支持增量式更新的大数据特征学习模型

卜范玉[1,2],陈志奎[1],张清辰[1]
BU Fanyu[1,2], CHEN Zhikui[1], ZHANG Qingchen[1]

1.大连理工大学 软件学院,辽宁 大连 116620
2.内蒙古财经大学 职业学院,呼和浩特 010010
1.School of Software Technology, Dalian University of Technology, Dalian, Liaoning 116620, China
2.College of Vocation, Inner Mongolia University of Finance and Economics, Hohhot 010010, China

BU Fanyu, CHEN Zhikui, ZHANG Qingchen. Incremental updating method for big data feature learning. Computer Engineering and Applications, 2015, 51(12):21-26.

Abstract: Data are generating at extremely high speed in the era of big data, whose contents and features are in the dynamic changes. Thus, the learning algorithm for neural networks should not only be able to adapt new instances, but also preserve the prior knowledge. However, the feed-forward neural network trained by typically Back-Propagation(BP) algorithm is not incremental in nature. This paper proposes an incremental back-propagation model for training neural networks. The goal of incremental leaning is achieved by adjusting the parameters and structures of the feed-forward neural network. The parameters are incrementally adapted by optimizing an objective function. The network topology is adapted by increasing the number of hidden neurons only if the parameters adaption perturbs the prior knowledge severely. After updating the model, the Singular Value Decomposition(SVD) of the weight matrix is performed to remove the redundant connections of each newly added hidden unit. Experimental results demonstrate that the proposed model can adjust its parameters and structure depending on the requirement of the big data process in real time with preserving the prior knowledge as much as possible in evolving environments.

Key words: big data; feed-forward neural networks; incremental learning; Singular Value Decomposition(SVD)

摘 要: 大数据具有高速变化特性,其内容与分布特征均处于动态变化之中,目前的前馈神经网络模型是一种静态学习模型,不支持增量式更新,难以实时学习动态变化的大数据特征。针对这个问题,提出一种支持增量式更新的大数据特征学习模型。通过设计一个优化目标函数对参数进行快速增量式更新,为了在更新过程中保持网络的原始知识,最小化平方误差函数。对于特征变化频繁的数据,通过增加隐藏层神经元数目网络对结构进行更新,使得更新后的网络能够实时学习动态变化大数据的特征。在对网络参数与结构更新之后,通过权重矩阵SVD分解对更新后的网络结构进行优化,删除冗余的网络连接,增强网络模型的泛化能力。实验结果表明提出的模型能够在尽可能保持网络模型原始知识的基础上,通过不断更新神经网络的参数与结构实时学习动态大数据的特征。

关键词: 大数据;前馈神经网络;增量式学习;奇异值分解(SVD)
文献标志码: A **中图分类号:** TP391 doi:10.3778/j.issn.1002-8331.1503-0165

1 引言

近年来,随着电子商务、科学研究等领域的迅速发展,数据量正以惊人的速度增长[1-3]。大数据时代已经到来,大数据的内容与分布特征均处于高速动态变化之中,并要求被实时分析和处理,这与传统数据挖掘有着本质不同,尤其是在欺诈检测,大规模在线监测与检测等;需要做到数据的实时分析和处理[4]。大数据的高速变化特性为其特征学习带来了巨大的挑战[5]。

深度学习的出现为大数据特征学习提供了新的解决思路[6-7]。如今,深度学习在图像分析、语音识别及自

基金项目: 国家自然科学基金重点项目(No.U1301253);辽宁省自然科学基金(No.201202032)。
作者简介: 卜范玉(1980—),男,博士研究生,讲师,研究领域为大数据、机器学习;陈志奎(1968—),男,博士,教授,研究领域为大数据、物联网;张清辰(1988—),男,博士研究生,研究领域为大数据、深度学习。E-mail:623759909@qq.com
收稿日期: 2015-03-16 **修回日期:** 2015-05-04 **文章编号:** 1002-8331(2015)12-0021-06

然语言处理方面取得了巨大的进展[8]。大数据具有高速变化特性,数据以极快的速度产生,其内容和分布特征均处于高速动态变化之中。同时,大数据要求得到及时分析与处理。典型的深度学习采用前馈神经网络堆叠而成,前馈神经网络是一种静态学习模型,导致深度学习难以学习处于高速动态变化中的大数据特征。

针对这个问题,本文提出一种支持增量更新的大数据特征学习模型。提出的模型在更新网络结构过程中,只需要加载当前数据实例,无需将整个大数据加载到内存中,因此能够有效学习规模巨大的大数据特征。提出的模型同时对网络模型的参数和结构进行更新,在参数更新过程中,在原始参数基础上引入一阶近似思想,避免通过迭代的方式求参数增量,提高参数更新效率,使得更新的模型能够快速学习动态变化的大数据的特征,最大程度上满足大数据特征学习的实时性要求,同时尽可能地保持神经网络模型的原始知识;在结构更新过程中,将新引入的参数与原始参数结合作为网络参数的初始值,充分利用原始参数提供的知识,加快参数求解的收敛速度,使得模型快速收敛,实时学习大数据特征;最后通过SVD优化网络结构和权重矩阵,删除网络模型冗余的连接,提高网络的泛化能力。

2　相关工作

作为深度学习典型的模块,双层前馈神经网络包括输入层数据,隐藏层特征和网络的实际输出值[9]。双层前馈神经网络的第一层通过编码函数 f 将输入层数据 $\boldsymbol{x}$ 映射到隐藏层特征 h,如公式(1)所示。

$$h=f(\boldsymbol{x})=s_f(\boldsymbol{w}^{(1)}\boldsymbol{x}+\boldsymbol{b}^{(1)}) \tag{1}$$

其中,s_f 为一个非线性激活函数,常用的激活函数为sigmoid函数,即 $f(z)=1/(1+\mathrm{e}^{-z})$。

第二层通过解码函数将隐藏层特征 h 映射到网络的实际输出 $\boldsymbol{y}'$,如公式(2)所示。

$$\boldsymbol{y}'=g(h)=s_g(\boldsymbol{w}^{(2)}\boldsymbol{x}+\boldsymbol{b}^{(2)}) \tag{2}$$

其中,$\theta=\{\boldsymbol{w}^{(1)},\boldsymbol{b}^{(1)},\boldsymbol{w}^{(2)},\boldsymbol{b}^{(2)}\}$ 为网络参数,$\boldsymbol{w}^{(1)}$ 和 $\boldsymbol{w}^{(2)}$ 分别代表网络的权重矩阵,$\boldsymbol{b}^{(1)}$ 和 $\boldsymbol{b}^{(2)}$ 是偏置向量。

前馈神经网络通过最小化网络实际输出值 $\boldsymbol{y}'$ 和理想输出 $\boldsymbol{y}$ 训练模型参数,目标函数定义如公式(3)所示。

$$J_{\mathrm{AE}}(\theta)=\sum_{\boldsymbol{x}\in D}L(\boldsymbol{y},g(f(\boldsymbol{x}))) \tag{3}$$

其中,L 为损失函数,最常用的损失函数为平方差函数,即 $L(\boldsymbol{y},\boldsymbol{y}')=\|\boldsymbol{y}-\boldsymbol{y}'\|^2$。

在求解参数过程中,通常先采用反向传播算法(BP)求解目标函数对权重矩阵与偏置向量的梯度[10],然后利用各种优化方法对参数进行更新。典型的参数更新优化算法包括梯度下降法、基于二阶大数据的参数更新算法以及最小二乘法等[11]。尽管以上这些方法在一定程度上都能够求解神经网络的参数,但是这些方法都属于静态的参数求解方法。静态的参数求解方法使得前馈神经网络难以有效学习处于高速动态变化之中数据的特征。针对这个问题,研究人员相继提出了多种增量式神经网络模型。两种典型的增量式神经网络模型分别是基于在线学习方式的参数更新模型和基于增加隐藏层神经元的结构更新模型。增量式BP算法、随机梯度下降算法和随机二阶导数算法均属于基于在线学习方式的参数更新模型[12],这种模型每次学习一个实例并逐步更新神经网络的参数,使网络模型能够不断学习新数据的特征。近年来在线学习方式获得了学术界的巨大关注。然而在线学习方式对参数的更新速度过快,使得整个模型对原始参数的遗忘速率过快,导致更新后的模型无法再学习历史数据的特征。尽管增量BP算法通过约束区间来限制参数的变化速度,实现对网络原始知识的保持,但该方法计算复杂,不适合用于快速变化的大数据特征学习。文献提出一种基于实例敏感度的线性学习方法,通过对新的实例分配一个权重来加快网络模型对新数据的适应速度,该算法能够有效学习动态环境中的数据特征,然而却无法有效保持模型的原始知识。另一种典型的增量式学习模型采用增加网络隐藏层神经元的方式对模型进行更新,有代表性的算法包括基于资源分配网络的径向基神经网络模型、增量式支持向量机及自适应拓扑结构的增量式神经网络模型(OANN)等[13]。当新的数据实例到来后,增加网络隐藏层的神经元,并重新求解网络参数,使网络模型能够学习新数据的特征。这种更新方法会不断增加网络隐藏层神经元数目,使得网络规模迅速增大。随着网络规模的增大,冗余连接数目随之增多,模型的训练速度、学习能力和泛化能力急剧降低。

3　支持增量式更新的大数据特征学习方法

本文提出的支持增量式更新的大数据特征学习方法通过参数更新、结构更新和结构优化三个步骤完成。一个自动编码神经网络的知识通过参数和结构体现,其中网络拓扑结构所隐藏的知识比参数知识更为丰富。因此本文提出的方法优先更新参数,对于特征变化频繁的数据集,才通过结构的更新来学习快速变化的数据特征,这样确保在最大的程度上保持网络的原始知识。

3.1　基于一阶近似的参数更新算法

考虑如图1所示的双层前馈神经网络,其中 $\boldsymbol{x}=\{x_1,x_2,\cdots,x_n\}$ 代表数据实例的输入,$\boldsymbol{y}=\{y_1,y_2,\cdots,y_m\}$ 代表数据实例的真实输出,$f(\boldsymbol{x},\theta)$ 表示当参数为 θ 时,$\boldsymbol{x}$ 经过非线性映射 f 得到的实际输出。

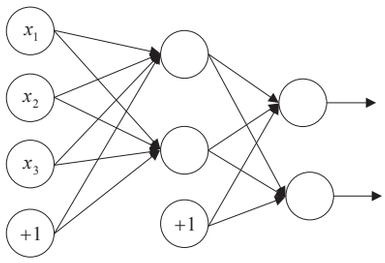

图1 双层前馈神经网络

对于新的数据实例 $\{\boldsymbol{x},\boldsymbol{y}\}$,为了使得模型能够学习新数据的特征,即网络模型的适应性,定义平均平方误差函数:

$$J(\boldsymbol{x},\theta+\Delta\theta)=\frac{1}{2}\Delta\boldsymbol{y}_{\text{new}}^{\text{T}}\cdot\Delta\boldsymbol{y}_{\text{new}} \tag{4}$$

其中 $\Delta\boldsymbol{y}_{\text{new}}$ 表示实际值 $\boldsymbol{y}$ 和预测值之间的误差:

$$\Delta\boldsymbol{y}_{\text{new}}=f(\boldsymbol{x},\theta+\Delta\theta)-\boldsymbol{y} \tag{5}$$

其中 θ 代表模型原始参数(模型原始知识),$\Delta\theta$ 代表参数增量(更新的知识)。网络的激活函数采用 sigmoid 函数,即 $f(z)=1/(1+\mathrm{e}^{-z})$,其对自变量的导数为 $f'(z)=f(z)(1-f(z))$。则 $\Delta\theta$ 的求解过程如下:根据误差函数式(4)可以定义代价函数如式(6)所示:

$$J(\boldsymbol{x},\theta+\Delta\theta)=\frac{1}{2}\Delta\boldsymbol{y}_{\text{new}}^2=\frac{1}{2}(f(\boldsymbol{x},\theta+\Delta\theta)-\boldsymbol{y})^2 \tag{6}$$

代价函数式(6)的一阶近似如式(7)所示:

$$J(\boldsymbol{x},\theta+\Delta\theta)=\frac{1}{2}(f(\boldsymbol{x},\theta+\Delta\theta)-f(\boldsymbol{x},\theta)+f(\boldsymbol{x},\theta)-\boldsymbol{y})^2\cong \frac{1}{2}(\frac{\partial f(\boldsymbol{x},\theta)}{\partial\theta}\Delta\theta+\Delta\boldsymbol{y})^2 \tag{7}$$

代价函数式(7)对 $\Delta\theta$ 求导,令导数等于0,求得 $\Delta\theta$ 的计算公式如式(8)所示:

$$\frac{\partial f(x,\theta)}{\partial\theta}\Delta\theta=-\Delta\boldsymbol{y}$$

$$\Delta\theta=-u\cdot(\frac{\partial f(\boldsymbol{x},\theta)}{\partial\theta})^{-1}\Delta\boldsymbol{y} \tag{8}$$

其中 u 为学习效率。

对于参数更新而言,给定一个新的数据样本 $\{\boldsymbol{x},\boldsymbol{y}\}$,寻找参数增量 $\Delta\theta$ 来更新原始参数 θ,使得误差函数 $J(\boldsymbol{x},\theta+\Delta\theta)$ 最小。基于一阶近似的参数更新算法描述如下。

算法1 基于一阶近似的参数更新算法

(1)通过前向传播计算隐藏层神经元及神经网络模型的输出值 $f(\boldsymbol{x},\theta)$。

(2)计算网络模型的输出值与理想输出值 $\boldsymbol{y}$ 的差值 $\Delta\boldsymbol{y}$。

(3)通过反向传播算法计算网络模型输出值 $f(\boldsymbol{x},\theta)$ 对原始参数 θ 的偏导数 $\frac{\partial f(\boldsymbol{x},\theta)}{\partial\theta}$。

(4)计算偏导数矩阵 $\frac{\partial f(\boldsymbol{x},\theta)}{\partial\theta}$ 的逆矩阵 $(\frac{\partial f(\boldsymbol{x},\theta)}{\partial\theta})^{-1}$。

(5)根据公式(8)计算网络模型的参数增量 $\Delta\theta$,将网络模型的参数更新为 $\theta+\Delta\theta$。

(6)将 $\theta+\Delta\theta$ 作为更新的网络模型的参数;否则转至网络模型的结构更新算法。

通过算法1的步骤可知,基于一阶近似的参数更新算法不需要进行迭代,主要运算是计算偏导数矩阵的逆矩阵,因此算法1的时间复杂度为 $O(m)$,m 表示网络模型的参数数目。

值得注意的是,算法1在更新模型的过程中,只需要将当前新的数据实例加载到内存之中,这点有利于学习规模巨大的大数据特征,除此之外,算法1避免通过迭代更新参数,因此参数更新速度能够在最大程度上满足大数据特征学习的实时性要求。

3.2 基于增加隐藏层神经元的结构更新

对于特征变化频繁的数据,本文通过增加隐藏层神经元数目来更新网络模型,使得更新的模型能够学习动态变化的大数据特征,同时尽可能地保持模型原始知识。对于图1所显示的自动编码网络,当增加一个隐藏层神经元后,其结构变成如图2所示。

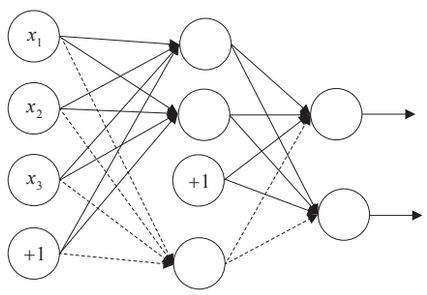

图2 更新的双层前馈神经网络

每当增加一个隐藏层神经元,需要调整权重矩阵和偏置向量的形式,权重矩阵 $\boldsymbol{w}^{(1)}$ 和 $\boldsymbol{w}^{(2)}$ 需要分别增加一行和一列;同时,两个偏置向量 $\boldsymbol{b}^{(1)}$ 和 $\boldsymbol{b}^{(2)}$ 分写需要增加一个分量。同理,如果增加 m 个隐藏层神经元,则每个权重矩阵需要增加 m 行 m 列,偏置向量需要增加 m 个分量,以适应网络连接。

对于增加隐藏层神经元之后的网络参数,本文将增加的分量其初始值设为0。设原始网络结构的当前参数为 $\theta=\{\boldsymbol{w}^{(1)},\boldsymbol{b}^{(1)};\boldsymbol{w}^{(2)},\boldsymbol{b}^{(2)}\}$,更新后的网络结构参数形式如公式(9)所示:

$$\boldsymbol{w}^{(i)'}=\begin{bmatrix} & & & (m) & \\ & \boldsymbol{w}^{(i)} & 0 & \cdots & 0 \\ & 0 & 0 & \cdots & 0 \\ (m) & \vdots & \vdots & & \vdots \\ & 0 & 0 & \cdots & 0 \end{bmatrix}\quad \boldsymbol{b}^{(i)'}=\begin{bmatrix} & b^{(i)} \\ & 0 \\ (m) & \vdots \\ & 0 \end{bmatrix} \tag{9}$$

其中 $i=1,2$;m 为新增加的隐藏层神经元的数目。

更新结构后,以 $\theta'=\{\boldsymbol{w}^{(i)'},\boldsymbol{b}^{(i)'}\}$ 为初始参数,利用BP算法和梯度下降法求得最终参数。给定一个新的数据

样本 $\{\boldsymbol{x},\boldsymbol{y}\}$，基于增加隐藏层神经元的结构更新算法描述如下。

算法2 基于增加隐藏层神经元的结构更新

(1)对网络参数 θ' 进行初始化。

(2)利用前向传播计算隐藏层输出值与网络模型输出值 $f(\boldsymbol{x},\theta')$。

(3)计算网络模型输出值 $f(\boldsymbol{x},\theta')$ 与理想输出值 $\boldsymbol{y}$ 的差值 $\Delta\boldsymbol{y}$ 及误差函数 $J(\boldsymbol{x},\theta')=\frac{1}{2}\Delta\boldsymbol{y}^{\mathrm{T}}\cdot\Delta\boldsymbol{y}$。

(4)利用反向传播计算误差函数 $J(\boldsymbol{x},\theta')$ 对网络参数 θ' 的偏导数。

(5)利用梯度下降法更新网络参数。

(6)重复步骤(2)到步骤(5)直至收敛。

算法2将新引入的参数与原始参数结合作为网络参数的初始值，充分利用原始参数提供的知识，加快参数求解的收敛速度，使得模型快速收敛，实时学习大数据特征；从算法2的步骤中可知，基于增加隐藏层神经元的结构更新算法主要采用反向传播算法求解网络误差函数对网络参数的偏导数，因此算法的时间复杂度与方向传播算法时间复杂度相同，每次迭代的运算量为 $O(m)$，m 为网络中的参数数目。

在更新结构过程中，随着隐藏层神经元数目的增加，连接数目会随之增加，会产生大量的冗余连接。这些冗余连接会降低网络的泛化能力，容易使得模型发生过度拟合。因此在自动编码网络更新之后，需要对网络连接结构进行优化。

网络连接的重要性可以通过权重矩阵中元素的取值反映，权重矩阵中元素值大的元素对应的连接相对更为重要。因此本文通过权重矩阵的SVD分解[14-15]寻找网络中的冗余连接并删除冗余连接，通过SVD分解优化权重矩阵和网络结构。

假设权重矩阵为 $\boldsymbol{w}$，对 $\boldsymbol{w}$ 进行SVD分解得到：

$$\boldsymbol{w}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} \tag{10}$$

根据SVD分解的定义，对角矩阵 $\boldsymbol{\Sigma}$ 中，那些等于0或者接近于0的权重表示该连接在整个网络中是冗余的，或者所起的作用是微乎其微的，因此可以删除这些连接，达到连接的优化。

当某个隐藏层神经元的连接都被删除后，则删除该隐藏层神经元，对整个自动编码网络进行优化。标准SVD分解的时间复杂度为 $O(n^3)$。尽管标准SVD分解的时间复杂度很高，但是并不是每次对网络结构更新后，都要执行SVD分解对网络结构进行优化，只有当增加神经元特征学习性能下降时，即网络连接出现大量冗余时，才执行SVD分解对网络结构进行优化。因此SVD分解并不影响增量式特征学习模型的实时性和高效性。

4 实验结果与分析

本文通过两个专业的图像数据集验证提出算法的有效性，将本文的算法同下列两个具有代表性的增量式神经网络学习算法进行对比，即增量BP算法和OANN算法。其中增量BP算法通过一个自适应区间来限制权重的变化量，并通过不断调整网络结构来适应新的数据。OANN算法通过引入遗忘函数来加快网络对新数据的适应性。两个典型的分类数据集来验证算法的有效性，分别是MNIST和STL-10。

4.1 MNIST数据集

MNIST是一个典型的分类数据集，包含50 000张训练图片和10 000张测试图片，每张图片代表一个0~9之间的数字，其大小为28像素×28像素[16]。首先应用训练数据集初始化参数(包括权重矩阵和偏置向量)，获得网络模型的原始参数 θ，然后从测试集中不断地随机选取图片作为新增数据实例，训练网络参数，并通过其他测试数据作为测试数据集，验证模型的有效性。由于MNIST数据集的训练集和测试集数据具有相同的数据分布特征，因此参数更新量很小，不需要通过增加隐藏层神经元数目即可使得更新的网络模型学习新数据的特征。

本文采用平均平方误差(*MSE*)和参数增量误差(*PIE*)作为指标衡量本文提出算法的有效性。其中平均平方误差用于度量网络的泛化能力，即更新模型对在学习处于动态变化中的大数据的特征方面的适应性，平均平方误差越小，说明更新模型对新数据的适应性越强；参数增量误差用来度量网络参数更新前后的变化量，即更新模型对网络原始知识的保持性，参数增量误差越小，表明更新模型对网络原始支持的保持性越好。每个算法重复执行10次，实验结果如表1和表2所示。

从表1可以看出，本文提出的算法充分利用函数的一阶近似性质，使得参数更新能够充分反映新数据的特征，在10次运行结果中，有6次平均平方误差最低，这表明本文的算法在对参数进行更新后，能够更有效地适应新数据的变化，更新模型能够有效对新数据进行分类学

表1　三种算法在MNIST数据集的*MSE*

	1	2	3	4	5	6	7	8	9	10
BP算法	0.14	0.16	0.13	0.15	0.16	0.15	0.17	0.15	0.09	0.18
OANN算法	0.12	**0.11**	0.15	0.12	0.11	0.09	**0.12**	0.16	0.09	**0.11**
本文算法	**0.09**	0.13	**0.11**	**0.11**	**0.08**	**0.11**	0.15	**0.09**	**0.07**	0.12

表2 三种算法在MNIST数据集的*PIE*

	1	2	3	4	5	6	7	8	9	10
BP算法	0.26	0.27	**0.23**	0.27	0.20	**0.25**	0.22	**0.22**	0.1	0.27
OANN算法	0.29	0.32	0.38	0.29	0.31	0.32	0.21	0.29	0.3	0.41
本文算法	**0.10**	**0.21**	0.25	**0.22**	**0.18**	0.29	**0.15**	0.23	**0.1**	**0.21**

习。此外,OANN算法采用遗忘函数来加快更新模型对新数据的适应性,因此在10次运行结果中,有4次平均平方误差最小,其总体结果仅次于本文提出的算法。

表2显示的结果表明,在10次运行结果中,本文的算法有7次参数增量误差最小,这是由于本文的算法在更新参数过程中直接在网络结构原始参数的基础上求其增量,通过增量的一阶近似来体现新数据的特征,能够尽力保持网络模型的原始参数。实验结果充分表明本文提出的算法在更新参数过程中,能够使得模型较好地保持网络结构的原始参数。与此同时,增量BP算法通过参数更新区间,限定参数的更新尺度,在10次运行结果中,有3次参数增量误差最小,结果非常接近本文提出的算法。然后通过表1可以看出,增量BP算法对参数更新量的限制,降低了更新模型学习新数据特征的能力。

表1和表2结果表明本文提出的算法通过参数的更新使得更新模型能够有效学习新数据的特征,同时有效保持了网络结构的原始参数。

图3显示了三个算法的运算时间。图3的显示结果表明,本文提出的参数更新算法所花费的时间最少,这是因为本文提出的算法在网络结构原始参数之上,通过一次近似直接求出参数的增量对参数进行更新,无需进行迭代。相比之下,增量BP算法和OANN算法对参数进行更新过程中,要重新迭代,不同的迭代次数导致算法运行的时间不同,取决于数据实例的特征。OANN算法将非线性的求微分问题转换为求解线性系统方程,尽管在更新参数过程中需要迭代运算,但是每次迭代所花费的时间小于增量BP算法,因此算法的整体运行时间复杂度低于增量BP算法。总之,对于只需要对参数进行更新的网络更新模型而言,本文提出的算法在更新效率上远远高于增量BP算法和OANN算法。

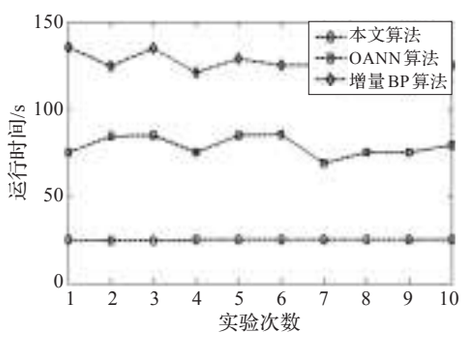

图3 三种算法运行时间对比

4.2 STL-10数据集

STL-10数据集是一个专业的图像识别数据集,包含500个训练图像和800个测试图像,另外包含100 000张未标签的图像。每张图像大小为96×96的RGB图像[17]。由于STL-10数据集每个子集图像的特征相差明显,因此需要通过调整结构的方式来更新模型,使得更新的网络结构能够有效学习新数据的特征。在实验过程中,首先利用训练集训练网络结构,获得网络结构的初始化参数,即网络结构的原始知识,设定网络结构隐藏层神经元数目为120。在对新的数据实例进行学习过程中,逐步增加隐藏层神经元的数目,更新网络结构,学习新的数据特征。本文首先测试三个算法增加不同隐藏层神经元数目对平均平方误差的影响,实验结果如图4所示。

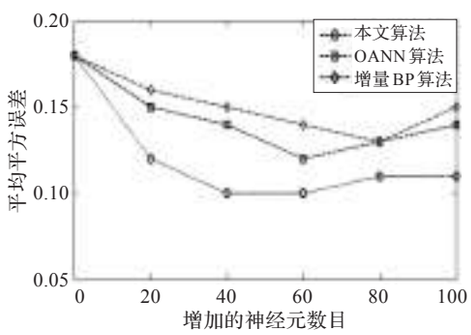

图4 三种算法平均平方误差对比

图4显示,随着隐藏层神经元数目的增加,网络结构对新数据分类的平均误差在逐渐降低,充分说明增加隐藏层神经元的数目可以降低对新数据分类的平均平方误差,即基于增加隐藏层神经元数目的网络结构更新能够有效学习动态变化数据的特征。从图4中可以看出,当增加相同隐藏层神经元数目时,本文提出算法的平均平方误差最小,说明本文提出网络结构更新算法对新数据的适应性最强。

从对网络结构的原始知识的保持性角度看,增加的神经元数目越少,说明对网络结构的原始知识保持性越好。从图4中可以看出,本文提出的算法当隐藏层神经单元增加到40个时,平均平方误差最小,而OANN算法需要增加60个神经元,增量式BP算法需要增加80个神经元。因此本文提出的模型更新算法对网络结构的保持性最好。

当隐藏层神经元数目增加到一定数据量之后,OANN算法和增量式BP算法的平均平方误差均有所提高,这是因为这两种方法未能对网络结构进行优化,随着隐藏

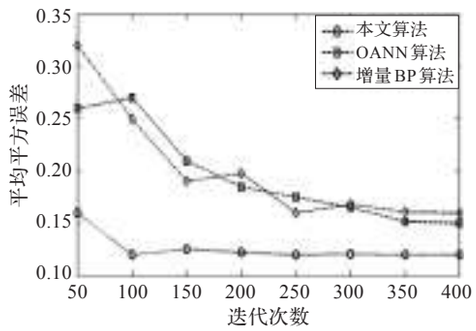

图5 增加40个隐藏层神经元三种算法的收敛结果

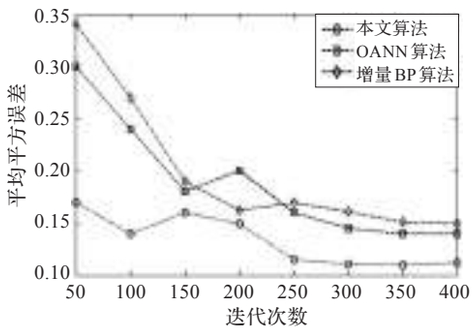

图6 增加60个隐藏层神经元三种算法的收敛结果

层神经元数目的增加,网络的冗余连接增多,网络结构的泛化能力变差。本文提出的算法在隐藏层神经元数目增加到40个时,平均平方误差达到最低,由于本文提出的算法时刻对网络结构进行优化更新,即便继续增加隐藏层神经元数目,网络结构的冗余连接并不会增加,因此网络的泛化能力和对新数据特征学习能力并未降低。

图5,6显示了三种算法增加不同隐藏层神经元数目时的收敛速度结果。

从图5和图6可以看出,当增加40个隐藏层神经元时,本文提出的算法在迭代100次时,达到收敛,而OANN算法和增量式BP算法需要300次和250次达到收敛;当增加60个隐藏层神经元时,本文提出的算法需要迭代250次达到收敛,而OANN算法和增量BP算法则分别需要300次收敛。本文提出的算法在增加隐藏层神经元后,初始化网络参数时充分结合原始参数,充分利用原有知识,因此在学习新的数据特征的时候,能够快速收敛。

5 结束语

本文针对大数据的内容与特征处于不断变化的特性,提出一种支持增量式更新的神经网络模型,该模型能够实时学习处于高速变化的大数据特征。在模型的更新过程中,每次根据一条新的数据实例进行更新,不需要将大规模数据加载到内存当中,因此能够有效学习大规模数据的特征。另外,在参数更新过程中,在原始参数基础上引入一阶近似思想,避免通过迭代的方式求参数增量,提高参数更新效率。本文重点讨论如何设计并实现支持快速有效的增量式更新的神经网络模型,使得更新模型能够满足大数据特征学习的实时性,没有关注数据的异构性,因此采用增量更新的图像数据集验证本文提出模型的有效性,在MNIST和STL-10两个典型的分类数据集上的实验表明,本文提出的算法能够有效学习动态变化且规模巨大的大数据特征,同时最大程度保持模型原始知识,满足大数据特征学习的实时性要求。下一步工作将扩展本文提出的模型,使得提出的模型能够有效学习异构数据的特征。

参考文献:

[1] 孟小峰,慈祥.大数据管理:概念、技术与挑战[J].计算机研究与发展,2013,50(1):146-169.

[2] Wu Xindong, Zhu Xingquan, Wu Gongqing, et al.Data mining with big data[J].IEEE Transactions on Knowledge and Data Engineering,2014,26(1):97-107.

[3] Zhang Qingchen, Chen Zhikui.A weighted kernel possibilistic c-means algorithm based on cloud computing for clustering big data[J].International Journal of Communication Systems,2014,27(9):1378-1391.

[4] 孙大为,张广艳,郑纬民.大数据流式计算:关键技术与系统实例[J].软件学报,2014,25(4):839-862.

[5] Chen Xuewen, Lin Xiaotong.Big data deep learning:challenges and perspectives[J].IEEE Access,2014,2:514-525.

[6] Hinton G E, Salakhutdinov R R.Reducing the dimensionality of data with neural networks[J].Science, 2006, 313(5786):504-507.

[7] Bengio Y, Courville A, Vincent P.Representation learning: a review and new perspectives[J].IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798-1828.

[8] 余凯,贾磊,陈雨强,等.深度学习的昨天、今天和明天[J].计算机研究与发展,2013,50(9):1799-1804.

[9] Liang N, Huang G, Saratchandran P, et al.A fast and accurate online sequential learning algorithm for feed-forward networks[J].IEEE Transactions on Neural Networks, 2006, 17(6):1411-1423.

[10] Li Guang, Na Jing, Stoten D P, et al.Adaptive neural network feed-forward control for dynamically sub-structured systems[J].IEEE Transactions on Control Systems Technology, 2014, 22(3):944-954.

[11] Rumelhart D E, Hinton G E, Willian R J.Learning representations of back-propagation errors[J].Nature, 1986, 323.

[12] Schraudolph N N.Fast curvature matrix-vector products for second order gradient descent[J].Neural Computation, 2012, 14(7).

(下转48页)