ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 33,No. 1,Jan. 2018,pp. 65－74
DOI:10.16337/j.1004-9037.2018.01.008

http://sjcj.nuaa.edu.cn
E-mail:sjcj@nuaa.edu.cn
Tel/Fax:＋86-025-84892742

基于 Spark 的高效并行自动编码机

庄福振[1]　钱明达[1]　申恩兆[1,2]　张大鹏[2]　何　清[1]

(1.中国科学院计算技术研究所智能信息处理重点实验室,北京,100190; 2.燕山大学信息科学与工程学院,秦皇岛,066004)

摘　要:机器学习中一个非常关键的问题就是如何获取良好的数据特征表示,许多经典的特征提取方法是基于数据间关系或利用简单线性组合降维后得到数据的特征表示。其中深度学习算法在各种学习任务中都可以取得良好的效果,而且可以学到很好的数据特征表示。但现有深度学习算法或模型大多为单机串行实现,不能处理较大规模的数据且运行时间较长。本文设计实现了一种基于 Spark 分布式平台的高效并行自动编码机,该编码机可以有效地进行特征表示学习,并且利用分布式计算平台 Spark 对算法进行加速,优化了对稀疏数据的操作,大大提升了运行效率。本文通过在文本数据特征学习以及协同过滤两个任务上的实验,表明本文所实现的并行自动编码机的有效性和高效性。

关键词:自动编码机;Spark;机器学习;深度学习;特征学习

中图分类号:TP181　　**文献标志码**:A

Efficient Parallel Auto-encoder Based on Spark

Zhuang Fuzhen[1], Qian Mingda[1], Shen Enzhao[1,2], Zhang Dapeng[2], He Qing[1]

(1. Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China; 2. School of Information Science and Engineering, Yanshan University, Qinhuangdao, 066004, China)

Abstract: How to find the good representation from raw data is a key and very important issue in machine learning. Most traditional approaches are based on the relationship among data or utilize simple linear combination, in which deep learning algorithm can perform very well in various machine learning tasks and achieve very good representations. However, most existing algorithms are implemented in serial, which cannot handle large-scale data. This paper proposes an effective parallel auto-encoder (PAE) based on Spark. The proposed PAE not only can learn satisfying representation, but also can speed up the executing time based on Spark. And then the paper adapts PAE to deal with the sparse data. Experiments conducted on two tasks, i.e., classification and collaborative filtering, demonstrate the effectiveness and efficiency of the proposed PAE.

Key words: auto-encoder; Spark; machine learning; deep learning; feature learning

---

基金项目:国家重点研发计划(2017YFB1002104)资助项目;国家自然科学基金(61773361,61473273,91546122,61573335,61602438)资助项目;广东省省级科技计划(2015B010109005)资助项目。
收稿日期:2016-04-03;**修订日期**:2016-04-17

引 言

机器学习作为人工智能的主体与核心，各种机器学习算法已经应用到各个领域，如语音识别、图片分类、推荐系统、手写输入识别和漏洞发现等，极大地提高了计算机智能化程度和各种系统的性能。实际应用中许多数据并不能很好地表示数据规律，甚至对系统的学习产生副作用，这就需要获得好的特征表示来改进算法的性能。然而随着计算机以及互联网的不断发展，信息量不断增大，信息复杂度也不断提升，数据包含的信息日益庞杂、冗余，良好的特征工程成为影响机器学习成败的关键因素。如何取得数据良好的特征表示，进而提升机器学习算法性能已经成为机器学习中一个关键的问题。

特征学习(Feature learning)又称作表示学习(Representation learning)，是机器学习领域中的一个重要研究问题，它的目标是自动地学习一个从原始输入到新特征表示的变换，使得新特征表示能够有效应用在各种学习任务中，从而把人从繁琐的特征工程中解放出来。根据训练过程是否需要有标签数据，可以把表示学习算法分为两类：有监督特征学习和无监督特征学习。在有监督特征学习中，通常需要利用有标签数据，比如有监督字典学习。无监督特征学习中，不需要数据具有标签，比如主成分分析、独立成分分析、自动编码机、矩阵分解[1]以及众多形式的聚类算法[2-3]。

近几年来神经网络深度结构对特征的多层次抽象吸引了研究人员的广泛关注，深度学习模型在各类学习任务中都取得了不错的性能，且许多深度模型在实际应用场景中都表现良好[4-6]。自动编码机已经被证明在数据降维和特征提取上有着非常好的表现。目前自动编码机已经发展出众多变体，除了自动编码机外，还有稀疏自动编码机、降噪自动编码机、堆叠自动编码机及卷积自动编码机等[7-9]。此外神经元的屏蔽、正则化、边缘化等方法[10]都被运用于自动编码机。在应用上，异常检测中自动编码机取得良好效果，深度自动编码机也被运用到迁移学习中[11]。但以上的自动编码机都是串行实现，不能满足目前大数据环境下模型复杂、计算量大的实际问题。因此，单机串行实现的模型在应用层面具有很大的局限性。

随着大数据时代的来临，处理海量数据、建立大规模计算模型成为必然需求。本文提出了一种基于Spark的高效自动编码机，利用Spark分布式内存计算的固有优势高效处理海量数据。现有数据种类繁多，许多数据有非常稀疏的特点，比如推荐系统关系矩阵、文本词袋模型等。而现有基于自动编码机的特征提取算法没有对稀疏数据做针对性优化，计算过程中大量无效运算和存储空间开销使时间复杂度和空间复杂度随数据维度成二次甚至三次增长，与有效数据成指数级增长。本文编码算法是一种对稀疏数据针对性操作的优化方法，使得稀疏数据处理开销与有效数据成正比，极大优化了算法运行的时间效率和空间效率。

1 基于Spark的高效并行自动编码机

1.1 自动编码机的概念

自动编码机(Auto-encoder)是一种无监督学习算法，作为神经网络的一种变体也被称作自编码神经网络，它使用反向传播算法，将输入值作为目标输出，比如 $\boldsymbol{y}^{(i)}=\boldsymbol{x}^{(i)}$。自动编码机主要由两部分组成，分别是编码部分和解码部分。训练中，编码部分和解码部分同时优化，使得 $h_{W,b}(\boldsymbol{x})\approx\boldsymbol{x}$。亦即，它试图逼近一个恒等函数，从而使输出 $\hat{\boldsymbol{x}}$ 接近于输入 $\boldsymbol{x}$。一般的，自动编码机隐含层节点少于输入层节点，迫使其学习输入数据的压缩表示，换言之自动编码机需要由较少的隐层神经元输出重构出原始高维数据。通常情况下，数据中隐含着一些特定结构，使得不同维度间彼此相关，那么就使得算法可以发现输入数据中的相关性，进而得到数据的有效低维表示，即数据压缩表示，或者说数据去除冗余信息后的特征表示。图1为一个简单自动编码机的结构[12]。

图 1 中的自动编码机,作为前馈神经网络,分别由输入层、隐含层和输出层组成。其中输入层为数据直接表示,隐含层和输出层中每个节点代表神经网络的一个神经元。神经元由上层网络输出的加权和通过激活函数得到输出,并提供下一层网络使用。神经元可以简单表示为如图 2 所示,激活函数一般为非线性函数,本文使用 Sigmoid 函数,其函数表达式为

$$y=\frac{1}{1+\exp(-x)} \tag{1}$$

一般地,一个自动编码机包括编码部分和解码部分。两部分可以概括为

$$\text{Encoding}:\boldsymbol{\xi}=f(\boldsymbol{W}_1\boldsymbol{x}+\boldsymbol{b}_1) \tag{2}$$

$$\text{Decoding}:\hat{\boldsymbol{x}}=f(\boldsymbol{W}_2\boldsymbol{\xi}+\boldsymbol{b}_2) \tag{3}$$

式中:f 为非线性激活函数 Sigmoid,权值矩阵分别为 $\boldsymbol{W}_1\in\mathbf{R}^{k\times m}$,$\boldsymbol{W}_2\in\mathbf{R}^{m\times k}$,偏置项为 $\boldsymbol{b}_1\in\mathbf{R}^{k\times 1}$,$\boldsymbol{b}_2\in\mathbf{R}^{1\times k}$,隐含层输出为 $\boldsymbol{\xi}\in\mathbf{R}^{k\times 1}$。给定一组输入数据 $\{\boldsymbol{x}\}_{i=1}^{n}$,那么重构误差可以表示为 $\sum_{i=1}^{n}\|\hat{\boldsymbol{x}}_i-\boldsymbol{x}\|^2$。整个自动编码机的最终目标就是通过最小化重构误差,得到恰当的权值 $\boldsymbol{W}_1$,$\boldsymbol{W}_2$ 以及偏置项 $\boldsymbol{b}_1$,$\boldsymbol{b}_2$。整个网络的重构误差可表示为 $\min_{\boldsymbol{W}_1,\boldsymbol{b}_1,\boldsymbol{W}_2,\boldsymbol{b}_2}\sum_{i=1}^{n}\|\hat{\boldsymbol{x}}_i-\boldsymbol{x}_i\|^2$。

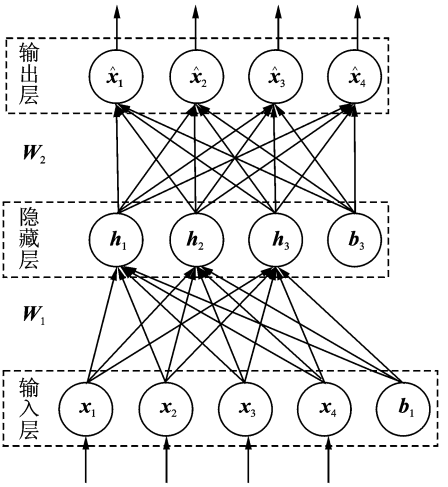

图 1 简单的自动编码机结构

Fig. 1 Structure of a simple auto-encoder

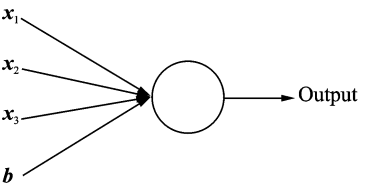

图 2 神经元

Fig. 2 Single neuron

1.2 Spark 分布式计算平台

Apache Spark,2014 年成为 Apache 基金顶级项目,以快速、通用、简单等特点成为当前流行大数据处理模型。Spark 提出的分布式内存计算框架,既保留了 MapReduce 的可扩展性、容错性和兼容性,又弥补了 MapReduce 在这些应用上的不足。由于采用基于内存的集群计算,所以 Spark 在这些应用上相对 MapReduce 有 100 倍左右的加速[13]。此外 Spark 可以部署在 Hadoop 集群环境下,拥有直接访问 HDFS 文件系统的能力。但不同于 MapReduce 中间过程和计算结果需要读写 HDFS,Spark 将计算结果保存在内存中,从而不必频繁读写 HDFS,减少了 IO 操作,提升了算法运行效率,使算法运算时间大幅缩短。

值得强调的是,分布式系统需要一个统一调度角色,一般称为管理者(Manager),而其他各运算节点称为工作者(Worker)。一般情况下,并行程序运行过程中由 Workers 分别计算相对独立的并行步骤,Manager 统一调度、管理和统计。

1.3 基于 Spark 的高效并行自动编码机基本原理

基于 Spark 的高效并行自动编码机(Parallel auto-encoder,PAE)是对现有自动编码机完成针对稀疏数据的修改后,将其迁移到 Spark 计算平台上的成果。系统的结构包括一个负责分发 Job、收集数据、调度任务的 Manager 以及若干个负责具体计算的 Worker。系统结构如图 3 所示。图中,系统于分布式系统的各计算节点初始化同样的神经网络。每计算一定量的数据,对模型的参数进行收集归并,不断迭代得到最终权值。

上述过程基于神经网络的性质:相同初始值的神经网络,对同分布下不同数据样本进行拟合,最终所拟合数据分布相同,即得到网络参数分布相近。本文算法以此为基础,在初始化时保证网络完全一致,达到自动编码机并行运算的目的。

此外,本系统针对自动编码机低效处理稀疏数据的缺陷,提出避免计算过程中无效计算和无效存储开销的方案。使用指示器函数对算法进行改进,指示器函数为

$$I_j=\begin{cases}1 & x_j\neq 0\\ 0 & x_j=0\end{cases}$$

通过指示器函数的选择,在计算中系统只关注有效值及其对应神经元,且反向传播时避免无效计算与操作。本文系统成功地将时间复杂度由二次复杂度降为线性复杂度,在保证计算正确性的前提下,大幅度提高了模型训练速度。

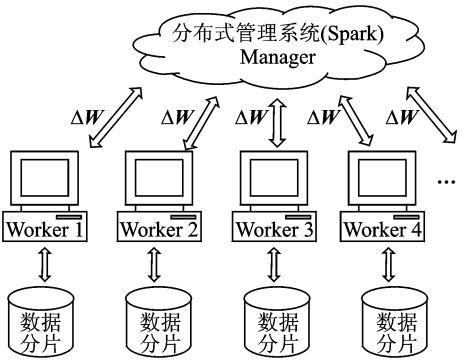

图 3 系统结构图

Fig. 3 Structure of system

2 基于 Spark 的高效并行自动编码机算法

2.1 编码机算法模型

为实现基于 Spark 的高效并行自动编码机,本文选择反向传播来进行学习和优化,此外目标函数学习使用了梯度下降思想。另外,为提高算法运行效率,本文采用随机梯度下降算法,即计算每一条样本,实时更新学习模型。

当模型训练开始,Manager 首先进行 Map 操作,将初始化参数分发至各 Worker 节点,包括数据规模、隐藏层配置、输入数据路径、正则化参数及随机数种子等。此后每个 Worker 独立读取数据,利用随机数种子分别初始化自动编码机的各项参数,其中由相同的随机种子保证各 Worker 上神经网络初始状态一致。

随后 Worker 分别利用各自数据分片训练自动编码机,Worker 每次处理一条数据样本,将读入样本进行正向传播,即利用输入数据计算隐藏层和输出层的结果。完成后进行反向传播,利用得到的输出数据误差计算出神经网络的参数梯度。利用所得梯度,完成网络参数更新,至此完成模型的一次迭代。整个训练过程需要重复迭代,保证所分得数据分片中的每一例样本至少计算一遍。

各 Worker 利用一部分数据完成模型训练后,进行 Reduce 操作,将各节点计算结果进行约减。其中约减操作定义为 Worker 将计算所得模型参数上传至 Manager,由 Manager 对各参数取平均。至此完成系统的一次迭代。

对系统进行迭代,保证所有 Worker 均完全训练,至少保证训练集中所有数据均参与到系统模型的训练中。最终在 Manager 上得到一个训练完毕的模型。值得指出的是,实际操作中可以根据集群实际情况,对数据进行多种划分模式,包括但不限于差异化节点数目、节点间不同数据条数、各次迭代处理的数据条数以及系统整体对数据比例划分。

以上模型可具体表示为:

正向传播

$$\boldsymbol{h}=\mathrm{sigmoid}(\boldsymbol{W}_1\boldsymbol{x}+\boldsymbol{b}_1) \tag{4}$$

$$\hat{\boldsymbol{x}}=\mathrm{sigmoid}(\boldsymbol{W}_2\boldsymbol{h}+\boldsymbol{b}_2).*\boldsymbol{I} \tag{5}$$

反向传播

$$\boldsymbol{E}_{\text{out}} = (1-\hat{\boldsymbol{x}})\hat{\boldsymbol{x}}(\boldsymbol{x}-\hat{\boldsymbol{x}}) \tag{6}$$

$$\boldsymbol{E}_{\boldsymbol{h}} = (1-\boldsymbol{h})\boldsymbol{h}\sum\nolimits_{E_{\text{out}}\in \text{down stream}} \boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{E}_{\text{out}} \tag{7}$$

$$\Delta\boldsymbol{W}_2 = \boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{E}_{\text{out}} \tag{8}$$

$$\Delta\boldsymbol{W}_1 = \boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{E}_{\boldsymbol{h}} \tag{9}$$

权值更新

$$\boldsymbol{W} = \boldsymbol{W} + \phi(\boldsymbol{x}_{\text{in}}\Delta\boldsymbol{W} - \alpha\boldsymbol{W}) \tag{10}$$

式(4～10)中符号表示含义如表 1 所示。

表 1 公式符号含义

Tab. 1 Meaning of formula symbols

符号	含义
$\boldsymbol{x}$	$M\times 1$ 列向量,输入数据
$\boldsymbol{W}$	$K\times M$,$M\times K$ 矩阵,权值矩阵
$\boldsymbol{B}$	$K\times 1$,$M\times 1$ 列向量,权值向量
sigmoid	$1/(1+\mathrm{e}^{-x})$激活函数
$\boldsymbol{h}$	$K\times 1$ 列向量,隐藏层输出
$\hat{\boldsymbol{x}}$	$M\times 1$ 列向量,输出
$\boldsymbol{I}$	$M\times 1$ 指示函数
$\boldsymbol{E}$	$M\times 1$,$K\times 1$ 列向量,误差
$\Delta\boldsymbol{W}$	$K\times M$,$M\times K$ 矩阵,更新值
ϕ	常数,步长
α	正则化参数

表中:N 为输入数据样本数量;M 为输入数据维度;K 表示隐藏层节点目。

2.2 自动编码机的实现

根据 2.1 节描述可以实现基于 Spark 的高效并行自动编码机。Worker 上随机梯度下降(Stochastic gradient descent,SGD)伪代码如算法 1 所示,基于 Spark 的高效并行自动编码机伪代码如算法 2 所示。

算法 1 随机梯度下降算法

输入:给定输入 $\boldsymbol{x}_{\text{in}} = \{\boldsymbol{x}_i\}\mid_{i=1}^{N}$,步长 ϕ。

输出:自动编码机学习到的参数 $\boldsymbol{W}_1$,$\boldsymbol{b}_1$,$\boldsymbol{W}_2$,$\boldsymbol{b}_2$

For:对输入数据集中的每一个样本:

(1)根据式(4,5),进行正向传播。并计算网络中每个单元的输出;

(2)根据式(6,7),利用已求有输出分别求得网络各节点的残差;

(3)由式(8,9),利用已有残差分别计算网络模型各参数的梯度;

(4)按照式(10),更新自编码机权值。

算法 2 基于 Spark 的高效并行自动编码机

输入:给定输入 $\boldsymbol{x}_{\text{in}} = \{\boldsymbol{x}_i\}\mid_{i=1}^{N}$,步长 ϕ,自动编码机网络规模参数 size,系统初始随机数种子 τ,集群参与运算节点数目 N_{worker}。

输出:自动编码机学习到的参数 $\boldsymbol{W}_1$,$\boldsymbol{b}_1$,$\boldsymbol{W}_2$,$\boldsymbol{b}_2$

(1)初始化,由 Manager 根据 N_{worker} 将网络参数 size,τ 进行分发,并根据实际情况,将 $\boldsymbol{x}_{\text{in}}$ 随机划分为若干部分,分别派发给 Worker。Worker 根据 Manager 下发的 size 来构建网络,τ 来初始化矩阵;

(2)并行运算,在每个 Worker 中:使用 Manager 分发的数据分片,进行 SGD 操作,训练得到模型参数,第 n 个 Worker 得到的结果表示为 $W_1^n, b_1^n, W_2^n, b_2^n$;

(3)约减收集操作,Worker 计算完毕,向 Manager 提交 $W_1^n, b_1^n, W_2^n, b_2^n$,Manager 收集到所有工作节点的结果后,根据 $W_1, b_1, W_2, b_2 = \dfrac{1}{N_{\text{worker}}} \sum\limits_n W_1^n, b_1^n, W_2^n, b_2^n$ 对参数做取平均操作。

算法 2 流程如图 4 所示。由算法 2 得到的所有 Worker 上的参数,可以得到最终的模型参数。

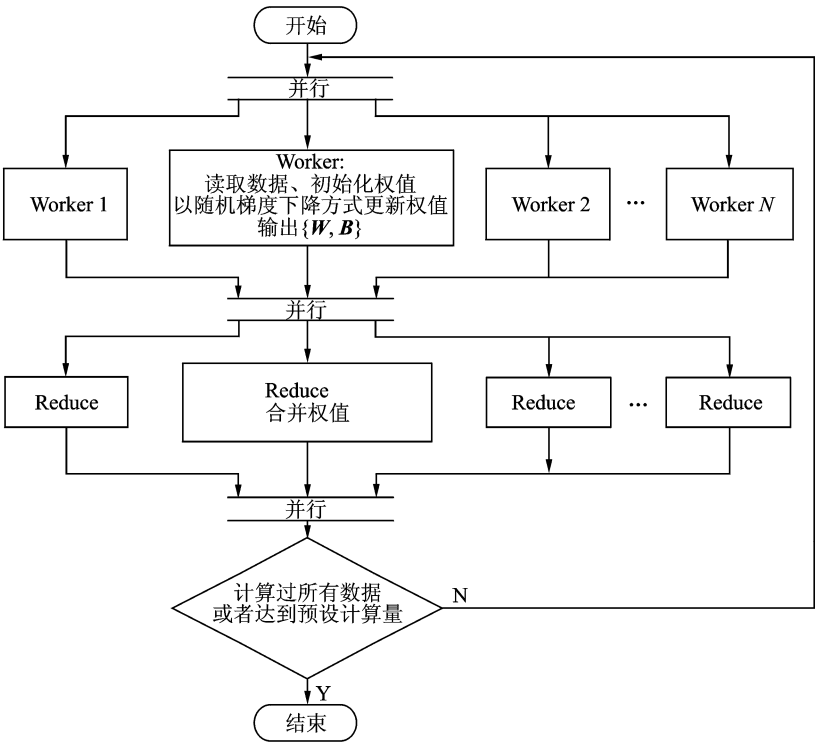

图 4 基于 Spark 高效并行自动编码机

Fig. 4 Efficient parallel auto-encoder based on Spark

3 实验与结果分析

本节在两个学习任务上验证算法的有效性以及高效性,即分类任务和协同过滤。本文实验包括:(1)直接对源数据利用支持向量机(SVM)得到分类结果与 PAE 分类结果进行比较。(2)利用 Matlab Deeplearning Tool(http://www.mathworks.com/matlabcentral/fileexchange/38310-deep-learning-toolbox)中 SAE 单机压缩数据后,再利用 SVM 分类结果与 PAE 得到的压缩数据分类结果进行比较,分别验证算法不同方面的能力和表现。其中 SAE 串行算法的有效性已经被众多事实所验证,公认其获得的特征可以有效提升机器学习算法性能,故而在此不再重复讨论。(3)将 PAE 应用到推荐系统中,得到推荐结果分别与基准算法进行比较,进而验证 PAE 在推荐系统中的有效性。

3.1 数据集和预处理

为了验证 PAE 算法的有效性和并行效率,本文采用文本数据集 20Newsgroups(http://qwone.

com/~jason/20Newsgroups/),该数据集包含 20 个主题,每个主题有大约 1 000 个样本文件,数据维度为 61 188,数据以稀疏数据的方式存储。20 个主题按照题材简单分类后如表 2 所示。为了构造分类问题,本文以文档主题作为分类预期结果,总共 20 类。

表 2 20Newsgroups 数据集主题

Tab. 2 Topic of 20Newsgroups

comp. graphics comp. os. ms-windows. misc comp. sys. ibm. pc. hardware comp. sys. mac. hardware comp. windows. x	rec. autos rec. motorcycles rec. sport. baseball rec. sport. hockey	sci. crypt sci. electronics sci. med sci. space
misc. forsale	talk. politics. misc talk. politics. guns talk. politics. mideast	talk. religion. misc alt. atheism soc. religion. christian

在 PAE 应用于推荐系统的实验中,本文基于两个个性推荐系统公共数据集 MovieLens(http://grouplens. org/datasets/movielens/)和 NetFlix(http://www. prea. gatech. edu/download. html)进行实验。

MovieLens 是一套公共推荐系统数据集,数据为用户对自己看过的电影进行的评分,分值为 1~5。显然,数据集中用户和电影构成的评分矩阵极度稀疏。数据集中包括 3 个不同大小的库,100 K 数据集中包括 1 000 位用户对 1 700 部电影进行评分的 100 000 个评分记录;1 M 数据集中包括 6 000 位用户对 4 000 部电影打分的 100 万条记录;10 M 数据集中包括了 72 000 位用户对 10 000 部电影进行评分的 1 000 万条记录和 10 万标签记录。

NetFlix 和 MovieLens 一样也是用户电影评分数据集,同样分值分布为 1~5,但前者数据量远远高于后者。该数据集反映了自 1998 年 10 月到 2005 年 12 月期间的评分记录分布。NetFlix 中包括了从 48 万名用户对 17 000 部电影的评分中随机选取的 1 亿条评分记录。此外还包括了电影的发行时间和发行标题。

3.2 特征表示学习

特征表示学习实验中主要分为两部分,分别验证 PAE 的有效性和高效性。

(1)在验证 PAE 有效性实验中,本文首先训练 SVM 分类器,得到源数据分类准确率基准值;随后由 PAE 在不同数目的 Worker 下对数据进行压缩降维,利用压缩后的降维特征训练分类器得到分类准确率。其中 SVM 作为基准分类器使用 LibSVM[14] 库。实验从所有数据中分别随机选取 60%,70%,80%,90%作为训练集,剩下的作为测试集。为了保证分布式算法在不同节点情况下充分测试,本文分别测试了 2,3,4 个 Worker 的情况。

(2)本文也比较了 Matlab Deeplearning Tool 中 SAE 作为特征的提取方法。由 SAE 对数据进行降维压缩,得到的特征表示由 SVM 进行学习并构造分类器,通过测试集得出分类准确率。实验中利用自动编码机将原有 61 188 维稀疏数据提取特征降维到 100 维。为了验证 PAE 对海量数据处理性能,本文将 18 774 个样本的数据集进行扩展、复制,进而得到海量数据。通过海量数据测试将 PAE 与 SAE 进行对比,验证算法运行效率。

3.3 特征表示学习实验结果

3.3.1 PAE 正确性分析

表 3 给出了 PAE 有效性实验的实验结果,所列数据均为 5 次实验平均值。表中还记录了不同数目

的 Worker 下本文并行算法执行结果。由表 3 中结果可以看到,SVM 分类准确率随着训练比的不断提高而提高,经过 PAE 降维后的数据在 SVM 分类器的表现普遍优于同训练比例的原始数据。这是由于原始数据维度非常高,通过降维可以得到更好的特征表示。实验结果证明了 PAE 提取特征的正确性和有效性。

表 3 SVM 分类准确率

Tab. 3 Accuracy of SVM %

训练比例	源数据	2-worker	3-worker	4-worker
0.6	70.57	75.42	77.39	77.75
0.7	72.14	76.22	78.15	77.69
0.8	73.42	77.95	76.99	77.39
0.9	75.31	75.69	78.02	77.65

3.3.2 PAE 效率分析

在保证正确性的情况下,与 Matlab Deeplearning Tool 中 SAE 做效率对比,同样地将源数据集由 61 188维度压缩为 100 维度。SAE 及 PAE 算法运行时间统计结果如图 5 所示。值得指出的是,SAE 在数据量达到 8 000 条时,报出了内存不足的异常,亦即单机 Matlab Deeplearning Tool 中 SAE 处理数据的极限不足 8 000 条。并且由图 5 显示,SAE 时间开销是随数据量的变化成二次变化。运行环境为:Intel Core™ i7-4790 3.6 GHz CPU,8 GB 内存,500 GB 硬盘,操作系统为 Windows 7,64 位专业版。

同样的数据,在 PAE 实验过程中设置不同的 Worker 数。由图 5 看到在数据量较小时,PAE 时间开销相对较大,变化趋势与数据量并无明显联系,这是因为数据量较小时,分布式计算中平台通讯开销所占系统时间开销比例较大。随着数据量的不断增大,可以看出当数据量达到 60 000 条时,系统时间开销开始有明显的增长,此时系统的计算时间占比逐渐变大,计算时间的变化对系统总时间开销的影响开始显现出来。与单机运行 SAE 对比,首先 PAE 能够处理的数据量远远大于 SAE;其次由图线对比得出,在数据量逐渐增大的过程中,系统时间开销随数据量增大而线性增长,相比 SAE 二次增长的时间曲线,具有明显优势。

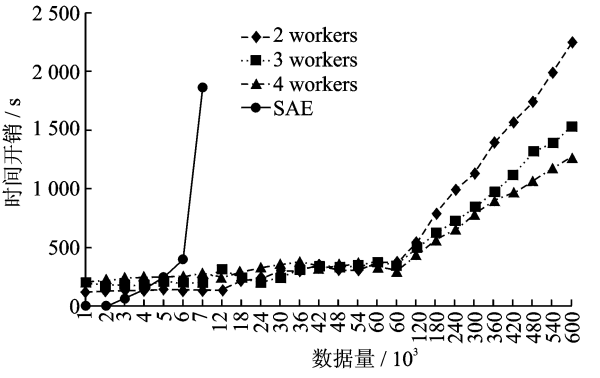

图 5 SAE 和 PAE 运算时间开销对比

Fig. 5 Time cost comparison between SAE and PAE

3.4 PAE 在推荐系统中的应用

在推荐系统的应用实验中,本文使用了非负矩阵分解(Non-negative matrix factorization, NMF)与概率矩阵分解(Probabilistic matrix factorization, PMF)作为与 PAE 比较的参考算法,其中 NMF 与 PMF 均来自于 PREA[15] 工具包。

个性化推荐算法工具包(Personalized recommendation algorithms toolkit, PREA)是一个个性化推荐算法的工具包,其中包括了大量推荐算法的实现,提供了便捷的接口可供调用。学术界提出的各种推荐算法常与它进行对比,验证准确率及效率。

在实验过程中,本文利用均值绝对误差(Mean absolute error, MAE)和均值平方根误差(Root mean

squared error, RSME)作为衡量算法效果的度量值。实验中利用 PAE 分别以 user 和 item 作为维度坐标进行压缩降维。为表述方便本文将两种方法表示为 PAE_u 和 PAE_i。不同算法 5 次实验结果的平均值如表 4 所示。

表 4 推荐系统实验数据

Tab. 4 Experimental result of recommendation system

数据集	度量值	NMF	PMF	PAE_u	PAE_i
100 K	MAE	0.769 7	0.830 2	0.833 6	0.823 7
	RMSE	0.977 3	1.013 8	1.031 2	1.028 1
	TIME/s	2.3	26.9	22.0	38.0
1 M	MAE	0.704 0	0.807 3	0.849 3	0.821 7
	RMSE	0.897 0	0.986 5	1.030 6	1.030 5
	TIME/s	200.9	407.2	115.0	63.6
10 M	MAE	—	—	0.835 2	0.794 2
	RMSE	—	—	1.032 3	0.990 5
	TIME/s	—	—	115.2	88.4
NetFlix	MAE	0.789 1	0.829 2	0.797 0	0.766 6
	RMSE	1.019 8	1.009 7	0.992 6	0.965 6
	TIME/s	6.6	108.4	42.6	12.4

由表 4 可以看出,当数据量较小时,在推荐结果的准确性和推荐效果的稳定性方面 NMF 表现相对较好,而且在算法运行时间上也有一定优势。但随着实验数据量的增大,NMF 与 PMF 均出现了内存不足的情况,在基于海量数据推荐任务中,两种算法并不能很好地给出结果。而 PAE 的两种算法,尽管在准确性和稳定性方面稍弱于 NMF 和 PMF,但可以应对海量数据的推荐任务。结果充分表明 PAE 可以应用于推荐系统中,特别是在海量数据下的应用场景。

4 结束语

本文提出了一种基于 Spark 的高效并行自动编码机。通过对自动编码机进行并行化,使之能够高效处理稀疏大数据,同时具备正常处理非稀疏数据能力。此外,将其迁移到 Spark 分布式平台,面向稀疏数据进行针对性优化,使其大幅减少 IO 操作,提升算法运行效率,算法运算时间明显缩短。本文做了大量实验来验证 PAE 的可行性与有效性,实验建立在两个实际应用任务上,其中分类任务验证了 PAE 的高效性和 PAE 对特征提取的有效性。在保证特征提取有效性的前提下本文将 PAE 应用在推荐系统中,利用自动编码机提取到的数据特征对原有数据进行填充预测,进而得出推荐结果,通过与对比算法的比较进一步验证本算法的有效性和高效性。

参考文献:

[1] Srebro N, Rennie J D M, Jaakkola T. Maximum-margin matrix factorization[J]. Advances in Neural Information Processing Systems, 2004,37(2):1329-1336.

[2] Coates A, Ng A Y, Lee H. An analysis of single-layer networks in unsupervised feature learning[J]. Journal of Machine Learning Research, 2011,15:215-223.

[3] Dance C, Willamowski J, Fan L, et al. Visual categorization with bags of key points[C]// ECCV International Workshop on Statistical Learning in Computer Vision. Prague:[s. n.], 2004:1-22.

[4] 卢宏涛,张秦川.深度卷积神经网络在计算机视觉中的应用研究综述[J].数据采集与处理,2016,31(1):1-17.
Lu Hongtao, Zhang Qinchuan. Applications of deep convolutional neural network in computer vision[J]. Journal of Data Ac-

quisition and Processing, 2016,31(1):1-17.

[5] 李思雯,吕建成,倪胜巧. 集成的卷积神经网络在智能冰箱果蔬识别中的应用[J]. 数据采集与处理,2016,31(1):205-212.
Li Siwen, Lü Jiancheng, Ni Shengqiao. Integrated convolutional neural network and its application in fruits and vegetables recognition of intelligent refrigerator[J]. Journal of Data Acquisition and Processing, 2016,31(1):205-212.

[6] 杨阳,张文生. 基于深度学习的图像自动标注算法[J]. 数据采集与处理,2015,30(1):88-98.
Yang Yang, Zhang Wensheng. Image auto-annotation based on deep learning[J]. Journal of Data Acquisition and Processing, 2015,30(1):88-98.

[7] Le Q V, Ranzato M A, Monga R, et al. Building high-level Ieatures using large scale unsupervised learning[C]//Proceedings of the International Conference on Machine Learning (ICML). Edinburgh, UK:[s. n.],2012:107-114.

[8] Vincent P, Larochelle H, Bengio Y, et al. Extracting and composing robust features with denoising autoencoders[C]//Proceedings of the 25th International Conference on Machine Learning. Canada:[s. n.], 2008:1096-1103.

[9] Gupta A, Maida A S, Ayhan M. Natural image bases to represent neuroimaging data[C]//30th International Conference on Machine Learning. Atlanta, Georgia, USA: International Machine Learning Society, 2013:987-994.

[10] Chen M, Xu Z, Weinberger K, et al. Marginalized denoising autoencoders for domain adaptation[J]. Computer Science, 2012: 2012arXiv1206.4683C.

[11] Zhuang F, Cheng X, Luo P, et al. Supervised representation learning: Transfer learning with deep autoencoders[C]//Proc of the 24th International Joint Conference on Artificial Intelligence. Buenos Aires, Argentina: AAAI Press, 2015:4119-4125.

[12] Bengio Y. Learning deep architectures for AI[J]. Foundations & TrendsR in Machine Learning, 2009,2(1):1-127.

[13] 黎文阳. 大数据处理模型 Apache Spark 研究[J]. 现代计算机:普及版,2015(8):55-60.
Li Wenyang. Research on apache spark for big data processing[J]. Modern Computer, 2015(8):55-60.

[14] Chang C C, Lin C J. LIBSVM: A library for support vector machines[J]. ACM Transactions on Intelligent Systems & Technology, 2011,2(3):389-396.

[15] Lee J, Sun M, Lebanon G. PREA: Personalized recommendation algorithms toolkit[J]. Journal of Machine Learning Research, 2012,13(3):2699-2703.

作者简介:

庄福振(1983-),男,副研究员,硕士生导师,研究方向:机器学习、数据挖掘、迁移学习、推荐系统,E-mail:zhuangfz@ics.ict.ac.cn。

钱明达(1994-),男,本科,研究方向:机器学习与数据挖掘。

申恩兆(1992-),男,硕士研究生,研究方向:人工智能、机器学习。

张大鹏(1979-),男,讲师,研究方向:人工智能(机器学习与数据挖掘)。

何清(1965-),男,研究员,研究方向:机器学习、数据挖掘、文本挖掘、基于云计算的分布式并行数据挖掘等。

(编辑:张黄群)