深度学习编程框架

王秉睿[1,2]，兰慧盈[2,3]，陈云霁[2]
1. 中国科学技术大学，安徽 合肥 230026；2. 中国科学院计算技术研究所，北京 100190；
3. 中国科学院大学，北京 100049

摘要
近年来，深度学习算法日益流行，在各种领域的应用都取得了出色的效果，受到工业界和学术界的广泛关注。越来越多的研究者开始利用深度学习算法解决实际问题（如图像分类、图像识别、语音识别、自然语言处理等）。人们提出了各种各样的深度学习编程框架，便于研究者们开发新的深度学习算法。这些深度学习库的设计原则、抽象层次各有不同。对常见的深度学习编程框架进行了分类介绍，针对深度学习编程框架设计中的关键问题进行了分析，并且对未来深度学习编程框架的发展方向做了展望，为以后深度学习编程框架的设计提供了设计思路和方向。

关键词
机器学习；编程框架；大数据

中图分类号：F830 文献标识码：A doi: 10.11959/j.issn.2096-0271.2018040

Programming frameworks for deep learning algorithms

WANG Bingrui[1,2], LAN Huiying[2,3], CHEN Yunji[2]
1. University of Science and Technology of China, Hefei 230026, China
2. Institute of Computing Technology Chinese Academy of Sciences, Beijing 100190, China
3. University of Chinese Academy of Sciences, Beijing 100049, China

Abstract
In recent years, deep learning algorithms became increasingly pervasive. It has drew extensive attentions from both researchers and industries, as it achieves very promising results on many applications of various fields. More and more researchers began to use deep learning algorithms to solve practical problems (e.g., image classification, image recognition, speech recognition, and natural language processing). Many deep learning frameworks and libraries were proposed so that researchers can develop new deep learning algorithms in a more convenient fashion. These frameworks and libraries were different in many aspects (e.g., design principles and abstraction). Firstly, several pervasive deep learning frameworks were introduced, and then the critical issue of designing such frameworks was analyzed. At last, the future challenges of designing deep learning frameworks were discussed. The study provides ideas and directions for future design.

Key words
machine learning, programming framework, big data

1 引言

机器学习是一类研究通过计算机模拟人类学习行为，从而解决实际问题的学科。机器学习算法涉及多学科，如概率论、统计学、逼近论、凸分析等。传统的机器学习算法包括k近邻（k-nearest neighbor，k-NN）分类、k均值（k-means）聚类算法、支持向量机（support vector machine，SVM）、决策树（decision tree）等。最近，机器学习中的一个子类——深度学习，在图像处理、语音处理等多个领域上都取得了长足的进步。深度学习的快速发展得益于两点，首先是硬件计算设备性能的提升，比如图形处理器（graphics processing unit，GPU）的广泛应用，其次是大规模数据集的构建，大量的训练数据减轻了过拟合的问题，使得训练复杂的模型成为可能。深度学习算法的结构通常非常复杂，包含上百兆的可训练参数。比如2016年由He K等人[1]提出的ResNet网络，最大的一种模型包含了上千层。

由于算法的复杂性，构建算法的计算过程需要耗费大量的时间。值得庆幸的是，深度学习算法的复杂网络结构是由有限的基本算子通过各种复杂连接关系搭建起来的，常用的算子包括卷积、池化、全连接等。因此，可以提取出机器学习算法中共性的部分，将其抽象出来，便于反复调用。用于机器学习算法的编程库层出不穷，各有特色。比如，Google公司提出的Tensorflow[2]，是一款基于数据流图的深度学习库，支持多种计算设备（CPU、GPU、张量处理单元（tensor processing unit，TPU）[3]），同时还可以直接运行在包含多种计算设备的分布式结构上。除了深度学习算法库，还有针对传统机器学习算法的库。比如，Scikit-Learn用Python语言作为接口，提供大量的机器学习算法，包括有监督学习算法、无监督学习算法、数据降维算法、图像预处理算法等。

现有的深度学习库主要面临两个方面的挑战。一是对更多的计算设备的支持，尤其是对机器学习加速器的支持。目前，多数深度学习库支持的设备是CPU和GPU（如MXNet、Caffe），一些深度学习库仅支持CPU（如Scikit-Learn），只有很少的一部分深度学习库可以支持深度学习加速器，如TensorFlow可以支持TPU。但是，这种支持还是有局限性的，其他的加速器要集成到TensorFlow中会非常困难，同时，TPU也无法被其他编程库利用。另一个挑战是机器学习库的模块化。现在的机器学习库所有模块之间耦合紧密，不利于代码的重用。开发者如果想构建一个新的机器学习库，很难利用现有库的模块进行开发。

2 机器学习算法

2.1 经典机器学习算法

经典机器学习算法指主要依赖统计方法的机器学习算法。和深度学习算法相比，这类机器学习算法的模型较为简单，在小数据集上可以达到很好的预测效果，但是对于大数据集，如ImageNet，预测能力不足。**表1**列举了常见的经典机器学习算法以及它们可以完成的机器学习任务。

2.2 深度学习算法

深度学习算法是机器学习算法中的一类，近年来，随着计算设备的计算能力

表1 常见的机器学习算法

类别	算法	应用
经典机器学习算法	k-NN[4]	聚类（无监督学习）
	k-means[5]、决策树、朴素贝叶斯[6]、逻辑回归、线性回归、支持向量机[7]	分类、回归（有监督学习）
	主成分分析（principle component analysis, PCA）	数据降维
深度学习算法	卷积神经网络（convolutional neural network, CNN）	图像识别、图像分类、语音识别
	递归神经网络（recurrent neural network, RNN）/长短期记忆（long short-term memory, LSTM）网络	自然语言处理

的增强以及大规模数据集的建立，神经网络算法开始从单层逐渐发展到多层，到现在，一个神经网络可能包含上百层以及上百兆的可训练参数。深度神经网络模型主要有：卷积神经网络[8]以及递归神经网络。卷积神经网络[8]是近年来迅速发展起来，并获得了广泛关注的一种前馈人工神经网络算法。它在大规模的图像识别任务上取得了出色的识别率。2012年提出的AlexNet[9]取得了ImageNet2012图像分类任务比赛的第一名，Top 5的识别错误率低至15.3%。之后的几年，这一错误率不断被新提出的CNN刷新，2014年提出的VGG-Net[10]取得了89.3%的平均正确率，2016年He K等人[1]提出的ResNet，又将分类错误率降低到3.57%。通常，卷积神经网络是由很多层构成的，常见的层包括卷积层、池化层、激活层、全连接层。卷积神经网络非常善于处理图像数据，通过卷积层，可以提取出各种图像特征，经过多次卷积处理后，可以提取出比较抽象的图像特征，这些高级的抽象特征则作为分类器（全连接层）的输入，用于图像的分类。递归神经网络是一类用于处理序列数据的神经网络，可以展现输入数据在时序上的行为。不同于前馈神经网络的是，RNN可以利用其内部的记忆来处理任意时序的输入序列，更容易处理输入数据长度不定的情况，比如手写识别、语音识别等。

3 深度学习编程框架

为了帮助深度学习开发者更加快速、方便地开发深度学习算法，各种针对深度学习算法开发的编程库被提出。**图1**展示了一般的深度学习编程库的层次。最上层是编程库提供的编程接口，程序员通过调用编程接口来描述算法的计算过程。对于开发者来说，编程接口的易用性以及接口的表达能力非常重要，对算法的描述会映射到计算图上，对计算图进行优化和调度后，图中的每一个算子会调用硬件的编程接口，如Nvidia厂商发布的、在其GPU设备上处理深度学习算法的高性能库——cuDNN。最后，这些硬件调用接口（高性能库）再进一步生成硬件指令，以在硬件设备上运行。比如，在TensorFlow和MXNet中，它们的编程接口是基于图的，源文件是一个Python文件。而Caffe基于层的框架采用了自定义的prototxt文件构建神经网络。

深度学习库的编程接口主要可以分为3类：一类是基于数据流图的编程接口，流行的基于数据流图的机器学习编程框架包括Tensorflow[2]、MXNet[11]、Theano[12]、Torch7[13]等；另一类是基于层的编程接口，如Caffe[14]；还有一类是基于算法的编程接口，主要用于传统机器学习算法的实现，如Scikit-Learn。

3.1 基于数据流图的编程框架

基于数据流图(data flow graph)[2]的机器学习编程框架利用节点(node)和边(edge)构造的有向图来描述计算过程。节点可以表示一个运算操作,或者表示一块数据的输入起点或者输出终点,边则表示节点之间的输入/输出关系。数据被表示为多维数组(张量)的形式,可以在这些边上进行传输。通过一个节点时,数据就会作为该节点运算操作的输入被计算,计算的结果则顺着该节点的输出边流向后面的节点。一旦输入端的所有数据准备好,节点将被分配到各种计算设备,完成异步并行的执行运算。下面介绍4种流行的基于数据流图的机器学习库。

Theano[12]是一个用Python语言写成的基于图的机器学习库。用户可以定义、优化和评估数学表达式,尤其是包含多维数组的表达式,多维数组是机器学习算法中最常用的一种数据结构。当数据量很大的时候,使用Theano可以获得与手工优化的C代码相媲美的运行速度。Theano也支持跨平台执行,在GPU上,可以获得超过CPU几个数量级的运行速度。Theano将计算机代数系统(computer algebra system,CAS)的各个方面与优化编译器的各个方面结合起来。它还可以为许多数学运算生成定制的C代码。CAS与优化编译的结合,对于复杂的数学表达式被反复计算且计算速度很关键的任务来说特别有用。

Tensorflow[2]是Google公司开发的一种基于数据流图的机器学习编程框架。它具有编程灵活、支持跨平台运行的特点。程序员只需要修改很少量的代码,就可以将在CPU上执行的Tensorflow移植到GPU平台上进行运算。同时,Tensorflow

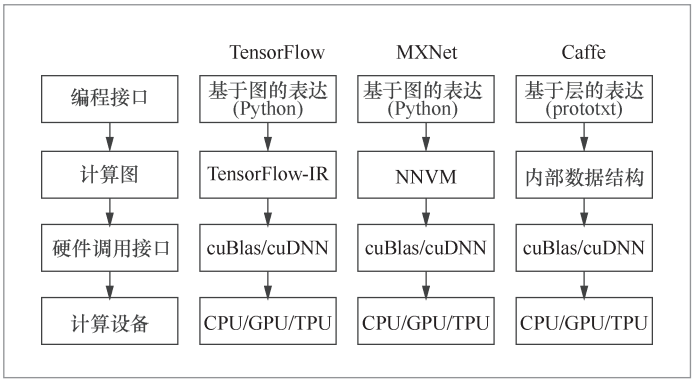

图1 深度学习编程框架结构抽象层次

还可以支持自动异构分布式计算,它的模型能够运行在不同的分布式系统上,系统可以包括多个GPU、CPU、手机节点等。

MXNet[11]也是一个基于计算图模型的机器学习编程库,类似Theano和TensorFlow。它同样可以支持多GPU配置。MXNet包含了类似Lasagne 和Blocks更高级别的模型构建块,并且可以在常见的硬件设备上运行(包括手机、服务器等)。除了支持Python的编程接口外,MXNet还提供了对 R、Julia、C++、Scala、Matlab和JavaScript 的编程接口。

Torch7[13]是一款用于科学计算的编程框架,为机器学习算法提供了充分的支持。Torch7采用数据流图的编程方法,便于构建计算模型。它采用Lua语言作为接口,主要原因是Lua和C/C++语言之间的调用接口友好,开销小,可以更好地支持内嵌CUDA-C优化代码。

3.2 基于层的编程框架

基于层的编程框架为用户提供一组表示各种层(比如卷积层、池化层、全连接层等)的函数作为接口。用户通过反复调用这些层的函数接口构成网络。通过将计算单位限制到层上,库的开发者们可以对各种层进

行充分的性能优化，因此可以提供更好的运行效率。Caffe[14]是一种常用的卷积神经网络框架，它采用层的调用方式，用一个prototxt文件对每一个层进行定义和配置，之后程序分析这个文件，获得这个网络结构的信息。

3.3 基于算法的编程框架

基于算法的编程库提供了大量的机器学习算法，涵盖各种任务和算法。和前两种框架不同，基于算法的编程库不需要构建复杂的网络结构，只需要通过设置库里面提供的算法函数接口中的参数即可完成任务。比如Scikit-Learn机器学习库，它是一种基于Python语言的机器学习库，提供了数据分析和数据挖掘中针对各种任务（包括分类、聚类、数据降维、模型选择、预处理等）的算法，提供了各种不同类别（如k-NN、k-means）的决策树（如C4.5、ID3等）的算法接口，开发者只需要设置接口中的参数，并且将数据传入，就可以得到训练和预测的结果。

3.4 深度学习编程框架中的关键问题

本节将针对3个深度学习编程框架中的问题进行分析。

（1）添加新算子

深度学习算法是由不同的算子构成的，在实现一个新算法的时候，有两种实现方式。首先可以利用深度学习框架中已经有的基本算子（如矩阵计算（矩阵乘法、矩阵转置等）、代数计算（包括标量数据的加减乘除））实现一个新算子。这种处理方法的劣势在于，由于一个算法中包含的算子非常多，算子的调度开销会增大。同时每一个算子都需要调用一次硬件设备，而调用硬件设备需要一定的启动开销，算子越多，整体启动开销越大，导致运行效率低下。因此，通常采用另一种更加高效的实现方式，即添加一个新的算子，算子内部是通过直接调用下一层的硬件调用接口实现的，以此节省算子的调度开销。深度学习算法的发展非常迅速，这使得添加新算子成为一种常见任务。如何方便地添加新算子成为深度学习框架需要研究的一个重要问题。

（2）资源映射策略

神经网络中很多算子的计算规模很大，需要较多的计算资源和内存资源。这可能超过了硬件能够提供的资源，因此需要深度学习框架对硬件上的计算资源和内存资源进行调配，通过将一个操作拆分成更小的子操作，使得硬件的资源可以支持该操作。此外，不同的算子对资源的需求也是不平衡的。这种不平衡使资源的映射和分配变得更加困难。

（3）分布式执行

现在神经网络结构规模越来越大，一些研究人员开始使用包含多种计算设备的集群（cluster）来执行。分布式执行面临的多种困难（如不同算子之间的同步、算子和设备之间的映射、算子到设备上的分配等）也是深度学习架构需要解决的，尤其是针对训练算法的分布式计算。

4 未来研究方向

本文对现有的常用机器学习库进行了介绍和分析。随着机器学习领域的发展，新的需求逐渐被提出，现有的机器学习库要兼容这些新的需求，需要进行进一步的改善。未来，对机器学习库的研究可从以下发展方向展开。

4.1 多样化计算设备支持

机器学习算法的发展越来越快，尤其

是在各种实际情景中，机器学习的应用场景日益增多。人们逐渐发现，传统的计算设备已经无法支持机器学习的计算需求。比如，在手机上，许多应用都需要用到人脸识别技术、语音识别技术。然而，虽然使用深度学习算法进行人脸识别、语音识别任务效果较好，但是计算量巨大，手机的计算资源有限，无法支持这样大规模的计算任务。在这样的情况下，研究人员提出了专门针对机器学习算法的加速器，利用机器学习算法中大量的数据重用，有效降低了功耗，提高了计算性能[15-20]。机器学习加速器虽然有巨大的计算潜力，但是由于缺乏合适的编程框架，机器学习开发者实际上很难在机器学习加速器上进行开发。因此，机器学习库除了需要支持CPU、GPU之外，还需要进一步支持机器学习加速器，尤其是对跨平台（多加速器）的支持。现在，虽然有一些加速器可以被集成到框架中，如TensorFlow可以支持同是Google公司研发的TPU[3]，但是，其他加速器要想集成到TensorFlow中则非常困难。

4.2 模块化系统定制

模块化是机器学习库发展的另一个方向。一个机器学习库一般由几个层次构成，各层次都是非模块化的，与自己的系统是紧耦合的，开发者如果想要定制自己的机器学习系统，很难复用现有库中的模块，这导致本来可以重用的模块还需要重新实现一次。现在，已经有一些开发者提出针对机器学习算法的中间语言（intermediate representation，IR）[21-22]以及相应的编译器，如由Chen T等人[22]提出的TVM，是一个针对深度学习的编译软件栈，其中用到的NNVM，就是一个开放的模块化的计算图的中间语言。

5 结束语

机器学习算法已经被学术界和工业界广泛接受，成为一种实用有效的算法，应用在了很多任务上，如图像识别、图像分类等。本文首先介绍了常用的机器学习算法，包括经典的基于统计学的机器学习算法以及深度学习算法（如CNN、RNN/LSTM）。由于机器学习算法编程的复杂性，人们提出了各种深度学习编程框架，从而为深度学习算法的开发提供便利。本文主要介绍了3类深度学习开发库：基于数据流图的编程库、基于层的编程库以及基于算法的编程库。

参考文献:

[1] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 26-July 1, 2016, Las Vegas, USA. Piscataway: IEEE Press, 2016: 770-778.

[2] ABADI M, BARHAM P, CHEN J, et al. TensorFlow: a system for large-scale machine learning[C]//The 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), November 2-4, 2016, Savannah, USA. Berkeley: USENIX Association, 2016: 265-284.

[3] JOUPPI N P, YOUNG C, PATIL N, et al. In-datacenter performance analysis of a tensor processing unit[C]// The 44th Annual International Symposium on Computer Architecture, June 24-28, 2017, Toronto, Canada. New York: ACM Press, 2017: 1-12.

[4] ZHANG M, ZHOU Z. ML-KNN : a lazy learning approach to multi-label

learning[J]. Pattern Recognition, 2007, 40(7): 2038-2048.

[5] YANG S L, LI Y S, HU X X, et al. Optimization study on k value of kmeans algorithm[J]. Systems Engineering-Theory & Practice, 2006, 26(2): 97-101.

[6] MADIGAN D, YORK J. Bayesian graphical models for discrete data[J]. International Statistical Review, 1995 , 63(2): 215-232.

[7] OSUNA E, FREUND R, GIROSI F. Training svm: an application to face detection[R]. 1997.

[8] LECUN Y, KAVUKCUOGLU K, FARABET C. Convolutional networks and applications in vision[C]// International Symposium on Circuits and Systems, May 30-June 2, 2010, Paris, France. Piscataway: IEEE Press, 2010: 253-256.

[9] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]// The 25th International Conference on Neural Information Processing Systems, December 3-6, 2012, Lake Tahoe, Nevada. New York: Curran Associates Inc., 2012: 1-9.

[10] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. Computer Science, 2014: arXiv:1409.1556.

[11] CHEN T, LI M, LI Y, et al. MXNet : a flexible and efficient machine learning library for heterogeneous distributed systems[J]. Statistics, 2015: arXiv:1512.01274.

[12] TEAM T D, ALRFOU R, ALAIN G, et al. Theano: a Python framework for fast computation of mathematical expressions[J]. Computer Science, 2016: arXiv:1605.02688.

[13] COLLOBERT R, KAVUKCUOGLU K, FARABET C. Torch7: a matlab-like environment for machine learning[C]// The 25th Annual Conference on Neural Information Processing Systems, December 12-14, 2011, Granada, Spain. [S.l.:s.n.], 2011: 1-6.

[14] JIA Y, SHELHAMER E, DONAHUE J, et al. Caffe: convolutional architecture for fast feature embedding[J]. Computer Science, 2014: arXiv:1408.5093.

[15] CHEN T, DU Z, SUN N, et al. DianNao: a small-footprint high-throughput accelerator for ubiquitous machine-learning[C]//The 19th International Conference on Architectural Support for Programming Languages and Operating Systems, March 1-5, 2014, Salt Lake City, USA. New York: ACM Press, 2014: 269-284.

[16] CHEN Y, LUO T, LIU S, et al. DaDianNao: a machine-learning supercomputer[C]//The 47th Annual IEEE/ACM International Symposium on Microarchitecture, December 13-17, 2014, Cambridge, UK. Washington DC: IEEE Computer Society, 2014: 609-622.

[17] DU Z, FASTHUBER R, CHEN T, et al. ShiDianNao[C]//The 42nd Annual International Symposium on Computer Architecture, June 13-17, 2015, Portland, USA. [S.l.:s.n.], 2015: 92-104.

[18] LIU D, CHEN T, LIU S, et al. PuDianNao: a polyvalent machine learning accelerator[C]//The 20th International Conference on Architectural Support for Programming Languages and Operating Systems, March 14-18, 2015, Istanbul, Turkey. New York: ACM Press, 2015: 369-381.

[19] ZHANG S, DU Z, ZHANG L, et al. Cambricon-X: an accelerator for sparse neural networks[C]// The 49th Annual IEEE/ACM International Symposium on Microarchitecture, October 15-19, 2016, Taipei, China. Piscataway: IEEE Press, 2016: 1-12.

[20] LIU S, DU Z, TAO J, et al. Cambricon: an instruction set architecture for neural networks[C]//The 43rd International Symposium on Computer Architecture, June 18-22, 2016, Seoul, Korea. New York: ACM Press, 2016: 393-405.

[21] WEI R, SCHWARTZ L, ADVE V. DLVM: a modern compiler infrastructure for deep learning systems[J]. Computer Science, 2017: arXiv:1711.03016.

[22] CHEN T, MOREAU T, JIANG Z, et al. TVM: end-to-end optimization stack for deep learning[J]. Computer Science, 2018: arXiv:1802.04799v1.

作者简介

王秉睿(1994-),男,中国科学技术大学硕士生,主要研究方向为计算机体系结构、机器学习编程方法、人工智能。

兰慧盈(1990-),女,中国科学院计算技术研究所博士生,主要研究方向为计算机体系结构、领域专用编程语言和编译器、人工智能。

陈云霁(1983-),男,博士,中国科学院计算技术研究所研究员(正教授)、博士生导师,主要研究方向为计算机体系结构、人工智能。

收稿日期: 2018-05-15

基金项目: 国家重点研发计划基金资助项目(No.2017YFB1003101);国家自然科学基金资助项目(No.61472396, No.61432016, No.61473275, No.61522211, No.61532016, No.61521092, No.61502446, No.61672491, No.61602441, No.61602446, No.61732002, No.61702478);北京市科技计划基金资助项目(No.Z151100000915072);中国科学院STS计划;国家重点基础研究发展计划("973"计划)基金资助项目(No.2015CB358800)

Foundation Items: National Key Research and Development Program of China (No.2017YFB1003101), The National Netural Science Foundation of China (No.61472396, No.61432016, No.61473275, No.61522211, No.61532016, No.61521092, No.61502446, No.61672491, No.61602441, No.61602446, No.61732002, No.61702478), Beijing Science and Technology Projects (No.Z151100000915072), STS Project of Chinese Academy of Sciences, The National Key Basic Research Program of China(973 Program) (No.2015CB358800)