

基于分布式内存计算的深度学习的方法

李抵非¹, 田地¹, 胡雄伟²

(1. 吉林大学 仪器科学与电气工程学院, 长春 130021; 2. 国家标准化管理委员会 标准信息中心, 北京 100088)

摘要: 为了提升深度学习技术并行化学习效率, 设计了一种面向计算机集群的分布式内存计算方法。构建分布式内存环境, 建立数据分片处理和多任务调度机制, 使模型参数和神经元节点的计算和存储并行运行于该环境中, 避免了磁盘 I/O 对训练速率的影响; 采用深度信念网络模型, 以多个副本异步并行计算的方式进行模型训练, 并使用 dropout 方法防止模型训练过拟合。对 CIFAR-10 图像数据集进行分类训练, 试验结果表明, 该方法可以明显提高深度神经网络的训练效率, 并具备良好的可扩展性。

关键词: 人工智能; 并行深度学习; 分布式内存计算; 深信度网络

中图分类号: TP183 **文献标志码:** A **文章编号:** 1671-5497(2015)03-0921-05

DOI: 10.13229/j.cnki.jdxbgxb201503034

A method of deep learning based on distributed memory computing

LI Di-fei¹, TIAN Di¹, HU Xiong-wei²

(1. College of Instrumentation & Electrical Engineering, Jilin University, Changchun 130021, China;
2. Standardization Administration Information Center, Standardization Administration of the People's Republic of China, Beijing 100088, China)

Abstract: To improve the efficiency of deep neural network distributed training, a new method is proposed, which makes neural network model running on distributed memory computing system. A framework of distributed memory is built, which contains functions of data partition and multi-task schedule. It can avoid the impact of I/O on the training rate and makes the training process run at memory-speed across cluster. Within the framework, multiple model replicas of deep belief network are trained in an asynchronous way. In addition, the dropout algorithm is employed to prevent over-fitting. The proposed method is evaluated using CIFAR-10 dataset. Experiment results show that the new method improves the efficiency of training deep neural network and enables scalability.

Key words: artificial intelligence; distributed deep learning; distributed memory computing; deep belief network

近年来随着对深度学习领域研究的深入, 已经证实通过扩大训练样本的数量和模型参数的规

模可以有效提升模型的精度^[1-3]。为了克服由于扩大训练规模造成的时间成本问题, 许多基于深

收稿日期: 2014-10-09.

基金项目: 科技部创新方法工作专项项目(2011IM010400)。

作者简介: 李抵非(1986-) 男, 博士研究生。研究方向: 人工智能技术。E-mail: lidf12@mails.jlu.edu.cn

通信作者: 田地(1958-) 男, 教授, 博士生导师。研究方向: 分析仪器测控技术及软件。E-mail: tiandi@jlu.edu.cn

度神经网络的并行化计算方法被提出,其中大致分为两类方法,一是采用集群的方式对神经网络进行分布式训练^[4-5],二是利用图形处理单元(GPU)特性进行并行加速^[6-9]。采用GPU计算的方法虽然有效地提升了模型的训练效率,但是由于现阶段显卡显存的限制和其不易扩展的特点,GPU的运用还主要集中在训练中、小规模深度神经网络;使用大规模集群的方式可以训练大型深度神经网络,并取得了良好的效果,但所有训练模型的副本以及参数需保存在存储设备中,训练过程中频繁发生存储设备的读写操作,I/O的速率将会影响训练的效率。

本文采用分布式内存计算的方式,使整个深度神经网络的训练过程运行在集群的分布式内存之上,既保证了深度神经网络规模可扩展又避免了I/O读写速率对系统的影响,提高了训练效率。

1 深度神经网络模型

目前深度神经网络模型存在很多种类,本文以深度信念网络(Deep belief networks, DBNs)模型^[10-12]为例,将其运行在分布式集群之上。DBNs是由多层限制玻尔兹曼机(RBM)堆叠形成,采用无监督贪婪逐层训练的方式进行学习。DBNs是一个概率生成模型,目的是建立可视层和隐藏层之间的联合分布。

$$p(\mathbf{v}|\mathbf{h}) = \frac{1}{Z(\boldsymbol{\theta})} e^{-E(\mathbf{v}|\mathbf{h}|\boldsymbol{\theta})} \quad (1)$$

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}|\mathbf{h}|\boldsymbol{\theta})} \quad (2)$$

$$E(\mathbf{v}|\mathbf{h}|\boldsymbol{\theta}) = -\mathbf{a}^T \mathbf{v} - \mathbf{b}^T \mathbf{h} - \mathbf{v}^T \mathbf{W} \mathbf{h} \quad (3)$$

式中: \mathbf{v} 和 \mathbf{h} 为可视层单元和隐藏层单元; $E(\mathbf{v}|\mathbf{h}|\boldsymbol{\theta})$ 为系统所具备的能量表达式; $Z(\boldsymbol{\theta})$ 为归一化因子, $\boldsymbol{\theta} = \{\mathbf{W} \mathbf{a} \mathbf{b}\}$; \mathbf{W} 为可视层和隐藏层之间的权重; \mathbf{a} 和 \mathbf{b} 分别是可视层和隐藏层的偏置向量。

由于RBM层级内节点间无连接,各神经单元激活状态是条件独立,则第 i 个可视单元和第 j 个隐藏单元被激活的概率表达式如式(4)(5)所示:

$$p(h_j = 1 | \mathbf{v}, \boldsymbol{\theta}) = \text{sigmoid}(b_j + \sum_i v_i W_{ij}) \quad (4)$$

$$p(v_i = 1 | \mathbf{h}, \boldsymbol{\theta}) = \text{sigmoid}(a_i + \sum_j W_{ji} h_j) \quad (5)$$

根据Hinton提出的对比散度^[13]的方法,利用式(4)(5),使用训练数据 \mathbf{v}_0 输入系统,并通过 k

步吉布斯采样对可视单元和隐藏单元进行重构。使用随机梯度下降法训练数据时,各参数更新的准则如式(6)~(8)所示:

$$\Delta W_{ij} = \varepsilon (\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{recon}}) \quad (6)$$

$$\Delta a_i = \varepsilon (\langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{recon}}) \quad (7)$$

$$\Delta b_j = \varepsilon (\langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{recon}}) \quad (8)$$

式中: $\langle \cdot \rangle_{\text{data}}$ 表示关于分布 $p(h = 1 | \mathbf{v}, \boldsymbol{\theta})$ 的数学期望; $\langle \cdot \rangle_{\text{recon}}$ 表示经过吉布斯采样重构后关于分布 $p(h' = 1 | \mathbf{v}, \boldsymbol{\theta})$ 的数学期望; ε 表示模型的学习速率。

2 分布式内存计算结构

为了使深度神经网络模型能够分布式地运行于集群之上,需要构建一个并行计算的框架。目前的分布式计算方式是采用MapReduce架构来运行,在计算迭代的过程中需要不断地与存储设备发生读写操作。而如果采用分布式内存的方式来承载工作数据集,模型的计算和存储都发生在分布式内存系统当中,则可以提升计算性能。采用集群分布式内存计算的结构能可扩展地在内存集群计算中缓存数据集,以缩短访问时间,利于存在迭代过程的深度学习算法,节省了大量的I/O操作,提升了算法的效率。

集群的分布式内存计算架构采用主从模式,如图1所示。控制节点主要保存集群中计算节点信息,并建立任务调度机制、数据分片调度和追踪机制,以及并行计算状态追踪机制;计算节点则通过与控制节点通信,开辟内存空间,创建任务线程池,运行控制节点分派的任务。

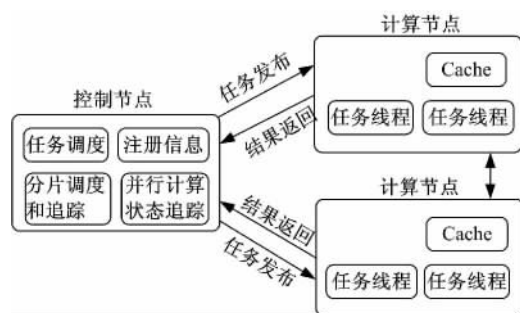


图1 分布式内存计算集群架构

Fig. 1 Distributed memory computing cluster architecture
程序运行于分布式内存集群的流程大致可分为5个阶段:

(1) 初始化集群管理程序。检测集群可用CPU和内存等状态信息。集群管理程序是控制

枢纽,可为后续计算任务分配资源。同时初始化任务调度器和任务追踪器,其功能为分发任务以及收集任务反馈。

(2) 初始化应用运算实例。依据用户提交的程序描述创建分布式的数据集对象,计算数据集的分片,创建数据分片信息列表、数据分片之间依赖关系列表。依据数据的本地性原则,分配相应数据分片存储于指定的计算节点之上。

(3) 构建运算的有向无环图(Directed acyclical graphs, DAG)。将运算过程中涉及到的map、sort、merge、shuffle等计算过程以序列的方式增量累加成DAG图,而后依据DAG图将整个运算过程分解成多个任务集合^[14]。

(4) 任务调度器按照任务执行自顶向下的顺序,将任务集合中的子任务通过集群管理器分发到指定的计算节点之上,每个任务对应着一个数据分片。如果任务失败则重新发布。

(5) 计算节点接收到任务后,为任务分配计算资源,创建进程池开始执行计算,并向控制节点反馈进程分配情况。

集群运行计算过程中需要保证任务的最优调度,即将任务分配到相应计算节点之上,该节点缓存了任务计算所需的数据分片,确保数据的本地性。同时当某个任务运行速度低于一定阈值时,则在其他节点之上重新开启任务。

3 深度神经网络并行结构

整个深度神经网络分布式运行的总体结构如图2所示。系统中的所有数据都经过数据分片操作,图2中小方块即代表一个数据分片,图中的虚线框表示一个数据集,一个数据集由多个数据分片组成,数据分片分布于集群中多台计算机内存之上。训练数据保存在集群的分布式文件存储系统当中,本文采用的是hadoop分布式文件系统(HDFS)。如果训练模型直接从HDFS上面读取数据,则会受到磁盘I/O的限制,所以系统将一部分训练数据用随机采样的方式缓存到集群的分布式共享内存当中,模型训练的过程中直接从采样数据集中提取数据。采样数据集以固定的时间间隔从HDFS文件系统中刷新采样样本集。

系统将深度信念网络模型的所有参数构建成一个分布式的数据集。为了实现模型训练的并行化,系统采用创建多个神经网络模型副本的方式,同时训练各副本模型,即使承载训练任务的计算

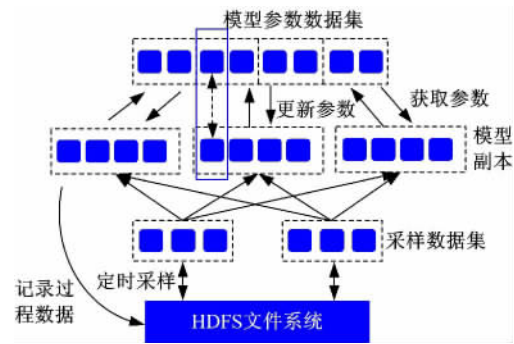


图2 模型并行训练结构

Fig. 2 model distributed training architecture

节点失效,也不会影响训练的整体进程。每个副本也是由多个数据分片组成,利用分布式内存的速度优势,提高模型的训练速度。各副本结束训练之后,将计算后的参数调整值异步地传送给模型参数数据集,并向数据集申请新的参数,进行下一步训练。由于参数数据集与模型副本都分布于集群的多台机器上,副本与参数集的数据传输量即被分散,有效地缓解了计算节点之间的通信压力。为了保证系统的稳定性,在模型的训练过程中设置check point机制,将模型中数据分片计算的过程信息,根据任务进度异步保存到HDFS,防止由于计算节点发生错误需重新运算的情况。

为了提高系统的鲁棒性,采用Adagrad^[15]自适应学习速率方法,随着训练的进行自动调整学习速率,学习速率如式(9)所示:

$$\varepsilon_{ik} = \frac{\mu}{\sqrt{\sum_{j=1}^K \Delta w_{ij}^2}} \quad (9)$$

式中: ε_{ik} 为第*i*个参数在第*k*次迭代时的学习速率; Δw_{ij} 为目标函数的梯度值; μ 为常数。

学习速率的计算仅与参数历史梯度值的平方和有关,所以Adagrad易于在每个参数服务器节点上单独实现。

为了防止模型训练过拟合,本文采用dropout方法^[9]。模型训练时随机让网络某些隐藏层节点的权重不工作,但是它的权重保留下来参与下一轮训练。每次有输入时,网络采样一个随机结构,这样每个神经元都不能依赖于其他神经元的存在,使模型神经元间减少了相互作用,迫使神经元与其他神经元共同学习更多、更有用的特征。

4 试验结果及分析

4.1 试验设置

试验所使用的训练样本为CIFAR-10图片数

数据集。有 60 000 张带分类标签的彩色图片组成。图片的分辨率为 32×32 。使用 50 000 张图片作为训练集,10 000 张作为测试集。试验集群由 9 台虚拟机构成,其中 1 台控制节点,8 台计算节点。每台计算节点虚拟机包含 8 核 CPU、26 G 内存。

深度神经网络包括:输入层、四层堆叠限制玻尔兹曼机作为隐藏层和一个 softmax 输出层。层间采用全连接结构。模型包含超过 1.2×10^6 个的神经元,参数集大小约为 110 M。

4.2 结果及分析

图 3 为分布式内存计算与传统分布式训练速率对比。由图 3 可以看出:采用分布式内存计算的方法训练神经网络模型,模型精度的变化更为陡峭,说明采用分布式内存计算的训练方式能够明显提升模型的训练速率。

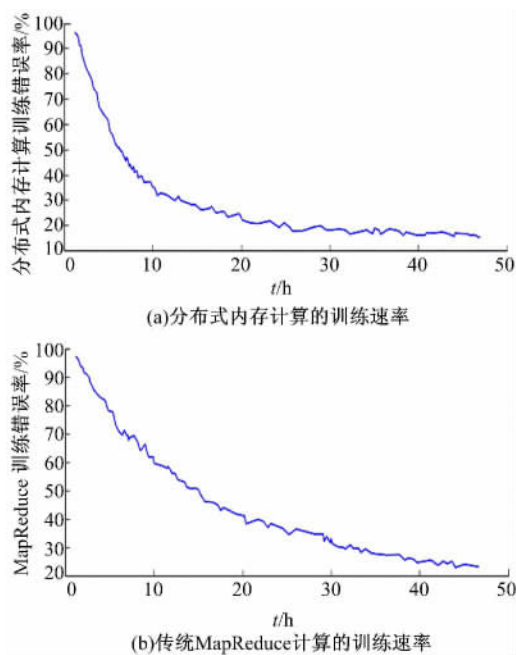


图3 分布式内存计算与传统分布式训练速率对比
Fig.3 Comparison of distributed memory computing and traditional map-reduce

为了说明集群的可扩展性,通过递增训练节点数量来观察集群对模型的加速比值变化。加速比值为单独使用一个计算节点完成一批次采样训练所需的时间与使用 K 台计算节点所用时间的比值。

由图 4 可以看出:当集群不断增加计算节点时,模型的训练效率显著提升,说明这种并行深度学习的方法具备良好的扩展性。

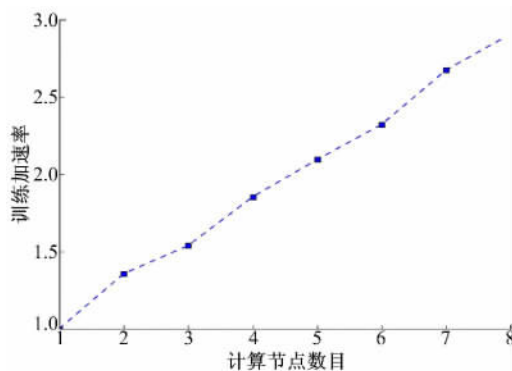


图4 扩展节点对加速比的提升

Fig.4 Speedup calculation by extending worker node

5 结束语

提出了一种分布式内存计算的方法,并应用于深度神经网络的训练,设计了深度神经网络模型在分布式内存中的运行结构,以 CIFAR-10 数据集为训练样本,完成对网络模型的并行训练。实验结果表明,该方法能够有效提升深度神经网络的训练速率,并具备良好的扩展性。理论上集群可以不断扩展以提高训练效率,直到节点间数据交换通信成为瓶颈,由于试验环境的限制,本文无法给出进一步证实。

参考文献:

- [1] Ciresan D C, Meier U, Gambardella L M, et al. Deep big simple neural nets excel on handwritten digit recognition [J]. *Neural Computation*, 2010 (12): 3207-3220.
- [2] Coates A, Lee H L, Ng A Y. An analysis of single-layer networks in unsupervised feature learning [C] // *Proceeding of the 14th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, USA, 2011: 215-223.
- [3] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors [DB/OL]. [2014-05-17]. <http://arxiv.org/abs/1207.0580>.
- [4] Raina R, Madhavan A, Ng A Y. Large-scale deep unsupervised learning using graphics processors [C] // *International Conference on Machine Learning*, Montreal, QC, Canada, 2009: 873-880.
- [5] Le Q V, Monga R, Devin M, et al. Building high-level features using large scale unsupervised learning [C] // *International Conference on Acoustics, Speech and Signal*, Vancouver, Canada, 2013: 8595-8598.
- [6] Glorot X, Bordes A, Bengio Y. Domain adaptation for

- large-scale sentiment classification: a deep learning approach [C]//Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, 2011: 513-520.
- [7] Bengio Y, Courville A C, Vincent P. Representation learning: a review and new perspectives [DB/OL]. [2014-05-23]. <http://arxiv.org/abs/1206.5538>.
- [8] Ngiam J, Coates A, Lahiri A, et al. On optimization methods for deep learning [C]//Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA 2011: 265-272.
- [9] Martens J. Deep learning via hessian-free optimization [C]//Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel 2010: 735-742.
- [10] 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.
- [11] Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups [J]. Signal Processing Magazine 2012, 29(6): 82-97.
- [12] Hinton G, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets [J]. Neural Computation, 2006, 18(7): 1527-1554.
- [13] Hinton G. A practical guide to training restricted Boltzmann machines [J]. Momentum 2010, 9(1): 926.
- [14] Zaharia M, Chowdhury M, Das T, et al. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing [DB/OL]. [2014-01-19]. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-82.pdf>.
- [15] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization [J]. Journal of Machine Learning Research 2011, 12(7): 2121-2159.