基于动态 Gibbs 采样的 RBM 训练算法研究

李 飞 [1] 高晓光 [1] 万开方 [1]

摘 要 目前大部分受限玻尔兹曼机 (Restricted Boltzmann machines, RBMs) 训练算法都是以多步 Gibbs 采样为基础的采样算法. 本文针对多步 Gibbs 采样过程中出现的采样发散和训练速度过慢的问题, 首先, 对问题进行实验描述, 给出了问题的具体形式; 然后, 从马尔科夫采样的角度对多步 Gibbs 采样的收敛性质进行了理论分析, 证明了多步 Gibbs 采样在受限玻尔兹曼机训练初期较差的收敛性质是造成采样发散和训练速度过慢的主要原因; 最后, 提出了动态 Gibbs 采样算法, 给出了对比仿真实验. 实验结果表明, 动态 Gibbs 采样算法可以有效地克服采样发散的问题, 并且能够以微小的运行时间为代价获得更高的训练精度.

关键词 受限玻尔兹曼机, Gibbs 采样, 采样算法, 马尔科夫理论

引用格式 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

DOI 10.16383/j.aas.2016.c150645

Research on RBM Training Algorithm with Dynamic Gibbs Sampling

LI Fei[1] GAO Xiao-Guang[1] WAN Kai-Fang[1]

Abstract Currently, most algorithms for training restricted Boltzmann machines (RBMs) are based on the multi-step Gibbs sampling. This article focuses on the problems of sampling divergence and the low training speed associated with the multi-step Gibbs sampling process. Firstly, these problems are illustrated and described by experiments. Then, the convergence property of the Gibbs sampling procedure is theoretically analyzed from the prospective of the Markov sampling. It is proved that the poor convergence property of the multi-step Gibbs sampling is the main cause of the sampling divergence and the low training speed when training an RBM. Furthermore, a new dynamic Gibbs sampling algorithm is proposed and its simulation results are given. It has been demonstrated that the dynamic Gibbs sampling algorithm can effectively tackle the issue of sampling divergence and can achieve a higher training accuracy at a reasonable expense of computation time.

Key words Restricted Boltzmann machine (RBM), Gibbs sampling, sampling algorithm, Markov theory

Citation 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

自 2006 年 Hinton 等[1] 提出第一个深度置信网络开始, 经过十年的发展, 深度学习已逐渐成为机器学习研究领域的前沿热点. 深度置信网络[2]、深度卷积神经网络[3]、深度自动编码器[4] 等深度网络也广泛应用于机器学习的各个领域, 如图像识别、语音分析、文本分析等[5−7]. 相对于传统的机器学习网络, 深度网络取得了更好的效果, 极大地推动了技术发展水平 (State-of-the-art)[8]. 尤其在大数据背景下, 针对海量无标签数据的学习, 深度网络具有明显的优势[9].

受限玻尔兹曼机 (Restricted Boltzmann machine, RBM)[10] 是深度学习领域中的一个重要模型, 也是构成诸多深度网络的基本单元之一. 由于 RBM 较难训练, 所以在很多大数据量任务上使用较少. 但相对于其他基本模型, RBM 具备较强的理论分析优势和可解释性, 是帮助我们理解深度网络和其他基本模型内在机理的重要模型, 而且在某些特殊数据集上, RBM 可以获得更好的学习效果. 所以, 研究 RBM 仍然很有意义. RBM 具有两层结构, 在无监督学习下, 隐层单元可以对输入层单元进行抽象, 提取输入层数据的抽象特征. 当多个 RBM 或 RBM 与其他基本单元以堆栈的方式构成深度网络时, RBM 隐层单元提取到的抽象特征可以作为其他单元的输入, 继续进行特征提取. 通过这种方式, 深度网络可以提取到抽象度非常高的数据特征. 当采用逐层贪婪 (Greedy layer-wise)[11] 训练方法对深度网络进行训练时, 各个基本单元是逐一被训练的. 因此, RBM 训练的优劣将直接影响整个深度网络的性能.

收稿日期 2015-10-19 录用日期 2016-05-03
Manuscript received October 19, 2015; accepted May 3, 2016
国家自然科学基金 (61305133, 61573285) 资助
Supported by National Natural Science Foundation of China (61305133, 61573285)
本文责任编委 柯登峰
Recommended by Associate Editor KE Deng-Feng
1. 西北工业大学电子信息学院 西安 710129
1. School of Electronics and Information, Northwestern Polytechnical University, Xi′an 710129

2006 年, Hinton 等提出了对比散度 (Contrastive divergence, CD) 算法[12] 用以训练 RBM 网络. 在每次训练迭代时, CD 算法以数据样本为初始值, 通过多步 Gibbs 迭代获得目标分布的近似采样, 然后通过该近似采样来近似目标梯度, 取得了较好的效果, 是目前 RBM 训练的标准算法. 但研究表明, CD 算法对目标梯度的估计是有偏估计[13], 而且每次迭代时都需要重新启动 Gibbs 采样链, 这降低了 CD 算法的训练性能. 为此, Tieleman 等以 CD 算法为基础, 于 2008 年提出了持续对比散度 (Persistent contrastive divergence, PCD) 算法[14]. 在学习率足够小的前提下, 每次参数更新后, RBM 模型的变化不大, 可以认为 RBM 网络分布基本不变. 基于此假设, PCD 算法只运行一条独立的采样链, 以上次采样迭代的采样值作为下次采样迭代的初值继续迭代, 而不是像 CD 算法那样每次采样都以样本数据为采样初值, 取得了比 CD 算法更好的训练效果. 为了加速 PCD 算法, Tieleman 又于 2009 年提出了加速持续对比散度 (Fast persistent contrastive divergence, FPCD) 算法[15], 引入了额外的加速参数来提高训练速度. PCD 算法和 FPCD 算法虽然训练性能较 CD 算法有所提高, 但并没有从本质上提高 CD 算法的混合率[16]. 不管是 CD 算法, 还是以 CD 算法为基础的 PCD 算法、FPCD 算法, 都是通过一条 Gibbs 采样链来逼近目标分布, 对于目标分布较简单的数据, 可以取得较好的效果. 但当数据分布复杂, 尤其为多模分布时, 即目标分布函数存在多个峰值, Gibbs 采样链很容易陷入局部极小域, 导致样本不能描述数据分布的整体结构[17]. 为克服这个问题, Desjardins (2010) 等[18]、Cho (2010) 等[19]、Brakel (2012) 等[20] 等分别提出应用并行回火算法 (Parallel tempering, PT) 来训练 RBM. PT 算法并行化多条温度链, 每条温度链上进行多步 Gibbs 迭代. 高温链采样目标总体分布的结构信息, 低温链采样目标局部分布的精确信息. 不同温度链之间以一定的交换概率进行交换, 不断迭代, 最后低温链就可以精确获得目标分布的总体信息. 对于多模分布数据, PT 算法的训练效果要明显优于 CD 算法[21].

通过以上描述可知, 不管是 CD 算法还是 PT 算法, 本质上都是以 Gibbs 采样来获得关于目标分布的采样样本. 因此, Gibbs 采样性能的优劣将直接影响以上算法的训练效果. 本文研究发现, 当采用多步 Gibbs 采样时, 在训练初期会发生采样发散现象, 严重影响网络收敛速度, 而且算法运行速度较慢; 当采用单步 Gibbs 采样时, 前期网络收敛性质较好, 且算法运行速度较快, 但后期采样精度不高. 如何在前期保证良好的收敛性质, 同时在后期保证网络训练精度并提高算法运行速度, 是目前基于 Gibbs 采样的 RBM 训练算法亟需解决的问题, 从现有文献来看, 尚无人对以上问题进行研究. 因此, 本文将从马尔科夫采样理论的角度对以上问题进行分析, 并提出了动态 Gibbs 采样算法, 最后给出了仿真验证.

1 问题描述

受限玻尔兹曼机是一个马尔科夫随机场模型[22], 它具有两层结构, 如图 1 所示. 下层为输入层, 包含 m 个输入单元 v_i, 用来表示输入数据, 每个输入单元包含一个实值偏置量 a_i; 上层为隐层, 包含 n 个隐层单元 h_j, 表示受限玻尔兹曼机提取到的输入数据的特征, 每个隐层单元包含一个实值偏置 b_j. 受限玻尔兹曼机具有层内无连接, 层间全连接的特点. 即同层内各节点之间没有连线, 每个节点与相邻层所有节点全连接, 连线上有实值权重矩阵 w_{ij}. 这一性质保证了各层之间的条件独立性.

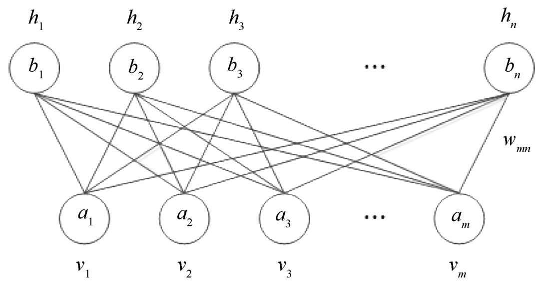


图 1 RBM 结构

Fig. 1 Configuration of RBM

本文研究二值受限玻尔兹曼机[23], 即随机变量 (V,H) 取值 $(v,h)\in\{0,1\}$. 由二值受限玻尔兹曼机定义的联合分布满足 *Gibbs* 分布 $P(v,h)=\frac{1}{Z_\theta}\mathrm{e}^{-E_\theta(v,h)}$, 其中 θ 为网络参数 $\theta=\{a_i,b_j,w_{ij}\}$, $E_\theta(v,h)$ 为网络的能量函数:

$$E_\theta(v,h)=-\sum_{i=1}^{n}\sum_{j=1}^{m}w_{ij}v_ih_j-\sum_{i=1}^{m}a_iv_i-\sum_{j=1}^{n}b_jh_j \tag{1}$$

Z_θ 为配分函数: $Z_\theta=\sum_{v,h}\mathrm{e}^{-E_\theta(v,h)}$. 输入层节点 v 的概率分布 $P(v)$ 为: $P(v)=\frac{1}{Z_\theta}\sum_h\mathrm{e}^{-E_\theta(v,h)}$. 由受限玻尔兹曼机各层之间的条件独立性可知, 当给定输入层数据时, 输出层节点取值满足如下条件概率:

$$\begin{aligned}P(h_k=1|v)=&\frac{1}{1+\exp(-b_j-\sum\limits_{i=1}^{n}w_{ij}v_i)}=\\&\mathrm{sigmoid}\left(b_j+\sum\limits_{i=1}^{n}w_{ij}v_i\right)\end{aligned} \tag{2}$$

相应地, 当输出层数据确定后, 输入层节点取值的条件概率为

$$P(h_k=1|v)=\frac{1}{1+\exp(-a_i-\sum\limits_{i=1}^{n}w_{ij}h_j)}= \\ \text{sigmoid}\left(a_i+\sum_{i=1}^{n}w_{ij}h_j\right) \tag{3}$$

给定一组训练样本 $S=\{v^1,v^2,\cdots,v^n\}$, 训练 RBM 意味着调整参数 θ, 以拟合给定的训练样本, 使得该参数下由相应 RBM 表示的概率分布尽可能地与训练数据的经验分布相符合. 本文应用最大似然估计的方法对网络参数进行估计. 这样, 训练 RBM 的目标就是最大化网络的似然函数: $L_{\theta,w}=\prod_{i=1}^{n}P(v^i)$. 为简化计算, 将其改写为对数形式: $\ln L_{\theta,w}=\sum_{i=1}^{n}\ln P(v^i)$. 进一步推导对数似然函数的参数梯度

$$\begin{aligned}
\frac{\partial\ln P(v)}{\partial a_i}&=-\sum_h P(h|v)\frac{\partial E(v,h)}{\partial a_i}+ \\
&\quad\sum_{v,h}P(v,h)\frac{\partial E(v,h)}{\partial a_i}=v_i-\sum_v P(v)v_i \\
\frac{\partial\ln P(v)}{\partial b_j}&=-\sum_h P(h|v)\frac{\partial E(v,h)}{\partial b_j}+ \\
&\quad\sum_{v,h}P(v,h)\frac{\partial E(v,h)}{\partial b_j}=P(h_i=1|v)- \\
&\quad\sum_v P(v)P(h_{i=1}|v) \\
\frac{\partial\ln P(v)}{\partial w_{ij}}&=-\sum_h P(h|v)\frac{\partial E(v,h)}{\partial w_{ij}}+ \\
&\quad\sum_{v,h}P(v,h)\frac{\partial E(v,h)}{\partial w_{ij}}=P(h_j=1|v)v_i- \\
&\quad\sum_v P(v)P(h_{j=1}|v)v_i
\end{aligned} \tag{4}$$

得到对数似然函数的参数梯度后, 可以由梯度上升法求解其最大值. 但由于数据分布 $P(v)$ 未知, 且包含配分函数 Z_θ, 因此, 无法给出梯度的解析解. 现有训练算法主要是基于采样的方法, 首先, 构造以 $P(v)$ 为平稳分布的马尔科夫链, 获得满足 $P(v)$ 分布的样本; 然后, 通过蒙特卡洛迭代来近似梯度:

$$\begin{aligned}
\nabla a_i&=v_i^{(0)}-v_i^{(k)} \\
\nabla b_j&=P(h_j=1|v^{(0)})-P(h_j=1|v^{(k)}) \\
\nabla w_{ij}&=P(h_j=1|v^{(0)})v_i^{(0)}-P(h_j=1|v^{(k)})v_i^{(k)}
\end{aligned} \tag{5}$$

其中, $v_i^{(0)}$ 为样本值, $v_i^{(k)}$ 为通过采样获得的满足 $P(v)$ 分布的样本. 最后, 参数更新方程如下:

$$\begin{aligned}
a_i&=a_i+\nabla a_i \\
b_i&=b_i+\nabla b_i \\
w_{ij}&=w_{ij}+\nabla w_{ij}
\end{aligned} \tag{6}$$

现有 RBM 训练算法, 包括 CD_k 算法、并行回火 (PT) 算法, 这两类算法都是以 Gibbs 采样为基础的, 都是通过多步 Gibbs 采样获得一定精度的目标采样, 然后分别通过其他后续操作获得最终的目标梯度近似值. CD_k 算法是 RBM 训练的主流算法, 因此, 本节以 CD_k 算法为例, 通过仿真的方式, 揭示了作为以上算法基本操作单元的 Gibbs 采样在网络训练过程中出现的问题, 研究了它对网络收敛速度和训练精度的影响.

首先给出 CD_k 算法的操作步骤:

步骤 1. 设定网络参数初值.

步骤 2. 将训练数据输入到输入层节点, 由式 (2) 对隐层节点值进行采样,

步骤 3. 根据式 (3) 对输入层节点进行采样. 再以此采样值作为输入层节点的值重复步骤 2, 这样就完成了一步 Gibbs 采样.

步骤 4. 步骤 2 和步骤 3 重复 k 次, 完成 k 步 Gibbs 采样, 即 CD_k.

步骤 5. 将步骤 4 获得的采样值带入式 (5) 中, 计算参数梯度.

步骤 6. 将步骤 5 中获得的参数梯度带入式 (6) 中, 对参数进行更新.

步骤 7. 更新训练数据, 重复步骤 2～6, 直到达到额定迭代次数.

相应的伪代码如算法 1 所示:

算法 1. CD_*k* 算法伪代码

Input: $RBM(V_1,\cdots,V_m,H_1,\cdots,H_n)$, training batch S
Output: w_{ij}, a_j and b_i for $i=1,\cdots,n,\ j=1,\cdots,m$
Init $\nabla w_{ij}=\nabla a_j=\nabla b_i=0$ for $i=1,\cdots,n, j=1,\cdots,m$
For all the $v\in S$ do
$v^{(0)}\leftarrow v$
for $t=0,\cdots,k-1$ do
    for $i=1,\cdots,n$ do sample $h_i^{(t)}\sim p(h_i|v^{(t)})$
for $j=1,\cdots,m$ do sample $v_j^{(t+1)}\sim p(v_j|h^{(t)})$
for $i=1,\cdots,n, j=1,\cdots,m$ do
$\nabla w_{ij}=p(H_i=1|v^{(0)})\cdot v_j^{(0)}-p(H_i=1|v^{(k)})\cdot v_j^{(k)}$
$\nabla a_j=v_j^{(0)}-v_j^{(k)}$
$\nabla b_i=p(H_i=1|v^{(0)})-p(H_i=0|v^{(k)})$
$w_{ij}=w_{ij}+\eta\nabla w_{ij}$
$a_i=a_i+\eta\nabla a_i$
$b_i=b_i+\eta\nabla b_i$
End For

其中, a 为可见层偏置向量, b 为隐层偏置向量, w 为网络权值矩阵, η 为学习率.

1.1 问题实验描述

1) 实验设计

本文采用的数据集是 MNIST 数据集, 它是二值手写数据集, 也是目前训练 RBM 网络的标准数据集. 它总共包含 60 000 个训练样本和 10 000 个测试样本, 每个样本是一幅 28 像素 ×28 像素的灰度图. 所采用的 RBM 网络有 784×500 个节点, 输入层有 784 个可见单元, 对应灰度图的 784 个像素点; 输出层有 500 个隐层节点, 这是目前实验显示的训练效果较好的隐层节点数目. 具体的网络参数初始值设定如表 1.

表 1 网络参数初值

Table 1 Initial value of parameters

网络参数	初始值
a	$zeros(1,784)$
b	$zeros(1,500)$
w	$0.1\times randn(784,500)$
η	0.1

本文设计了 6 组对比实验, 用 60 000 个训练样本对 RBM 进行训练, 分别迭代 1 000 次, 如表 2 所示. 其中 CD_k 表示进行 k 步 Gibbs 迭代. 用于显示的样本数据的原始图片如图 2 所示. 实验结束后, 我们比较了各组实验的重构误差, 并给出了最终的误差图.

表 2 实验分组

Table 2 Experimental grouping

数据集	算法	迭代次数
MNIST	CD_1	1 000
MNIST	CD_5	1 000
MNIST	CD_10	1 000
MNIST	CD_100	1 000
MNIST	CD_500	1 000
MNIST	CD_1000	1 000

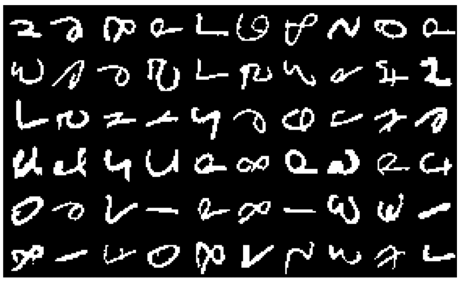


图 2 原始数据灰度图

Fig. 2 Gray image of initial data

2) 仿真结果图 3 表示整个迭代过程中各组 CD 算法的重构误差图, 图 4 给出了各组实验的训练时间, 图 5～图 10 分别给出了各组实验的采样灰度图.

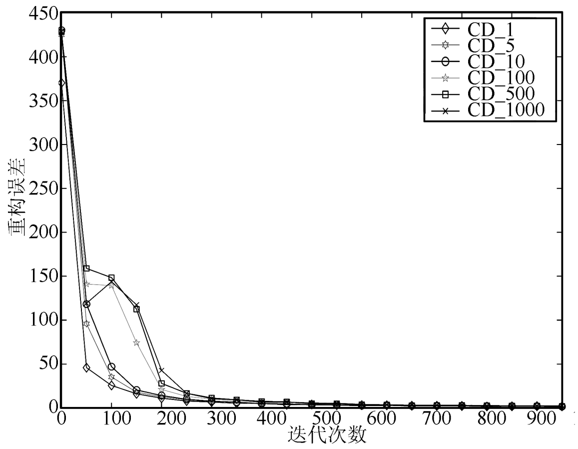


图 3 重构误差图

Fig. 3 Reconstruction error diagram

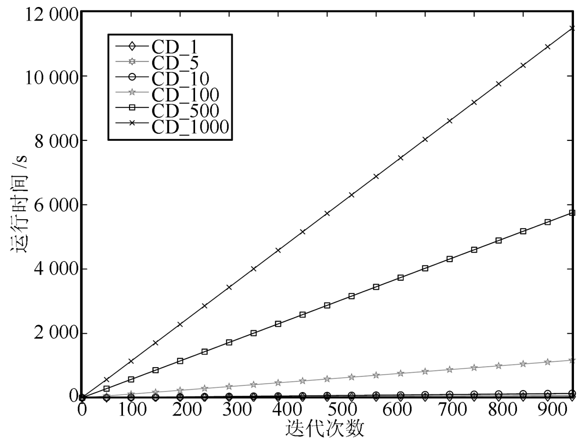


图 4 运行时间图

Fig. 4 Runtime diagram

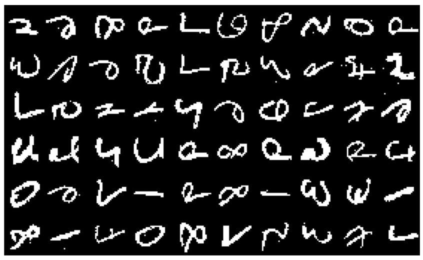


图 5 CD_1 采样灰度图

Fig. 5 Gray image of CD_1 sampling

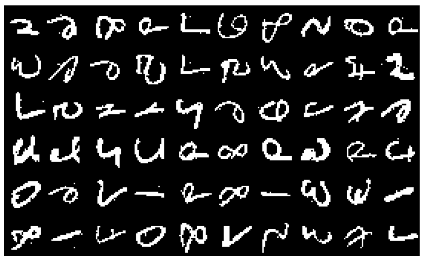


图 6 CD_5 采样灰度图

Fig. 6 Gray image of CD_5 sampling

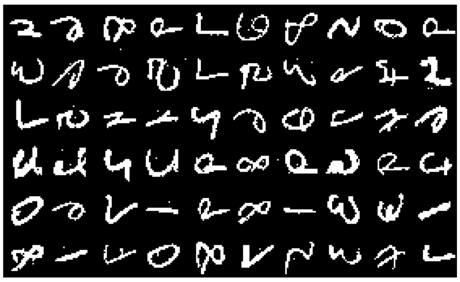


图 7　CD_10 采样灰度图

Fig. 7　Gray image of CD_10 sampling

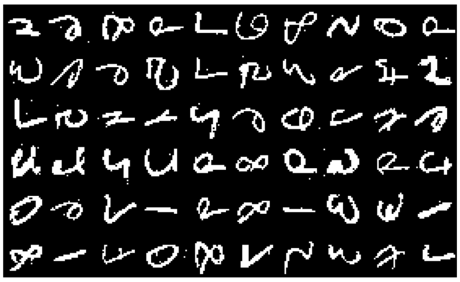


图 8　CD_100 采样灰度图

Fig. 8　Gray image of CD_100 sampling

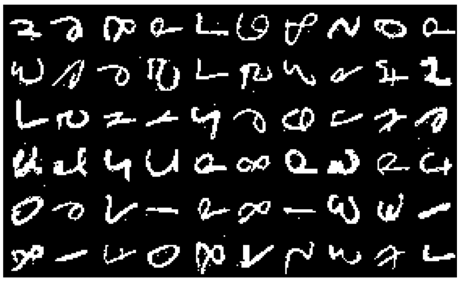


图 9　CD_500 采样灰度图

Fig. 9　Gray image of CD_500 sampling

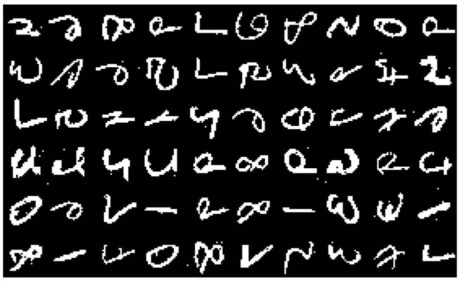


图 10　CD_1000 采样灰度图

Fig. 10　Gray image of CD_1000 sampling

1.2　问题归纳描述

上节实验给出了 CD 算法在不同 Gibbs 采样步数下的仿真图, 可以看出, 当 RBM 网络采用多步 Gibbs 算法进行采样迭代时, 会出现如下问题:

问题 1. 训练初始阶段, 得到的每幅重构采样图几乎完全相同.

如图 11、图 12 所示, 在训练初始阶段, 多步 Gibbs 采样出现了各组采样数据同分布的现象, 这表明各组样本几乎完全相同, 这与事实相左. 在训练初期, 大约 0～100 次迭代期间, 这种现象持续存在.

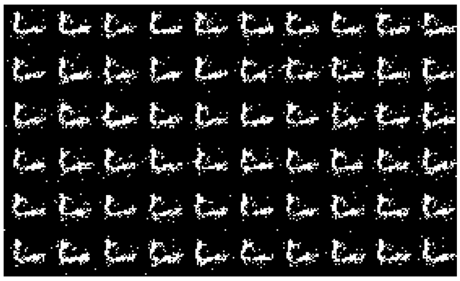


图 11　CD_500 采样灰度图

Fig. 11　Gray image of CD_500 sampling

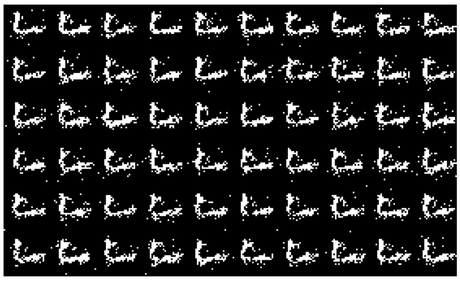


图 12　CD_1000 采样灰度图

Fig. 12　Gray image of CD_1000 sampling

问题 2. 采样误差分布集中, 在批量训练时, 存在全 0 全 1 现象.

如图 13、图 14 所示, 当进行多步 Gibbs 采样时, 出现了误差分布集中的现象: 有些样本采样几乎全为 1, 而其他的样本采样几乎全为 0. 由仿真实验可知, 在 0～100 次迭代期间, 这种现象在迭代初期持续存在.

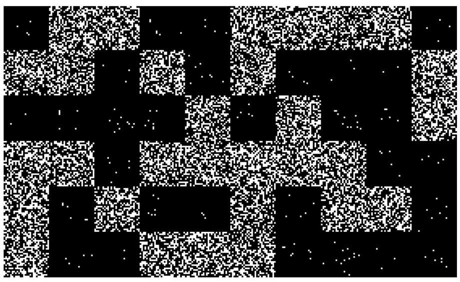


图 13　CD_500 采样灰度图

Fig. 13　Gray image of CD_500 sampling

问题 3. 一步 Gibbs 采样初始误差小, 训练速度快, 但后期训练精度低; 多步 Gibbs 采样初始误差大, 训练速度慢, 但后期训练精高.

如图 15、图 16 所示, 只进行一步 Gibbs 采样的 CD_1 算法在开始时训练误差较小, 很快便收敛

到较好值, 但训练后期精度不如 CD_10 等进行多步 Gibbs 迭代的 CD 算法; 进行多步 Gibbs 采样的 CD_k 迭代算法, 在训练初期误差较大, 且不断振荡, 而且训练时间较慢, 但到训练后期, 它们可以达到极高的精度.

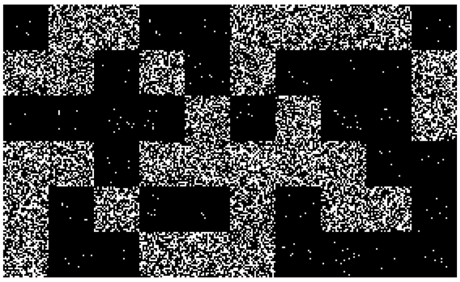


图 14　CD_1000 采样灰度图

Fig. 14　Gray image of CD_1000 sampling

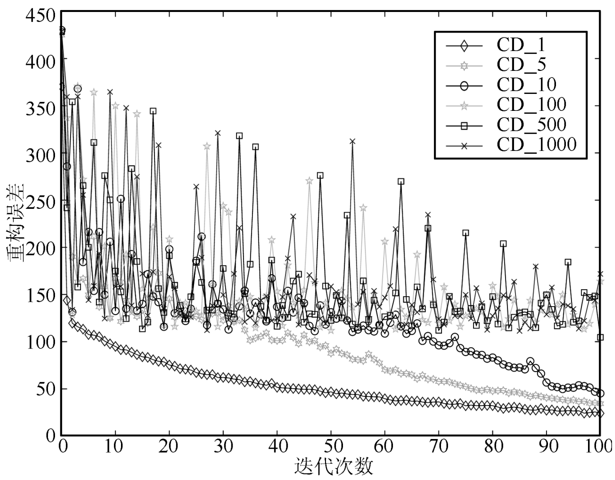


图 15　采样误差局部放大图

Fig. 15　Local enlarged drawing of reconstruction error in initial phase

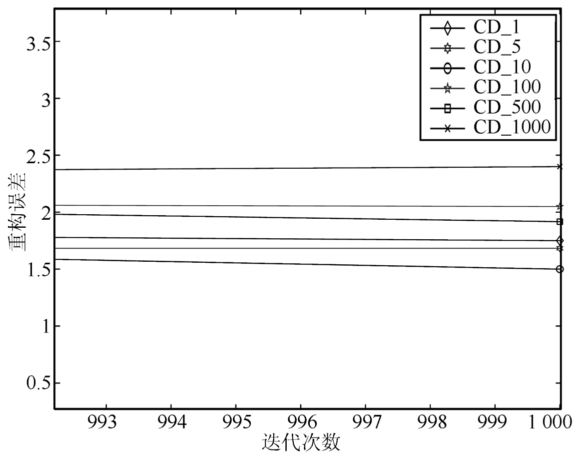


图 16　采样误差局部放大图

Fig. 16　Local enlarged drawing of reconstruction error in later stage

以上实验表明, CD 算法虽然对 RBM 具有良好的训练能力, 但 Gibbs 采样的步数对训练性能造成了明显的影响. 我们将在下节研究这种影响, 并对以上问题给出理论分析.

2　Gibbs 采样误差的理论分析

Gibbs 采样是马尔科夫链蒙特卡洛 (Markov chain Monte Carlo, MCMC) 采样算法的一种. 在 RBM 训练中, 它的转移核是 Sigmoid 函数. 隐层节点和输入层节点交替采样, 公式如下:

$$\begin{aligned} P(h_j=1|V) &= \text{sigmoid}(b_j+\sum_{i=1}^{n} w_{i,j}v_i) \\ P(v_i=1|H) &= \text{sigmoid}(a_i+\sum_{i=1}^{n} w_{i,j}h_j) \end{aligned} \tag{7}$$

由马尔科夫链收敛定理可知, 当 $n\rightarrow+\infty$ 时, Gibbs 采样链会收敛到平衡分布, 即:

$$\pi_i(x)=\pi_{i-1}(x)P=\pi_0P^n \tag{8}$$

其中, $\pi(x)$ 为样本 x 的平衡分布. 同时, 由细致平衡准则可得:

$$\pi(i)P_{ij}=\pi(j)P_{ji},\ \ \forall i,j \tag{9}$$

即 Gibbs 采样的平稳分布与迭代初始值无关, 只与转移概率有关. 由上面给出的 RBM 交替采样概率公式可知, 当用 Gibbs 采样对 RBM 进行采样训练时, 其平稳分布是网络参数的函数:

$$\pi(x)=f(a,b,w) \tag{10}$$

从这个角度讲, 训练 RBM 的目的就是调节网络参数, 使由网络参数确定的平稳分布等于样本的真实分布.

基于以上描述, 下面对第 2 节中提出的问题给出理论解释.

问题 1. 训练初始阶段, 得到的每幅重构采样图几乎完全相同.

初始时刻, 网络参数初值相同, 在早期迭代过程中, 网络参数值的变动也不大, 满足如下公式:

$$\begin{cases} a_i-a_j<\varepsilon \\ b_i-b_j<\varepsilon \\ w_i=w_j \end{cases} \tag{11}$$

ε 为一极小正值. 由网络参数决定的平稳分布也近乎相同:

$$\begin{aligned} &f(a_i,b_i,w)\approx f(a_j,b_j,w)\Rightarrow \\ &\quad\pi(x_i)\approx\pi(x_j) \end{aligned} \tag{12}$$

即各样本的平稳分布相等. 因此, 当进行多步 Gibbs 采样时, 各训练样本的采样样本逐渐收敛到相同的平稳分布, 这时就出现了问题 1 描述的现象, 各样本的重构采样图几乎完全相同.

问题 2. 采样误差分布集中, 在批量训练时, 存在全 0 全 1 现象.

由上一部分分析可知, 在训练初期, 网络参数改变不大, 由 RBM 参数决定的平衡分布几乎同构, 即各采样概率收敛到相同平衡分布值. 上述对比实验中, 网络参数的初始值为 $\theta=(a,b,w)=(0,0,0.1)$, 此时网络平衡分布收敛在 0.5 附近, 样本数据的收敛概率将在 0.5 附近浮动, 即一部分样本的采样概率略小于 0.5, 另一部分样本的采样概率略大于 0.5, 即:

$$\begin{aligned}&\pi(\theta)\to 0.5\\&p(v_i|H)=0.5+\varepsilon\\&p(v_{n-i}|H)=0.5-\varepsilon\end{aligned}\tag{13}$$

其中, ε 为一极小正值. 这时基于随机数对样本进行采样, 一部分样本的采样值将全为 0, 另一部分的采样值将全为 1, 即全 0 全 1 现象.

问题 3. 一步 Gibbs 采样初始误差小, 训练速度快, 但后期训练精度低; 多步 Gibbs 采样初始误差大, 训练速度慢, 但后期训练精高.

设网络参数期望值为 $\hat{\theta}=\left(\hat{a},\hat{b},\hat{w}\right)$, 它代表参数的真实值: 设网络参数实际值为 $\theta=(a,b,w)$, 这是我们在训练网络过程中, 网络参数的实际值, 训练的目标就是使网络参数实际值逐渐逼近其真实值. 定义网络参数差 $\Delta\theta(\Delta a,\Delta b,\Delta w)$:

$$\begin{aligned}&\Delta a=\hat{a}-a\\&\Delta b=\hat{b}-b\\&\Delta w=\hat{w}-w\end{aligned}\tag{14}$$

在网络训练早期, 网络参数差较大, 由网络参数定义的平稳分布与真实分布相差也较大, 即 $\Delta\pi=|\pi_{\hat{\theta}}(x)-\pi_{\theta}(x)|\gg 0$. 此时, 如果对样本进行多步迭代采样, 采样样本将偏离真实分布, 从而不能收敛到真实分布, 而是收敛到与真实分布相差较大的其他分布. 因此, 在迭代初期, CD_1000、CD_500 等算法的采样误差非常大, 而且运行时间较长. 而 CD_1 算法由于只进行了一次采样迭代, 不仅运行速度加快, 而且由于采样样本的分布没有偏离真实分布太多, 使得这时候的 CD_1 算法的采样误差非常小. 由实验可知, 此时采样误差的大小关系为: CD_1 < CD_5 < CD_10 < CD_100 < CD_500 < CD_1000. 到了网络训练后期, 由于网络参数差非常小, 网络参数的实际值已经非常接近真实值, 这时候进行多步 Gibbs 迭代能很好地逼近样本真实分布, 所以这一阶段, CD_k 算法的采样精度要比 CD_1 高. 但由于网络参数差一直存在, 所以, Gibbs 迭代步数也不宜过高, 如实验所示, CD_1000 在采样到最后, 采样误差仍高于 CD_10.

3 动态 Gibbs 采样

在现有以 Gibbs 采样为基础的 RBM 训练算法中, Gibbs 采样的采样步数多为固定值, 即在整个训练过程中, 每次迭代采样时都进行固定步数的 Gibbs 采样, 这样就难以兼顾训练精度和训练速度这两个训练指标. 当进行多步 Gibbs 采样时, 容易在训练前期发生误差发散的现象, 且算法运行时间较长; 一步 Gibbs 采样算法运行较快, 但后期训练精度不高, 基于此, 本文提出了动态 Gibbs 采样 (Dynamic Gibbs sampling, DGS) 算法.

定义 1. 动态 Gibbs 采样是指在迭代训练过程中的不同阶段, 根据网络的训练误差, 动态地调整 Gibbs 采样的步数, 以达到最优训练效果.

通过上节分析可知, 在网络训练初期, 网络参数几乎相等, 各样本的平稳分布也近乎相等, 而且网络参数差较大, 样本的平稳分布与真实分布相差也较大, 因此, 这一阶段应尽量减少采样次数, 克服多步 Gibbs 采样引起的误差发散, 提高训练速度, 使网络参数尽快逼近真实值; 当网络参数逼近真实值时, 此时应加大采样迭代次数, 提高训练精度.

基于以上定义和描述, DGS 算法的操作步骤如下:

步骤 1. 设定网络参数初值和动态策略 M.

步骤 2. 在 $1\sim m_1$ 迭代范围内, 设置 Gibbs 采样步数 $k_1=Gibbs\_N_1$.

步骤 3. 将训练数据输入到输入层节点, 由式 (2) 对隐层节点值进行采样.

步骤 4. 根据式 (3) 对输入层节点进行采样. 再以此采样值作为输入层节点的值重复步骤 3, 这样就完成了一步 Gibbs 采样.

步骤 5. 步骤 3 和步骤 4 重复 k_1 次, 完成 k_1 步 Gibbs 采样.

步骤 6. 将步骤 5 获得的采样值带入式 (5) 中, 计算参数梯度.

步骤 7. 将步骤 6 中获得的参数梯度带入式 (6) 中, 对参数进行更新.

步骤 8. 更新训练数据, 重复步骤 3 到步骤 7, 直到迭代次数达到 m_1.

步骤 9. 在 $m_1\sim m_2$ 迭代范围内, 设置 Gibbs 采样步数 $k_2=Gibbs\_N_2$.

步骤 10. 重复步骤 3 到步骤 8, 直到迭代次数达到 m_2.

步骤 11. 在 $m_2 \sim Iter$ 迭代范围内, 设置 Gibbs 采样步数 $k_3 = Gibbs\_N_3$.

步骤 12. 重复步骤 3 到步骤 8, 直到迭代次数达到最大迭代次数 $Iter$.

相应的伪代码如算法 2 所示.

算法 2. DGS 算法伪代码

Input: $RBM(v_1, v_2, v_3, \cdots, v_n, h_1, h_2, h_3, \cdots, h_m)$, training batch S
Output: w_{ij}, a_i and b_j for $i = 1, \cdots, n,\ j = 1, \cdots, m$
Init: $\nabla w_{ij} = \nabla a_j = \nabla b_i = 0$ for $i = 1, \cdots, n, j = 1, \cdots, m$
For all the S do
for $iter = 1 : m_1$ do
for $t = 0, \cdots, k-1$ do $Gibbs\_N_1$
for $i = 1, \cdots, n$ do sample $h_i^{(t)} \sim p(h_i|v^{(t)})$
for $j = 1, \cdots, m$ do sample $v_j^{(t+1)} \sim p(v_j|h^{(t)})$
for $iter = m_1 : m_2$ do
for $t = 0, \cdots, k-1$ do $Gibbs\_N_2$
for $i = 1, \cdots, n$ do sample $h_i^{(t)} \sim p(h_i|v^{(t)})$
for $j = 1, \cdots, m$ do sample $v_j^{(t+1)} \sim p(v_j|h^{(t)})$
for $iter = m_2 : Iter$ do
for $t = 0, \cdots, k-1$ do $Gibbs\_N_3$
for $i = 1, \cdots, n$ do sample $h_i^{(t)} \sim p(h_i|v^{(t)})$
for $j = 1, \cdots, m$ do sample $v_j^{(t+1)} \sim p(v_j|h^{(t)})$
for $i = 1, \cdots, n, j = 1, \cdots, m$ do
$\nabla w_{ij} = p(H_i = 1|v^{(0)}) \cdot v_j^{(0)} - p(H_i = 1|v^{(k)}) \cdot v_j^{(k)}$
$\nabla a_j = v_j^{(0)} - v_j^{(k)}$
$\nabla b_i = p(H_i = 1|v^{(0)}) - p(H_i = 0|v^{(k)})$
$w_{ij} = w_{ij} + \eta \nabla w_{ij}$
$a_i = a_i + \eta \nabla a_i$
$b_i = b_i + \eta \nabla b_i$
End For

其中, $M = (m_1, m_2)$ 为动态策略, 且满足 $m_2 > m_1$. $Iter$ 为总的迭代次数, $iter$ 为当前迭代次数. $Gibbs\_N_i$ 为 Gibbs 采样, N_i 表示采样次数, 且满足 $N_n > N_{n-1}$. 其中 Gibbs 采样次数 N 与网络训练迭代次数 M 之间的大致关系如下:

$$\begin{aligned} &Gibbs\_N_1 = 1 && 若\ \ iter \in (1 \sim m_1) \\ &Gibbs\_N_2 = 2 \sim 10 && 若\ \ iter \in (m_1 \sim m_2) \\ &Gibbs\_N_3 > 10 && 若\ \ iter \in (m_2 \sim Iter) \end{aligned} \tag{15}$$

4 仿真实验

本节设计了 7 组对比实验, 第 1～6 组实验采用固定 Gibbs 采样步数的 CD_k 算法进行训练仿真, 第 6 组实验用 DGS 算法对网络进行训练仿真, 如表 3 所示. 两组实验使用相同的数据集 MNIST, 网络结构相同, 网络参数初始值相同, 如表 4 所示. 本文设计的动态采样策略如表 5 所示. 下面给出仿真实验结果和分析.

表 3 实验分组

Table 3 Experimental grouping

数据集	训练算法	$Iter$
MNIST	CD_1	1 000
MNIST	CD_5	1 000
MNIST	CD_10	1 000
MNIST	CD_100	1 000
MNIST	CD_500	1 000
MNIST	CD_1000	1 000
MNIST	DGS	1 000

表 4 网络参数初值

Table 4 Initial values of parameters

算法参数	CD_k	DGS
a	$zeros(1, 784)$	$zeros(1, 784)$
b	$zeros(1, 500)$	$zeros(1, 500)$
w	$0.1 \times randn(784, 500)$	$0.1 \times randn(784, 500)$
η	0.1	0.1
V	784	784
H	500	500

表 5 DGS 迭代策略

Table 5 Iterative strategy of DGS

M	$Gibbs\_N$
$(1 : m_1) = (1 : 300)$	$Gibbs\_N_1 = 1$
$(m_1 : m_2) = (300 : 900)$	$Gibbs\_N_2 = 5$
$(m_2 : Iter) = (900 : 1\,000)$	$Gibbs\_N_3 = 10$

4.1 重构误差对比分析

图 17 给出了所有算法的重构误差对比图. 对比结果显示, 本文设计的 DGS 算法可以很好地训练 RBM 网络, 从而证明了本文算法的有效性.

在迭代初期, DGS 算法只进行一次 Gibbs 采样迭代, 避免了采样发散, 从而迅速收敛到较好的值, 由误差对比图初始阶段的局部放大图 (图 18) 可以看出, 此时误差满足:

$$\begin{aligned} \mathrm{DGS} = \mathrm{CD\_1} > \mathrm{CD\_5} > \mathrm{CD\_10} > \mathrm{CD\_100} > \\ \mathrm{CD\_500} > \mathrm{CD\_1000} \end{aligned} \tag{16}$$

在迭代后期, 网络参数值已非常接近真实值, 此时 DGS 逐步增大了 Gibbs 采样的迭代步数, 获得了采样精度更高的目标样本, 最终获得了更高的训练精度, 即:

$$\begin{aligned} \mathrm{DGS} > \mathrm{CD\_10} > \mathrm{CD\_5} > \mathrm{CD\_1} > \mathrm{CD\_100} > \\ \mathrm{CD\_500} > \mathrm{CD\_1000} \end{aligned} \tag{17}$$

如图 19 所示.

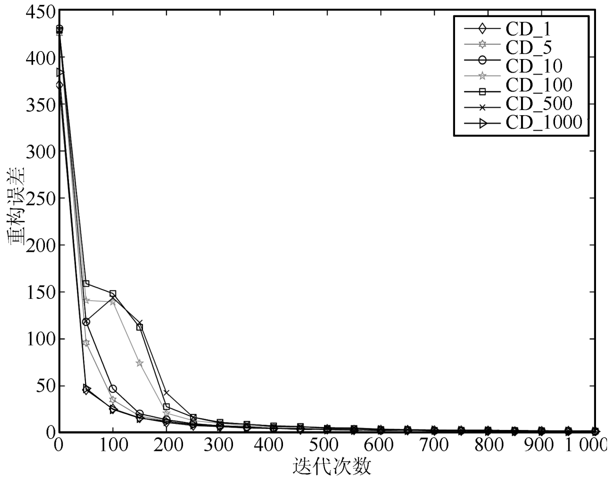


图 17　重构误差对比图

Fig. 17　Contrast of reconstruction error

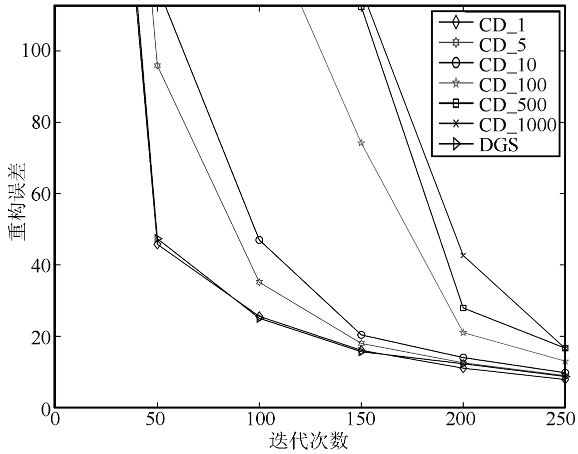


图 18　训练初期局部放大图

Fig. 18　Local enlarged drawing of reconstruction error in initial phase

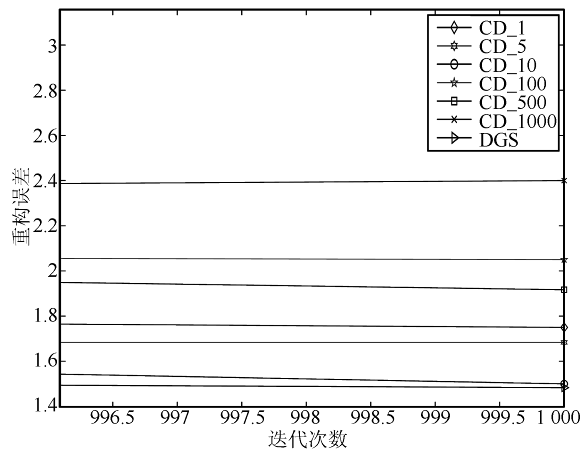


图 19　训练后期局部放大图

Fig. 19　Local enlarged drawing of reconstruction error in later stage

4.2　运行时间对比分析

图 20 给出了所有算法的运行时间对比图. 从图中可以看出, 在整个训练过程中, DGS 算法、CD_1 算法、CD_5 算法和 CD_10 算法的运行速度都明显比其他算法快. 因此, 下面根据本文设计的动态策略, 对各个迭代区间内这 4 种算法的运行速度进行分析:

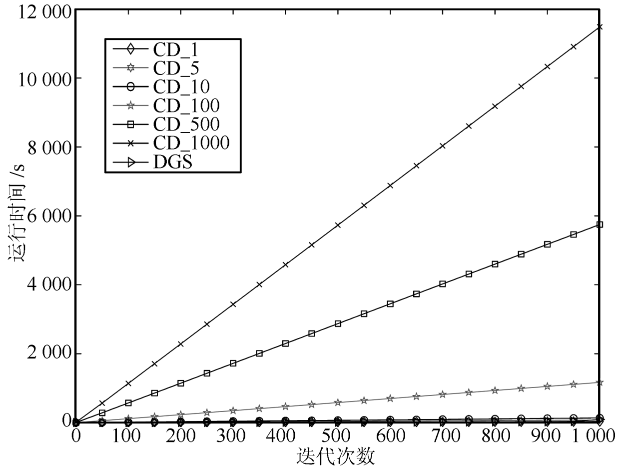


图 20　运行时间对比图

Fig. 20　Contrast of runtime

在 1 ~ 300 迭代范围内, DGS 算法的 Gibbs 采样步数 k 设为 1, 与 CD_1 算法相同. 所以, 此时的 DGS 算法的运行速度与 CD_1 相同, 且快于其他两种算法, 如图 21 所示.

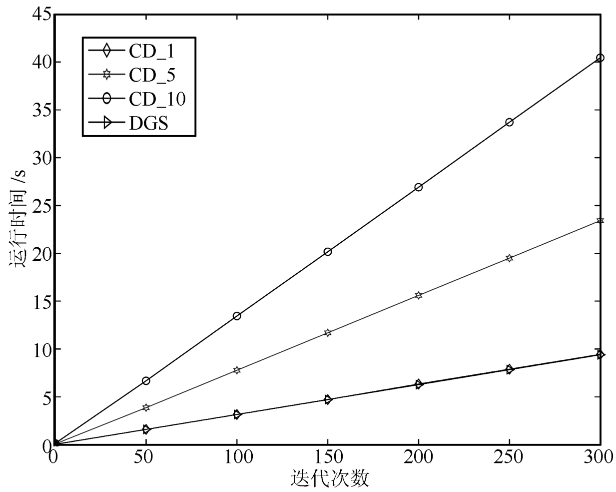


图 21　运行时间对比图

Fig. 21　Contrast of runtime

在 300 ~ 900 迭代范围内, DGS 算法的 Gibbs 采样步数 k 设为 5. 由图 22 可以看出, 此时 DGS 算法的运行速度逐渐放缓, 运行时间明显上升, 逐渐大于 CD_1 算法.

在 900 ~ 1 000 迭代范围内, DGS 算法的 Gibbs 采样步数 k 设为 10. 所以, 这个时期的 DGS 运行时间持续放缓. 但从图 23 中可以看出, 即便到了训练后期, DGS 算法的运行时间仍然小于 CD_5 算法和其他 CD_k ($k > 5$) 算法. 这说明, DGS 算法在后期提高训练精度的同时, 只付出了微小的时间代价.

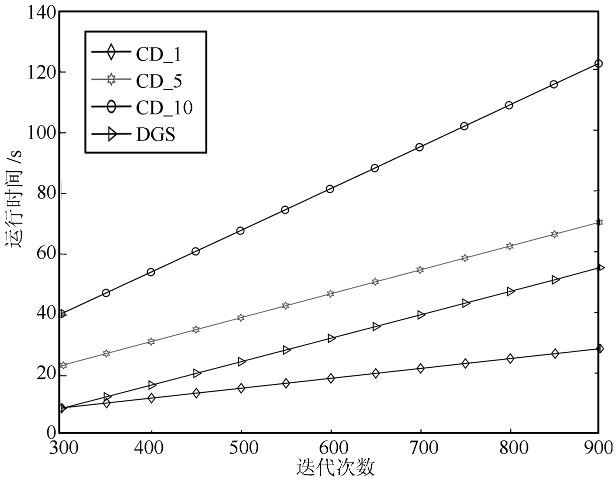


图 22 运行时间对比图

Fig. 22 Contrast of runtime

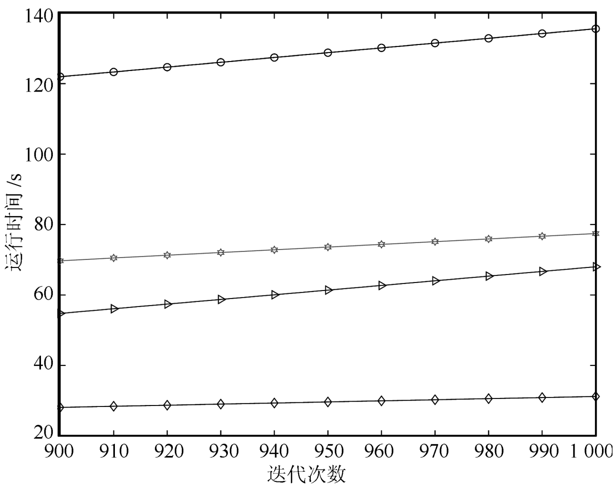


图 23 运行时间对比图

Fig. 23 Contrast of runtime

4.3 采样效果图

图 24～图 28 分别给出了 DGS 算法在不同迭代次数下的采样重构图. 对比图 11、图 12, 可以看出, DGS 在训练迭代 50 次以内就可以很好地重构输入样本, 而且没有出现全 0 全 1 现象和采样图同构现象, 从而克服了第 2.2 节问题 1 和问题 2 中描述的问题.

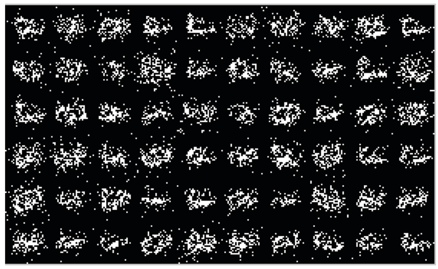


图 24 DGS 迭代 10 次采样灰度图

Fig. 24 Gray image of DGS by 10 iterations

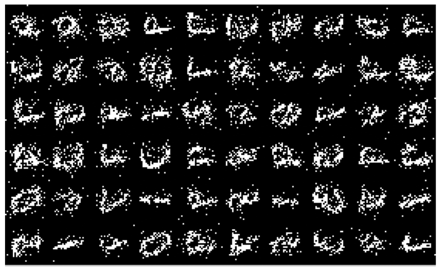


图 25 DGS 迭代 20 次采样灰度图

Fig. 25 Gray image of DGS by 20 iterations

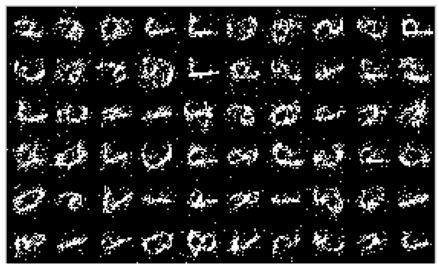


图 26 DGS 迭代 30 次采样灰度图

Fig. 26 Gray image of DGS by 30 iterations

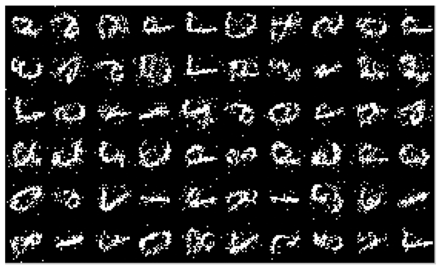


图 27 DGS 迭代 40 次采样灰度图

Fig. 27 Gray image of DGS by 40 iterations

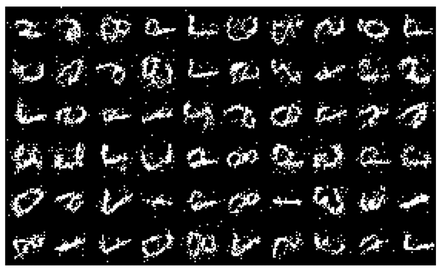


图 28 DGS 迭代 50 次采样灰度图

Fig. 28 Gray image of DGS by 50 iterations

图 29 显示了 DGS 训练结束后的重构灰度图, 图中几乎没有噪点. 可见, 采用 DGS 算法训练网络可以获得更高的训练精度, 从而解决了第 2.2 节中问题 3 描述的问题.

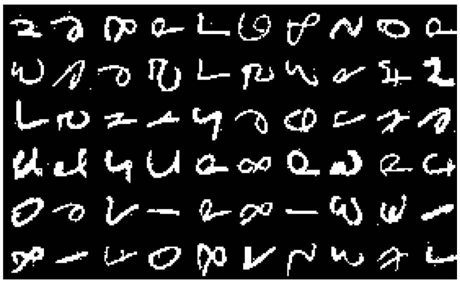


图 29　DGS 重构灰度图

Fig. 29　Gray image of DGS

综上所述, 本文设计的 DGS 算法在训练初期克服了多步 Gibbs 采样发散的缺点, 在训练后期获得更高的精度, 而且在保证收敛精度的情况下大幅度提高了训练速度, 获得了较好的效果.

5　总结

本文首先通过仿真实验, 给出了现有基于 Gibbs 采样的 RBM 训练算法在训练初期误差发散和后期训练精度不高等问题的具体描述, 然后从马尔科夫采样理论的角度对 Gibbs 采样误差进行理论分析. 证明在 RBM 网络下, 多步 Gibbs 采样较差的收敛性质是导致前期采样发散和算法运行速度较低的主要原因; 单步 Gibbs 采样是造成后期训练精度不高的主要原因. 基于此, 本文提出了动态 Gibbs 采样算法, 并给出了验证实验. 实验表明, 本文提出的动态 Gibbs 采样算法在训练初期克服了多步 Gibbs 采样引起的误差发散, 后期克服了单步 Gibbs 采样带来的训练精度低的问题, 同时提高了训练速度, 以上特点可以弥补现有以 Gibbs 采样为基础的 RBM 训练算法的不足.

关于 Gibbs 采样步数、训练迭代次数与训练精度之间的关系, 本文在理论分析部分只给出了定性分析; 在动态 Gibbs 采样算法设计阶段, 本文只是根据实验分析, 给出 Gibbs 采样步数和训练迭代次数之间的经验区间. Gibbs 采样步数、训练迭代次数以及网络训练精度之间是否存在精确的数学关系, 如果存在, 其数学模型如何构建. 以上问题仍有待进一步研究.

References

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

2 Le Roux N, Heess N, Shotton J, Winn J. Learning a generative model of images by factoring appearance and shape. *Neural Computation*, 2011, **23**(3): 593−650

3 Su Lian-Cheng, Zhu Feng. Design of a novel omnidirectional stereo vision system. *Acta Automatica Sinica*, 2006, **32**(1): 67−72
(苏连成, 朱枫. 一种新的全向立体视觉系统的设计. 自动化学报, 2006, **32**(1): 67−72)

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

5 Deng L, Abdel-Hamid O, Yu D. A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In: Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Vancouver, BC: IEEE, 2013. 6669−6673

6 Deng L. Design and learning of output representations for speech recognition. In: Proceedings of the Neural Information Processing Systems (NIPS) Workshop on Learning Output Representations [Online], available: http://research.microsoft.com/apps/pubs/default.aspx?id=204702, July 14, 2015

7 Chet C C, Eswaran C. Reconstruction and recognition of face and digit images using autoencoders. *Neural Computing and Applications*, 2010, **19**(7): 1069−1079

8 Deng L, Hinton G, Kingsbury B. New types of deep neural network learning for speech recognition and related applications: an overview. In: Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Vancouver, BC: IEEE, 2013. 8599−8603

9 Erhan D, Courville A, Bengio Y, Vincent P. Why does unsupervised pre-training help deep learning? In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010). Sardinia, Italy, 2010. 201−208

10 Salakhutdinov R, Hinton G. Deep Boltzmann machines. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS 2009). Florida, USA, 2009. 448−455

11 Swersky K, Chen B, Marlin B, de Freitas N. A tutorial on stochastic approximation algorithms for training restricted Boltzmann machines and deep belief nets. In: Proceedings of the 2010 Information Theory and Applications Workshop (ITA). San Diego, CA: IEEE, 2010. 1−10

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

13 Fischer A, Igel C. Bounding the bias of contrastive divergence learning. *Neural Computation*, 2011, **23**(3): 664−673

14 Tieleman T. Training restricted Boltzmann machines using approximations to the likelihood gradient. In: Proceedings of the 25th International Conference on Machine Learning (ICML). New York: ACM, 2008. 1064−1071

15 Tieleman T, Hinton G E. Using fast weights to improve persistent contrastive divergence. In: Proceedings of the 26th Annual International Conference on Machine Learning (ICML). New York: ACM, 2009. 1033−1040

16 Sutskever I, Tieleman T. On the convergence properties of contrastive divergence. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010). Sardinia, Italy, 2010. 789−795

17 Fischer A, Igel C. Parallel tempering, importance sampling, and restricted Boltzmann machines. In: Proceedings of 5th Workshop on Theory of Randomized Search Heuristics (ThRaSH), [Online], available: http://www2.imm.dtu.dk/projects/thrash-workshop/schedule.php, August 20, 2015

18 Desjardins G, Courville A, Bengio Y. Adaptive parallel tempering for stochastic maximum likelihood learning of RBMs. In: Proceedings of NIPS 2010 Workshop on Deep Learning and Unsupervised Feature Learning. Granada, Spain, 2010.

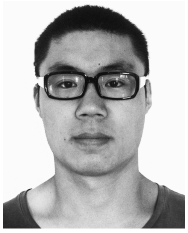
19 Cho K, Raiko T, Ilin A. Parallel tempering is efficient for learning restricted Boltzmann machines. In: Proceedings of the WCCI 2010 IEEE World Congress on Computational Intelligence. Barcelona, Spain: IEEE, 2010. 3246−3253

20 Brakel P, Dieleman S, Schrauwen B. Training restricted Boltzmann machines with multi-tempering: harnessing parallelization. In: Proceedings of the 22nd International Conference on Artificial Neural Networks. Lausanne, Switzerland: Springer, 2012. 92−99

21 Desjardins G, Courville A, Bengio Y, Vincent P, Delalleau O. Tempered Markov chain Monte Carlo for training of restricted Boltzmann machines. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010). Sardinia, Italy, 2010. 145−152

22 Fischer A, Igel C. Training restricted Boltzmann machines: an introduction. *Pattern Recognition*, 2014, **47**(1): 25−39

23 Hinton G E. A practical guide to training restricted Boltzmann machines. *Neural Networks: Tricks of the Trade* (2nd Edition). Berlin Heidelberg: Springer, 2012. 599−619



李 飞 西北工业大学电子信息学院博士研究生. 2011 年获得西北工业大学系统工程专业学士学位. 主要研究方向为机器学习和深度学习.
E-mail: nwpulf@mail.nwpu.edu.cn
(**LI Fei** Ph. D. candidate at the School of Electronics and Information, Northwestern Polytechnical University. He received his bachelor degree in system engineering from Northwestern Polytechnical University in 2011. His research interest covers machine learning and deep learning.)

高晓光 西北工业大学电子信息学院教授. 1989 年获得西北工业大学飞行器导航与控制系统博士学位. 主要研究方向为贝叶斯和航空火力控制. 本文通信作者. E-mail: cxg2012@nwpu.edu.cn
(**GAO Xiao-Guang** Professor at the School of Electronics and Information, Northwestern Polytechnical University. She received her Ph. D. degree in aircraft navigation and control system from Northwestern Polytechnical University in 1989. Her research interest covers Bayes and airborne fire control. Corresponding author of this paper.)

万开方 西北工业大学电子信息学院博士研究生. 2010 年获得西北工业大学系统工程专业学士学位. 主要研究方向为航空火力控制.
E-mail: yibai_2003@126.com
(**WAN Kai-Fang** Ph. D. candidate at the School of Electronics and Information, Northwestern Polytechnical University. He received his bachelor degree in system engineering from Northwestern Polytechnical University in 2010. His main research interest is airborne fire control.)