基于权值动量的 RBM 加速学习算法研究

李 飞 [1] 高晓光 [1] 万开方 [1]

摘 要 动量算法理论上可以加速受限玻尔兹曼机 (Restricted Boltzmann machine, RBM) 网络的训练速度. 本文通过对现有动量算法进行仿真研究, 发现现有动量算法在受限玻尔兹曼机网络训练中加速效果较差, 且在训练后期逐渐失去了加速性能. 针对以上问题, 本文首先基于 Gibbs 采样收敛性定理对现有动量算法进行了理论分析, 证明了现有动量算法的加速效果是以牺牲网络权值为代价的; 然后, 本文进一步对网络权值进行研究, 发现网络权值中包含大量真实梯度的方向信息, 这些方向信息可以用来对网络进行训练; 基于此, 本文提出了基于网络权值的权值动量算法, 最后给出了仿真实验. 实验结果表明, 本文提出的动量算法具有更好的加速效果, 并且在训练后期仍然能够保持较好的加速性能, 可以很好地弥补现有动量算法的不足.

关键词 深度学习, 受限玻尔兹曼机, 动量算法, 权值动量

引用格式 李飞, 高晓光, 万开方. 基于权值动量的 RBM 加速学习算法研究. 自动化学报, 2017, **43**(7): 1142−1159

DOI 10.16383/j.aas.2017.c160325

Research on RBM Accelerating Learning Algorithm with Weight Momentum

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

Abstract Momentum algorithms can accelerate the training speed of restricted Boltzmann machine theoretically. Through a simulation study on existing momentum algorithms, it is found that existing momentum algorithms for training restricted Boltzmann machine have a poor accelerating effect and they began to lose acceleration performance. In the latter part of training process. Focusing on this problem, firstly, this paper gives a theoretical analysis of the algorithms based on Gibbs sampling convergence theorem. It is proved that the acceleration effect of existing momentum algorithms is at the expense of enlarging network weights. Then, a further investigation on network weights shows that the network weights contain a lot of information of the true gradient direction which can be used to train the network. According to this, a weight momentum algorithm is proposed based on the weight of the network. Finally, simulation results demonstrate that the proposed algorithm has a better acceleration effect and has the accelerating ability even in the end of the training process. Therefore the proposed algorithm can well make up for the weaknesses of existing momentum algorithms.

Key words Deep learning, restricted Boltzmann machine (RBM), momentum algorithm, weight momentum

Citation Li Fei, Gao Xiao-Guang, Wan Kai-Fang. Research on RBM accelerating learning algorithm with weight momentum. *Acta Automatica Sinica*, 2017, **43**(7): 1142−1159

自 2006 年 Hinton 等[1] 提出第一个深度网络开始, 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]、深度自动编码机[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[5]、语音分析[6]、文本分析[7]、游戏[8−9]、控制[10]、环境保护[11]. 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)[12]. 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[13].

受限玻尔兹曼机 (Restricted Boltzmann machine, RBM)[14] 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. 它具有两层结构, 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)[2] 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.

2002 年, Hinton 提出了对比散度 (Contrastive divergence, CD) 算法[15] 用以训练 RBM 网络, 通过一条 Gibbs 采样链来近似目标梯度, 取得了良好

收稿日期 2016-04-11 录用日期 2016-09-30
Manuscript received April 11, 2016; accepted September 30, 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 魏庆来
Recommended by Associate Editor WEI Qing-Lai
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

的训练效果, 是目前 RBM 训练的标准算法. 为克服 CD 算法采样初值较差的缺点, 2008 年, Tieleman 以 CD 算法为基础, 提出了持续对比散度 (Persistent contrastive divergence, PCD) 算法[16], 它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, 加快了 Gibbs 采样链的收敛速度. 为了加速 PCD 算法, Tieleman 等又于 2009 年提出了加速持续对比散度 (Fast persistent contrastive divergence, FPCD) 算法[17], 引入了额外的加速参数提高采样速度. 为提高 Gibbs 采样链的混合率, Desjardins 等 (2010)[18]、Cho 等 (2011)[19]、Brakel 等 (2011)[20] 分别提出应用并行回火算法 (Parallel tempering, PT) 来训练 RBM. PT 算法在不同温度下并行化多条 Gibbs 采样链, 不同温度下的 Gibbs 采样以一定的交换概率进行交换, 相对于一条 Gibbs 采样链, PT 算法具有更高的采样混合率. 针对复杂分布, 尤其是多模分布, PT 算法的训练效果要明显优于 CD 算法[21].

动量算法作为梯度加速项可以与以上算法结合加速 RBM 网络的训练效果. 1964 年, Polyak[22] 提出了经典动量方法 (Classical momentum, CM), 它通过一个速度 v 来积累梯度, 在梯度下降算法中可以加快收敛速度, 该算法在整个机器学习领域已经取得了广泛的成功. 2010 年, Fischer 等[23] 发现经典动量算法在 RBM 网络训练中效果较差, 甚至在一些实验中根本无法起到加速效果. 2012 年, Hinton 在文献 [24] 中推荐使用动量算法来加速 RBM 网络的训练速度, 但并没有给出相应的实验结果. 2013 年, Sutskever 等[25] 为训练深度网络, 提出了基于 Nesterov 加速梯度算法的 Nesterov 动量 (Nesterov momentum, NM) 算法. 该算法在一定程度上克服了经典动量算法的不稳定性和方向误差, 在深度自动编码机和深度递归神经网络等深层网络中取得了良好的效果. 2015 年, Zarea 等[26] 将该动量算法应用到受限玻尔兹曼机训练中, 从仿真结果来看, 效果并不是很明显.

基于以上描述, 本文将在后续章节分别对下列问题进行研究: 第 1 节, 现有动量算法在训练 RBM 网络的过程中存在哪些问题; 第 2 节, 造成这些问题的原因是什么; 第 3 节, 如何弥补现有动量算法的这些不足; 最后, 第 4 节, 仿真实验部分给出了具体的仿真结果和详细分析.

1　背景知识

本节对本文所需背景知识进行简要介绍, 给出了受限玻尔兹曼机模型及其训练算法、传统动量算法的简要描述.

1.1　受限玻尔兹曼机

受限玻尔兹曼机是一个马尔科夫随机场模型[21], 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.

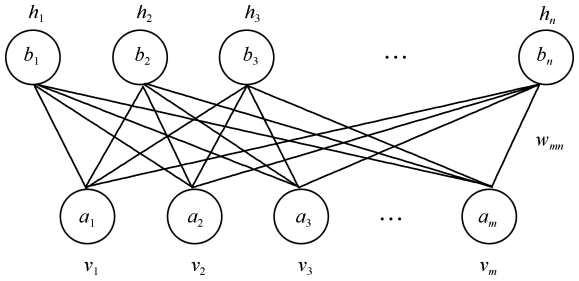

图 1　RBM 结构

Fig. 1　Configuration of RBM

本文研究二值 RBM, 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:

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

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:

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

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

$$P(v_k = 1|h) = d\frac{1}{1 + \exp\left(-a_i - \sum\limits_{i=1}^{n} w_{ij} h_j\right)} =$$

$$\text{sigmoid}\left(a_i+\sum_{i=1}^{n}w_{ij}h_j\right) \tag{3}$$

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: $L_{\theta,v}=\prod_{i=1}^{n}P(v^i)$. 为简化计算, 将其改写为对数形式: $\ln L_{\theta,v}=\sum_{i=1}^{n}\ln P(v^i)$. 进一步推导对数似然函数的参数梯度

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

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:

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

现有 RBM 训练算法, 包括对比散度 (Contrastive divergence, CD) 算法、并行回火 (PT) 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样本, 然后分别通过其他后续操作获得最终的目标梯度. CD 算法是 RBM 训练的主流算法, 下面首先给出 CD 算法[21]:

算法 1. Contrastive divergence

Input. $RBM(V_1,\cdots,V_m,H_1,\cdots,H_n)$, training batch S

Output. w_{ij},a_j and b_i for $i=1,\cdots,n,\quad j=1,\cdots,m$

1: Init $\nabla w_{ij}=\nabla a_j=\nabla b_i=0$ for $i=1,\cdots,n,j=1,\cdots,m$
2: For all the $v\in S$ do
3: $v^{(0)}\leftarrow v$
4: for $t=0,\cdots,k-1$ do
5: for $i=1,\cdots,n$ do sample $h_i^{(t)}\sim p(h_i|v^{(t)})$
6: for $j=1,\cdots,m$ do sample $v_j^{(t+1)}\sim p(v_j|h^{(t)})$
7: for $i=1,\cdots,n,j=1,\cdots,m$ do
8: $\nabla w_{ij}=p(H_i=1|v^{(0)})\cdot v_j^{(0)}-p(H_i=1|v^{(k)})\cdot v_j^{(k)}$
9: $\nabla a_i=v_i^{(0)}-v_i^{(k)}$
10: $\nabla b_j=p(H_j=1|v^{(0)})-p(H_j=0|v^{(k)})$
11: $w_{ij}=w_{ij}+\eta\nabla w_{ij}$
12: $a_i=a_i+\eta\nabla a_i$
13: $b_j=b_j+\eta\nabla b_j$
14: 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.

1.2 动量算法

经典动量方法 (Classical momentum, CM)[22] 如图 2 (a) 所示, 在梯度下降算法中, 其中 v 为累计速度, $\nabla g(\theta_t)$ 为目标函数在当前点的梯度, 它通过累计速度与当前梯度的差值来调整目标梯度, 从而加快收敛速度. RBM 模型是基于梯度上升法进行训练的, 因此经典动量方法在 RBM 模型下的参数更新公式为式 (7), 其中 μ 为累计速度参数, η 为学习率.

$$\begin{aligned}
v_{t+1}&=\mu v_t+\eta\nabla g(\theta_t)\\
\theta_{t+1}&=\theta_t+v_{t+1}
\end{aligned} \tag{7}$$

Nesterov 动量 (Nesterov momentum, NM)[25]

如图 2 (b) 所示. 与 CM 不同的是, NM 不是计算目标函数在当前点的梯度, 而是首先由当前时刻的累计速度对目标函数的参数进行调节, 计算 $\theta_t + \mu v_t$ 的梯度, 然后才对目标梯度进行调节, 在 RBM 模型下的 NM 参数更新公式如式 (8), 其中 μ 为累计速度参数, η 为学习率.

$$\begin{aligned} v_{t+1} &= \mu v_t + \eta \nabla g(\theta_t + \mu v_t) \\ \theta_{t+1} &= \theta_t + v_{t+1} \end{aligned} \tag{8}$$

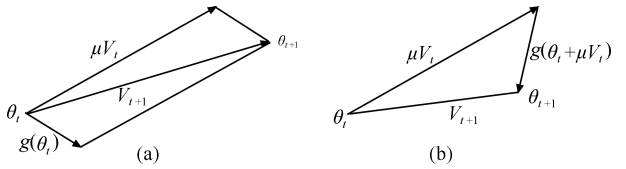

图 2　动量示意图

Fig. 2　Momentum diagram

2　问题描述

由第 5.1 节中给出的网络模型和训练策略, 得到 CD 算法和传统动量算法在 MNIST 数据集上的训练效果对比图.

其中 CD 表示原始 CD 算法, CM 表示加入经典动量项的 CD 算法, NM 表示加入 Nesterov 动量项的 CD 算法.

图 3 和图 4 给出了三种算法的仿真对比图, 由仿真结果可以看出, CM 算法和 NM 算法可以加快 RBM 训练过程的收敛速度, 但仍存在以下问题:

1) 加速效果不明显. 如图 3 所示, CM 算法和 NM 算法的收敛曲线与 CD 算法的收敛曲线间隔较小, 考虑每次迭代额外增加的计算量的前提下, 这两种动量方法并没有达到预想的效果.

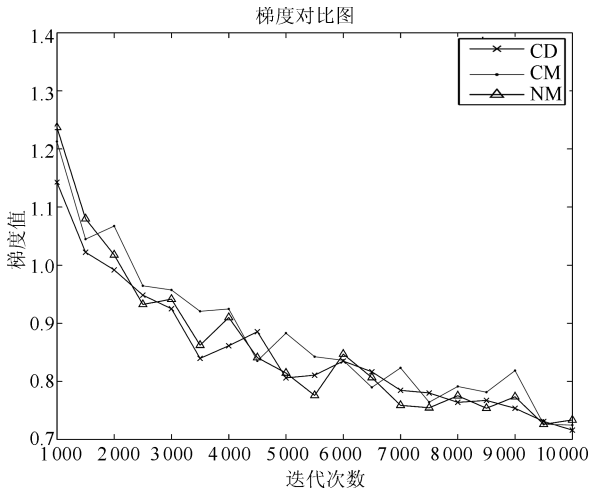

图 3　重构误差对比图

Fig. 3　Compassion of reconstruction errors

2) 训练后期加速失效. 随着迭代的进行, CM 算法和 NM 算法的误差曲线逐渐与 CD 算法重合. 如图 4 所示, CM 算法与 NM 算法与 CD 算法的重构误差的差值逐渐收敛到 0, 这说明在训练后期, CM 算法和 NM 算法逐渐失去了加速效果.

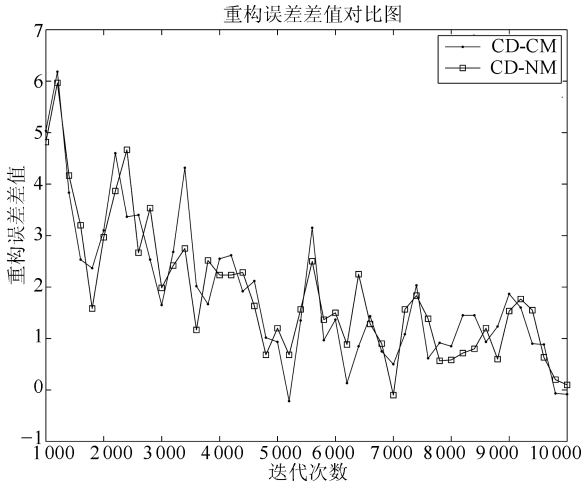

图 4　重构误差差值对比图

Fig. 4　Compassion of the difference of the reconstruction errors

3　问题分析

RBM 网络的训练算法都是基于 Gibbs 采样的, Gibbs 采样链的收敛性质, 即采样混合率, 是影响训练算法性能的本质因素. 因此, 本节首先基于 Gibbs 采样收敛性理论对以上算法和问题进行分析.

3.1　基于 Gibbs 采样收敛性的理论分析

3.1.1　Gibbs 采样链收敛性定理

由于在 RBM 网络中, 可见层偏置向量 a 和隐层偏置向量 b 都是极小值, 相对于网络权值 w 可以忽略不计. 因此, 本文在研究网络参数的时候, 只对网络权值 w 进行研究. 下面首先给出 Gibbs 采样收敛性定理:

定理 1.　RBM 网络下, Gibbs 采样链的混合率随网络权值量级的增大而逐渐降低[15, 21, 27−28].

证明.　RBM 网络基于式 (2) 和 (3) 进行 Gibbs 迭代, 以此完成参数更新. 当网络权值 w 的量级逐渐增大时, 即 $|w| \to +\infty$, 可推理出如下结果:

$$\begin{cases} -b_j - \sum\limits_{i=1}^{n} w_{ij} v_i \to +\infty \text{ 或 } -\infty \\ -a_i - \sum\limits_{i=1}^{n} w_{ij} h_j \to +\infty \text{ 或 } -\infty \end{cases} \tag{9}$$

则对应的隐层节点的取值概率 $P(h_k = 1|v)$ 和可见层节点的取值概率 $P(v_k = 1|h)$ 分别为

$$\begin{cases} P(h_k = 1|v) \to 0 \text{ 或 } 1 \\ P(v_k = 1|h) \to 0 \text{ 或 } 1 \end{cases} \tag{10}$$

当网络权值逐渐增大时, Gibbs 采样链的采样概率逐渐趋于 0 或 1. 即, 每次采样时, 各点的取值均为 0 或 1, 此时, 采样链的转移算子不再具备随机性 (Less randomness[27]).

由文献 [27] 可知: Gibbs 采样链的混合率随着其转移算子随机性的降低而逐渐降低. 因此, 当 Gibbs 采样链的采样概率逐渐趋于 0 或 1 时, 采样链的转移算子的随机性逐渐降低, 导致了采样链混合率的逐渐下降. 从而证明了当网络权值逐渐增大时, Gibbs 采样链的混合率逐渐下降. □

为验证上述推论的合理性, 我们给出了 Sigmoid 函数在不同权值下的曲线, 如图 5 所示.

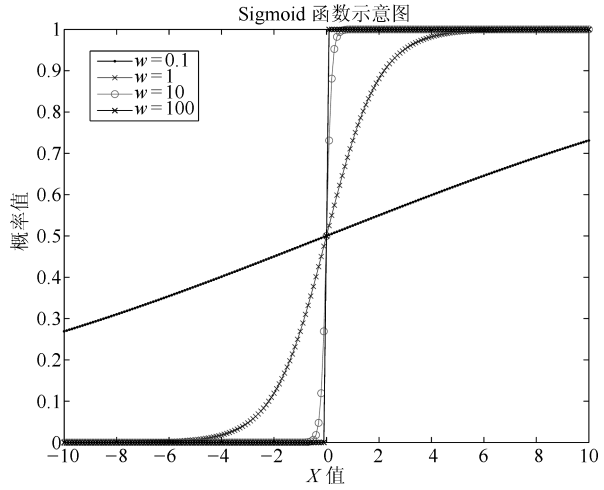

图 5 Sigmoid 函数示意图

Fig. 5 Sigmoid diagram

当网络权值量级逐渐增大时, Sigmoid 函数曲线变得越来越陡峭. 这时如果对状态进行采样, 大部分采样区域将为 0 或 1, 发生变化的区域, 即可调域 (图中曲线部分), 非常小, 由此导致采样前后的状态变化量非常小, 即转移算子的随机性非常小, 从而导致采样混合率降低. 当采样前后状态变化量不变时, 转移算子没有随机性, 采样率为 0, 此时, 网络将停止优化.

所以, 在 RBM 网络下, 随着网络权值量级的逐渐增大, Gibbs 采样链的混合率逐渐降低.

3.1.2 对第 2 节中的问题进行分析

由第 1.2 节给出的动量公式可以看出, CM 算法和 NM 算法通过速度项对梯度进行累积, 然后对参数梯度进行修正, 即, 除去参数本身的梯度外, CM 算法和 NM 算法还额外引入了一个梯度更新量, 这将加快网络权值的增加速度, 图 6 给出了 CD 算法和 CM、NM 算法的权值增长曲线对比图.

由图 6 可以看出, CM 算法和 NM 算法的权值增长速度明显快于 CD 算法, 图 7 给出的权值差值对比图也说明了这一点. 这说明, CM 算法和 NM 算法的加速效果是以增加权值量级为代价的. 在迭代训练初期, 由于三者的网络权值都较小, 这时, 虽然 CM 算法和 NM 算法的混合率略小于 CD 算法的混合率, 但由于累计速度 v 的存在, 使得 CM 算法和 NM 算法的综合梯度要大于 CD 算法, 如图 8 所示, 从而在前期, CM 算法和 NM 能够起到加速效果. 但由于它们之间的梯度差值差别不是很大, 所以加速效果并不是很明显, 即第 2 节问题 1) 中描述的现象.

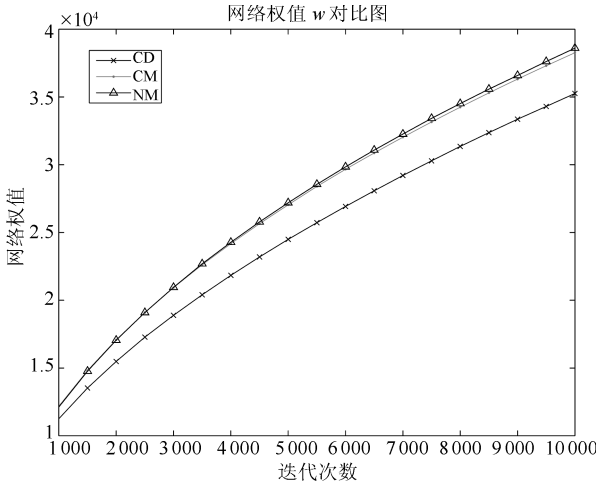

图 6 网络权值 w 对比图

Fig. 6 Compassion of w

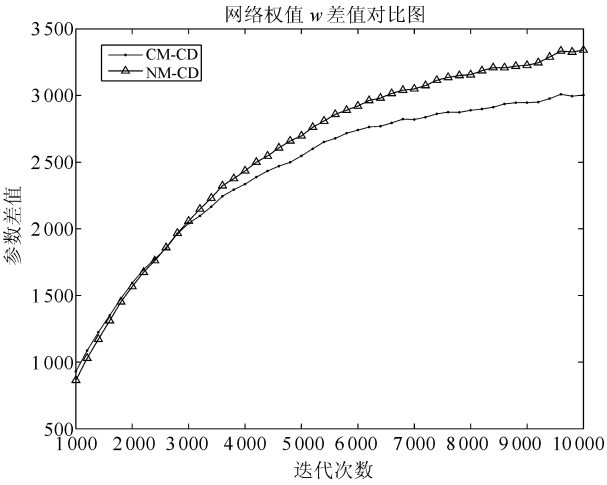

图 7 网络权值 w 差值对比图

Fig. 7 Compassion of the difference of w

但到了训练后期, CM 算法和 NM 算法训练下的权值明显大于 CD 算法训练下的权值, 这使得 CM 算法和 NM 算法下的采样链混合率急剧下降, 远小于 CD 算法下的采样链混合率, 这时, 即便存在累计速度 v, 也不能弥补混合率下降带来的梯度劣势, 从而, 使得最终 CM 算法和 NM 算法的梯度值接近, 如下图 9 中所示, CM 算法与 NM 算法与 CD 算法的梯度差值在训练后期逐渐降为 0, 甚至小于 0, 所以到了训练后期, CM 算法与 NM 算法逐渐失去了加速效果, 这就是第 2 节中问题 2) 的原因.

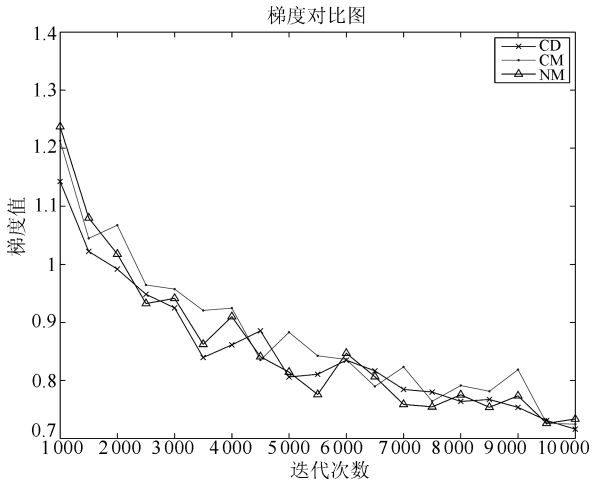

图 8　梯度对比图

Fig. 8　Compassion of gradients

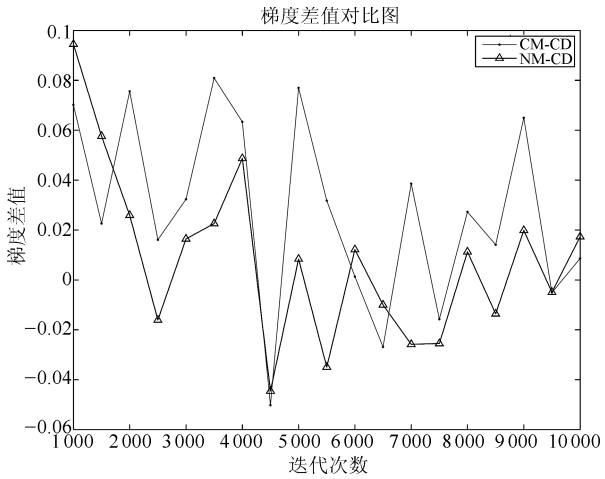

图 9　梯度差值对比图

Fig. 9　Compassion of the difference of gradients

3.2　通过权值衰减项控制网络权值

3.2.1　仿真实验

由于网络权值的量级是影响 Gibbs 采样链的关键因素, 所以一个很自然的想法是通过控制网络权值的量级来保证采样链的混合率. 因此, 本节在原有动量算法的基础上引入了权值衰减项 (Weight decay), 研究总体训练效果. 首先分别给出 CM 算法和 NM 算法与权值衰减项结合后总的梯度更新公式:

$$\begin{cases} \mathrm{CMD}\theta_{t+1} = \theta_t + \mathrm{CM}grad_t - \lambda\theta_t \\ \mathrm{NMD}\theta_{t+1} = w_t + \mathrm{NM}grad_t - \lambda\theta_t \end{cases} \tag{11}$$

其中, λ 为权值衰减项参数, 设为 0.00001. 以 CMD 表示 CM 算法与权值衰减项结合后的算法, CMDθ 表示 CMD 算法下的参数值, CM$grad$ 为由 CM 算法计算得到的更新梯度; 以 NMD 表示 NM 算法与权值衰减项结合后的算法, NMDθ 表示 NMD 算法下的参数值, NM$grad$ 为由 NM 算法计算得到的更新梯度. 以第 5.1 节中给出的实验设计为例, 进行仿真实验, 仿真结果如下.

图 10 给出了权值衰减下的网络权值对比图, 图 11 给出了各动量算法与原始 CD 算法的网络权值差值对比图. 通过以上两图可以看出, 权值衰减项可以有效地控制网络权值的变化, 使其不至于过大, 根据第 3.1 节给出的定理, 权值衰减项的引入可以一定程度上保证 Gibbs 采样链的混合率.

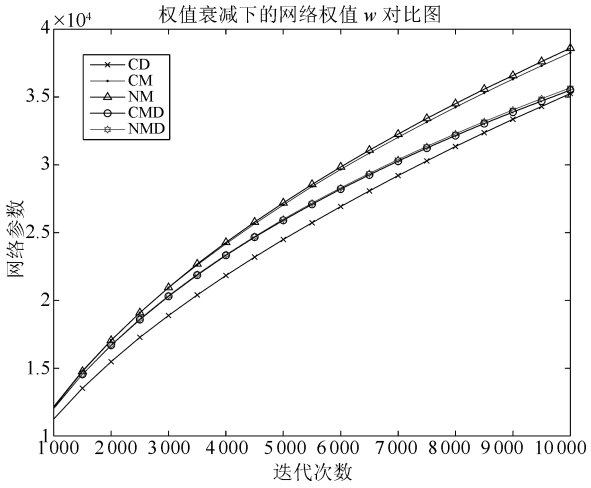

图 10　权值衰减下网络权值 w 对比图

Fig. 10　Compassion of w

图 12 给出了各算法的重构误差对比图, 从图 12 中可以看出, CMD 算法和 NMD 算法的训练曲线与 CM 算法和 NM 算法的训练曲线几乎重合, 即权值衰减项的引入并没有提高 CM 算法和 NM 算法的加速效果; 图 13 给出的重构误差差值对比图中, 各差值曲线也几乎重合, 都是逐渐收敛到 0, 这说明了在训练后期, CMD 算法和 NMD 算法也会失去加速效果. 加入权值衰减项后的动量算法并没有改善原始动量算法的性能, 仍然出现了第 2 节中描述的问题.

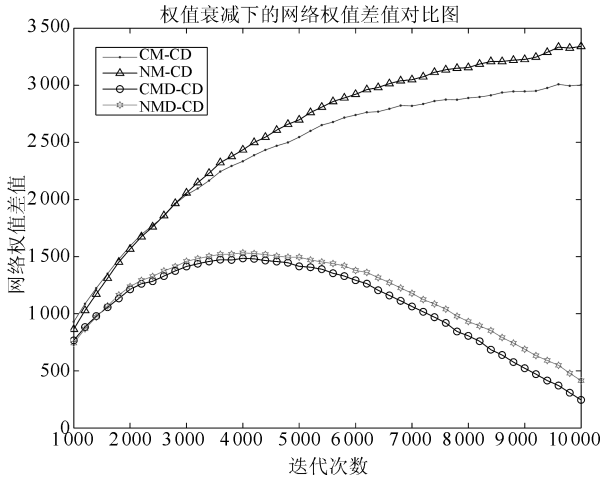

图 11 权值衰减下网络权值差值对比图

Fig. 11 Compassion of the difference of w

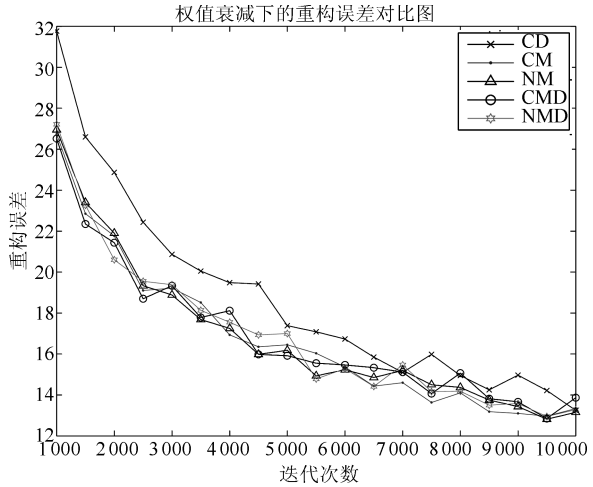

图 12 权值衰减下重构误差对比图

Fig. 12 Compassion of the reconstruction errors

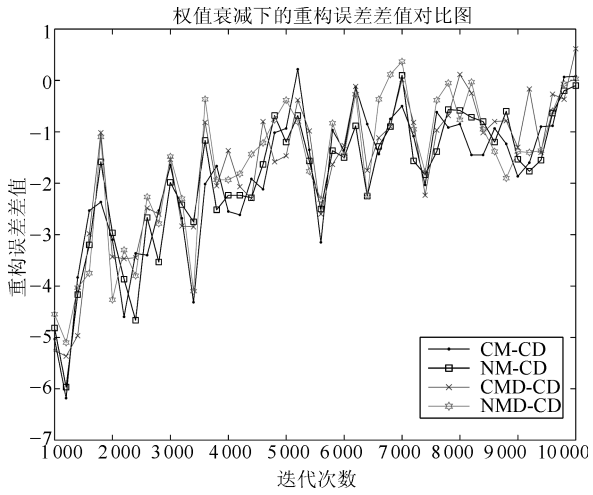

图 13 权值衰减下重构误差差值对比图

Fig. 13 Compassion of the difference of the reconstruction errors

3.2.2 问题解释

由式 (11), 可以得到引入权值衰减项后的权值梯度公式:

$$\nabla W = \mu v_t + \eta \nabla w - \lambda W \tag{12}$$

其中, ∇w 为 Gibbs 采样计算的梯度, ∇W 为总的梯度.

权值衰减项的引入虽然可以控制权值的量级, 保证采样链的混合率, 即提高了 ∇w 的值, 但由式 (12) 可以看出, 权值衰减项的方向与动量的方向相反, 也就是说, 权值衰减项对动量具有一定的抵消作用. 由提高混合率带来的加速效果和由方向相反引起的抵消作用相互牵制, 最终导致总的梯度 ∇W 与仅由动量算法计算的梯度大致相等, 如图 14 和 15 所示, 从而出现了第 2 节中描述的问题.

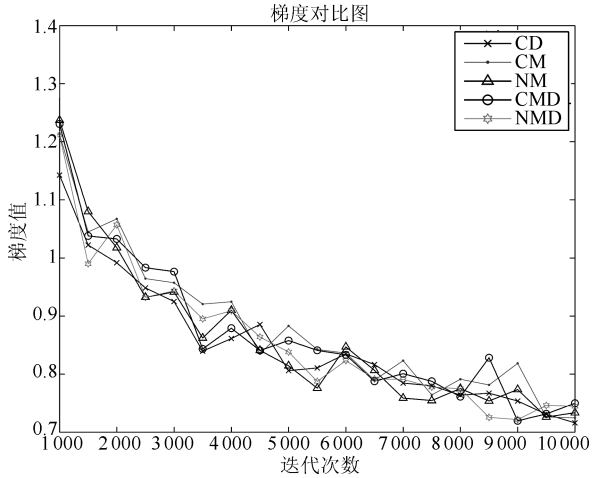

图 14 梯度对比图

Fig. 14 Compassion of the gradients

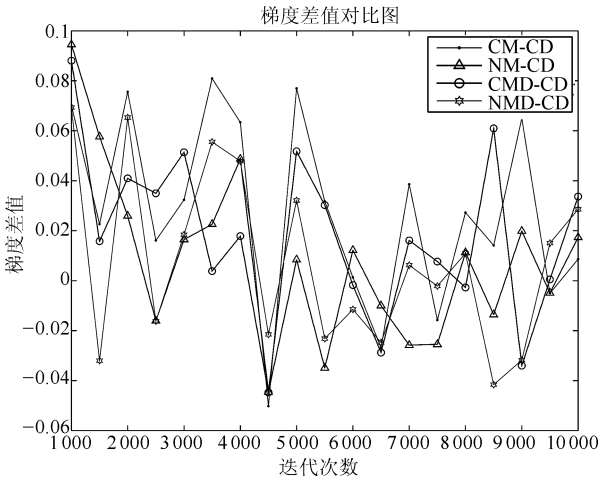

图 15 梯度差值对比图

Fig. 15 Compassion of the difference of gradients

3.2.3　衰减系数对仿真结果的影响

为研究不同衰减系数对训练效果的影响, 本节我们以 CMD 算法为例, 设计了 7 组对比实验, 在每组实验中, 衰减系数 λ 分别取不同的值, 仿真结果和相应的 λ 的取值如图 16 和 17 所示. 其中当 $\lambda = 0$ 时, 表示没有权值衰减项, 即经典动量算法 CM.

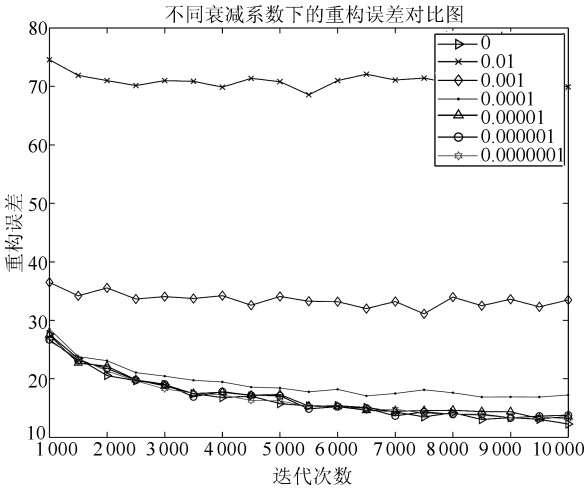

图 16　重构误差对比图

Fig. 16　Compassion of the reconstruction errors

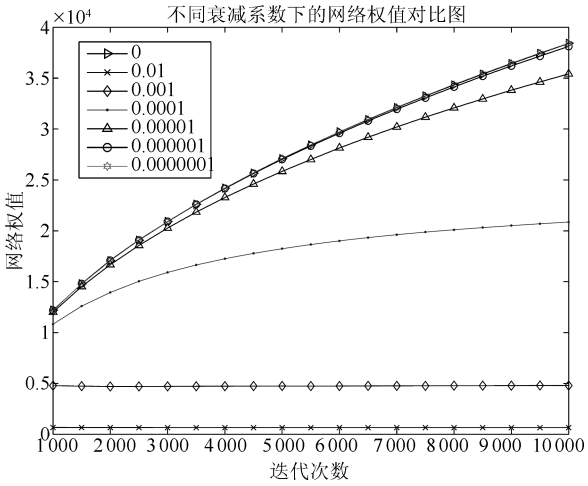

图 17　网络权值对比图

Fig. 17　Compassion of the difference of w

图 16 给出了不同衰减系数下的重构误差对比图; 图 17 给出了不同衰减系数下的网络权值对比图. 从图 17 中可以看出, 当衰减系数 λ 取较大值时 ($\lambda > 0.0001$), 衰减项对网络权值的控制较强, 此时网络权值较小, 但相应的重构误差较大, 这说明, 当 λ 取值较大时, 会降低网络的训练效果, 甚至导致网络无法训练; 当 λ 取值较小时 ($\lambda < 0.0001$), 衰减项对网络权值的控制较弱, 此时网络权值略小于无权值衰减下的网络权值, 对应的重构误差与无权值衰减下的重构误差差别不大; 随着 λ 的继续减小 ($\lambda < 0.00001$), 权值衰减项逐渐失去了控制作用, 此时网络权值与无权值衰减项下的网络权值几乎相等, 相应的重构误差与无权值衰减项下的网络重构误差几乎相等. 从图 17 中可以看出, λ 取 0.00001(即本文的取值) 是较为合适的.

由以上理论分析可知, 现有动量算法以提高权值量级为代价加快训练过程的收敛, 同时这也导致了后期加速性能的失效. 通过引入权值衰减项来控制网络权值并不能解决这个问题.

4　权值动量算法

4.1　网络权值信息

本文以网络权值作为动量项的想法来自于以下基本判断:

迭代训练一段时期后, 网络参数的方向与真实参数方向大致相同, 这时, 如果沿着该方向加大网络参数, 可以一定程度上提高网络训练效果.

下面给出分析证明.

4.1.1　理论分析

定理 2.　CD 算法的估计梯度的方向在大部分时间内与真实梯度的方向是相同的[27].

文献 [27] 中通过大量的实验分析表明, 在大部分时间内, CD 算法的近似梯度与真实梯度的方向是相同的. 所以, 我们有理由相信, 经过一段时间的训练调节之后, 网络权值的方向与期望权值的方向大致相同.

4.1.2　实验证明

本次试验的网络结构和参数初值与第 5.1 节中给出的设计一致. 本节采用的训练策略是在迭代进行到 500 次时, 对网络权值进行加倍, 即 $w = w \times 10$. 实验结果如图 18.

从以上对比结果可以看出, 在迭代次数为 500 处对网络权值进行加倍可以大幅度地提升训练效果. 但加倍后的重构误差值几乎不再变化, 而不加倍的重构误差值则逐渐减小. 我们做了大量的仿真实验, 分别在不同的迭代点, 对网络权值 w 分别乘以不同的倍数, 都会出现以上结果: 权值加倍后会提高网络训练效果, 但后续训练速度极其缓慢.

出现以上现象的原因是: 当对网络参数进行加倍后, 网络权值急剧增大, 如图 19 所示, 根据第 3.1 节给出的 Gibbs 采样收敛性定理, 此时, Gibbs 采样链的混合率急剧下降, 从而使得网络参数的梯度值急剧下降, 如图 20 所示, 最终导致网络训练曲线不再变化.

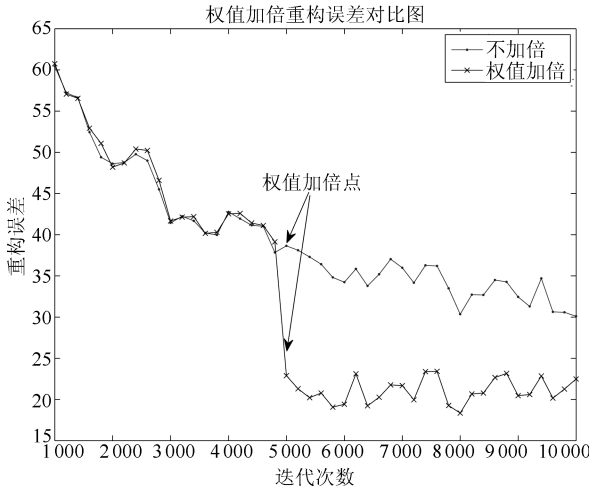

图 18 重构误差对比图

Fig. 18 Compassion of reconstruction errors

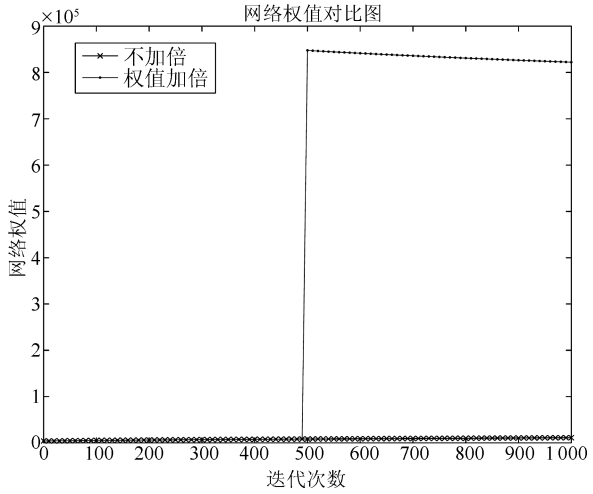

图 19 网络权值对比图

Fig. 19 Compassion of the value of w

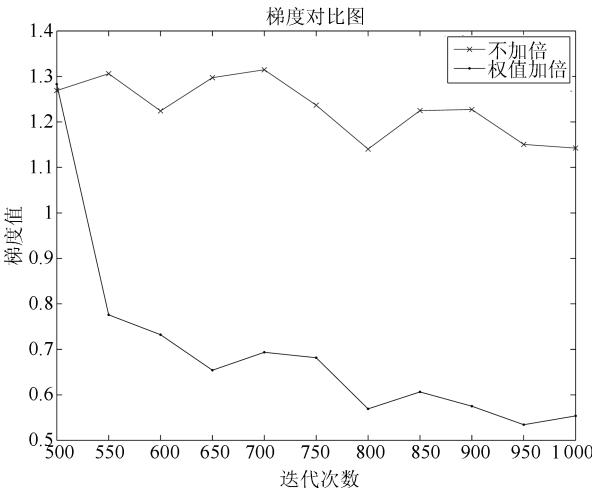

图 20 梯度对比图

Fig. 20 Compassion of gradients

以上分析表明, 网络权值在网络训练中保留了重要的真实梯度的方向信息. 当网络权值增大到采样梯度很小甚至接近于 0 的时候, 我们可以用权值中保存的梯度方向信息来继续训练网络, 基于此, 本文设计了基于权值动量的 RBM 加速训练算法.

4.2 权值动量算法

本文设计的动量项为网络权值 w, 该权值动量项可以与原始 CD 算法结合, 也可以与现有动量算法 CM 算法和 NM 算法结合, 在混合率降低时, 提供梯度方向信息, 加速网络训练. 下面分别给出权值动量项与上述三种算法结合后的参数更新公式.

权值动量与 CD 结合后的算法称为 CDW, 相应的参数更新公式为

$$\begin{cases} w_{ij} = w_{ij} + \eta\nabla w_{ij} + \alpha w_{ij} \\ a_i = a_i + \eta\nabla a_i \\ b_j = b_j + \eta\nabla b_j \end{cases} \tag{13}$$

其中, η 为学习率, α 为权值动量系数.

权值动量与 CM 算法结合后的算法称为 CMW, 相应的参数更新公式为

$$\begin{cases} w_{ij} = w_{ij} + \mu v_{ij}^{cm} + \eta\nabla w_{ij} + \alpha w_{ij} \\ a_i = a_i + \mu v_i^{cm} + \eta\nabla a_i \\ b_j = b_j + \mu v_j^{cm} + \eta\nabla b_j \end{cases} \tag{14}$$

其中, v^{cm} 为 CM 算法下的速度, μ 为速度参数.

权值动量与 NM 算法结合后的算法称为 NMW, 相应的参数更新公式为

$$\begin{cases} w_{ij} = w_{ij} + \mu v_{ij}^{nm} + \eta\nabla w_{ij} + \alpha w_{ij} \\ a_i = a_i + \mu v_i^{nm} + \eta\nabla a_i \\ b_j = b_j + \mu v_j^{nm} + \eta\nabla b_j \end{cases} \tag{15}$$

其中, v^{nm} 表示 NM 算法下的速度, μ 为速度参数.

5 仿真实验

为证明算法的有效性和普适性, 本文选取了 5 个常用的 Benchmark 标准数据集. 分别为 MNIST[29] 数据集、MNORB 数据集、CIFAR-10[30] 数据集、CIFAR-100[30] 数据集和 OLIVETTI_FACE[31] 数据集. 并在这 5 个 Benchmark 标准数据集上进行了测试.

MNIST 手写数据集共包含 60 000 个训练样本, 对应 0 ~ 9 十个数字, 每个样本是一幅 28×28 像素的灰度图. CIFAR-10 数据集是 Tiny image[32] 数据集的一部分, 共包含 60 000 幅 $32 \times 32 \times 3$ 彩色图片. 图片分为 10 类, 每个类由 5 000 幅训练图片和 1 000 幅测试图片组成. CIFAR-100 数据集与

CIFAR-10 数据集类似, 它包含 100 个类, 每个类对应 600 幅图片, 其中 500 幅用于训练, 100 幅用于测试. 整个数据集共包含 60 000 幅图片. MNORB 数据集是 NORB[33] 数据集的二值化子集, 总共包含 19 440 个数据, 每个数据对应一幅 32×32 的灰度图. OLIVETTI_FACE 数据集是贝尔实验室收集的人脸数据库, 用于人脸识别任务. 它总共包含 400 幅图片, 每幅图片为 64×64 的灰度图.

在每个数据集下, 本文分别设计了 7 组对比实验, 分别将上文给出的 7 种训练算法进行对比. 为避免重复叙述, 本文只在 MNIST 数据集下给出了详细的仿真实验, 并结合仿真结果, 对本文提出的算法进行了详细地分析, 在其他 4 个数据集只给出网络结构和参数设定, 以及相应的仿真结果和简要分析.

5.1 MNIST 数据集

MNIST 数据集内的数据为 28×28 的灰度图片, 因此, 本文设计的 RBM 网络模型为 784×500, 输入层有 784 个节点, 对应灰度图的 784 个像素点, 隐层有 500 个节点, 用来提取输入层数据的特征. 训练迭代次数为 10 000 次. 具体的网络结构如表 1 所示.

表 1　网络参数值

Table 1　The value of network parameters

网络参数	初始值
a	$\text{zeros}(1, 784)$
b	$\text{zeros}(1, 500)$
w	$0.1 \times randn(784, 500)$
η	0.1
μ	0.9

具体的网络参数初始值设定如表 2 所示.

表 2　训练参数

Table 2　Training parameters

算法参数	μ	λ	α
CD	0.9		
CM	0.9		
NM	0.9		
CMD	0.9	0.00001	
NMD	0.9	0.00001	
CDW	0.9		0.0001
CMW	0.9		0.0001
NMW	0.9		0.0001

5.1.1 训练精度对比分析

图 21 给出了所有算法的训练误差对比图, 其中 CDW 算法、CMW 算法和 NMW 算法的训练曲线与 CD 算法的间距较大, 远大于 CM 算法、NM 算法、CMD 算法和 NMD 算法. 仿真结果说明本文提出的权值动量算法相对于现有动量算法, 具有更好的加速效果, 且加速效果明显. 从而克服了第 2 节问题 1) 中描述的现有动量算法出现的问题.

图 22 给出了迭代后期各动量算法与 CD 算法重构误差的差值对比图, 为画图方便, 作如下记号, 如表 3 所示.

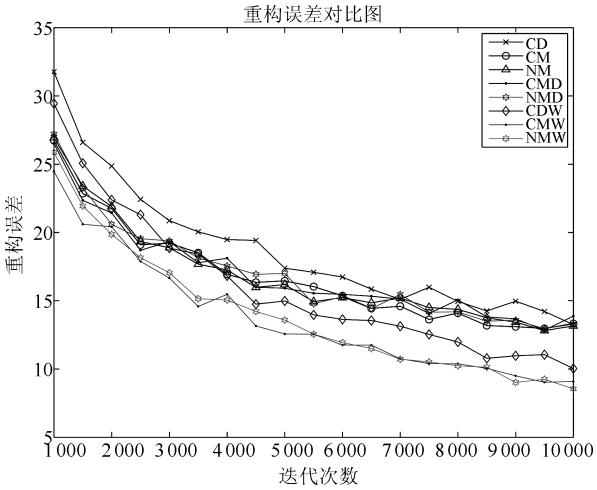

图 21　重构误差对比图

Fig. 21　Compassion of reconstruction errors

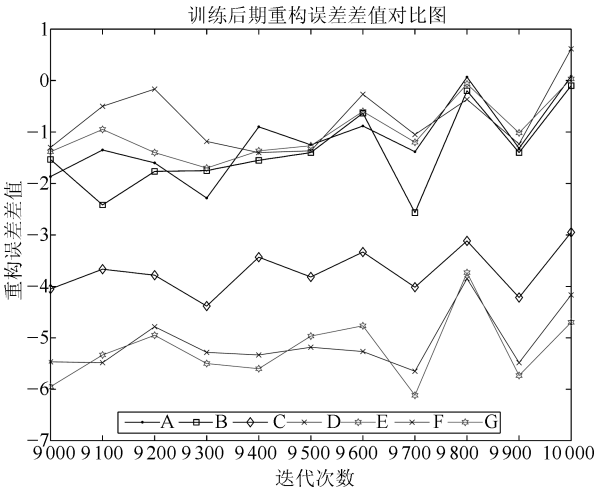

图 22　重构误差差值对比图

Fig. 22　Compassion of the difference of the reconstruction errors

由图 22 所示, 在迭代末期, CDW 算法、CMW 算法和 NMW 算法与 CD 算法的重构误差差值仍然较大, 而其他算法与 CD 算法的重构误差差值逐渐增大到 0, 甚至大于 0. 这说明, 其他算法在后期逐渐失去了加速效果, 甚至会产生副作用, 而本文提出的权值动量算法在迭代后期仍然具有良好的加速效

果. 从而克服了第 3 节问题 2) 中描述的现有动量算法出现的问题.

表 3 记号示意图

Table 3 Sign diagram

代号	差值项
A	CM-CD
B	NM-CD
C	CMW-CD
D	NMW-CD
E	CDW-CD
F	CMW-CD
G	NMW-CD

5.1.2 梯度对比分析

图 23 给出了迭代初期各动量算法与 CD 算法之间的梯度差值对比图. 从图 23 中可以看出, 在迭代初期, 各动量算法的梯度值基本上都大于 CD 算法的梯度值, 从而保证了加速效果.

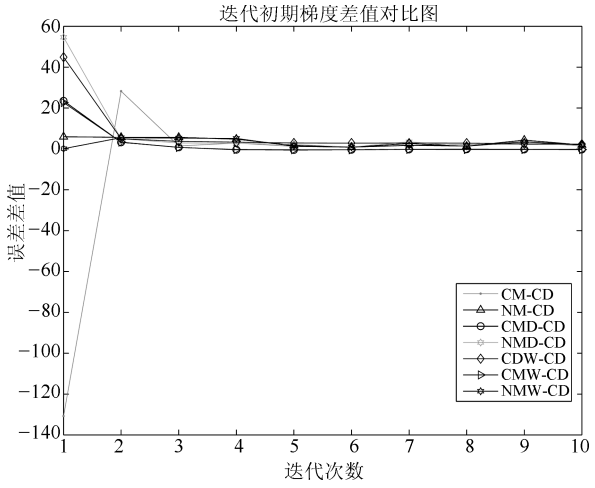

图 23 迭代初期梯度差值对比图

Fig. 23 Compassion of the difference of the gradients in initial stages of iteration

图 24 给出了迭代中期各动量算法与 CD 算法之间的梯度差值, 图 24 给出了迭代中期各算法的重构误差对比图, 对比分析这两个图, 可以发现以下两个问题:

问题 1. 图 24 中显示, 在迭代中期, CDW 算法的梯度远大于其他算法梯度. 理论上, 这一时期 CDW 算法的加速效果要优于其他算法. 但图 25 给出的迭代中期重构误差图中, CDW 算法的重估误差虽然比 CD 算法的重估误差小, 但却大于其他动量算法, 与图 24 中的现象相反.

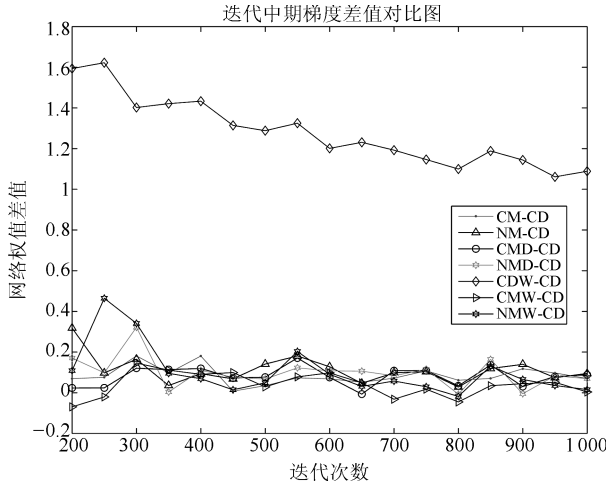

图 24 迭代中期梯度差值对比图

Fig. 24 Compassion of the difference of the gradients in mid-term of iteration

问题 2. 图 23 中, CMW 算法和 NMW 算法在中期的梯度值并不突出, 但图 24 中显示, 此时这两种算法仍然具有最优的训练效果, 且训练效果要明显好于其他算法.

下面对以上两个问题作出解答:

解答 1. 在迭代中期, 由于网络参数的初值为任意选取, 导致在迭代中期, 网络权值 w 的方向并没有完全调节好, 造成其中包含的方向信息不准确, 即一部分分量的方向为真实权值的方向, 另一部分分量的方向并不是真实权值方向. 而 CDW 算法仅仅是以网络权值作为动量项进行加速, 所以此时 CDW 算法的梯度虽然大, 但只有部分分量的方向是正确的, 而另一部分分量的方向是错误的. 方向正确的分量会加速训练效果, 方向错误的分量会削弱训练效果, 这两种效果相互补偿, 从而出现上述问题 1) 中的现象.

解答 2. 另外两种算法, NMW 算法和 CMW 算法, 因为有累计梯度 v, 包含了正确的梯度信息, 所以, 当权值分量方向正确时, 会进一步加强加速效果, 当权值分量方向错误时, 会补偿一部分削弱效果. 所以, 尽管这一时期 CMW 算法和 NMW 算法的梯度值没有明显优势, 仍然具有最优的训练效果.

在整个训练过程中, 网络逐渐进入一个较优的局部极大域. 这时, 较小的梯度值更有利于网络的微调, 从而使网络更加靠近局部极大点. 图 26 显示, 在网络靠近局部极大点的过程中, CDW 算法、CMW 算法和 NMW 算法的梯度值显著下降, 下降速度要远大于其他算法. 在迭代末期, 如图 27 所示, CDW 算法、CMW 算法和 NMW 算法的梯度值要远小于其他算法, 这说明, 当网络逐渐进入一个较优的局部极大域后, CDW 算法、CMW 算法和 NMW 算法

相对于其他算法, 具有更好的局部微调能力, 使得网络可以继续收敛. 而其他算法由于后期梯度值较大, 网络参数每次更新步长始终较大, 导致网络在局部极大域内发生振荡, 从而很难收敛.

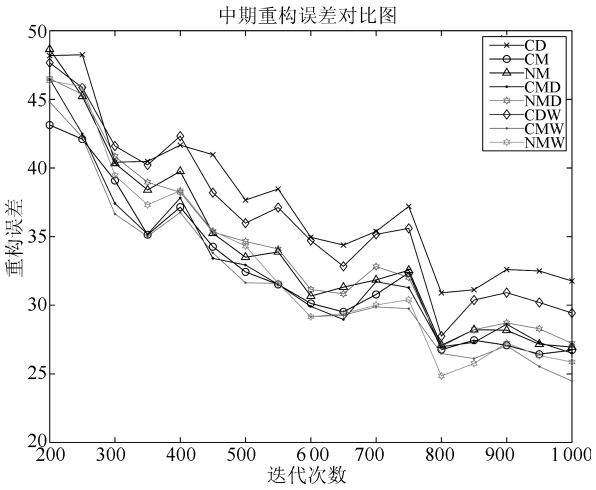

图 25　迭代中期重构误差对比图

Fig. 25　Compassion of reconstruction errors in mid-term of iteration

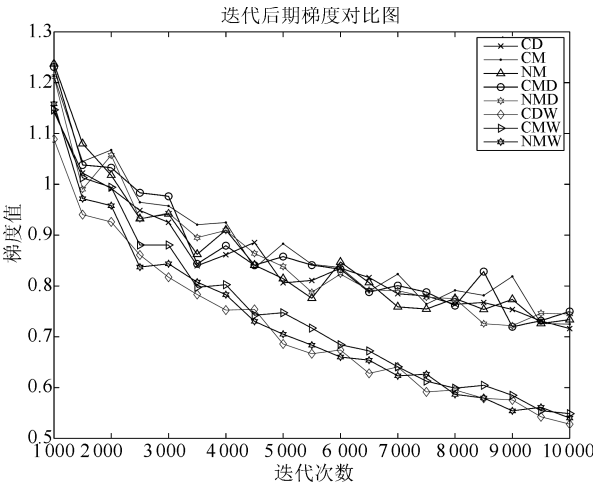

图 26　迭代后期梯度对比图

Fig. 26　Compassion of gradients in late-stage of iteration

5.1.3　网络权值对比分析

图 28 给出了各动量算法的网络权值对比图, 图 29 给出了各动量算法与 CD 算法的动量差值对比图. 从以上两图可以看出, 本文提出权值动量算法是以增大网络权值为代价的. 按照第 4.1 节给出的 Gibbs 采样链收敛性定理, 当权值增大后, Gibbs 采样链的混合率将显著下降, 甚至降为 0, 此时网络将无法训练. 但本文提出的权值动量算法在网络权值急剧增大的情况下仍然能有效地训练网络, 这是因为, 在训练后期, 网络权值增加, 虽然 Gibbs 采样的

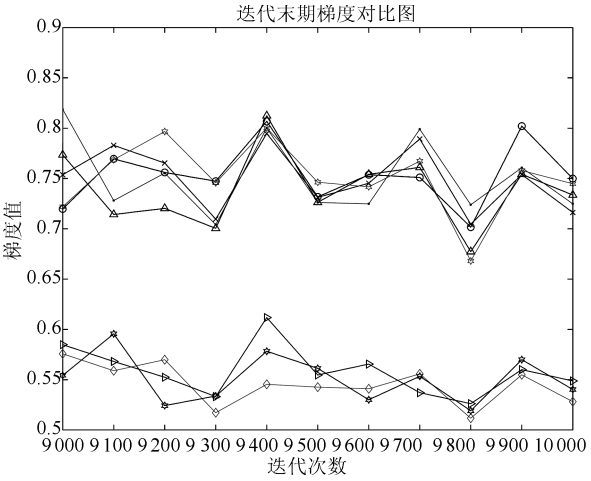

图 27　迭代末期梯度对比图

Fig. 27　Compassion of gradients in late-stage of iteration

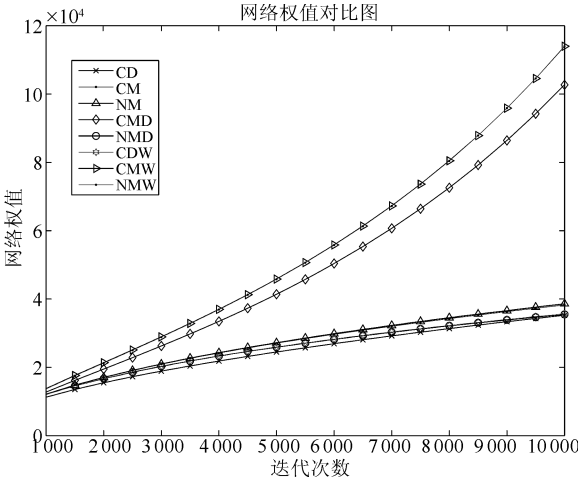

图 28　网络权值对比图

Fig. 28　Compassion of w

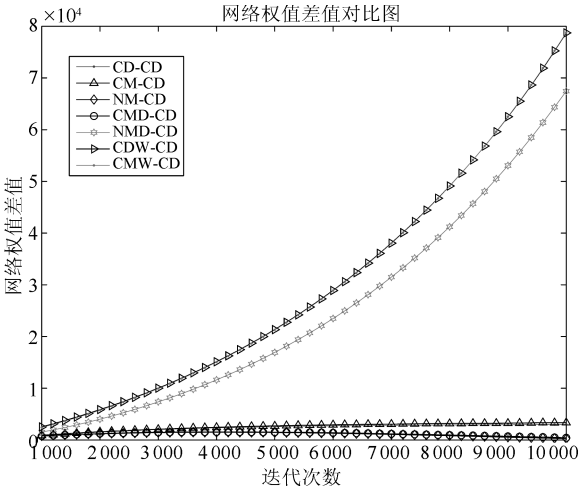

图 29　网络梯度差值对比图

Fig. 29　Compassion of the difference of the gradients

混合率显著降低, 但此时, 由于网络已经训练较好, 可以认为已经达到了较好的局部极大域, 此时, 网络权值 w 的方向已经于真实方向几乎相同, 再加上累积的梯度 v 的参与, 完全可以以这两项对网络进行调节, 所以, 训练后期, 虽然网络权值较大, 但网络仍然能够继续训练.

5.1.4 采样效果对比

图 30 为最后一幅训练图片的原始图, 图 31 ~ 36 分别为各训练算法对应的重构图. 从以上结果可以看出, 通过权值动量算法训练后的网络重构图, 噪点较少, 重构精度更高, 说明权值动量算法具有更好的加速训练效果.

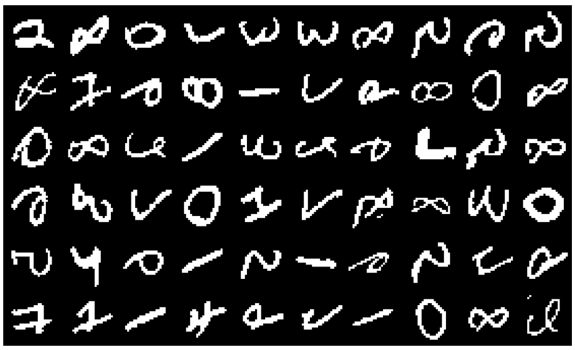

图 30 原始图片

Fig. 30 Original image

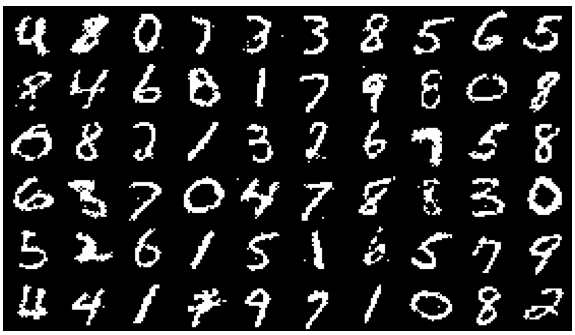

图 31 CD 算法重构图

Fig. 31 Reconstructed image by CD

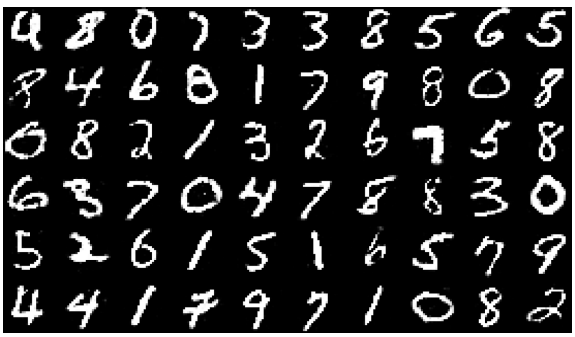

图 32 CM 算法重构图

Fig. 32 Reconstructed image by CM

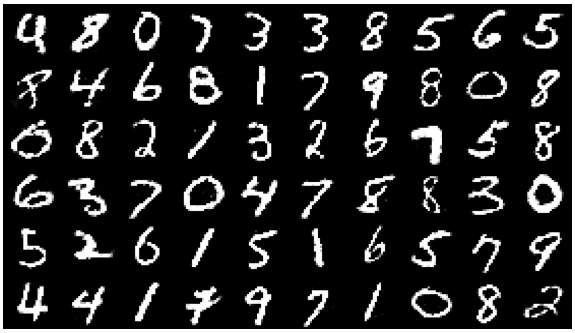

图 33 NM 算法重构图

Fig. 33 Reconstructed image by NM

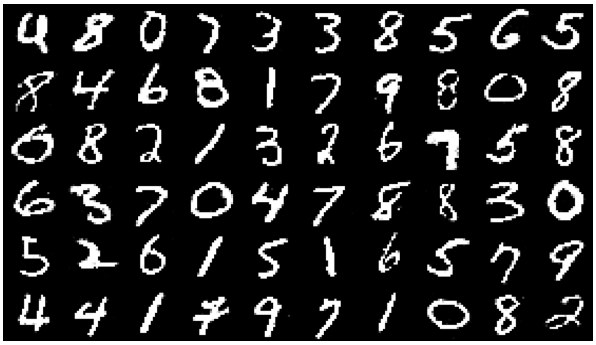

图 34 CDW 算法重构图

Fig. 34 Reconstructed image by CDW

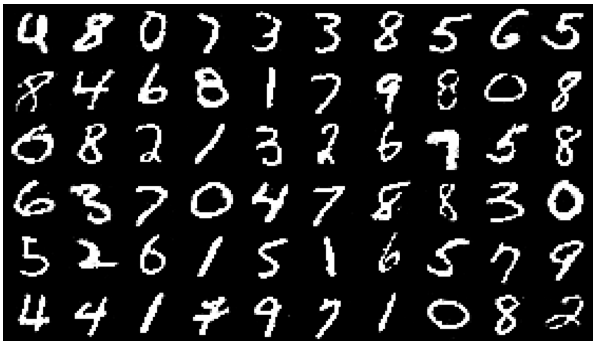

图 35 CMW 算法重构图

Fig. 35 Reconstructed image by CMW

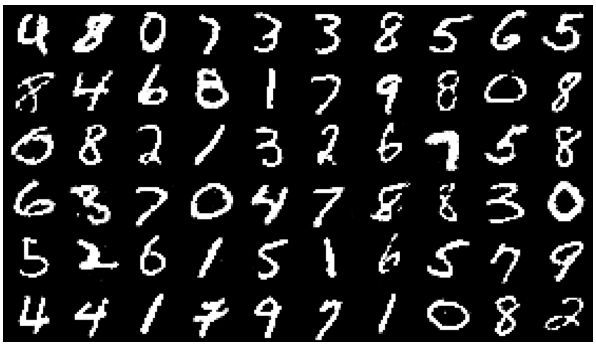

图 36 NMW 算法重构图

Fig. 36 Reconstructed image by NMW

5.2 MNOB 数据集

MNORB 数据集内的数据为 32×32 的灰度图, 如图 37 所示. 本文设计的 RBM 网络模型为 $1\,024 \times 800$, 输入层有 1 024 个节点, 对应灰度图的 1 024 个像素点, 隐层有 800 个节点, 用来提取输入层数据的特征. 训练迭代次数为 10 000 次. 网络结构和初始参数如表 4 所示. 仿真结果如图 38 所示.

图 37　原始图片

Fig. 37　Original image

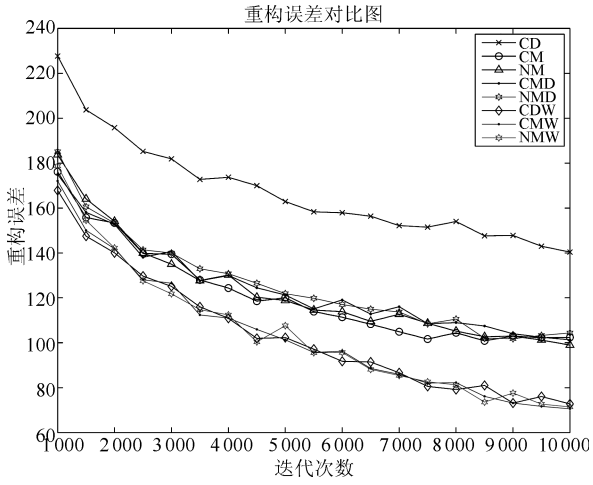

图 38　重构误差对比图

Fig. 38　Compassion of reconstruction errors

图 38 给出了 MNOB 数据集下各算法重构误差对比图. 从图 38 中可以看出, CDW 算法、CMW 算法和 NMW 算法的训练曲线与 CD 算法的间距较大, 远大于 CM 算法、NM 算法、CMD 算法和 NMD 算法. 仿真结果说明本文提出的权值动量算法相对于现有动量算法, 具有更好的加速效果, 且加速效果明显.

表 4　网络参数值

Table 4　The value of network parameters

网络参数	初始值
a	$\text{zeros}(1, 1\,024)$
b	$\text{zeros}(1, 800)$
w	$0.1 \times randn(1\,024, 800)$
η	0.01
μ	0.9

5.3 CIFAR10 数据集

CIFAR-10 数据集内的数据为 $32 \times 32 \times 3$ 的彩色图, 如图 39 所示. 本文设计的 RBM 网络模型为 $3\,072 \times 2\,000$, 输入层有 3 072 个节点, 对彩色图的 3 072 个像素点, 隐层有 2 000 个节点, 用来提取输入层数据的特征. 训练迭代次数为 10 000 次. 网络结构和初始参数设定如表 5 所示. 仿真结果如图 40 所示.

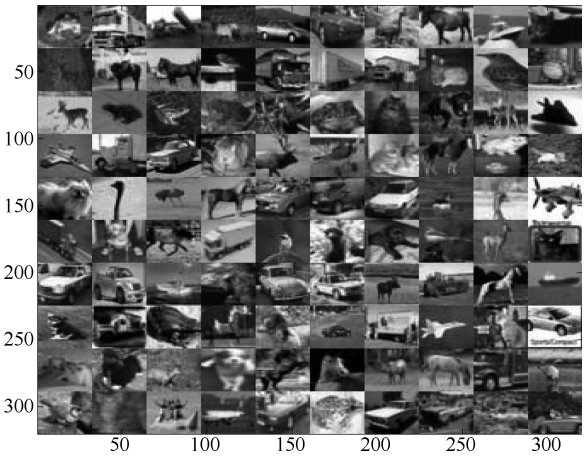

图 39　原始图片

Fig. 39　Original image

表 5　网络参数值

Table 5　The value of network parameters

网络参数	初始值
a	$\text{zeros}(1, 3\,072)$
b	$\text{zeros}(1, 2\,000)$
w	$0.1 \times randn(3\,072, 2\,000)$
η	0.01
μ	0.9

图 40 给出了 CIFAR10 数据集下各算法重构误差对比图. 从图 40 中可以看出, CDW 算法、CMW 算法和 NMW 算法的训练曲线与 CD 算法的间距较大, 远大于 CM 算法、NM 算法、CMD 算法和 NMD 算法. 仿真结果说明本文提出的权值动量算法相对于现有动量算法, 具有更好的加速效果, 且加速效果明显.

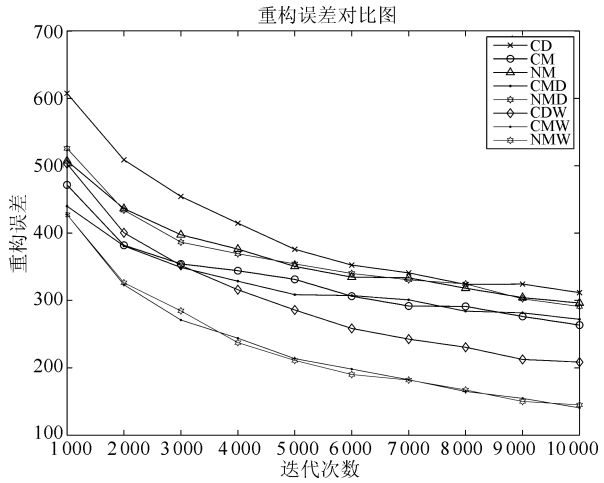

图 40 重构误差对比图

Fig. 40 Compassion of reconstruction errors

5.4 CIFAR100 数据集

CIFAR-100 数据集内的数据为 $32 \times 32 \times 3$ 的彩色图, 如图 41 所示. 本文设计的 RBM 网络模型为 $3\,072 \times 2\,000$, 输入层有 3 072 个节点, 对彩色图的 3 072 个像素点, 隐层有 2 000 个节点, 用来提取输入层数据的特征. 训练迭代次数为 10 000 次. 网络结构和初始参数设定如表 6 所示. 仿真结果如图 42 所示.

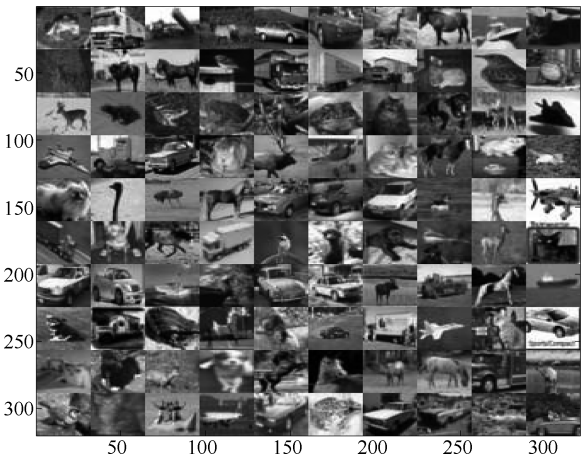

图 41 原始图片

Fig. 41 Original image

表 6 网络参数值

Table 6 The value of network parameters

网络参数	初始值
a	zeros$(1, 3\,072)$
b	zeros$(1, 2\,000)$
w	$0.1 \times randn(3\,072, 2\,000)$
η	0.01
μ	0.9

图 42 给出了 CIFAR100 数据集下各算法重构误差对比图. 从图 42 中可以看出, CDW 算法、CMW 算法和 NMW 算法的训练曲线与 CD 算法的间距较大, 远大于 CM 算法、NM 算法、CMD 算法和 NMD 算法. 在该数据集下, CMW 算法和 NMW 算法加速效果明显, 具有最好的加速效果. 仿真结果说明本文提出的权值动量算法相对于现有动量算法, 具有更好的加速效果, 且加速效果明显.

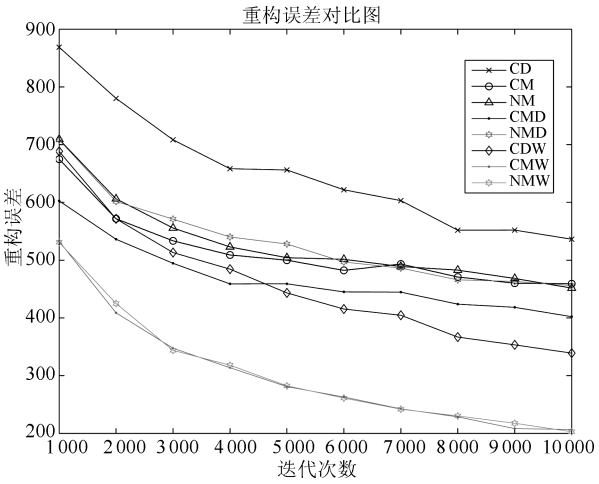

图 42 重构误差对比图

Fig. 42 Compassion of reconstruction errors

5.5 OLIVETTI_FACE 数据集

OLIVETTI_FACE 数据集内的数据为 64×64 的灰度图, 如图 43 所示. 本文设计的 RBM 网络模型为 $4\,096 \times 3\,000$, 输入层有 4 096 个节点, 对灰度图的 4 096 个像素点, 隐层有 3 000 个节点, 用来提取输入层数据的特征. 训练迭代次数为 10 000 次. 网络结构和初始参数设置如表 7 所示. 仿真结果如图 44 所示.

表 7 网络参数值

Table 7 The value of network parameters

网络参数	初始值
a	zeros$(1, 4\,096)$
b	zeros$(1, 3\,000)$
w	$0.1 \times randn(4\,096, 3\,000)$
η	0.01
μ	0.9

图 44 给出了 CIFAR10 数据集下各算法重构误差对比图. 从图中可以看出, CDW 算法、CMW 算法和 NMW 算法的训练明显好于 CM 算法、NM 算法、CMD 算法和 NMD 算法. 仿真结果说明本文提出的权值动量算法相对于现有动量算法, 具有更好的加速效果, 且加速效果明显.

图 43　原始图片

Fig. 43　Original image

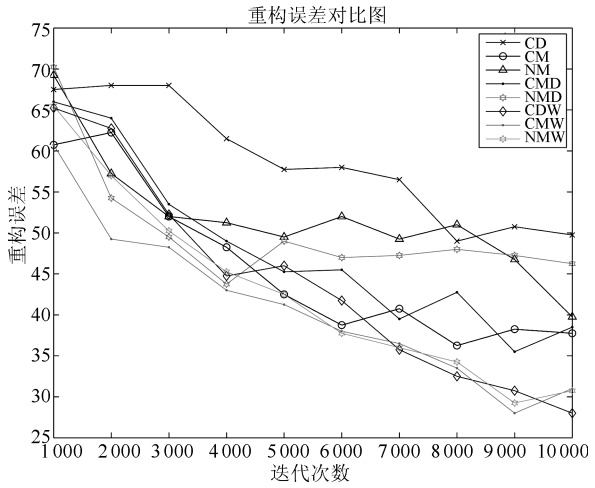

图 44　重构误差对比图

Fig. 44　Compassion of reconstruction errors

6　总结

本文针对现有动量算法在受限玻尔兹曼机网络训练中加速效果较差、训练后期加速性能显著降低的问题, 首先, 基于 Gibbs 采样收敛性定理对现有动量算法进行了理论分析, 证明了现有动量算法的加速效果是以增大网络权值为代价的, 随着网络权值逐渐增大, Gibbs 采样链的收敛性逐渐变差, 这也最终导致了现有动量算法在训练后期逐渐失去加速能力; 然后, 本文通过引入权值衰减项来对网络权值进行控制, 并进行了仿真实验, 实验结果表明, 权值衰减项并不能解决现有动量算法存在的问题, 而且权值衰减项的系数非常难以调节. 系数较大时, 算法不能有效地训练网络; 系数较小时, 不能起到很好的权值控制作用. 于是, 本文接着对网络权值进行研究, 发现网络权值中包含大量真实梯度的方向信息, 这些梯度方向信息可以与现有动量算法结合来加速网络训练, 尤其在训练后期, 当 Gibbs 采样链几乎无法收敛时, 仍然可以用网络权值中保存的梯度方向信息继续对网络进行训练. 基于此, 本文提出了基于网络权值的权值动量算法, 最后分别在 MNIST 数据集、MNOB 数据集、CIFAR10 数据集、CIFAR100 数据集和 FACE 数据集上给出了仿真实验. 实验结果表明, 相对于传统动量算法, 本文提出的权值动量算法在以上 5 个数据集上均具有更好的加速效果, 并且在训练后期仍然能够保持较好的加速性能. 同时, 5 个数据集上的仿真结果表明, 本文提出的动量算法具有一定的普适性.

本文提出的权值动量算法虽然可以很好地弥补现有动量算法的不足, 但仍然是以增大网络权值为代价的. 网络权值过大可能会导致网络的泛化性能降低, 如何在保证加速性能的同时保持网络的泛化性能, 仍有待于进一步研究.

References

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

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

3 Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems 25. Cambridge, MA: MIT Press, 2012.

4 Bengio Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2009, **21**(6): 1−27

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 International Conference on Acoustics Speech and Signal Processing (ICASSP). Vancouver, BC, Canada: IEEE, 2013. 6669−6673

6 Deng L. Design and learning of output representations for speech recognition. In: Neural Information Processing Systems (NIPS) Workshop on Learning Output Representations. Lake Tahoe, USA: NIPS, 2013.

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

8 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)

9 Tian Yuan-Dong. A simple analysis of AlphaGo. *Acta Automatica Sinica*, 2016, **42**(5): 671−675
(田渊栋. 阿法狗围棋系统的简要分析. 自动化学报, 2016, **42**(5): 671−675)

10 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)

11 Geng Jie, Fan Jian-Chao, Chu Jia-Lan, Wang Hong-Yu. Research on marine floating raft aquaculture SAR image target recognition based on deep collaborative sparse coding network. *Acta Automatica Sinica*, 2016, **42**(4): 593−604
(耿杰, 范剑超, 初佳兰, 王洪玉. 基于深度协同稀疏编码网络的海洋浮筏 SAR 图像目标识别. 自动化学报, 2016, **42**(4): 593−604)

12 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 International Conference on Acoustics, Speech and Signal Processing (ICASSP). Vancouver, BC, Canada: IEEE, 2013. 8599−8603

13 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). Chia Laguna Resort, Sardinia, Italy: AISTATS, 2010. 201−208

14 Smolensky P. Information processing in dynamical systems: foundations of harmony theory. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1: Foundations*. Cambridge: MIT Press, 1986. 194−281

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

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

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

18 Desjardins G, Courville A C, Bengio Y, Vincent P, Dellaleau O. Tempered Markov chain Monte Carlo for training of restricted Boltzmann machines. In: Proceedings of the 13th International Workshop on Artificial Intelligence and Statistics (AISTATS). Chia Laguna Resort, Sardinia, Italy: AISTATS, 2010. 45−152

19 Cho K, Raiko T, Ilin A. Parallel tempering is efficient for learning restricted Boltzmann machines. In: Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN). Barcelona, Spain: IEEE, 2010. 3246−3253

20 Brakel P, Dieleman S, Schrauwen B. Training restricted Boltzmann machines with multi-tempering: harnessing parallelization. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN). Belgium: Evere, 2012. 287−292

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

22 Polyak B T. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 1964, **4**(5): 1−17

23 Fischer A, Igel C. Empirical analysis of the divergence of Gibbs sampling based learning algorithms for restricted Boltzmann machines. *Artificial Neural Networks*. Berlin Heidelberg: Springer, 2010. 208−217

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

25 Sutskever I, Martens J, Dahl G, Hinton G. On the importance of initialization and momentum in deep learning. In: Proceedings of the 30th International Conference on Machine Learning. Atlanta, Georgia, USA: ICML, 2013. 1139−1147

26 Zaręba S, Gonczarek A, Tomczak J M, Świątek J. Accelerated learning for restricted Boltzmann machine with momentum term. *Progress in Systems Engineering*. Switzerland: Springer International Publishing, 2015. **330**: 187−192

27 Bengio Y, Delalleau O. Justifying and generalizing contrastive divergence. *Neural Computation*, 2009, **21**(6): 1601−1621

28 Carreira-Perpiñán M Á, Hinton G E. On contrastive divergence learning. In: Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS). Barbados: The Society for Artificial Intelligence and Statistics, 2005. 59−66

29 Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998, **86**(11): 2278−2324

30 Krizhevsky A. Learning multiple layers of features from tiny images [Master dissertation], University of Toronto, Toronto, Canada, 2009.

31 Roweis S. available: http://www.cs.nyu.edu/~roweis/, July 2, 2016.

32 Torralba A, Fergus R, Freeman W T. 80 million tiny images: a large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008, **30**(11): 1958−1970

33 LeCun Y, Huang F J, Bottou L. Learning methods for generic object recognition with invariance to pose and lighting. In: Proceedings of the 2004 IEEE Computer Society Conference Computer Vision and Pattern Recognition. Washington, DC, USA: IEEE, 2004. **2**(2): II-97-104

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 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.)