软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, [doi: 10.13328/j.cnki.jos.006064]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 面向移动终端智能的自治学习系统*

徐梦炜[1,2], 刘渊强[1,2], 黄 康[3], 刘譞哲[1,2], 黄 罡[1,2]

[1](北京大学 信息科学技术学院 软件研究所,北京 100871)

[2](高可信软件技术教育部重点实验室(北京大学),北京 100871)

[3](领规科技 北京有限公司,北京)

通讯作者: 刘譞哲, E-mail: xzl@pku.edu.cn

**摘 要**: 在移动终端设备中部署机器学习模型已经成为学术界和产业界的研究热点,其中重要的一环是利用用户数据训练生成模型.然而,由于数据隐私日益得到重视,特别是随着欧洲出台 GDPR、我国出台《个人信息保护法》等相关法律法规,导致开发者不能任意从用户设备中获取训练数据(特别是隐私数据),从而无法保证模型训练的质量.国内外学者针对如何在隐私数据上训练神经网络模型展开了一系列研究,我们对其进行了总结并指出其相应的局限性.为此,本文提出了一种新型的面向移动终端隐私数据的机器学习模型训练模式,将所有与用户隐私数据相关的计算任务都部署在本地终端设备,无需用户以任何形式上传数据,从而保护用户隐私.我们称这种训练模式为自治式学习(Autonomous Learning).为了解决自治式学习面临的移动终端数据量不足与计算能力不足两大挑战,我们设计实现了自治学习系统 AutLearn,通过云(公共数据,预训练)和端(隐私数据,迁移学习)协同的思想,以及终端数据增强技术,提高终端设备上模型的训练效果.进一步地,通过模型压缩,神经网络编译器优化,运行时缓存等一系列技术,AutLearn 可以极大地优化移动终端上的模型训练计算开销.我们基于 AutLearn 在两个经典的神经网络应用场景下实现了自治式学习,实验结果证明了 AutLearn 可以在保护隐私数据的前提下,训练模型达到甚至超过传统的集中式/联邦式模式,并且极大地减小了在移动终端上进行模型训练的计算和能耗开销.

**关键词**: 机器学习;移动计算;边缘计算;分布式系统

# Autonomous Learning System towards Mobile Intelligence

XU Meng-Wei[1,2], LIU Yuan-Qiang[1,2], HUANG Kang[3], LIU Xuan-Zhe[1,2], HUANG Gang[1,2]

[1] (Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

[2] (Key Labortory of High Confidence Software Technologies of Ministry of Education (Peking University), Beijing 100871, China)

[3] (Linggui Tech, Beijing, China)

**Abstract**: How to efficiently deploy machine learning models on mobile devices has drawn a lot of attention in both academia and industry, among which the model training is a critical part. However, with increasingly public attention on data privacy and the recently adopted laws and regulations, it becomes harder for developers to collect training data from users and thus cannot train high-quality models. Researchers have been exploring approaches of training neural networks on decentralized data. We will summarize those efforts and point out their limitations. To this end, this work presents a novel neural network training paradigm on mobile devices, which distributes all training computations associated with private data on local devices and requires no data to be uploaded in any form. We name such training paradigm autonomous learning. To deal with two main challenges of autonomous learning, i.e., limited data volume and insufficient computing power available on mobile devices, we design and implement the first autonomous learning system AutLearn. It incorporates the cloud (public data, pre-training) – client (private data, transfer learning) cooperation methodology and data augmentation techniques to ensure the model convergence on mobile devices. Furthermore, by utilizing a series of optimization techniques such as model compression, neural network compiler, and runtime cache reuse, AutLearn can significantly reduce the on-client training cost. We implemented two classical scenarios of autonomous learning based on AutLearn and carried out a set of experiments. The results showed that AutLearn can training the neural networks with comparable or even higher accuracy compared to traditional centralized/federated

收稿时间: 2020-02-07; 修改时间: 2020-04-04; 采用时间: 2020-05-09; jos 在线出版时间: 2020-06-10

training mode with privacy preserved. AutLearn can also significantly reduce the computational and energy cost of neural network training on mobile devices.

**Key words**:machine learning; mobile computing; edge computing; distributed system

近些年,以神经网络(Neural Networks)为代表的机器学习技术得到了快速的发展,提升了计算机系统的智能化程度,在计算机视觉、自然语言处理等多个重要研究领域取得了广泛应用.近年来,机器学习应用的发展趋势之一是由云端逐渐迁移到终端设备如智能手机.例如,近期的实证研究[1]发现,Google Play 应用商店中搭载神经网络并进行本地计算的移动应用数量在 2018 第三季度内增加了 27%,这些深度学习应用拥有千万级的用户下载与评论量,占应用商店中所有应用下载评论总量超过 10%.此外,各大互联网厂商纷纷自研了面向移动终端的深度学习框架,包括 Google 的 TF Lite,Facebook 的 Caffe2,苹果的 Core ML,高通的 SNPE,腾讯的 ncnn 等,以优化神经网络模型在这些资源受限设备上的部署.考虑到神经网络在性能上的优势以及部署上更具有挑战性,本文将以神经网络为代表研究终端设备上的机器学习部署问题.

开发面向移动终端的神经网络面临的核心挑战之一是:模型的训练需要大量的数据,而这些数据往往来自于终端设备本身且包含大量用户隐私信息.例如,输入法应用的词预测任务中需要用户的输入文本作为训练集,其中可能包含用户的信用卡号、聊天记录、邮件等隐私信息.近年来,为了更加规范地保护用户数据隐私,各个国家机构出台了相关的法律法规.例如,欧盟于 2018 年正式施行《通用数据保护条例》(GDPR)[2],其中详细规定了公司在涉及用户数据时需要遵循的原则;我国制订中的《个人信息保护法》对 APP 违规收集用户信息等行为下了法律定义,对用户隐私保护做到有法可依.

如何在保护用户隐私的前提下训练机器学习模型正在逐渐引起学术界和工业界的广泛关注.以联邦学习[3]、同态加密[7]、差分隐私[11]为代表的技术在某种程度上提供了解决方案,但都存在一定的局限性.例如,联邦学习通过将数据和训练分布在不同终端设备上,通过中心化的参数服务器来分发模型、聚合梯度,从而达到隐私保护的目的.但是在联邦学习的过程中,依旧存在着上传梯度被攻击、网络带宽开销大等问题.同态加密利用经典的密码学算法实现数据保护,但会造成大量的额外计算开销,导致其在复杂神经网络上无法用于实践.

为此,本文提出了**“自治式学习”**的神经网络训练模式:将与用户数据相关的模型计算任务完全部署在本地终端,保证数据不会以任何形式(原始数据、加密数据、模型梯度等)传到外部.本文主要关注其中的模型训练过程.与传统的集中式或联邦式学习相比,自治式学习能够更大程度地保护用户隐私之外,还具有另外两个优势:(1)个性化学习:自治式学习利用每个终端上产生的数据训练各自的模型,然后服务该特定的用户,意味着该模型是为每个用户定制化训练学习得到的.在很多移动场景下,用户的行为习惯(如输入法,语音助手)各异,定制化训练得到的模型相较所有用户共享的全局模型具有更高的准确率.(2)高扩展性:传统的集中式/联邦式的学习模式所需要的硬件资源,包括计算、存储、网络带宽等,与接入的终端设备数量呈近似线性关系,对于用户量巨大的应用而言,这会带来应用开发和维护成本的巨大增加.而自治式学习要求每一台接入设备只通过本地的计算存储资源进行模型训练,云端无需提供更多的硬件资源,因此具有更强的扩展性.

实现一个自治式学习系统面临两个主要挑战:首先,如何在终端上数据不传到外部的前提下,训练出高质量的神经网络模型?其次,每个终端设备上的硬件资源受限,如何在不影响用户体验的前提下,快速完成训练并进行部署?为了解决这两个问题,我们设计实现了第一个面向移动终端的自治式学习系统 AutLearn,其核心思想是采用**云端协同思想**,在云端服务器首先利用公共数据集训练一个泛化能力较强的模型,然后在终端利用迁移学习技术对模型进行调整,得到一个适用于该终端设备的可用于部署的模型.云端的预训练使用公共的非隐私数据集,而无需用户上传任何数据;终端的迁移学习使用了数据增强技术,以提高训练的效果.为了不影响用户在使用终端设备过程中的体验,我们将终端迁移学习进一步分成了两种模式:终端离线学习与终端在线学习,以适用于不同的训练场景与目标.此外,AutLearn 引入了一系列的优化技术,包括模型压缩,运行时缓存优化,神经网络编译器等.这些技术在保证模型精度(或极少精度损失)的前提下,极大地节省了终端设备上神经网络训练的资源开销.

我们基于 AutLearn 实现了两个自治式学习的典型应用场景:输入词预测与图像分类,并在大规模数据集和

经典神经网络模型上进行了验证.实验结果表明:AutLearn 可以在保护用户隐私的前提下,训练得到收敛至较高准确率的神经网络模型.与集中式/联邦式的训练模式相比,自治式训练得到的模型可以达到相近甚至更高的准确率,产生个性化定制的效果.此外,AutLearn 的终端训练优化技术可以节省最多超过 80%的训练时间和终端设备能耗.结合我们对真实智能手机用户使用行为的观察分析,AutLearn 可以在一天内完成新模型的训练用于部署.

## 1. 研究背景与相关工作

### 1.1 研究背景

以深度学习算法为代表的机器学习技术已经得到了学术界广泛的关注,并且在工业界也有了大量的部署.深度学习模型的质量依赖于训练数据,后者的数量与质量决定了最终训练得到模型的准确率是否满足要求.然而,这些数据往往需要从用户设备中采集,并包含用户的隐私信息.这些数据一旦离开终端设备就会造成隐私泄露的风险.例如,输入法应用的词预测任务中需要用户的输入文本作为训练集,其中可能包含用户的信用卡号、聊天记录、邮件等隐私信息.当数据被上传至服务器后,传统的信息安全技术如加密传输,用户匿名处理等都无法保证用户数据不会被泄露和恶意地使用.为了更加规范的保护用户数据,各个国家机构出台了相关的法律.例如,欧盟于 2018 年正式施行《通用数据保护条例》[1],详细规定了公司在涉及用户数据时需要遵循的原则.如何在保证用户隐私的前提下训练高质量的深度学习模型是一项巨大的挑战.为此,相关的研究人员提出了不同的解决方案.我们概述其中几个主要相关工作,总结各自的不足.

### 1.2 相关工作

联邦学习技术最初是由 Google AI 团队提出[3],其核心思想在于不直接上传用户数据,而是将模型训练任务部署到终端设备,后者在训练结束后上传模型的更新到云端,而云端只需要对来自不同终端设备的模型更新聚合,以及下发聚合后的模型.由于输入数据到梯度是多对多的高维空间映射,攻击者很难根据模型梯度获取原始数据.因此,联邦学习在一定程度上保护了用户隐私.后续有大量的工作关注于如何优化联邦学习的流程.例如,为了减小终端到服务器的通讯开销,Konečný 等人[4]提出了两种优化策略:结构化更新与速写式更新.前者将模型参数限制在一个相对更小的解空间里,使得模型可以使用更少的参数表示,从而减少了模型梯度的大小.后者仍旧使用原模型进行训练,但在上传之前对训练得到的模型梯度进行压缩如量化、降采样等;为了解决联邦学习过程中的终端设备异构性问题,Li 等人[5]提出的 SmartPC 系统使用动态步调同步策略,在每一轮训练根据上一轮收到终端模型更新的延迟来动态调整当前的等候期限.其次,该工作还提出了通过动态调整终端设备的 CPU 运行频率,在保证梯度可以在等候期限之前被上传,并最小化训练能耗;为了进一步优化联邦学习过程中的数据安全,Bonawitz[6]等人提出模型的更新通过安全多方计算技术在终端进行融合,从而保证服务器只能看到融合后的整体更新,而无法通过个体的模型更新来推断其训练数据.

同态加密算法允许对密文进行特定形式的代数运算后,得到仍然是加密的结果,将其解密所得到的结果与对明文进行同样的运算结果一致.换言之,这项技术令人们可以在加密的数据中进行诸如检索、比较等操作,得出正确的结果,而在整个处理过程中无需对数据进行解密.一些相关工作[7][45]探究了如何将同态加密运用于机器学习训练过程,但主要集中于一些简单算法如支持向量机、决策树等.由于深度学习运算的复杂性,为其设计同态加密算法具有更大的挑战.同时,在加密数据上进行训练也远远比预测[8]更加复杂.Nandakumar 等人[9]初步探讨并实现了如何利用同态加密技术在加密数据上进行深度学习训练.作者使用了 ciphertext packing 技术对训练进行加速,但训练的效率依旧不高,无法支持卷积层操作,在训练得到模型精度上甚至还有少量的损失.

差分隐私最初是针对数据库查询而设计的,其目的在于将用户隐私量化,保证用户只能够获取数据库的统计信息,而无法获得单独的个体信息,同时最大化数据查询的准确性.差分隐私的具体实现方式主要是在查询结果里加入随机性,如服从 Laplace 分布和指数分布的噪音.为了将差分隐私技术应用于机器学习训练过程中的数据保护,Abadi 等人[11]提出修改模型的训练算法,在每一个 batch 的数据训练得到的梯度之上加入扰动,然后将扰

动后的梯度应用于模型训练.在该过程中,还可以累加扰动的数值,从而在训练结束之后得到整体隐私的量化度.该技术也已经被集成到 TensorFlow 的开源项目中,以帮助开发者训练具有隐私保护的深度学习模型.Shokri等人[12]提出部分梯度共享的策略,即在每一轮训练结束之后,选择性的分享梯度中的一部分,然后将多个组织的梯度聚合,整体应用于旧模型参数上,得到更新后的模型.这种策略能够有效工作的主要原因在于梯度下降训练法本身对不可靠的带有随机性质的模型梯度具有很强的健壮性.在这种策略之上,作者进一步地采用差分隐私技术来量化并减少隐私泄露的风险.

**小结:**以上技术虽然能在某种程度上保护用户隐私,但同样存在着其局限.例如:联邦学习技术始终要求用户上传数据相关信息(模型梯度或者模型本身),导致其依旧有隐私泄露的风险[33],并且联邦学习本身会导致大量的网络通信开销,对带宽的要求较高; 同态加密技术会导致大量的计算开销,无法被应用于复杂的神经网络结构及实践之中; 差分隐私技术主要用于保护训练后得到的模型中蕴含的数据信息,而无法保护训练过程中训练数据不被获取以及恶意地使用.

## 2. 分布式自治学习:优势,挑战,及可行性分析

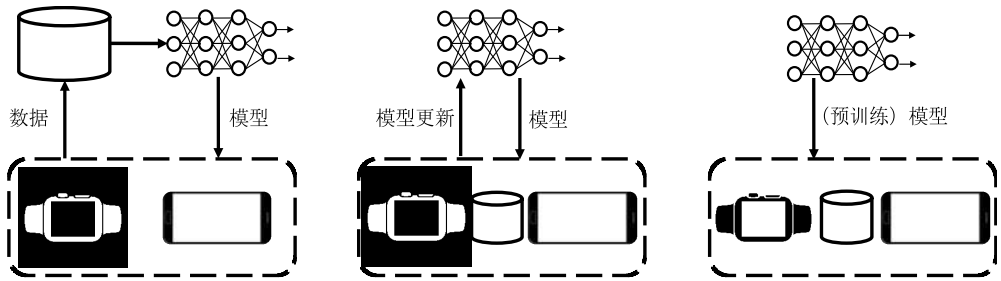


Fig.1 A comparison of different machine learning training paradigms

图 1 多种机器学习训练模式的比较示意图

本文提出分布式自治学习的概念,其核心思想是:*终端设备无需以任何形式上传用户数据,与用户数据相关的运算(模型训练)全部在本地进行*.在某种程度上,自治式学习可以视为集中式学习到近些年的联邦式发展方向的一种特殊情况,即,原始数据上传(集中式学习)->训练结果数据上传(联邦学习)->无数据上传(自治式学习),其对隐私的保护程度也依次递增.在自治式学习中,每个客户端设备完全不依赖于任何外部的数据与计算能力,从一个初始模型出发,通过本地数据和计算能力获得一个新的部署于本地的高精度定制模型.

这种自治式学习的计算模式也契合了近些年逐步引起重视的去中心化思想.联邦式学习虽然在某种程度上进行了去中心化(相较集中式学习),但是依旧存在一个中心化节点来协调各个终端的计算任务,汇总这些终端上传的模型参数.而自治式学习实现了更加彻底的去中心化机器学习训练模式.

**我们认为,**自治式学习的优势主要体现在以下三个方面.

- 用户隐私保护:由于终端设备没有任何数据上传操作,所有数据在本地终端设备上产生、存储、并消耗,极大地减小了隐私泄露的风险.相对应的,类似联邦式的机制依旧需要上传与用户隐私间接相关的数据(如模型更新梯度),依旧存在隐私泄露的风险.从最终用户的角度出发,将数据保留在本地显然更容易让人接受,更利于各国数据保护条例的实施.
- 个性化模型:自治式学习的过程是利用每个终端上产生的数据训练各自的模型,这个模型会被部署于服务该特定的用户,这即意味着模型是为每个用户定制化训练学习得到的.在很多场景下,用户的行为习惯(如输入法,语音助手)各异,一个全局统一的、由集中式或联邦式训练出的全局模型无法很好地服务于不同用户,为此需要自治式学习获得的个性化能力.

- 更强的扩展性:自治式学习意味着每一台新接入的设备都会提供自己的计算存储资源用于模型的训练,而云端无需提供更多的硬件资源,因此具有很强的扩展性.而集中式/联邦式的学习模式则要求云端为新接入的终端设备提供更多的硬件支持,包括计算、存储、网络带宽,对于用户量巨大的应用而言,这会造成极大的经济开销.

事实上,传统的观点认为用户隐私和用户个性化很难同时达到[20],例如推荐系统,因为两者在本质上存在着矛盾.自治式学习通过充分利用终端设备本地计算能力,在某种程度上可以解决这个问题.

**挑战**:实现一个自治式学习系统的挑战主要存在于两个方面:终端设备本地数据量不足与终端计算力不足.首先,训练机器学习模型,尤其是神经网络模型需要大量的数据,而每个终端设备上的训练数据往往有限,在不足量上训练模型可能导致模型无法收敛的问题,无法得到高质量的模型.其次,模型的训练需要大量的计算资源,而终端设备往往计算能力有限,并且考虑到终端设备的复杂运行环境和用户的交互,无法在任意时间用于模型训练.一个直接的问题是:当一个新的模型结构被设计出来,分发到各个终端设备后,需要多久才能完成训练并用于部署?

**可行性分析**:离线模型训练是一种最直接的自治学习方式,即客户端从云端获取一个新的模型结构后,选择适宜的时机对该模型进行训练,完成后用于部署.这里“适宜”的训练时机主要指的是不会影响用户的使用体验.在大部分联邦学习系统中[13],模型的训练被要求只在满足以下条件时才会进行:1. 设备充电, 2. 设备屏幕关闭(即没有被用户交互使用), 3. 设备处在不计费的网络环境下(如 WiFi).对于自治学习而言,由于没有模型上传的需求,因此条件 3 可以被省去.但即便如此,这些苛刻的条件仍然极大地限制了终端设备可以参与离线训练的时间.我们发现现有的工作缺乏相关的真实用户数据支持这种离线的深度学习训练模式.为此,我们采集了近 1500 名真实用户使用智能手机的行为数据,包括这些设备上硬件状态(网络状态、充电状态、屏幕状态)的转变.

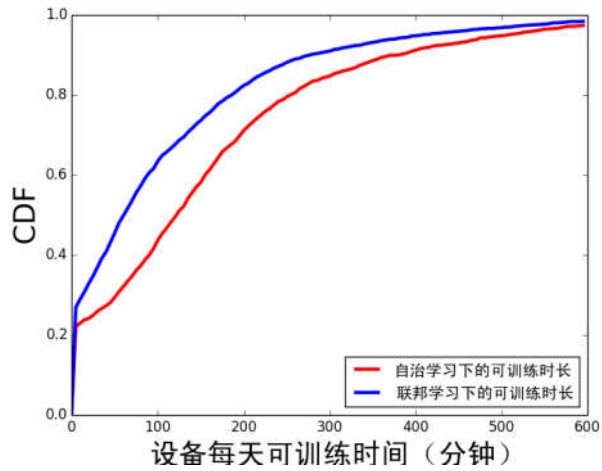


Fig.2 The trainable time per day under federated learning and autonomous learning

图 2 联邦学习与自治学习下,单个移动终端每天可用于模型训练的时间

图 2 展示了用户行为数据的分析结果:其中横坐标代表单个终端每天可用于模型训练的时间,即满足上述系统状态条件的时长;纵坐标代表该时长在不同设备与用户上的累积分布函数 CDF.我们发现每个用户每天可用于自治式训练的时间约为 120 分钟(中位数),而联邦学习的时间为 60 分钟(中位数),原因是联邦学习对设备参与训练的条件更加苛刻,即必须满足设备在不计费的网络环境下.我们认为 120 分钟的时间通常足以完成大量的模型训练任务,结合我们在下文提出的 AutLearn 系统与其优化技术,足以实现自治式学习的目的.我们

将在第 4.3 章中具体介绍了自治学习在终端设备上的计算开销.

## 3. 分布式自治学习系统 AutLearn:设计与实现

我们实现了第一个分布式自治学习系统框架 AutLearn.本节介绍 AutLearn 系统的设计和具体实现:首先介绍该系统的核心思想；然后,介绍总体架构以及系统中每一个功能模块的具体实现；最后,以输入法和图像处理为例介绍特定的应用如何在该系统上运行,以达到自治学习的目的.

### 3.1 核心思想

为了解决终端设备上训练数据不足的问题,AutLearn 采用了云端协同思想,在云端服务器上利用公共数据训练一个泛化能力较强的模型,然后在终端利用迁移学习[18]技术对模型进行调整,得到一个适用于该终端设备的可用于部署的模型.为了保证用户的隐私,云端服务器在预训练模型时不能要求用户上传数据,而是通过数据挖掘的方式从公共数据中寻找与用户数据分布相似的替代数据,达到公私合赢的目标.这里的公共数据包括那些获得用户授权的数据,通过合法途径购买的数据,利用爬虫技术从公网上爬取的数据等等.此外,我们还引入了数据增强技术用于提高终端上用于训练的数据量.

为了解决终端设备计算资源及电量受限的问题,AutLearn 引入了一系列的优化技术,包括模型层压缩,运行时缓存优化,神经网络编译器等.这些技术在保证模型精度(或极少精度损失)的前提下,极大地节省了终端设备上神经网络训练的资源开销.

### 3.2 总体架构

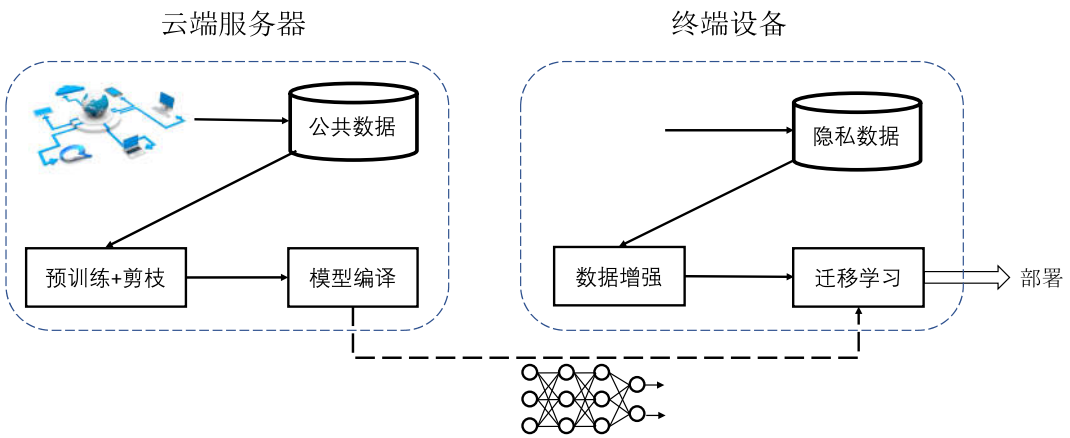


Figure 3 Architecture of AutLearn

图 3 AutLearn 系统架构图

本文提出的自治学习系统框架 AutLearn 如图 3 所示.在云端服务器,首先利用公共数据集对模型进行预训练.考虑到模型需要部署在终端,AutLearn 内置了基于奇异值分解[30]的模型剪枝算法.剪枝结束之后需对模型进行重新训练以恢复模型精度.接下来,AutLearn 会利用神经网络编译技术对剪枝后的模型进行编译,从而优化其在相应终端计算硬件(CPU、GPU 等)上的运算效率.关于编译的具体内容将第 3.4 章进行介绍.

编译后的模型被下载到各个终端设备,利用本地数据在适宜的时间段进行迁移学习,这个过程我们称之为离线学习.其中,AutLearn 会对本地数据进行增强以提高数据量.训练结束后模型会被用于部署,执行神经网络的前向推断.为了让模型可以持续学习用户行为的改变以获取更高的准确率,AutLearn 同时实现了在线学习,即在终端设备上生成训练数据之后立即进行训练.相较于离线学习,在线学习的优势是可以更快地对用户行为改变

进行适应,从而在接下来的推断中得到更合理的结果,且数据无需保存在本地,进一步地减少了隐私泄露的风险.具体地,使用离线学习还是在线学习,亦或是两者结合,需要取决于具体的应用场景.我们将在第 3.6 章中介绍如何选择合适的训练模式.

### 3.3 云端协同训练

云端协同训练的目的是在云端利用公共数据集训练得到一个泛化能力较强的模型,从而在各个终端可以利用本地数据集对该模型进行精调（Fine Tune）,达到收敛并个性化的目的.迁移学习技术的主要特点与优势在于它放宽了传统机器学习中的两个基本假设:(1)用于学习的训练样本与新的测试样本满足独立同分布的条件;(2)必须有足够可利用的训练样本才能学习得到一个好的模型.在自治式学习中同样存在这两个问题,即预训练数据很难与真实用户数据服从同样的分布；以及每个终端上可用于训练的数据有限.在 AutLearn 的设计中,获取合适的公共数据集是能否让模型在终端上收敛并得到高准确率的关键,其数据的语义和分布需要与终端设备上的整体数据较为接近,或呈包含关系,例如在输入法的例子中,我们用从 twitter 网站上爬取的语料数据集做预训练,终端设备上使用用户的输入法数据做迁移学习.在现实中,twitter 的数据很大一部分就是用户通过手机上传的,因此这些数据在分布和语义上存在着很强的关联性,意味着从 twitter 数据中训练得到的模型往往会学习到包含用户输入数据的特征.此外,迁移学习的常用方式是固定前面网络层的参数,这些参数只参与前向推断,而不会参与后向的训练,只有最后一层的参数会随着训练的进行而改变,这也是为什么迁移学习能利用小样本就可以完成训练,且对计算力要求不高的主要原因.

为了进一步提高本地迁移学习的效果,AutLearn 还使用了数据增强技术以提高训练数据量.对于图片数据,AutLearn 使用了常用的数据增强方法如翻转(Flip)、旋转(Rotation)、比例缩放(Scale)、裁剪(Crop)、移位(Translation)、添加高斯噪声(Gaussian Noise)等.对于自然语言类数据,AutLearn 使用了 EDA[15]开源项目中的技术,具体包含四种主要的操作:(1) 同义词替换(SR: Synonyms Replace):不考虑 stopwords,在句子中随机抽取 n 个词,然后从同义词词典中随机抽取同义词,并进行替换.(2) 随机插入(RI: Randomly Insert):不考虑 stopwords,随机抽取一个词,然后在该词的同义词集合中随机选择一个,插入原句子中的随机位置.该过程可以重复 n 次.(3) 随机交换(RS: Randomly Swap):句子中,随机选择两个词,位置交换.该过程可以重复多次.(4) 随机删除(RD: Randomly Delete):将句子中的每个词以概率 p 随机删除.具体地,使用何种数据增强技术或几种技术的组合需要根据具体的应用场景(分类任务)来决定.

Table 1 A comparison of different on-device transfer learning in AutLearn

表 1 AutLearn 中的两种终端迁移学习方式比较

| | 训练数据 | 训练时机 | 优势 |
|---|---|---|---|
| 离线学习 | 保存在本地的历史数据 | 离线空闲时间:手机充电且屏幕关闭 | • 历史积累下的数据通常量较多,训练结果明显,可以有效地解决冷启动问题.<br>• 离线进行不影响用户体验,通常有大量的时间可以用于训练. |
| 在线学习 | 用户在使用设备过程中实时产生的数据 | 在线进行,通常伴随用户与设备交互 | • 用户在线产生的数据通常比较缓慢,使用这些数据进行训练对用户体验影响较小<br>• 无需等待特定的训练时机,在线学习的过程中不断优化模型,快速提高模型精度. |

AutLearn 将终端设备上进行的迁移学习按照训练发起的时机和训练数据产生的时间分为两种:离线学习和在线学习(见表 1).离线学习是基于历史数据,选择适宜的时机在不影响用户体验(延迟、电量)的前提下进行

的训练,通常发生在开发者更新下发了一个新模型之后,对模型进行迁移学习,解决冷启动问题.由于历史数据通常较多,训练时间较长,无法确保在短时间内(如一天)完成,因此 AutLearn 会存储中间结果,将训练分散在多个时间窗口中.在线学习是基于设备上实时产生的数据进行的训练,主要目的是为了对新的数据做出及时的响应,例如用户行为改变之后,可以立即对模型进行定制化训练,是一个持续强化的过程.虽然在线学习会产生相应的计算开销,但是由于实时产生的数据较为有效,因此训练带来的开销通常不易察觉.在第 4 章中我们着重分析了在输入法应用中在线学习带来的系统资源开销.

### 3.4 模型训练编译器

现有的在移动终端上部署神经网络的方式主要有两种:一种是使用通用的机器学习框架,如 TensorFlow,这些框架实现了各种算子的前向/后向运算,以第三方库的方式加载进内存,然后读入模型与数据,进行运算;另一种方式是近些年兴起的以 TVM[19]为代表的神经网络编译器,这些编译器为特定的模型和硬件平台生成一个可执行文件,可直接在该平台上部署运行.例如,TVM 沿用并拓展了 Halide[44]的 compute-schedule 的概念,通过定义 schedule 来确定代码的优化过程,通过做自动代码生成来实现在各种设备上的部署深度学习模型.相较于通用的机器学习框架,神经网络编译器具有以下几个优点:1)可以根据硬件和算子的描述,自动生成底层的运算代码,具有更强的扩展性;2)可以自动搜索在特定硬件上的实现方式(包括 memory layout,SIMD),以达到最优的运算效率.在自治学习的场景下,引入模型训练编译器可以有效地减少终端设备上的计算开销.

然而,已有的神经网络编译器只服务于模型推断的场景,无法用于模型训练.为此,我们扩展了原有的 TVM 框架,以实现移动终端上的自治学习.

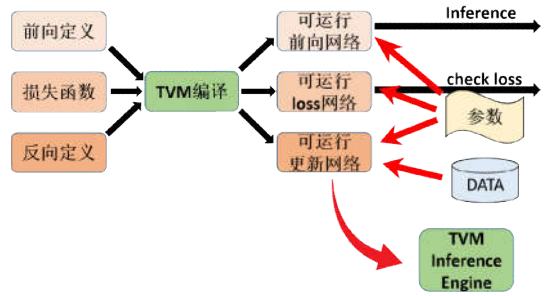


Fig.4 A deep learning compiler for autonomous learning

图 4 面向自治学习的模型编译框架

如图 4 所示,为了编译生成一个可用于训练的可执行文件,我们将用户输入的网络结构和损失函数通过编译得到三个网络:

1) 可运行的前向网络,该网络用于前向计算;
2) 可运行的 loss 网络,该网络与可运行的前向网络的区别在于增加了损失函数,通过这一个网络可以得到相关的训练参数,从而确定需要更新的参数;
3) 可运行的更新网络,该网络是用来更新网络的相关参数,根据输入参数和计算出来的相关梯度,利用 SGD 等算法则可以将参数进行更新,从而完成对计算图的参数更新.

生成以上三个不同功能的网络是为了更好地利用 TVM 模块化特性,减少对 TVM 代码库的修改.

最后,我们补充了 TVM 中缺失的各个算子的后向传播,从而可以对于任意模型结构,编译得到在移动端可以执行的二进制代码,最后在移动设备上运行更新网络即可以实现训练过程.由于模型训练中的反向传播与模型推断在底层实现上基本类似,例如最核心的操作都是围绕矩阵乘法展开,因此在编译出可执行代码的过程中,我们沿用了推断过程中已有 schedule 的设计思路,包括搜索空间与搜索算法.更具体地,我们针对上述第三个可运行的更新网络,增加了针对反向传播算子(如 conv2d_grad、dense_grad 等)的 schedule 的搜索空间,以 TVM 现

有的 template 形式实现;对于搜索算法,我们沿用了 TVM 中基于机器学习的 cost model 方法来预测 schedule 的运行时间,并在实际编译的过程中不断更新这个 cost model,最终使搜索过程快速达到收敛.

此外,针对不同的终端平台,TVM 内部实现了很多与硬件相关的 schedule 原语,我们利用这些原语来设计不同终端平台的代码搜索空间,经过自动调优的不断测试,最终得到针对后端硬件优化后的代码.本文扩展后的 TVM 框架在生成训练代码时,同样需要使用不同终端设备(或不同硬件 SoC)进行各自的调优,调优过程中需要真实设备提供运行时服务,以获取搜索过程中遍历结点的运行效率,作为搜索过程的反馈.对于每一台设备或硬件(本文实验中使用了 2 种设备),都需要进行以上搜索调优过程,时间成本开销较大.在未来工作中,我们将尝试使用更加有效的 cost model[40]来加速编译过程.

### 3.5 训练优化技术

模型压缩是常用的减少神经网络模型复杂度的方式,包括模型剪枝、量化、参数共享、知识蒸馏等具体方案.AutLearn 中采用了基于 magnitude 的参数剪枝方案[30]:每一轮选择一个卷积层(CONV)或者全连接层(FC),然后将 L2-norm magnitude 最小的 K 个卷积核(FC 可以视为特殊的 CONV,同样具有卷积核)去掉,这里 K 的大小取决于最后想要压缩得到模型的复杂度大小.在多次迭代之后,可以获取一个或多个压缩后的模型,适用于不同的终端设备.为了减少开发者负担,无需手动选择哪些网络层进行剪枝,AutLearn 中内置了一个模型自动剪枝模块,即一种特殊形式的神经网络结构自动搜索算法(NAS),可以从一个大模型出发,自动地生成一系列针对不同资源状况的优化后模型.如下公式所示,这里 Accuracy 计算一个模型的准确率,通过在验证集上测得;Resource 计算一个模型的资源消耗情况(例如延迟,能耗,内存占用等),可以通过根据模型结构建模的方式获得[39],也可以通过真机实测获得[16][41],本文为了获取更加真实准确的资源数据采用了后者;Budget 为用户设定的可使用资源上限.这里资源类型可以是模型的计算复杂度(FLOPs)、模型的推断时间、模型的推断能耗、模型的内存占用等,由开发者控制.

$$\underset{Model}{\operatorname{argmax}}\ Accuracy(Model)$$

$$\text{subject to Resource(Model)} < Budget$$

目前,AutLearn 的剪枝是在云端完成的,主要原因是云端具有更多的数据对剪之后的模型进行重新训练,同时减小终端的计算负载.但在终端进行模型剪枝也有一定的优势,主要在于可以直接利用终端的数据进行剪枝训练和测试,得到更加准确的剪枝方案.本文使用基于 magnitude 的剪枝技术的原因是这种技术已经被广泛证明且使用,但其他类型的剪枝技术同样与 AutLearn 系统兼容,例如[38]中使用的根据剪枝操作对 feature map 的影响来判定最优剪枝方案.相较于这些更加复杂的剪枝算法,基于 magnitude 的剪枝方案计算复杂度更低,因此更有利于在终端设备上进行.

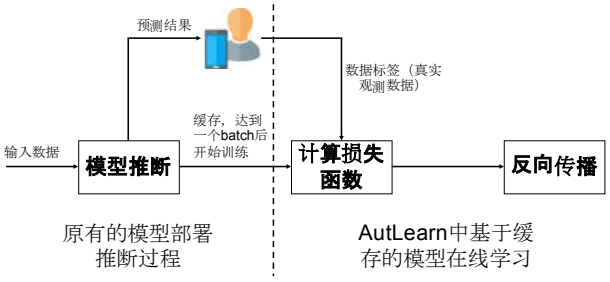


Figure 5 Workflow of inference results reuse in online learning

图 5 在线学习中的推断-训练缓存复用流程图

此外,AutLearn 针对在线学习场景,设计了推断-训练缓存复用技术以减少模型训练的开销.如图 5 所示,在线学习中,模型会在实时的输入数据上做推断运算,根据用户的反馈生成标签,与原数据组成训练数据,用于训练模型.推断与训练流水线进行.其中,神经网络训练的本质其实就是在推断结果的基础上计算损失函数,然后做后向传播.因此,AutLearn 会将预测的结果进行复用,直接在其基础上根据用户的反馈行为(真实标签)进行模型训练.具体地,需要在模型推断中进行缓存的数据为推断预测结果(用于生成 loss)以及产生的中间结果(即特征图 feature map,用于反向传播过程中的参数更新).由于自治学习中使用迁移学习技术,只需要更新模型尾部少部分参数,假设需要更新的网络层是从第 K 层起至最后一层,则只需要保存第 K-1 层至最后一层中产生的中间向量,极大地减小了运行时缓存的内存开销.为了进一步减小单个数据的性能开销,AutLearn 会等待多个数据产生达到一个 batch 后一起训练(默认的 batch size 是 16,可由开发者调节设置).

### 3.6 应用实例

我们基于 AutLearn 构建了两个应用实例验证其功能及效果.

#### 3.6.1 输入法中的词预测

输入法是移动设备上最为重要的应用之一.输入法应用的主要功能之一是输入词预测:以英文输入法为例,基于用户已有的输入序列(包括单词和字母),预测用户想要的输入词.词预测功能在大部分主流输入法应用如 Gboard 中都基于神经网络实现,是自然语言处理任务中最常见的任务之一.当前处理该任务的主流算法之一是循环神经网络(RNN).我们以 RNN 的常见变体 LSTM 网络结构为例,实现了一个基本的词预测神经网络模型,如图 6 所示.

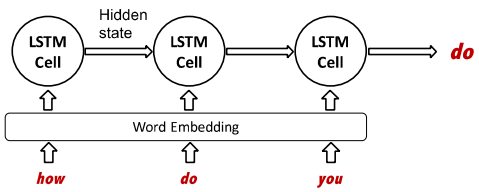


Fig.6 The LSTM model structure implemented in this work

图 6 本文实现的用于词预测功能的 LSTM 模型

输入法中的词预测功能契合了 AutLearn 的应用场景:1)数据隐私:用户输入数据包含聊天记录、账号密码等,不应上传至中心化的服务器;2)本地产生训练数据,用户的点击输入即为数据进行了标注,可直接用于训练;3)用户输入行为具有个性化,需要对模型进行定制以达到最佳的效果;4)本地通常存有历史输入数据,可用于离线的模型迁移学习;5)用户的输入行为不是一成不变的,可能会随着时间而改变,需要持续不断地进行学习更新(在线学习).

#### 3.6.2 图像分类

图片处理是神经网络最常见的应用场景之一.例如,在 iPhone 手机自带的相册应用中,已有卷积神经网络用于物体、场景、人脸的识别,帮助用户整理图片,便于搜索查找.与输入法中的文字输入数据类似,用户拍摄的图片包含大量的隐私信息,不应上传至服务器进行训练.

我们基于 AutLearn 构建了经典的卷积神经网络 MobileNet,以实现手机终端上的图片分类.MobileNet 是专门为移动设备设计的网络结构,其内部利用了 depthwise separable convolutions 结构,减少了卷积层的计算量.

## 4. 实验

本章介绍我们对 AutLearn 的实验设计与实验结论.实验的目的主要分为两个方面:1)验证 AutLearn 可以通过在移动终端上的迁移学习获得较高的神经网络模型准确率;2)常用的移动终端平台上的硬件资源足以支撑神经网络模型的训练,尤其是上文中提到的一系列优化技术对减小神经网络模型训练开销的效果.

### 4.1 实验环境与设计

**模型与数据** 针对上文提到的两种应用场景,我们分别使用了不同的公共数据集和模型进行测试.

- 输入法词预测应用中,我们使用 LSTM[22]模型,以 Twitter 数据作为预训练数据(公共数据集),Shakespeare 数据集作为各个终端设备上的本地数据集.两份数据都来自于 LEAF[17]联邦学习基准测试中的标准数据集,都预先切分好了不同的用户,不同用户的数据不满足独立同分布(non-iid).两个数据集都用于输入词预测训练.终端迁移学习中只有最后一层全连接层参与模型更新.本文所使用的 LSTM 模型具体参数为:10,000 个使用频率最高的词组成的词汇表大小,Step Size 为通过 bucketing 技术[23]实现的可变长大小,2 层 LSTM Cell 堆叠 (2-layer stacked).
- 在图像分类应用中,我们使用 MobileNet[35]模型,以 ImageNet 数据集作为预训练数据,FEMINST 数据集作为各个终端设备上的本地数据集.其中 FEMINST 同样来自原 LEAF 的标准数据集,数据为 non-iid.ImageNet 数据集用于训练 MobileNet 的 1000 类物体识别,FEMINST 用于迁移学习至手写体识别任务,迁移学习中只有最后一层全连接层与 Softmax 层会参与模型更新.两个数据集中,我们都只挑选了数据量最大的 10%用户用于训练与测试,因为现实中用户的历史数据积累往往大于这些标准数据集的数据量.

**基准线** 我们比较 AutLearn 与以下方案:1)联邦学习是近些年兴起的在分布式数据上进行模型训练的技术(具体见第 1 章),这里我们使用最常见的联邦学习算法 *FedAvg*[3].与自治学习技术相比,联邦学习没有对不同用户的模型进行定制化,且需要大量的通讯开销;2)中心化的云端训练技术(CloudTrain)[43]将用户的数据都收集到各个终端上,然后利用这些数据训练一个通用的模型.同样的,CloudTrain 的方法也没有对模型进行定制化训练.这种集中式的训练模式也是现在工业界通用的部署方式.; 3)中心化的定制训练技术(CloudCustomize)[42],是指在 CloudTrain 获得的模型的基础上,对新模型在不同设备上的数据进行定制化训练后的结果.从模型准确率的角度考虑,CloudCustomize 可以被认为是最优的情形,因为它同时使用了用户隐私数据进行预训练和迁移学习.但同时,CloudCustomize 的方式在用户隐私和可扩展性上都有较大的缺陷.AutLearn 和所有基准线都通过 mini-batch 的方式进行模型训练[25].

**终端设备** 我们使用 Samsung Note 8 和 Redmi Note 8 两种设备型号测试 AutLearn 在终端设备上的训练效率,包括训练时间与能耗.

### 4.2 模型准确率

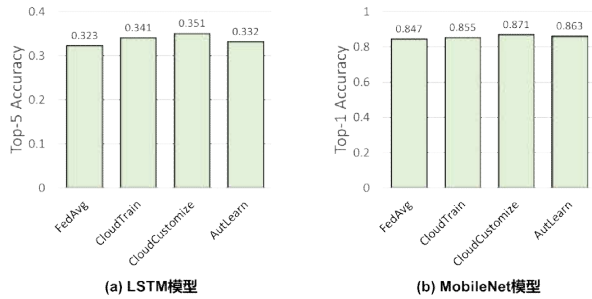


Fig.7 The accuracy of AutLearn compared with baselines

图 7 AutLearn 训练模型准确率与基准线对比

图 7 总结了离线学习中,AutLearn 在不同模型下和各个基准线的准确率对比.实验结果证明,AutLearn 和各个基准线都可以训练得到收敛的模型.与 CloudTrain 这种传统的训练模式比,FedAvg 联邦学习训练模式得到的模型有少量的精度损失(约 1%),而 AutLearn 的精度则与 CloudTrain 模式基本持平甚至更高:在 LSTM 模型上,AutLearn 相较于 CloudTrain 有 0.9%精度下降,而在 MobileNet 模型上则有 0.8%的模型精度提升.原因是

AutLearn 采用自治式学习的方式,在终端迁移学习过程中会对每个用户的模型使用该特定用户产生的数据进行训练,达到了个性化学习的效果.此外,CloudCustomize 的方式在两种模型上都得到最高的准确率,原因是它在 CloudTrain 集中式训练的基础上,再进行个性化训练,且使用相同的用户隐私数据集；而 AutLearn 使用分布不同的数据集首先进行云端预训练得到一个全局模型,然后再利用与测试数据集同源的数据集进行迁移学习.由于预训练与迁移学习中使用的数据集不同,难免导致预训练过程中模型学到的特征提取无法完全适用于用户隐私数据,因此效果不如 CloudCustomize 模式.需要强调的是:四种训练模式中,只有 FedAvg 和 AutLearn 考虑了用户隐私的保护.在这一系列实验中,我们发现模型定制化的效果有限,原因是数据集限制了每个设备上的数据量.

我们同时验证了数据增强技术对模型准确率的影响.对于 LSTM 模型,我们使用了同义词替换和的技术；对于 MobileNet 模型,我们使用了裁剪、旋转和移位技术.需要注意的是,两种技术都可以将训练数据量提升最高超过 5 倍,为了平衡训练的计算开销,本文的实验中都只在生成数据中采样不超过原数据量的 1 倍.实验证明数据增强对词预测和图像分类两种任务分别有 0.9%和 3.1%的准确率提升.后者的效果更明显:事实上自然语言处理中的数据增强比起图片数据而言确实更加困难.

此外,我们还通过实验探究了在线学习对模型准确率的提升效果.我们从预训练的模型出发,然后将数据集序列化,不断地输入到 AutLearn 系统中.对于每一条数据,AutLearn 会首先做出预测,然后利用该数据进行模型的强化训练,不断重复该过程,最后汇总预测的整体结果.这个过程即是模拟用户在使用该模型的过程中 AutLearn 对用户当前行为(产生数据)做出的适应性改变.我们发现,对于 LSTM 和 MobileNet 模型而言,相比只用预训练模型预测所有数据,在线学习可以在所有数据上平均提升 5.9%和 3.1%的模型准确率.并且该提升在后面输入的数据上更加明显,原因是随着更多数据的输入,模型得到了持续的学习,精度上升.实验结果证明了在线学习可以有效地根据用户行为改变(如输入模式)适应性地对模型进行调整,以达到更高的准确率.

### 4.3 终端训练开销

我们测试了在两种终端设备上进行迁移学习的性能开销(离线学习),结果如图 8 所示.其中 AutLearn 代表离线学习的训练速度,w/o compression 和 w/o compiler 分别代表去除前文提到的模型压缩与编译器加速技术后的性能,w/o cache 代表利用推断-训练缓存加速后的在线训练时间.这里,去掉编译器优化即使用统一的深度学习计算库进行模型训练(TensorFlow 库).其中,LSTM 模型我们使用 batch 大小为 16 进行训练,而 MobileNet 由于内存占用较大,在实验终端上只能使用 batch 大小为 1 进行训练.

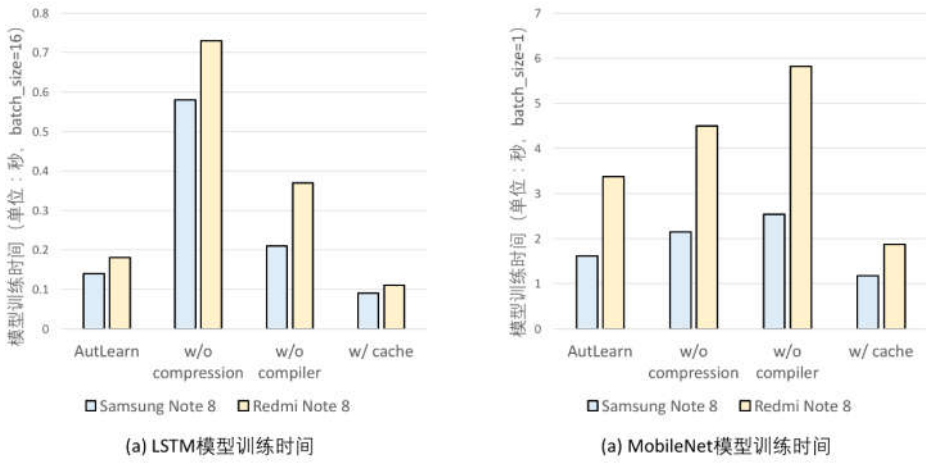


Figure 8 The speedup of AutLearn on different device models (including online/offline learning)

图 8 AutLearn 在不同模型和机型上的性能(包括离线学习与在线学习)

实验结果显示:1)模型压缩技术可以有效地减少训练时间,但在不同模型上的表现差别较大.这里,我们默认压缩模型至 1%精度损失.在 LSTM 模型上,模型压缩技术可以减少超过 80%的训练时间,但是在 MobileNet 上,只能减少 20%-30%.主要原因在于 MobileNet 本身就是专门为低计算能力设备设计的极为精简的模型,在其上做压缩更难.2)编译器技术同样可以有效地减少训练时间,在 LSTM 和 MobileNet 模型上都达到了将近 40%的节省.相较模型压缩,编译器技术基本不依赖于模型本身的结构,且又不会造成模型精度的损失.3)在线学习中使用推断-训练缓存机制可以进一步减小模型的训练时间(30%-40%),原因是节省了模型训练过程中的前向推断过程,只需要进行反向传播.需要注意的是,缓存机制只对在线学习有效,因为离线学习中,每一个 batch 训练之后会更新参数,导致原来缓存的预测结果在新模型上失效,无法复用.4)在不同终端设备上的训练时间有最高将近 2 倍的差异,原因是不同终端设备上计算资源的差异.但是 AutLearn 的优化技术在不同的设备上都有相应的性能提升,具有一定的普适性.

结合在第 2 章中提到,我们分析真实用户的行为数据发现,超过半数的终端每天都有约 120 分钟处于可训练阶段.这意味着 LSTM 模型和 MobileNet 模型都可以在一天内完成多轮迁移学习的训练,即新模型下发后的第二天就可以用于部署,并达到定制化后的高准确率.

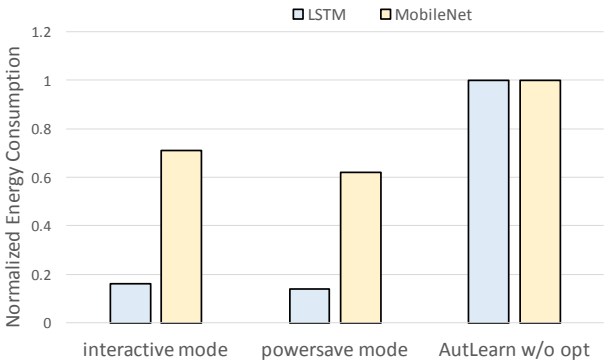


Figure 9 The energy consumption of AutLearn under different CPU status

图 9 AutLearn 在不同 CPU 状态下的能耗

我们还度量了 AutLearn 在终端设备上的能耗状况,这里我们以 Nexus6 型号为例.如图 9 所示,“interactive”和“powersave”代表不同的 CPU 状态调节器,其中前者会使 CPU 运行在较高频率上(Android 设备的默认调节器),而后者则会让 CPU 一直运行于最低频率以节省能耗.所有结果都归一化到同样的 baseline:即不使用 AutLearn 的优化加速技术且让 CPU 处在 interactive 调节器下.实验结果表明:1)AutLearn 的优化技术可以极大地减少终端模型训练的能耗,最多超过 80%(LSTM 模型),原因主要在于节省了模型训练的端到端时间;2)在 CPU 低频率运行状态下,虽然模型的训练需要花费更多的时间,但是整体的能耗却有所下降,原因是在低频率运行状态下单位时间内 CPU 能耗更低.这意味着在一些不需要迅速完成模型训练进行部署的场景下,可以通过将 CPU 频率调低来节省设备能耗.

## 5. 结束语

针对面向移动终端的数据隐私问题,本文提出了自治式的机器学习模式.区别于以往传统的集中式和联邦式,自治式学习将隐私数据相关的计算全都部署在本地,极大程度上提高了用户隐私保护能力,同时提供了模型定制化效果.为了解决自治学习中终端设备上数据量不足以及计算能力不足的两大挑战,我们设计实现了 AutLearn 自治学习框架,其中包含云端协同训练,本地数据增强,模型压缩以及缓存复用等技术.我们以经典的自然语言处理和图像识别任务为例,在真实数据集上验证了自治式学习的效果:AutLearn 相比传统的训练模式,可以达到相当甚至更高的准确率,同时其计算开销在普通智能手机可承受范围内.

**References**:


[1] Xu M, Liu J, Liu Y, Lin FX, Liu Y, Liu X. A first look at deep learning apps on smartphones. In The World Wide Web Conference 2019 May 13 (pp. 2125-2136).

[2] General Data Protection Regulation (GDPR), 2018. https://gdpr-info.eu/.

[3] McMahan HB, Moore E, Ramage D, Hampson S. Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629. 2016 Feb 17.

[4] Konečný J, McMahan HB, Ramage D, Richtárik P. Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527. 2016 Oct 8.

[5] Li L, Xiong H, Wang J, Xu CZ, Guo Z. SmartPC: Hierarchical Pace Control in Real-Time Federated Learning System. IEEE Real-Time Systems Symposium (RTSS 2019).

[6] Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K. Practical secure aggregation for privacy-preserving machine learning. InProceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security 2017 Oct 30 (pp. 1175-1191).

[7] Bost R, Popa RA, Tu S, Goldwasser S. Machine learning classification over encrypted data. InNDSS 2015 Feb 8 (Vol. 4324, p. 4325).

[8] Graepel T, Lauter K, Naehrig M. ML confidential: Machine learning on encrypted data. InInternational Conference on Information Security and Cryptology 2012 Nov 28 (pp. 1-21). Springer, Berlin, Heidelberg.

[9] Nandakumar K, Ratha N, Pankanti S, Halevi S. Towards Deep Neural Network Training on Encrypted Data. InProceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops 2019.

[10] Badawi AA, Chao J, Lin J, Mun CF, Sim JJ, Tan BH, Nan X, Aung KM, Chandrasekhar VR. The alexnet moment for homomorphic encryption: Hcnn, the first homomorphic cnn on encrypted data with gpus. arXiv preprint arXiv:1811.00778. 2018 Nov 2.

[11] Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L. Deep learning with differential privacy. InProceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security 2016 Oct 24 (pp. 308-318).

[12] Shokri R, Shmatikov V. Privacy-preserving deep learning. InProceedings of the 22nd ACM SIGSAC conference on computer and communications security 2015 Oct 12 (pp. 1310-1321).

[13] Bonawitz K, Eichner H, Grieskamp W, Huba D, Ingerman A, Ivanov V, Kiddon C, Konecny J, Mazzocchi S, McMahan HB, Van Overveldt T. Towards federated learning at scale: System design. SysML 2019.

[14] Denton EL, Zaremba W, Bruna J, LeCun Y, Fergus R. Exploiting linear structure within convolutional networks for efficient evaluation. InAdvances in neural information processing systems 2014 (pp. 1269-1277).

[15] Wei JW, Zou K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. EMNLP/IJCNLP (1) 2019: 6381-6387.

[16] Yang TJ, Howard A, Chen B, Zhang X, Go A, Sandler M, Sze V, Adam H. Netadapt: Platform-aware neural network adaptation for mobile applications. InProceedings of the European Conference on Computer Vision (ECCV) 2018 (pp. 285-300).

[17] Caldas S, Wu P, Li T, Konečný J, McMahan HB, Smith V, Talwalkar A. Leaf: A benchmark for federated settings. arXiv preprint arXiv:1812.01097. 2018 Dec 3.

[18] Zhuang F, Luo P, He Q, Shi Z. Survey on Transfer Learning Research. Journal of Software, 2015, 26(1): 26-39 (in Chinese).

[19] Chen T, Moreau T, Jiang Z, Zheng L, Yan E, Shen H, Cowan M, Wang L, Hu Y, Ceze L, Guestrin C. {TVM}: An automated end-to-end optimizing compiler for deep learning. In13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18) 2018 (pp. 578-594).

[20] Chellappa RK, Sin RG. Personalization versus privacy: An empirical examination of the online consumer's dilemma. Information technology and management. 2005 Apr 1;6(2-3):181-202.

[21] Ekberg JE, Kostiainen K, Asokan N. Trusted execution environments on mobile devices. InProceedings of the 2013 ACM SIGSAC conference on Computer & communications security 2013 Nov 4 (pp. 1497-1498).

[22] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov 15;9(8):1735-80.

[23] Khomenko V, Shyshkov O, Radyvonenko O, Bokhan K. Accelerating recurrent neural network training using sequence bucketing and multi-gpu data parallelization. In2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP) 2016 Aug 23 (pp. 100-103). IEEE.

[24] Lane ND, Bhattacharya S, Georgiev P, Forlivesi C, Jiao L, Qendro L, Kawsar F. Deepx: A software accelerator for low-power deep learning inference on mobile devices. In2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) 2016 Apr 11 (pp. 1-12). IEEE.

[25] Li M, Zhang T, Chen Y, Smola AJ. Efficient mini-batch training for stochastic optimization. InProceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining 2014 Aug 24 (pp. 661-670).

[26] Volokh E. Personalization and privacy. Communications of the ACM. 2000 Aug 1;43(8):84-88.

[27] Santos N, Raj H, Saroiu S, Wolman A. Using ARM TrustZone to build a trusted language runtime for mobile applications. In Proceedings of the 19th international conference on Architectural support for programming languages and operating systems 2014 Feb 24 (pp. 67-80).

[28] Vanhaesebrouck P, Bellet A, Tommasi M. Decentralized collaborative learning of personalized models over networks. In International Conference on Artificial Intelligence and Statistics (AISTATS'17), 2017.

[29] Xia Y, Liu Y, Tan C, Ma M, Guan H, Zang B, Chen H. TinMan: eliminating confidential mobile data exposure with security oriented offloading. In Proceedings of the Tenth European Conference on Computer Systems 2015 Apr 17 (pp. 1-16).

[30] Xue J, Li J, Gong Y. Restructuring of deep neural network acoustic models with singular value decomposition. InInterspeech 2013 Aug 25 (pp. 2365-2369).

[31] Lei J, Gao X, Song J, Wang XL, Song ML. Survey of Deep Neural Network Model Compression[J]. Journal of Software, 2018, 29(2): 251-266 (in Chinese).

[32] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? InAdvances in neural information processing systems 2014 (pp. 3320-3328).

[33] Bagdasaryan E, Veit A, Hua Y, Estrin D, Shmatikov V. How to backdoor federated learning. arXiv preprint arXiv:1807.00459. 2018 Jul 2.

[34] Smith V, Chiang CK, Sanjabi M, Talwalkar AS. Federated multi-task learning. InAdvances in Neural Information Processing Systems 2017 (pp. 4424-4434).

[35] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. 2017 Apr 17.

[36] Xiong P, Zhu T, Wang X. A Survey on Differential Privacy and Applications. Chinese Journal of Computers 2014;37(1):101-22.

[37] Xu M, Qian F, Mei Q, Huang K, Liu X. Deeptype: On-device deep learning for input personalization service with minimal privacy concern. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. 2018 Dec 27;2(4):1-26.

[38] Yang, T.-J., Chen, Y.-H., and Sze, V. Designing energy-efficient convolutional neural networks using energy-aware pruning. In Proceedings of the IEEE Confer- ence on Computer Vision and Pattern Recognition (2017), pp. 5687–5695.

[39] Xu, M., Qian, F., Zhu, M., Huang, F., Pushp, S. and Liu, X., 2019. Deepwear: Adaptive local offloading for on-wearable deep learning. IEEE Transactions on Mobile Computing, 19(2), pp.314-330.

[40] Zerrell, T. and Bruestle, J., 2019. Stripe: Tensor compilation via the nested polyhedral model. arXiv preprint arXiv:1903.06498.

[41] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A. and Le, Q.V., 2019. Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2820-2828).

[42] Taylor, S.A., Jaques, N., Nosakhare, E., Sano, A. and Picard, R., 2017. Personalized multitask learning for predicting tomorrow's mood, stress, and health. IEEE Transactions on Affective Computing.

[43] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[44] Ragan-Kelley, J., Barnes, C., Adams, A., Paris, S., Durand, F. and Amarasinghe, S., 2013. Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. Acm Sigplan Notices, 48(6), pp.519-530.

[45] Vaidya, J., Kantarcıoğlu, M. and Clifton, C., 2008. Privacy-preserving naive bayes classification. The VLDB Journal, 17(4), pp.879-898.

[46] Xu, M. , Zhu, M. , Liu, Y. , Lin, F. X. , & Liu, X. . (2018). DeepCache: Principled Cache for Mobile Deep Vision. the 24th Annual International Conference (MobiCom), pp. 129-144.

附中文参考文献:

[18] 庄福振,罗平,何清,史忠植.迁移学习研究进展.软件学报,2015,26(1):26-39.http://www.jos.org.cn/1000-9825/4631.html

[31] 雷杰,高鑫,宋杰,王兴路,宋明黎.深度网络模型压缩综述.软件学报,2018,29(2):251-266. http://www.jos.org.cn/1000-9825/5428.htm

[36] 熊平,朱天清,王晓峰.差分隐私保护及其应用.计算机学报,2014,37(1):101-22.

徐梦炜(1992－),男,博士生,主要研究领域为系统软件,移动/边缘计算.

刘渊强(1997－),男,硕士生,主要研究领域为系统软件,神经网络编译器.

黄康(1984－),男,博士,主要研究领域为机器学习,自然语言处理.

刘譞哲(1980－),男,博士,副教授,博士生导师,CCF 专业会员,主要研究领域为服务计算,Web 技术,软件工程.

黄罡(1975－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为软件中间件,软件体系结构,网构软件.