$$\begin{aligned}\alpha\propto P\big(&\left(\langle f_if_j\rangle_0-\langle f_if_j\rangle_k\right)\times\\&\left(\langle f_if_j\rangle_0'-\langle f_if_j\rangle_k'\right)>0\big)\end{aligned} \tag{31}$$

由式 (30) 和 (31) 可知, ξ 与 α 的数学定性关系式为

$$\alpha \propto \xi \tag{32}$$

同时, 在一个 Gibbs 采样过程中权值更新一次, 而中间状态的二值化采样要进行两次, 并且每次更新的权值与状态采样成正比, 所以得到 ξ 与 α 的近似关系式如下:

$$\alpha \approx 2\xi \tag{33}$$

ξ 是对隐含层神经元状态 (开启或者关闭) 进行判别的概率阈值, 取值范围是 [0.5, 1], 在实际应用中常取值为 0.7, 由式 (33) 可得 α 的取值范围是 [1, 2], 同理可知, β 的取值范围是 [0, 1].

3 实验及分析

为了验证所提方法的有效性, 本节对手写数字识别、CO_2 浓度预测和 Lorenz 时间序列预测等进行了仿真研究. 为了消除其他无关因素对实验效果的影响, 从而客观地反映所提方法的有效性, 所有仿真实验的编译软件和计算机运行环境设置情况如下: 编译软件为 Matlab 8.2 版本, 计算机处理器为 Intel(R) Core(TM) i7-4790, 主频为 3.6 GHz, RAM 为 8 GB.

3.1 手写数字识别

该实验数据源于 MNIST 手写数据库, 该数据库拥有 6 万个训练图像和 1 万个测试图像, 其中的数字都是手写体, 每一个数字均用很多数量的手写方式来显示. 随着模式识别和数据挖掘技术的不断发展, 很多理论方法应用到该数据库中, 所以该数据库被视为一种理想的、标准的测试新方法的经典对象.

取 5 000 个样本用于训练, 1 000 个样本用于测试. 每张样本图像为 0 ~ 9 手写体的阿拉伯数字, 像素为 28×28, 故可视层神经元个数设定为 784 个, 每个神经元接收每张图像中的一个像素点. 5 000 个样本分为 50 批次进行训练, 每批次包含 100 个样本, 每个 RBM 迭代 50 次, 故每层神经元个数默认为 100 个, 学习率增大和减小系数的根据经验取值为: $\alpha = 1.5, \beta = 0.5$. 实验中根据 RBM 的分布, 进行一次 Gibbs 采样后所获样本与原数据的差异称为重构误差, 重构误差变化曲线能直观地反映出无监督学习的效果和收敛速度, 重构误差可表示为

$$RE = \frac{\sum\limits_{i=1}^{s}\sum\limits_{j=1}^{d}\left|v_{ij}^{0} - v_{ij}^{1}\right|}{s \cdot d} \tag{34}$$

其中, s 为样本个数, d 为输入维数, v_{ij}^{1} 为可视层神经元的重构状态, v_{ij}^{0} 为真实值, 仿真结果如图 6、图 7 和图 8 所示.

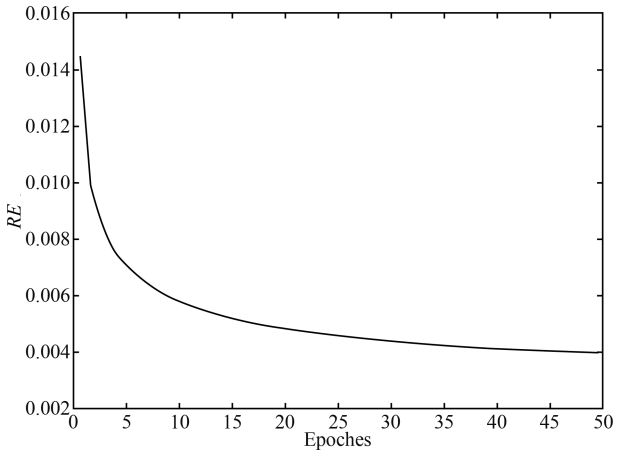


图 6 顶层 RBM 的重构误差

Fig. 6 The reconstruction error of top RBM

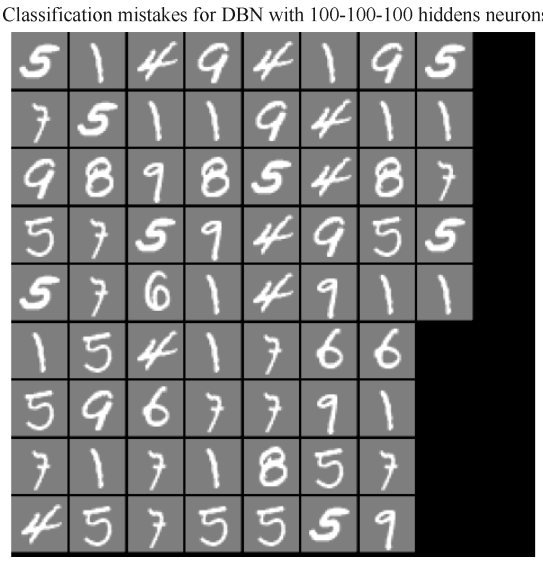


图 7 ALRDBN 错误识别原图像

Fig. 7 The original images with classification mistakes of ALRDBN

由图 6 可以看出, 重构误差一开始呈现急速下降趋势, 当迭代到第 30 次时基本达到稳定 (收敛), 由此可以看出自适应学习率提高了算法的收敛速度; 图 7 和图 8 显示, 对 1 000 个样本进行测试后产生了 68 处错误.

为了更好地展现 ALRDBN 的快速收敛性以及更高的识别精度, 在相同的实验环境和设置下将 ALRDBN 与其他算法相比较, 结果如表 1 所示, 图 9 是隐含层神经元数与收敛时间的关系曲线.

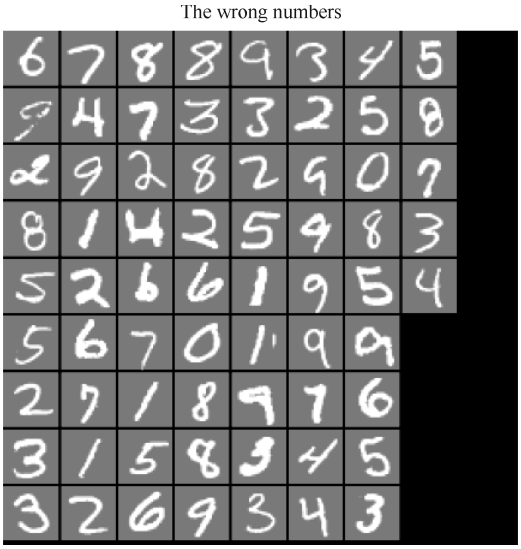


图 8　ALRDBN 错误识别图像

Fig. 8　The images with classification mistakes of ALRDBN

表 1　MNIST 手写数字实验结果对比

Table 1　Result comparison of MNIST experiment

方法	隐含层数	每层节点数	正确识别率	运算时间 (s)
ALRDBN	2	100	**93.1 %**	**20.0**
CDBN	2	100	93.0 %	34.3
DBN[21]	2	100	92.6 %	32.9

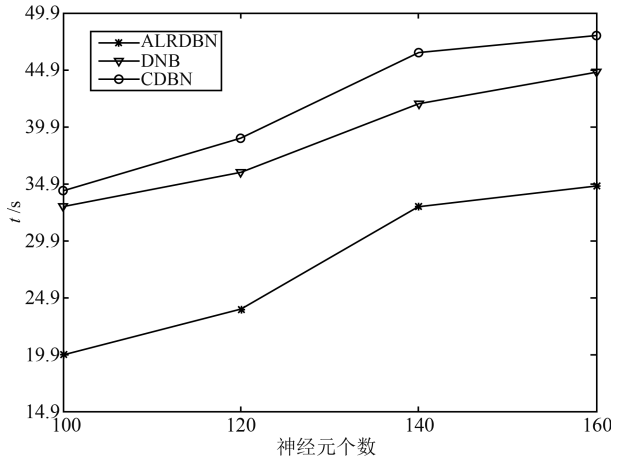


图 9　隐含层神经元数对收敛时间的影响

Fig. 9　Effect of the number of hidden neurons on convergence time

由表 1 可以看出, 在隐含层个数和每层神经元个数相同的情况下, ALRDBN 的收敛速度最快, 且对手写数字识别的正确率略有提高. 同时, 从图 9 可知, 在相同的隐含层神经元数条件下 ALRDBN 的收敛时间最快. 该 MNIST 手写数字识别实验证明, ALRDBN 具有相对较好的学习能力和较快的收敛速度.

同时, 为了更充分地说明学习率增大和减小系数 α 和 β 对加快算法收敛的影响, 在确保其他实验参数不变以及 ALRDBN 性能稳定的前提下, 将 α 和 β 取若干组不同的值并再次实验. 图 10 是不同的 α 和 β 对应的算法收敛时间. 可以看出, 当 α 和 β 在第 2 节所确定的范围内取值时, 算法收敛速度变化很微弱, 收敛时间基本维持在 20 s 左右, 此时 ALRDBN 处于平稳状态. 当 α 和 β 的取值超出稳定范围时, 算法收敛时间出现跳跃式增加, 此时 ALRDBN 处于不稳定状态. 由此可见, 第 2 节所确定的学习率增大和减小系数 α 和 β 的取值范围在理论和应用上得到了有效的验证.

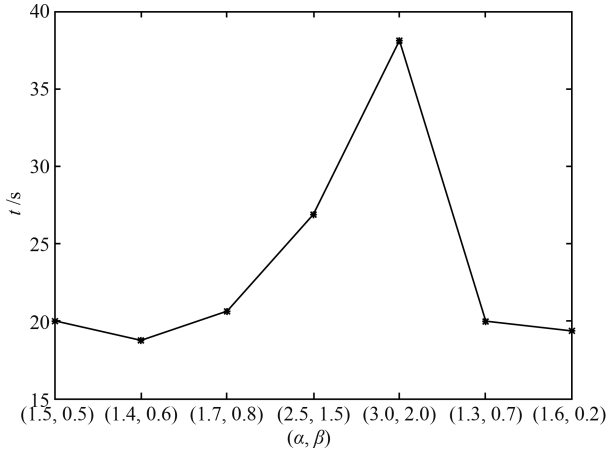


图 10　α 和 β 对收敛时间的影响

Fig. 10　Influence of α and β on convergence time

3.2　预测大气 CO_2 浓度变化

环境问题一直是世界各国关心的焦点之一, 而 CO_2 浓度的变化恰恰能反映出环境质量的好坏. 大气中 CO_2 浓度的增加导致全球变暖; 反过来, 气候的变化也会影响到碳循环, 从而影响到大气中的 CO_2 浓度. 大气中 CO_2 浓度的增加一方面取决于人为排放的增加 (这也取决于世界人口增长速度、能源需求量的增加速度和开发速度以及替代能源的开发速度等等), 另一方面又取决于自然 CO_2 贮库对人为排放的 CO_2 的响应, 特别是生物圈和海洋的响应. 因此为了客观地反映空气中 CO_2 的浓度变化情况, 选取西太平洋某海域的一个海岛 CO_2 的浓度变化进行预测. 实验使用此海岛 1965 年到 1980 的 CO_2 浓度数据共 187 组, 前 100 组数据作为训练样本对网络进行训练, 后 87 组数据作为测试样本对网络进行测试, 网络使用 3 个历史数据作为输入来进行一步预测. 为了较好地学习到 CO_2 数据的内在特征提高预测效果, 采用的 ALRDBN 网络结构为 3-20-40-1, 每个 RBM 训练次数为 200 次, 学习率增大和减小系数的根据经验取值为: $\alpha = 0.5$, $\beta = 1.5$. 有监督的微调阶段训练步数为 10 000 次, 仿真结果

如图 11、图 12 和图 13 所示.

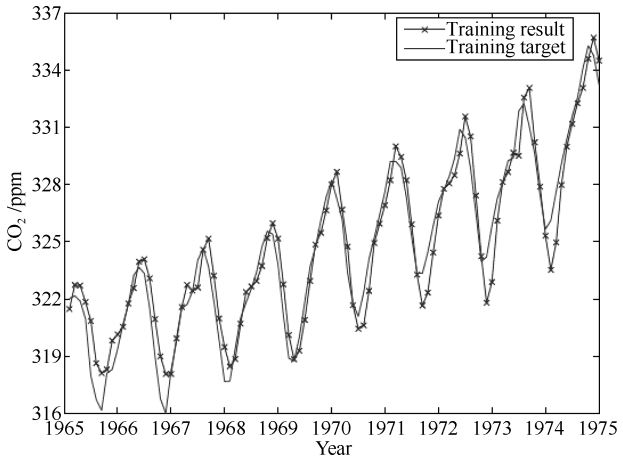


图 11 ALRDBN 训练结果

Fig. 11 The training results of ALRDBN

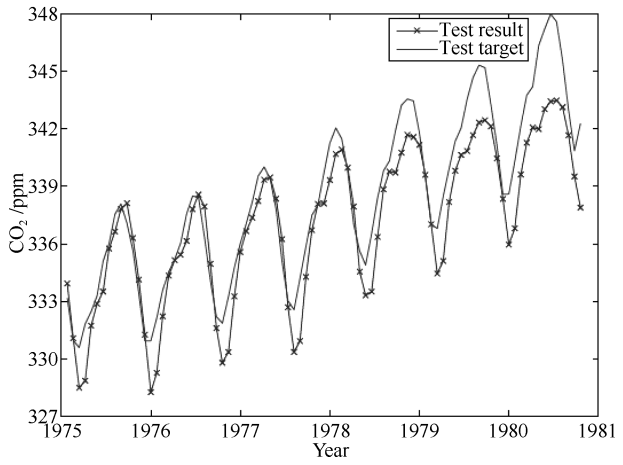


图 12 ALRDBN 测试结果

Fig. 12 The test results of ALRDBN

图 11 是对网络的训练结果, 图 12 是对网络的预测结果, 可以看出该海岛所在地区 CO_2 浓度在不同季节有规律地变化, 这主要是由不同季节海岛居民的有规律活动引起的, 但总体上呈现出逐年上升的趋势, 这也符合当今全球 CO_2 浓度逐年走高的现实情况 (温室效应导致的全球变暖). 图 13 是有监督阶段均方根误差的变化曲线, 可以看出, 一开始均方根误差急速下降后出现波动, 这是因为一开始无监督学习过程中某两次参数更新方向不同而导致的, 随着迭代次数的增加, 学习率以自适应的方式不断加大, 从而加速了学习过程. 当迭代到 1 000 次时基本趋于稳定, 这说明自适应学习率提高了无监督学习算法的收敛速度并最终实现收敛.

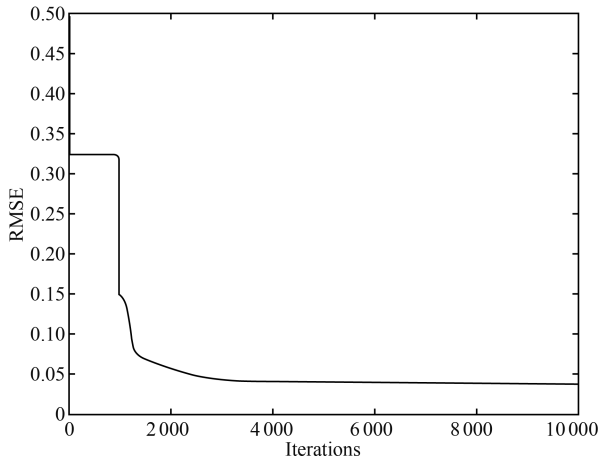


图 13 ALRDBN 训练 RMSE

Fig. 13 The training RMSE of ALRDBN

为了更进一步说明实验效果, 在相同的实验环境和参数设置下将 ALRDBN 与其他算法相比较, 结果如表 2 和图 14 所示.

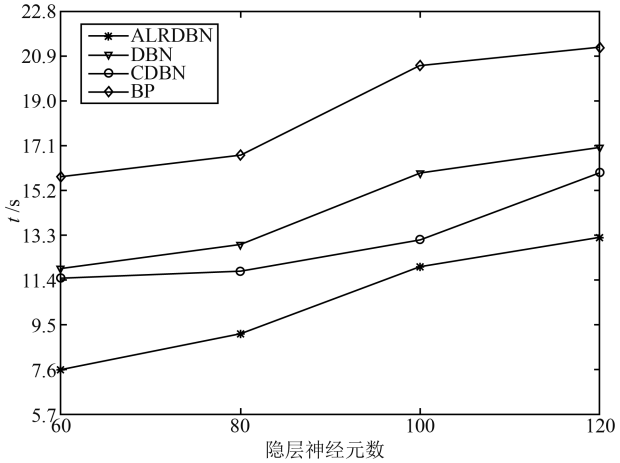


图 14 隐含层神经元数对收敛时间的影响

Fig. 14 Effect of the number of hidden neurons on convergence time

表 2 CO_2 浓度变化实验结果对比

Table 2 Result comparison of CO_2 forecasting

方法	网络结构	RMSE (训练)	RMSE (测试)	运算时间 (s)
ALRDBN	3-20-40-1	**0.9164**	**1.1671**	**7.6**
DBN	3-20-40-1	0.9487	1.2830	11.9
CDBN[22]	3-20-40-1	0.9133	1.1507	11.5
BP	3-60-1	>0.1	$1.3\sim6.6$	15.8

由表 2 可以看出, BP 网络在采用 3-6-1 的结构时网络性能相对稳定在一定的范围, 这说明 BP 网络耗时且易陷入局部最优. 在隐含层个数和每层神经元个数相同的情况下, 深度神经网络中 ALRDBN 的运算速度 (收敛速度) 最快, 同时均方根误差相对略有降低. 实际经验表明, 有规律的增加或者减少

隐含层神经元个数可以获得优于表 2 的均方根误差. 图 14 表明尽管随着隐含层神经元数的增加收敛时间延长, 但是 ALRDBN 的收敛时间仍然是最快的. 该 CO_2 浓度变化预测实验证明, ALRDBN 具有较快的收敛速度. 另一方面, 为了更充分地验证第 2 节所确定的 α 和 β 取值范围的有效性, 在确保其他实验参数不变以及 ALRDBN 性能稳定的前提下, 将 α 和 β 取若干组不同的值并再次实验. 图 15 是不同的 α 和 β 对应的算法收敛时间, 可以看出, 当 α 和 β 在第 2 节所确定的范围内取值时算法收敛时间变化很微弱, ALRDBN 处于平稳状态. 当 α 和 β 的取值超出稳定范围时算法收敛时间出现跳跃式增加, 此时 ALRDBN 处于不稳定状态.

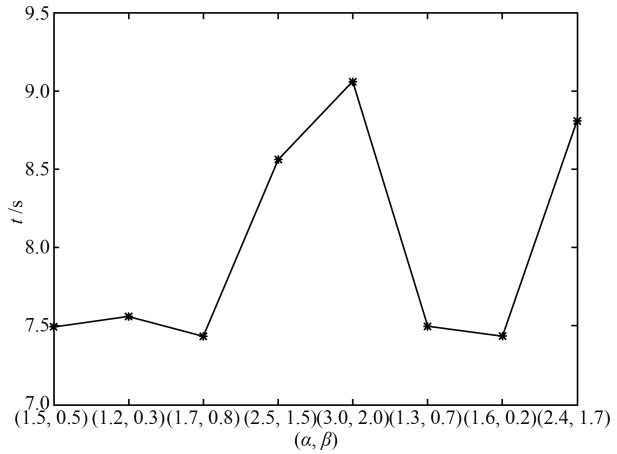


图 15　α 和 β 对收敛时间的影响

Fig. 15　Influence of α and β on convergence time

3.3　Lorenz 混沌时间序列预测

本实验目的是验证 ALRDBN 能够以较快的速度完成预测任务, 即能够提高算法收敛的速度. 混沌时间序列预测是检验神经网络结构和方法有效性的基准问题之一, Lorenz 效应因其相位图酷似蝴蝶又称蝴蝶效应. 实验中, 取 1 500 个样本用于训练, 1 000 个样本用于测试, 在网络中使用 3 输入进行一步预测, 即用 $t-2$, $t-1$ 和 t 时刻的信息预测 $t+1$ 时刻的信息, ALRDBN 网络结构为 3-3-3-1 每个 RBM 的迭代 100 次, 微调阶段训练步数为 5 000 次, 实验结果如图 16、图 17 和图 18 所示.

图 16 和图 17 分别是 ALRDBN 对 Lorenz 混沌时间序列的训练结果和预测结果, 可以看出, ALRDBN 很好地学习了 Lorenz 时序状态, 并实现了较好的预测. 图 18 是 ALRDBN 在有监督微调过程中均方根误差变化曲线, 从图中可以看出, 均方根误差一开始急速下降, 这是因为无监督学习过程的初始阶段参数更新方向一致, 学习率以自适应的方式不断加大, 从而加速了学习过程. 当迭代到 500 次左右时均方根误差基本趋于稳定, 最终训练误差为 0.0326. 这说明基于自适应学习率算法收敛速度较快. 为了更进一步展现实验效果, 在相同的实验环境和参数设置下将 ALRDBN 与其他算法相比较, 结果如表 3 和图 19 所示. 由图 19 和表 3 可以看出,

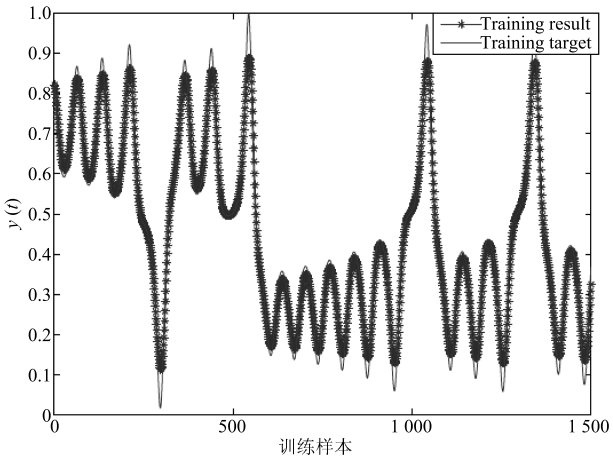


图 16　ALRDBN 训练结果

Fig. 16　The training results of ALRDBN

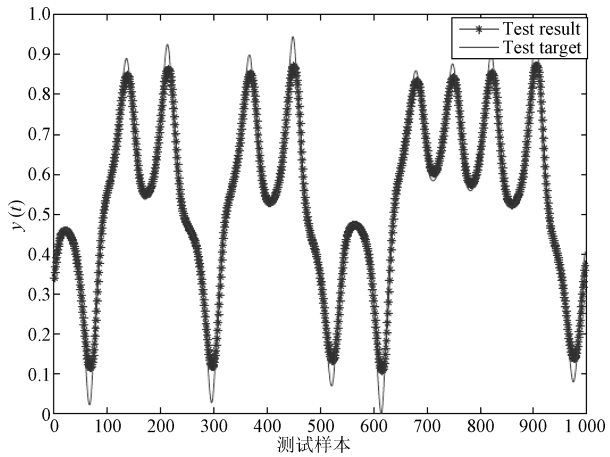


图 17　ALRDBN 测试结果

Fig. 17　The test results of ALRDBN

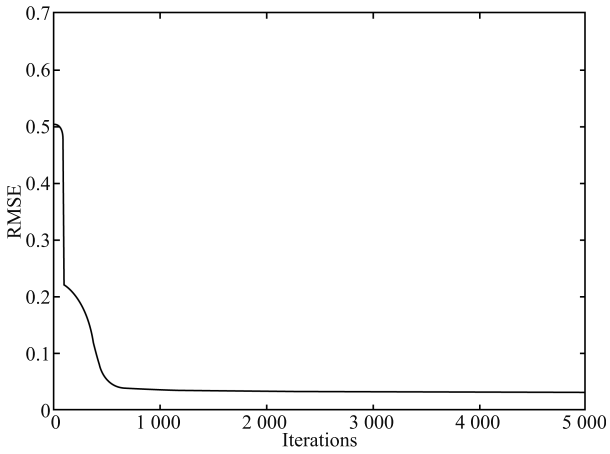


图 18　ALRDBN 训练 RMSE

Fig. 18　The training RMSE of ALRDBN

在隐含层个数和每层神经元个数以及网络总神经元个数相同的情况下, ALRDBN 的收敛速度最快, 同时均方根误差也略有降低.

表 3 Lorenz 时序预测实验结果对比

Table 3 Result comparison of Lorenz forecasting

方法	网络结构	RMSE (训练)	RMSE (测试)	运算时间 (s)
ALRDBN	3-3-3-1	**0.0210**	**0.0225**	**2.9**
DBN	3-3-3-1	0.0371	0.0388	3.6
CDBN	3-3-3-1	0.0208	0.0223	3.2
BPNN[23]	3-6-1	0.0700	0.0835	>10
SRNN[24]	3-6-1	0.0232	0.0302	6.7

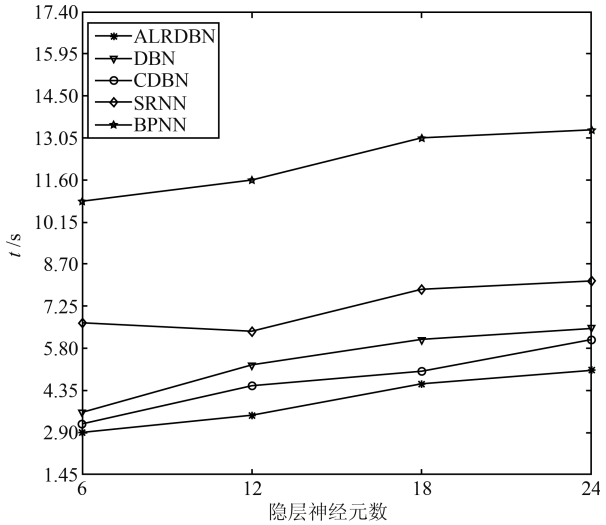


图 19 隐含层神经元数对收敛时间的影响

Fig. 19 Effect of the number of hidden neurons on convergence time

相应的, 为了更充分的验证第 2 节所确定的 α 和 β 取值范围的有效性, 在确保其他实验参数不变以及 ALRDBN 性能稳定的前提下, 将 α 和 β 取若干组不同的值并再次实验. 图 20 是不同的 α 和 β 对应的算法收敛时间, 可以看出, 当 α 和 β 在第 2 节所确定的范围内取值时算法收敛时间变化很微弱, ALRDBN 处于平稳状态. 当 α 和 β 的取值超出稳定范围时算法收敛时间出现跳跃式增加, 此时 ALRDBN 处于不稳定状态.

4 总结与展望

针对 DBN 预训练耗时长的问题, 文中将基于自适应学习率的思想引入到传统深度信念网中, 提出了自适应学习率深度信念网的设计方法, 并通过网络性能分析证明了网络的稳定性和有效性, 同时给出了自适应学习率增大和减小系数的取值范围. 最后, 通过大量实验结果可以看出, 该方法在提高算法收敛速度方面效果明显.

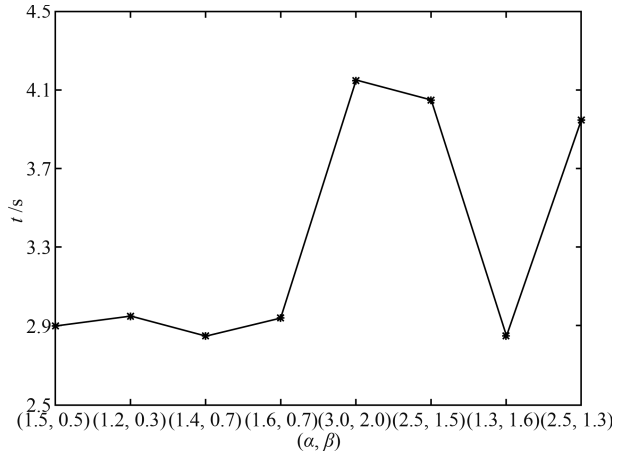


图 20 α 和 β 对收敛时间的影响

Fig. 20 Influence of α and β on convergence time

同时, 文中也存在两方面的问题: 1) 在 DBN 隐含层个数和每层神经元个数的确定方面尚没有行之有效的统一理论方法, 目前都是凭借经验设定, 文中也是通过经验确定隐含层层数和每层神经元个数, 这对 DBN 的应用效果带来很大的随机性. 2) 要想加快 DBN 的收敛速度, 仅凭在学习算法上的改进 (自适应学习率) 是远远不够的, 应该在考虑经济成本的基础之上试图更新硬件设备的性能来加速算法收敛. 因此, 如何解决这两个方面存在的问题将是 DBN 研究的一个热点方向, 也是下一步工作的重点方向.

References

1 Bengio Y H, Delalleau O. On the expressive power of deep Architectures. In: Proceeding of the 22nd International Conference. Berlin Heidelberg, Germany: Springer-Verlag, 2011. 18−36

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

3 Guo Xiao-Xiao, Li Cheng, Mei Qiao-Zhu. Deep learning applied to games. *Acta Automatica Sinica*, 2016, **42**(5): 676−684
(郭潇逍, 李程, 梅俏竹. 深度学习在游戏中的应用. 自动化学报, 2016, **42**(5): 676−684)

4 LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015, **521**(7553): 436−444

5 He Yu-Yao, Li Bao-Qi. A combinatory form learning rate scheduling for deep learning model. *Acta Automatica Sinica*, 2016, **42**(6): 953−958
(贺昱曜, 李宝奇. 一种组合型的深度学习模型学习率策略. 自动化学报, 2016, **42**(6): 953−958)

6 Ma Shuai, Shen Tao, Wang Rui-Qi, Lai Hua, Yu Zheng-Tao. Terahertz spectroscopic identification with deep belief network. *Spectroscopy and Spectral Analysis*, 2015, **35**(12): 3325−3329
(马帅, 沈韬, 王瑞琦, 赖华, 余正涛. 基于深层信念网络的太赫兹光谱识别. 光谱学与光谱分析, 2015, **35**(12): 3325−3329)

7 Geng Zhi-Qiang, Zhang Yi-Kang. An improved deep belief network inspired by glia chains. *Acta Automatica Sinica*, 2016, **42**(6): 943−952
(耿志强, 张怡康. 一种基于胶质细胞链的改进深度信念网络模型. 自动化学报, 2016, **42**(6): 943−952)

8 Abdel-Zaher A M, Eldeib A M. Breast cancer classification using deep belief networks. *Expert Systems with Applications*, 2016, **46**: 139−144

9 Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors. *Nature*, 1986, **323**(6088): 533−536

10 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

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

12 Lopes N, Ribeiro B. Improving convergence of restricted Boltzmann machines via a learning adaptive step size. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Lecture Notes in Computer Science*. Berlin Heidelberg: Springer, 2012. 511−518

13 Raina R, Madhavan A, Ng A Y. Large-scale deep unsupervised learning using graphics processors. In: Proceedings of the 26th Annual International Conference on Machine Learning. New York, NY, USA: ACM, 2009. 873−880

14 Ly D L, Paprotski V, Danny Y. Neural Networks on GPUs: Restricted Boltzmann Machines, Technical Report, University of Toronto, Canada, 2009.

15 Lopes N, Ribeiro B. Towards adaptive learning with improved convergence of deep belief networks on graphics processing units. *Pattern Recognition*, 2014, **47**(1): 114−127

16 Le Roux N, Bengio Y. Representational power of restricted boltzmann machines and deep belief networks. *Neural Computation*, 2008, **20**(6): 1631−1649

17 Hinton G E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002, **14**(8): 1771−1800

18 Yu X H, Chen G A, Cheng S X. Dynamic learning rate optimization of the backpropagation algorithm. *IEEE Transactions on Neural Networks*, 1995, **6**(3): 669−677

19 Magoulas G D, Vrahatis M N, Androulakis G S. Improving the convergence of the backpropagation algorithm using learning rate adaptation methods. *Neural Computation*, 1999, **11**(7): 1769−1796

20 Lee H, Ekanadham C, Ng A. Sparse deep belief net model for visual area V2. In: Proceedings of the 2008 Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2008. 873−880

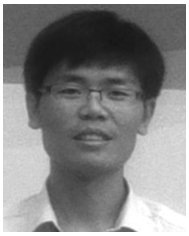
21 Ji N N, Zhang J S, Zhang C X. A sparse-response deep belief network based on rate distortion theory. *Pattern Recognition*, 2014, **47**(9): 3179−3191

22 Qiao Jun-Fei, Pan Guang-Yuan, Han Hong-Gui. Design and application of continuous deep belief network. *Acta Automatica Sinica*, 2015, **41**(12): 2138−2146
(乔俊飞, 潘广源, 韩红桂. 一种连续型深度信念网的设计与应用. 自动化学报, 2015, **41**(12): 2138−2146)

23 Chang L C, Chen P A, Chang F J. Reinforced two-step-ahead weight adjustment technique for online training of recurrent neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2012, **23**(8): 1269−1278

24 Chen Q L, Chai W, Qiao J F. A stable online self-constructing recurrent neural network. *Advances in Neural Networks − ISNN 2011*. Berlin Heidelberg: Springer, 2011, **6677**: 122−131

乔俊飞 北京工业大学教授. 主要研究方向为智能控制, 神经网络分析与设计.
E-mail: junfeq@bjut.edu.cn
(**QIAO Jun-Fei** Professor at Faculty of Information Technology, Beijing University of Technology. His research interest covers intelligent control, analysis and design of neural networks.)



王功明 北京工业大学博士研究生. 主要研究方向为深度学习, 神经网络结构设计和优化. 本文通信作者.
E-mail: xiaowangqsd@163.com
(**WANG Gong-Ming** Ph. D. candidate at Faculty of Information Technology, Beijing University of Technology. His research interest covers deep learning, analysis and design of neural networks. Corresponding author of this paper.)

李晓理 北京工业大学教授. 1997 年获得大连理工大学控制理论与工程硕士学位, 2000 年获得东北大学博士学位. 主要研究方向为多模型自适应控制, 神经网络控制.
E-mail: lixiaolibjut@bjut.edu.cn
(**LI Xiao-Li** Professor at Faculty of Information Technology, Beijing University of Technology. He received his master degree in control theory and control engineering from Dalian University of Technology in 1997, and Ph. D. degree from Northeastern University in 2000, respectively. His research interest covers multiple model adaptive control and neural network control.)

韩红桂 北京工业大学教授. 主要研究方向为污水处理工艺复杂建模与控制, 神经网络分析与设计.
E-mail: rechardhan@sina.com
(**HAN Hong-Gui** Professor at Faculty of Information Technology, Beijing University of Technology. His research interest covers modelling and control in waste water treatment process, analysis and design of neural networks.)

柴 伟 北京工业大学讲师. 主要研究方向为系统辨识和状态估计研究.
E-mail: chaiwei@bjut.edu.cn
(**CHAI Wei** Lecturer at Faculty of Information Technology, Beijing University of Technology. His research interest covers system identification and state estimation.)