中国科学: 信息科学 2020 年 第 50 卷 第 11 期: 1629–1644
SCIENTIA SINICA Informationis
可成长软件专题 · 论文

深度学习模型终端环境自适应方法研究

郭斌*, 仵允港, 王虹力, 王豪, 刘思聪, 刘佳琪, 於志文, 周兴社

西北工业大学计算机学院, 西安 710072
* 通信作者. E-mail: guob@nwpu.edu.cn

收稿日期: 2020–03–21; 修回日期: 2020–07–07; 接受日期: 2020–08–31; 网络出版日期: 2020–11–09

国家重点研发计划 (批准号: 2017YFB1001800) 和国家自然科学基金 (批准号: 61772428, 61725205) 资助项目

摘要 随着人工智能和物联网的快速发展与融合, 智能物联网 AIoT 正成长为一个极具前景的新兴前沿领域, 其中深度学习模型的终端运行是其主要特征之一. 针对智能物联网应用场景动态多样, 以及物联网终端 (智能手机、可穿戴及其他嵌入式设备等) 计算和存储资源受限等问题, 深度学习模型环境自适应正成为一种新的模型演化方式. 其旨在确保适当性能的条件下, 能自适应地根据环境变化动态调整模型, 从而降低资源消耗、提高运算效率. 具体来说, 它需要主动感知环境、任务性能需求和平台资源约束等动态需求, 进而通过终端模型的自适应压缩、云边端模型分割、领域自适应等方法, 实现深度学习模型对终端环境的动态自适应和持续演化. 本文围绕深度学习模型自适应问题, 从其概念、系统架构、研究挑战与关键技术等不同方面进行阐述和讨论, 并介绍我们在这方面的研究实践.

关键词 智能物联, 环境自适应, 模型演化, 深度模型压缩, 云边端模型分割, 领域自适应

1 引言

随着物联网和人工智能技术的快速发展与加速融合, 智能物联网 (AI in IoT, AIoT) [1] 正成长为一个具有广泛发展前景的新兴前沿领域. 近年来, 基于深度计算模型的物联网智能应用和服务已经逐步融入国家重大需求和民生的各个领域, 例如智慧城市、智能制造、无人驾驶、健康卫生等. 面向这些智能物联场景, 将深度学习模型 (如实时视频数据处理) 离线部署在资源受限且环境多变的物联网终端设备执行逐渐成为一种趋势, 其具有低计算延时、低传输成本、保护数据隐私等优势 [2]. 然而, 在资源受限的移动端运行深度学习模型仍面临着两项关键挑战, 制约了其落地和大规模的运用.

- **硬件资源限制.** 深度学习模型通常是计算密集型的大规模网络, 往往需要较高的存储、计算和能量资源, 而终端设备的有限硬件资源成为了在终端部署和运行深度模型的技术瓶颈.

引用格式: 郭斌, 仵允港, 王虹力, 等. 深度学习模型终端环境自适应方法研究. 中国科学: 信息科学, 2020, 50: 1629–1644, doi: 10.1360/SSI-2020-0067
Guo B, Wu Y G, Wang H L, et al. Context-aware adaptation of deep learning models for IoT devices (in Chinese). Sci Sin Inform, 2020, 50: 1629–1644, doi: 10.1360/SSI-2020-0067

• **应用环境复杂.** 物联终端计算具有环境动态变化 (如电源消耗、存储资源变化等)、应用场景多样、数据分布不同等特点. 而深度学习模型的训练过程是基于特定数据集的知识学习过程, 对终端复杂应用场景的适应能力差.

为了解决上述挑战, 部分研究者已经以不同的方式作出模型自适应压缩和加速的初步尝试. 例如 AdaDeep [3] 提出了压缩技术组合的方法来解决对深度学习模型的资源消耗自适应的问题; 哈佛大学的 Teerapittayanon 等 [4] 通过选择网络的不同 "提前退出" 分支实现深度模型的运行时加速研究; 浙江大学提出的 DeepThing [5] 方法将深度学习模型分割为多个分块然后分配给不同的移动边缘终端, 在整体资源消耗和终端运行性能上实现自适应的最优折中. 然而上述工作对于如何使模型根据动态环境变化实现自适应演化还没有做深入的探索和研究.

鉴于此, 本文提出面向物联网终端智能发展需要研究一种新的自适应演化方式, 即 "深度学习模型的终端环境自适应演化": 深度学习模型需要自适应于不同的物联网终端 "环境", 包括运行环境、部署环境, 以及可以直接或间接影响其运行演化方式的各种因素等. 具体来说, "环境" 包含但不限于感知数据、应用领域、应用任务的性能需求、目标部署平台的资源约束, 以及所处系统的软硬件上下文环境; "自适应能力" 则是主动捕获环境变化能力和自动化决策能力的融合; "演化" 旨在为学习模型针对不断新增的数据、新增用户的个性特征、跨领域/跨实体间模型的知识迁移 [6,7] 等需求提供持续性的学习和更新方案, 即终身学习 (lifelong learning) 能力 [8~10]. 综上, 深度学习模型的 "环境自适应" 旨在主动感知所处环境状态、实时捕获环境变化, 并对深度学习模型的运算和结构进行自动化调优以及持续性演化.

相比深度学习模型的环境自适应而言, 传统的深度学习模型关注集中式训练高性能模型 [11,12] 或通过一些模型压缩方法减少资源消耗 [13], 主要采用固定训练方式而较少考虑模型的自适应能力和持续性演化问题, 对智能物联网应用系统的长期生存和成长是不利的. 在智能物联环境下, 深度学习模型的环境自适应需要主动感知终端环境变化、自动调整运算模式和模型结构, 因此需要探索新的计算范式. 智能物联终端环境自适应演化的研究内涵非常丰富, 本文将主要从两个方面阐述和讨论深度学习模型的环境自适应: (1) 面向硬件资源的模型自适应; (2) 跨领域、跨实体模型自适应.

第 2 节先介绍包括深度模型压缩、模型分割技术在内的面向硬件资源的自适应模型及其特点, 以域自适应方法为代表的迁移演化模型, 以及深度学习模型终端环境自适应的典型架构. 第 3 节阐述深度学习模型的终端环境自适应面临的挑战及关键技术发展. 第 4 节介绍我们针对这一领域的最新研究成果. 最后对全文进行总结和展望.

2 深度学习模型环境自适应: 概念与系统架构

本节首先从面向硬件资源和面向域变化两方面, 介绍深度学习模型的环境自适应模式和相关探索. 接着对深度学习模型的环境自适应问题进行定义, 并给出具有推广性的深度学习模型终端环境自适应系统架构.

2.1 深度学习模型环境自适应模式及特点

深度学习模型对于环境的自适应包含两个方面: (1) 硬件资源状态的自适应; (2) 领域自适应. 硬件资源状态的自适应旨在满足目标任务性能约束的前提下, 根据硬件资源 (如存储、计算算力、电量) 约束, 自适应调整深度模型的架构、参数及运算方式. 领域自适应模型演化, 则是在满足硬件资源约束的前提下, 根据目标任务的需求变化和应用场景变化, 实现模型的重训练和自适应演化. 硬件自适应

与领域自适应的输出会相互影响彼此的约束条件. 总的来说二者相互影响, 相互辅助.

2.1.1 面向硬件资源状态的自适应

深度学习模型对硬件资源约束的自适应是指深度学习模型应该根据目标平台上硬件资源的变化, 自适应调整其模型参数以适应新的需求. 具体来说, 硬件资源状态的自适应包含两个层面的内涵: (1) 在满足目标性能需求的前提下, 自适应于不同的硬件资源 (如存储、算力、电量) 约束. (2) 根据新的硬件资源变化在线自适应. 对于如何自适应于不同的硬件资源, 研究人员已经进行了一些研究, 例如, AdaDeep [3] 根据计算任务需求和平台的资源约束自动选择并组合不同的压缩技术对深度学习模型进行简化, 从而实现自适应的轻量级网络架构搜索. AMC [14] 则是以任务的需求作为约束条件, 自动地压缩深度学习的每一层. 当压缩率过高不满足任务性能需求时会自动调整其压缩率, 实现了满足任务性能的硬件资源需求自适应. 此外, 如何根据新的硬件资源变化进行在线自适应, 也有了相关的研究工作. 例如, Once-For-All [15] 通过一次训练一个超级网络, 在针对特定任务时从该超网中搜索对应的网络结构而不需要对网络重训练. 在搜索算法层面他们通过建立结构与精度的匹配对实现了快速适应的需求. 这种方式可以根据环境的动态变化快速找到最佳的网络结构, 以实现在线自适应. 总的来说, 为使深度学习模型能够长期生存, 应该根据硬件资源的变化调整模型的计算单元、组成结构和运行设置等参数. 目前, 研究者们从模型压缩与模型分割两个主流方向探索如何改变模型参数.

- **终端上的深度模型压缩.** 模型压缩技术可以简化深度学习模型结构, 降低模型参数量与运算量, 从而可以将其部署在资源受限的移动设备上. 目前已经有很多模型压缩的方法被提出, 例如, 结构化剪枝技术, 旨在调整模型的计算单元, 删除模型中冗余的通道和过滤器; 全连接层替换技术, 旨在调整模型的组成结构; 早退出技术, 旨在运行时动态调整模型的相关设置. 但是这些纯软件层面压缩技术的硬件友好性受限, 因此制约了其性能. 最近的一些研究工作探索了软硬件协同优化技术, 并取得了非常好的结果. Chen 等 [16] 提出模型张量化的概念, 将计算流图的每个节点定义为一个张量表达式, 并利用机器学习找到张量表达式到底层程序的最佳映射. Ma 等 [17] 提出了一种新的剪枝模式, 并针对这种具有视觉特性的卷积内核开发了一种新颖的编译器来辅助 DNN 进行推理, 实现了实时执行 PCONV 模型而不会影响精度的效果. 但是这些方法的目标是优化模型执行时间, 并不关注深度学习模型自适应.

- **云边端融合的深度模型分割.** 模型分割技术是将完整的深度学习模型进行分块, 并根据性能需求 (如时延、精度) 和资源消耗 (如网络传输、终端设备上存储和能耗等) 自动寻找最佳分割点, 将模型中不同的层部署到云、边、端的不同设备上. 模型分割根据整体或终端的关注点倾向, 通常有两种方式: (1) 降低整体模型的资源消耗. 因为深度网络某些中间层间的传输数据量要远小于原始数据量, 因此选取合适的模型分割点能够降低数据传输量, 并且减少整个模型的全局资源消耗. (2) 降低模型在单台设备上的资源消耗. 深度学习模型在分割之后, 每块网络对硬件资源的需求将大幅度减少, 可以在资源受限的硬件设备上运行. 目前模型分割主要集中在 "端云分割", 即将深度学习模型在某一点切分后, 一部分部署在终端设备上, 一部分部署在云端, 二者共同完成推断任务. 研究的热点集中在选取最佳的模型分割点上, 使得模型在分割之后依然能够保持良好的推断性能. 例如, Neurosurgeon 框架 [18] 提出训练回归模型以估算深度学习模型网络层的执行时延以及能耗, 继而选择其中最佳的模型分割点使得深度学习模型能够满足时延需求或能耗需求. Edgent 框架 [19] 提出了基于边缘与终端协同的深度学习模型优化框架, 联合优化模型分割和模型压缩两种策略. 然而针对深度模型的训练时和运行时自适应分割, 以及针对多个终端的自适应分割和协作方面的研究还有待深入探索.

2.1.2 域自适应模型演化

基于深度学习的物联终端智能应用的应用场景和任务需求总在不断变化. 例如, 在智能制造场景中, 面向产品质量检测的设备端视频识别算法会受到场景布置、拍摄角度、光线变化等外部因素的影响; 在柔性制造背景下, 感知的类别和对象也在不断变化, 如新的产品或零件加入后会由于缺少标注和训练数据而导致既有模型无法发挥好的效用. 此外, 一般来说深度学习模型的鲁棒性较差, 传统的解决办法是在训练阶段加入适量的噪声, 以提高模型的鲁棒性. 但是这一方法在训练完成后仍不能抵御新噪声. 因此, 利用域自适应的方法训练模型来抵御这种环境或需求变化正在成为智能物联领域的新发展方向.

域自适应技术旨在寻找一个空间映射, 将源域和目标域映射到同一特征子空间中, 使得源域和目标域的分布差距最小. 基于这一映射, 还可以利用两个域的数据学习分类器或者预测器. 域自适应是迁移学习领域的一个重要分支, 特别是随着生成式对抗网络、元学习等新兴学习算法的引入, 使用深度学习模型找到最佳映射的方法已被广泛研究使用. DANN [20] 框架提出对抗的域自适应方法, 利用对抗思想训练一个更好的特征提取器, 使得经过此特征提取器的源域和目标域的数据分布差异最小化. TADA [21] 框架是 DANN 的延续, 进一步提出更细粒度的局部特征选择方法实现域自适应. Nagabandi 等 [22] 则提出借助元学习解决在线快速适应新任务问题.

2.1.3 硬件自适应与域自适应的协作

硬件自适应与域自适应是深度学习模型应对环境变化进行自适应演化的两种方式, 二者相互影响, 相互辅助. 在实际应用中, 新设备的加入或离开都会对整体的可用计算资源造成影响. 对于这些硬件资源的变化需要模型自动做出适应性的改变, 也即硬件自适应. 但是新设备的加入往往会带来新的数据, 甚至改变原有任务. 因此为保持深度学习模型的性能, 在硬件自适应过程中也需要域自适应根据新数据对深度学习模型进行重训练. 反之进行域自适应时对硬件的资源需求也有变化, 这需要硬件自适应进行协调. 物联终端智能应用的场景变化和任务需求变化导致深度学习模型需要进行域自适应. 但是域自适应的过程即是对模型进行重训练的过程, 这需要硬件设备提供比推断阶段更为强劲的算力及相关硬件资源, 也就意味着在域自适应阶段需要消耗更多的电量、内存, 以及 CPU 占用率. 因此在硬件资源约束下, 需要硬件自适应协调硬件资源的调配, 以顺利完成域自适应过程. 因此, 硬件自适应与域自适应是相互关联, 相互影响的.

2.2 深度学习模型环境自适应的系统架构

图 1 给出一个参考性的深度学习模型终端环境自适应系统架构 A^2IoT (adaptive AIoT). 其中, 环境感知层主动捕获环境状态, 并将环境状态描述输入给硬件自适应层和域自适应层. 硬件自适应层和域自适应层按需调用, 分别提供面向硬件和域动态变化的自适应压缩、分割和可持续性演化方案. 自适应方案将作为应用层的支撑, 实现用户定义的智能物联应用和服务. 各层的功能和协作详述如下.

- **环境感知层.** 目标应用设备一方面感知自身的硬件资源状态 (如摄像头、处理器、剩余电量、存储资源等), 另一方面要感知周围环境的上下文变化状态 (如光线、场景、感知目标变化等); 通过目标设备和周围边缘设备和云端连接获取通信传输能力信息; 通过自定义的交互接口, 获取目标应用的性能 (如精度、时延) 需求; 以上获取的环境描述将输入给硬件自适应层和域自适应层.
- **硬件自适应层.** 面向硬件资源、网络传输的实时状况和预算, 自适应选择和组合各种模型压缩技术, 通过云边端融合选择模型最佳分割点, 从而对模型的协作运算模式和模型结构进行调整, 在满足硬件资源约束的前提下实现深度模型性能的最大化.

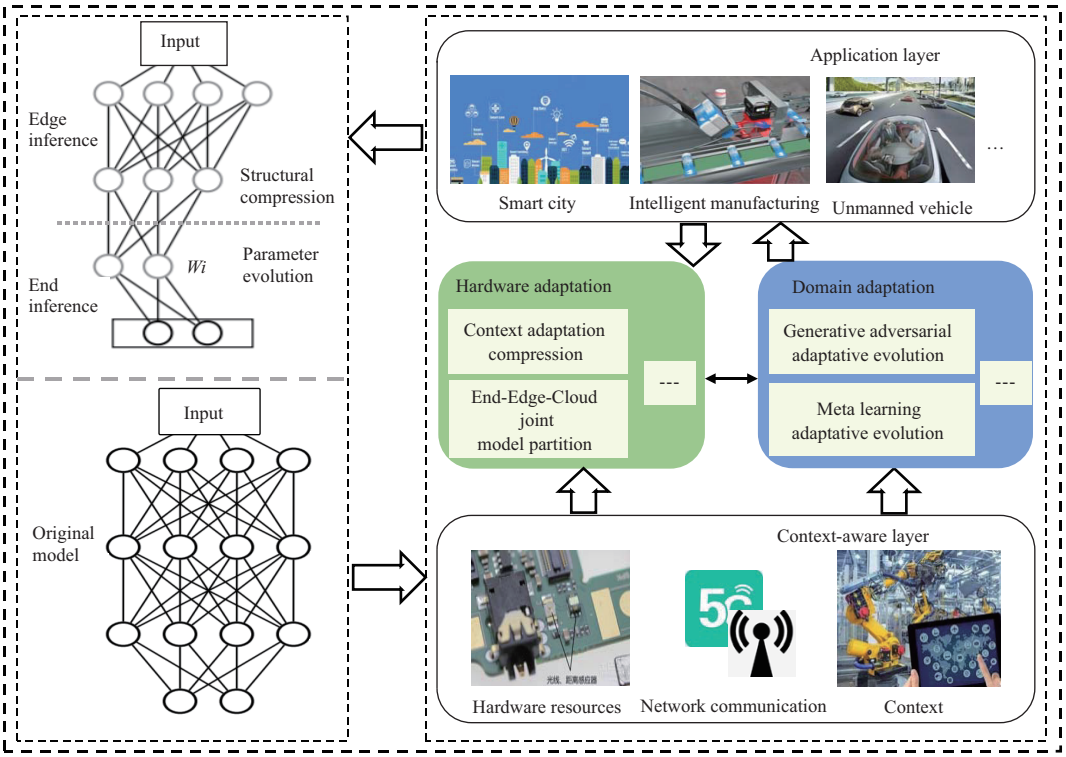

图 1 (网络版彩图) A^{2}IoT 系统架构

Figure 1 (Color online) A^{2}IoT system architecture

- **域自适应层.** 融合运用生成式对抗学习、迁移学习、元学习等算法, 调整基于源域训练的深度模型, 借助跨场景、跨实体知识迁移体系, 在不遗忘旧知识的同时, 使深度学习模型快速有效地适应新的目标域任务 [23].
- **应用层.** 面向目标应用数据模态、目标应用需求和应用场景, 按需调用硬件自适应层和域自适应层, 融合深度模型的压缩、分割和持续性演化的自适应方案, 建立满足当前应用需求的深度模型, 并提供持续性的模型更新和维护方案.

3 研究挑战及关键技术

作为一个新的研究领域, 深度学习模型环境自适应面临一系列新的研究挑战和问题, 下面我们将分别介绍其关键技术研究进展.

3.1 终端自适应模型压缩

深度学习模型随着网络结构的加深, 如 VGG [11] 和 ResNet [12] 等, 其准确率获得大幅提升, 但是也导致网络参数量与推断计算量急剧增加. 硬件设备与深度学习模型在硬件资源上的矛盾表现得愈发突出, 下面表 1 和 2 分别给出了近年来主流物联网终端设备的资源状况以及深度学习模型的参数和资源利用状况. 例如, Sony Smartwatch SW3 的运行内存为 512 MB, 但 AlexNet 模型的存储量已经高达 233 MB. 通过对比可知, 主流移动终端的硬件资源非常有限, 难以支撑深度学习模型的终端运行.

因此, 为将深度学习模型实际部署在资源受限的终端设备上, 需进行终端模型压缩 [24~30] 来降低

表 1　主流物联网终端资源状况

Table 1　Internet of Things terminal resources

Type	Device	Processor	DRAM	Battery (mAh)
Smart phone	Redmi 7A	Snapdragon 439	2 GB	4000
	Redmi 8A	Snapdragon 439	3 GB	5000
	Redmi Note 8	Snapdragon 665	4 GB	4000
	Redmi K30	Snapdragon 730	6 GB	6400
	Huawei changxiang 9 PLUS	Hisilicon Kirin 710	4 GB	4000
	Huawei nova 5z	HUAWEI Kirin 810	6 GB	4000
Smart watch	Sony Smartwatch SW3	ARM CortexA7	512 MB	420
	HUAWEI WATCH 2 Pro	Snapdragon wear 2100	768 MB	420
	Xiaomi watche	Qualcomm 3100	1 GB	570
Smart bracelet	Huawei bracelet B5	ARM Cortex M4	384 KB	108
	Xiaomi bracelet4	Dialog DA14681	512 KB	135

表 2　深度学习模型参数

Table 2　Deep learning model parameters

Network name	Parameter number (M)	Needed storage capacity (MB)	FLOPs
AlexNet	60	233	727 MFLOPs
GoogleNet	6.8	51	1.5 GFLOPs
ResNet-18	33	44.7	1.8 GFLOPs
ResNet-50	25.5	97.8	4.1 GFLOPs
ResNet-152	117	230	11 GFLOPs
VGG-16	138	528	16 GFLOPs
VGG-19	144	548	20 GFLOPs

推理阶段的计算和存储成本. 模型压缩已有多种技术类型, 其中应用较为广泛的是网络剪枝技术. 网络剪枝, 旨在剔除权重矩阵中冗余参数. 早在 1990 年, LeCun 等 [31] 从人类大脑的突触修剪得到灵感, 提出通过类似突触修剪的方式对模型进行剪枝, 以修剪模型的 “过度能力”. 但是网络剪枝在过去的几十年并未受到太多关注. 随着移动设备的普及以及模型深度的大幅度提升, 网络剪枝再次活跃在了人们眼前. 经过系统的发展, 网络剪枝细分为了两种方式: (1) 结构化剪枝, (2) 非结构化剪枝. 例如, Luo 等 [32] 针对单个神经元进行非结构化剪枝, 这样虽然可以保持较高的精度, 但是目前的软硬件基础很难支持非结构化的矩阵运算. Polyak 等 [33] 采用结构化剪枝方式, 针对卷积神经网络中的滤波器或通道进行结构化剪枝, 可以简化底层运算, 但是会引入较大的精度损失. Liu 等 [34] 将结构化修剪方法的组合应用到自动化过程中, 并以 ADMM 剪枝技术为核心结合退火算法自动搜索超参数进行结构化剪枝优化. 目前传统的压缩技术已较为成熟, 可以大幅度降低模型的参数量与运算量, 但它们往往通过预训练提供固定的压缩模式, 不能根据环境的动态变化实现自适应压缩和持续性演化.

为了提高模型的生存适应性和自动应对环境变化的能力, 需进一步研究终端模型的自适应压缩方法这一挑战性问题. 它将环境变化作为约束条件, 通过自动化搜索算法找到最佳的压缩超参数, 以实现模型结构对环境变化的自适应. 目前在这方面已经有一些初步的研究. 例如, DeepIoT [35] 采用了额外的压缩器网络, 将待压缩模型隐藏层输入其中, 由压缩器网络计算冗余概率并自动剔除冗余

连接. 该方法是最早提出的适用于所有类型神经网络层的自动化模型压缩技术之一. 但是该方法的压缩器采用序列网络结构, 为模型压缩的自动化搜索过程引入更高的计算复杂度和并行运算难度. AdaDeep [3] 首次将自适应模型压缩问题与 DNN 的超参数优化框架相结合, 将压缩技术看作是粗粒度的 DNN 超参数, 利用强化学习对不同的计算任务需求和平台资源约束进行自动化选择, 从而实现自适应的轻量级模型架构搜索. 虽然该方法通过超参数的自动化调优过程实现了粗粒度的模型自适应压缩, 但是难以实现更细粒度的自适应模型压缩. 同时, 谷歌在自适应模型压缩率搜索上开展了深入研究, 例如, AMC [14] 采用深度策略梯度逐层实现自动化的最佳稀疏率计算, 从而实现模型的自适应压缩. 特别地, ChamNet [36] 基于 AutoML 思想提出根据硬件资源的情况自动化组合基本的神经网络模块, 实现自动化神经网络架构设计. 然而, 上述工作没有将深度模型的动态综合性能指标 (如终端能耗) 纳入自适应反馈机制. 综上, 虽然自动化模型压缩技术已经取得很多进展, 但是目前终端模型压缩仍然面临很多挑战.

(1) 更灵活的自适应压缩策略. 目前压缩技术已经实现了一定程度的自适应, 但这种自适应的程度还较低. 虽然其中有些方法实现了精度和资源消耗的动态调节, 但是模型压缩的比例在压缩过程中依然是提前设定的. 事实上, 终端设备的硬件资源是动态变化的, 这导致固定压缩比的方式不能满足硬件资源的变化情况, 进而影响终端运行性能. 如何根据硬件资源的变化自动调整模型的压缩比例, 使其能够满足硬件资源的变化仍然是有挑战的.

(2) 压缩评估模型的构建. 深度学习模型每一层、每个计算单元的硬件资源消耗, 特别是压缩后所形成的新模型的资源消耗, 在部署前很难进行实际测量. 因此, 需要一个较为精确的资源消耗评估模型以对模型的压缩比提供指导. 因为深度学习模型与硬件设备都具有异构性, 如何构建一个评估模型, 在不同硬件设备上对各种深度学习模型都有良好的评估效果存在极大挑战. 这方面密歇根大学 (The University of Michigan) 的 Kang 等已经做了初步成果, Neurosurgeon [18] 提出通过高斯回归评估模型的资源消耗, 但是其只考虑了在固定硬件设备上进行评估.

3.2 云边端融合模型自适应分割

云边端融合模型自适应分割, 是指在云端上通过高斯回归等方法自主搜索模型最佳分割点, 并据此将深度学习模型进行切分, 分别部署在云端设备、边缘设备与终端设备上. 在保证云端设备、边缘设备与终端设备之间数据有效传输的情况下, 实现深度学习模型的推断过程. 由于在终端只执行一部分模型, 所以能有效降低计算成本, 但由于需要进行数据传输, 会增加传输的能源消耗和时间延迟代价.

一般来说, 终端设备所拥有的计算资源不能支持深度学习模型完成推断过程. 因此, 传统的解决方案是将深度学习模型的计算部分托管在云服务器上 [37~40], 计算结果传回端设备. 例如, Kang 等 [18] 和 Osia 等 [41] 的工作是在终端设备与云服务器之间拆分深度学习模型的计算部分, 并将其分别部署在终端设备与云服务器上. Mao 等 [42] 则选择在卷积神经网络第 1 个卷积层后做切分, 并分别部署在终端与云端, 目的是最小化边缘端的计算负载. 这种方法虽然可以解决资源限制问题, 但是也带来很多额外的问题. 例如, 提高了网络通信负载, 增加了数据泄露的风险, 不能实时处理数据等. 因此, 将深度学习模型进行云边端融合自适应分割近来得到了广泛关注. Neurosurgeon 框架 [18] 提出以层为粒度在移动设备与数据中心之间搜索最佳分割点, 划分深度神经网络, 以实现对硬件资源变化的自适应. Ko 等 [43] 则提出将模型分割与有损编码结合, 通过对中间数据进行有损编码, 以减少传输时间, 从而进一步降低整体的模型推理时延. JALAD [44] 通过建立一套基于归一化的层内数据压缩策略, 自动寻找深度学习模型结构的最佳分割点. DeepWear [45] 则提出通过蓝牙等本地网络, 将深度学习任务从可穿戴设备转移到配对的手持设备. 综上, 云边端融合模型分割研究通过扩展计算架构和资源解决本地

资源不足问题, 但是目前的研究仍然存在很多挑战.

(1) 自适应分割点选取. 在不同的硬件资源和网络通信环境下, 同一个深度学习模型的最佳分割点有时也是不同的. 此外, 在处理不同的任务时, 深度学习模型的最佳分割点也不尽相同. 所以, 在面临任务不同、硬件不同、网络不同、模型不同等问题时, 如何用统一的算法自适应外部条件变化, 确定最佳模型分割点仍然是具有挑战的.

(2) 多设备的自适应调度. 在深度学习模型部署环境中, 终端和边缘设备的数量是动态变化的. 为使这些终端设备得到充分利用, 模型在分割时应当自适应地调整分割策略, 动态选取分割点位与参与计算的设备数量, 并将模型动态分配给各设备以完成分布式协同学习任务. 虽然目前已有一些调度策略, 但是如何应对多设备的自适应调度仍然存在极大挑战, 特别是在当前物联网快速发展、终端设备不断丰富的背景下.

3.3 域自适应模型演化

深度学习模型能达到较好推断效果有两个基本假设: (1) 大量有标签的训练样本; (2) 训练和测试数据同分布. 但是在实际应用中, 这两个基本假设常常得不到保证. 例如, 不同设备、不同用户、不同场景、不同任务, 其数据分布往往具有差异性. 若将深度学习模型不加修缮而直接部署在这种环境下, 会大幅降低模型的推断准确率. 因此, 如何解决新环境下数据标签稀疏与概率分布不一致的问题一直是研究的热点, "域自适应" 在此背景下应运而生. 针对制造业环境下, 电动机工作环境的时变特性导致的数据分布差异问题, Xiao 等 [46] 提出将最大化平均差异的正则化合并到 CNN 的训练过程的方法, 能够自适应地演化模型以适应环境的变化. TCNN [47] 则针对工业设备故障检测中, 采集训练数据的实体与部署实体不同而产生的数据分布差异, 提出基于深度迁移卷积神经网络的在线故障诊断方法, 通过离线 CNN 与在线 CNN 的有机结合解决了跨实体的自适应演化问题. Li 等 [48] 则提出基于深度生成神经网络, 人工生成用于领域自适应的伪造样本来提高滚动轴承的故障诊断的精度, 以使诊断模型更加适应实际的工作环境. Shao 等 [49] 提出通过预训练的神经网络提取低级特征, 并用标记时频图像微调神经网络的高层结构, 以缩短深度神经网络的训练时间, 并提高模型对于不同机器故障的适应能力. DADA 框架 [50] 则构建了一个对抗性适应网络, 以解决源域和目标域的分布不一致问题, 提高了滚动轴承故障诊断的精确性, 以适应模型推断环境的快速变化. FNA [51] 针对相似新任务, 通过将种子网络扩充为超级网络, 并采用参数重映射技术, 从超网中搜索出具有不同深度、宽度或内核的网络结构, 以实现对新任务的快速适应. 然而, 对于跨场景、跨实体的深度学习模型域自适应仍然存在着很多未解决的问题.

(1) 模型终端域自适应. 如何在资源受限的硬件设备上解决深度学习模型的域自适应问题. 由于深度学习模型进行域自适应的过程实际是对模型进行重训练的过程, 这需要大量的硬件资源的支持, 但是实际的物联网终端并不能提供如此多的资源给模型, 进行域自适应训练. 因此在资源受限的硬件设备上同时环境自适应和域自适应是有很大挑战的.

(2) 跨域迁移模型的理论保证. 虽然目前有很多域自适应方法, 但是在标准数据集中这些模型仍然存在推断准确率较低的现象. 在环境复杂多变的物联网计算场景下, 基于现有方法的深度学习模型的域自适应性能无法保证. 因此如何在理论层面确保知识迁移的正向性, 减少负向知识迁移, 以及在物联网海量运行终端情况下从多个终端源域对目标域进行群体智能迁移等方面值得进一步深入研究.

4 我们的研究实践

在国家重点研发计划项目 “可持续演化的智能化软件理论、方法和技术 (前沿基础类)” 等支持下, 我们开展了深度学习模型环境自适应方向的研究, 下面就目前的工作进展进行介绍.

4.1 X-ADMM: 边端融合增强的模型压缩

在中国制造业升级迭代的历史机遇中, 对智能制造场景下的产品质量检测提出了更高的要求, 需要其拥有自适应感知、自适应检测的能力. 随着计算机视觉的发展, 利用图像识别的相关算法, 代替人工进行产品质量检测逐渐得到了应用. 但是制造业现场的嵌入式设备通常性能较差, 不能单独完成图像识别模型的推断过程. 因此需要降低模型的参数量与计算量, 帮助其完成部署. 若单独采取模型压缩技术虽然可以降低模型的资源消耗, 但是压缩后模型的资源消耗通常仍然不能被嵌入式设备所接受. 若是单独采取模型分割技术, 则模型大量的中间数据传输会产生额外的通信和能源消耗. 因此我们提出将自适应的模型压缩与模型分割技术相结合, 共同完成模型的部署. 首先将进行原始模型在训练阶段, 按照不同的压缩阈值, 训练出不同资源消耗的图像识别模型, 将其统一部署在边缘服务器上, 根据任务环境及资源环境变化, 自适应选择模型类型与模型分割点, 实现终端与边端的自适应联合推断. 我们首先对模型压缩与模型分割的联合应用进行了探索, 下面将详细介绍.

(1) 模型设计. 本文提出的 X-ADMM 方法融合了模型剪枝和分割的优势, 整体模型框架如图 2 所示. 基于 ADMM (alternating direction method of multipliers) [52] 的模型压缩方案, 采用结构化剪枝方法并基于 ADMM 进行精细修剪. 首先将 DNN 的权值剪枝问题归结为一个非凸优化问题, 用组合约束来指定稀疏性要求, 然后采用 ADMM 框架进行系统的权值修剪. 利用 ADMM, 将原非凸优化问题分解为两个子问题进行迭代求解. 其中一个子问题可以用随机梯度下降法求解, 另一个子问题可以用解析法求解. 通过对这两个子问题的求解完成剪枝过程.

模型在推断过程中, 传递的信息流是输入数据的特征矩阵. 理论上只要保证模型前后传递特征矩阵的准确性, 可以将模型的不同部分放置在不同的设备运行. 模型分割理论就是基于这样的一种观察. 我们发现模型压缩技术与模型分割技术是相互独立的, 两者互不影响, 且能够从不同的层面降低模型对于硬件资源的消耗. 基于这种观察我们将模型压缩方法与模型分割方法进行结合.

终端模型压缩完成之后, 已经实现了对模型的缩小. 在此基础上, 进一步基于 Neurosurgeon [18] 的分割方法, 以层为粒度进行模型的划分, 并将分割后的模型分别部署在不同的终端设备于边缘设备上. 首先, 预测在模型每层分割可能产生的延迟. 其次, 结合任务的延迟要求综合考虑模型的推断延迟和资源消耗, 选择最佳的模型分割点. 总的来说, 我们在完成终端模型压缩工作之后, 进一步使用模型分割算法将深度学习模型进行分割, 并分别部署在边缘设备与终端设备上.

(2) 实验细节及结果. 我们采用了树莓派 4 作为终端设备, 戴尔 Inspiron 13-7368 作为边缘设备, 对本文提出的 X-ADMM 和最新扩展的 ADMM 方法 —— RAP-ADMM [53] 方法进行了性能对比. 神经网络模型则分别选择 AlexNet, GoogleNet, ResNet-18, VGG-16, MobileNet, ShuffleNet 等图像处理领域常用的 CNN 网络进行实验. 模型压缩实验过程中利用 CIFAR-10 数据集进行模型的训练过程, CIFAR-10 是普适的彩色图片小型数据集, 数据集中包含 10 个类别, 分别是: 飞机、汽车、鸟类、猫、鹿、狗等. 每个类中包括多达 5000 张训练图片和 1000 张测试图片. 模型中的网络结构代码基于 Pytorch 实现, 实验过程中网络实现及模型参数设置如下: 模型压缩阶段, 训练过程的 batch 大小设为 64, momentum 设置为 0.9, 权重衰减指数为 0.0005. 利用均值为 0, 方差为 0.01 的高斯分布进行权重初始化, 优化器选择 Adam 优化器, 在每个卷积层和之后激活之前进行批归一化, 初始学习率设置为

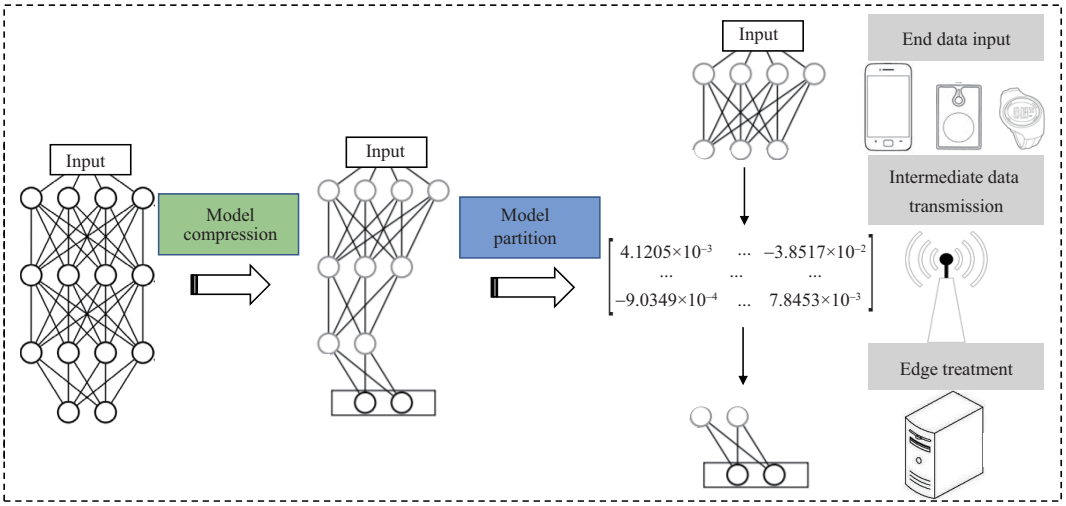

图 2 (网络版彩图) X-ADMM 模型框架图

Figure 2 (Color online) X-ADMM model framework

表 3 模型压缩与分割实验结果

Table 3 Experimental results of model compression and partition

Network name	Original accuracy (%)	Original inference time (ms)	Model compression and partition				
			Compression rate (%)	RAP-ADMM inference time (ms)	RAP-ADMM accuracy (%)	X-ADMM inference time (ms)	X-ADMM accuracy (%)
Alexnet	85.82	46.8	16.0	38.78	84.21	32.1	84.17
GoogleNet	87.48	943.6	16.0	883.95	84.91	532.6	84.60
ResNet-18	91.60	285.5	16.0	267.7	90.01	213.5	89.80
VGG-16	91.66	203.7	16.0	186.9	89.59	88.3	89.20
MobileNet	89.60	219.2	2.0	207.89	87.96	179.2	87.90
MobileNet	89.60	219.2	4.0	196.69	80.60	160.3	80.57
ShuffleNet	88.14	202.8	4.0	183.79	84.25	153.4	84.19

0.1, 并以 10 倍的速率进行衰减. 结果表明, 相比较于 RAP-ADMM, X-ADMM 在 6 种模型上都取得了较好效果, 具体结果见表 3.

实验结果表明, X-ADMM 对所有 6 种 DNN 模型都具有加速效果, 相比于原始模型, 推断时间分别减少了 31.41%, 43.56%, 25.22%, 56.65%, 18.25%, 26.87%, 24.36%. 而精度则分别下降了 1.65%, 2.88%, 1.80%, 2.46%, 1.7%, 8.77%, 3.95%. X-ADMM 与 RAP-ADMM 相比, 在 6 种 DNN 模型上的推断时间分别下降了 17.22%, 39.75%, 20.25%, 52.76%, 13.8%, 18.5%, 16.54%, 而精度分别下降了 0.04%, 0.31%, 0.21%, 0.39%, 0.06%, 0.03%, 0.06%. 从结果可以看出, 对于深度学习模型 X-ADMM 都可以取得优于 RAP-ADMM 的加速效果, 而精度下降则非常有限. 因此, X-ADMM 方法对于面向硬件资源的模型自适应是具有实际意义的.

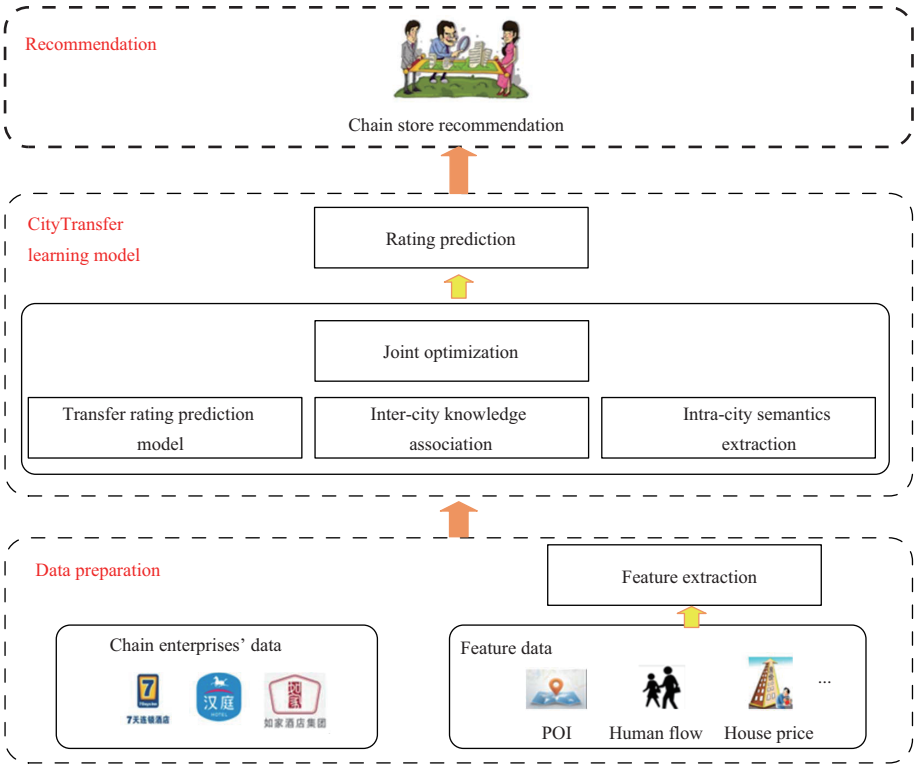

图 3 (网络版彩图) CityTransfer 系统框架

Figure 3 (Color online) CityTransfer system framework

4.2 CityTransfer: 跨城市跨实体知识迁移

域自适应模型演化存在的一个重要挑战是跨域迁移模型的理论保证. 本文结合智能商业选址的实际问题对这一问题进行初步的探索. 在当前城市大数据和人工智能的背景下, 智能商业选址一直是一个热门的研究领域. 而对于一些新的企业或者该企业进军新的城市时, 原有城市的推荐模型将会失效. 在智能选址时, 两个不同城市的数据分布是不同的. 因此, 对于城市选址模型进行域自适应是十分有必要的. 从其他企业、其他城市中提取类似的知识, 辅助提高推荐的准确度. 目前已经出现了很多利用机器学习、深度学习进行零售店选址推荐的研究 [54~56], 但是这些均是基于大量丰富的城市数据而进行的推荐. 这些方法并不适用于新的企业或者该企业进军新的城市时的商业选址. 此时, 需要从其他企业、其他城市中提取类似的知识, 辅助提高推荐的准确度. 然而利用多源的城市数据存在 3 个挑战: (i) 如何将源城市学习到的知识迁移到目标城市来处理冷启动的问题; (ii) 不同城市的数据在特征和评分上具有不同的分布, 如何比较两个不同城市的位置. (iii) 从多源城市提取的局部特征或许存在噪声, 如何去掉冗余提高知识迁移的效果.

(1) 模型设计. 针对这些挑战, 我们提出 CityTransfer 框架 [57], 如图 3 所示, 实现了同城市跨实体的知识迁移和同实体跨城市的知识迁移. 它主要从以下 3 方面考虑.

(i) 数据处理: 分别爬取连锁店数据、城市 POI 的数据和房价数据. 首先将每个城市划分为同等大小的网格, 然后对于每个网格, 提取多样性、人流量、交通便利性、POI 集合等地理特征和密度、竞争性、互补性、房价等商业特征作为城市和店铺的特征表示.

(ii) 知识迁移: 对于每个城市, 使用 AutoEncoder 方法重构地理网格的原始特征, 剔除冗余和噪

表 4 城市间知识关联和城市内语义特征的作用结果

Table 4 The results of inter-city knowledge association and semantic characteristics

	Hanting Inn		7 Days Inn		Home Inn	
	NDCG	RMSE	NDCG	RMSE	NDCG	RMSE
MF	0.663	1.469	0.652	1.592	0.628	1.435
MF_ SE	0.683	1.413	0.788	1.098	0.736	1.346
MF_ KA	0.741	1.689	0.623	1.716	0.782	1.735
CityTransfer	0.769	1.548	0.812	1.205	0.701	1.261

表 5 从不同的源城市迁移知识到西安的作用结果

Table 5 The effect of transferring knowledge from different source cities to Xi'an

Source city	Hanting Inn	7 Days Inn	Home Inn
Beijing	0.859	0.826	0.814
Shanghai	0.729	0.798	0.729

表 6 从不同的源城市迁移知识到南京的作用结果

Table 6 The effect of transferring knowledge from different source cities to Nanjing

Source city	Hanting Inn	7 Days Inn	Home Inn
Beijing	0.754	0.821	0.764
Shanghai	0.801	0.848	0.765

音, 使得新特征变得更加健壮和信息化. 对于同城市跨实体的知识迁移, 利用扩展的 SVD 模型进行迁移. 对于同实体跨城市的知识迁移, 通过测量特征之间的 Pearson 相关系数建立城市间对应的网格之间的关联, 并将其用于之后的训练过程中, 以此消除城市间的不可比性. 最后联合优化这些目标, 使用梯度下降的方法得到每个网格的预测评分.

(iii) 地址推荐: 根据训练好的模型对每个网格给出预测评分, 选取得分最高的 Top K 个网格进行地址的推荐.

(2) 实验结果. 我们使用归一化折扣累积增益 (normalized discounted normalized gain, NDCG) 和均方根误差 (root mean square error, RMSE) 作为实验的评估指标. 一方面表明回归问题的优劣, 另一方面阐述排序的质量好坏.

首先, 为了分析我们模型的有效性, 我们将其与基于目标城市单源数据预测的 MF 方法、同城市跨实体的 MF_SE 方法以及同实体跨城市的 MF_KA 方法比较, 由表 4 可以看出, 我们的方法 CityTransfer 取得最好的性能. 表明了城市间的知识关联和城市内语义特征对与城市间、实体间的知识迁移具有较大的作用.

其次, 我们分析了从不同的源城市进行知识迁移的效果. 将北京和上海作为源城市, 西安或南京作为目标城市. 从表 5 和 6 中, 我们可以观察到: (i) 北京比上海迁移的知识更有效, 可能是因为北京有更多数量的连锁店. (ii) 当西安作为目标城市时, 北京迁移的知识更有效, 或许是因为北京和西安均在北方, 有相似的地理特征. 当南京作为目标城市时, 上海迁移的知识更有效, 原因是类似的, 即上海的地理特征同南京的相似.

因此, 我们在选取源城市时, 一方面要考虑两个城市的相似性 (地域邻近性、特征相似性等), 另一

方面要考虑源城市中目标企业的数量.

5 结论与展望

本文对深度学习模型的终端环境自适应这一新兴研究领域进行了阐述, 并进一步阐述了其中应用模式和通用型系统架构. 从该领域问题出发, 论述了深度学习模型环境自适应面临的一些挑战, 包括更灵活的自适应压缩、压缩评估模型的构建、自适应分割点选取、多设备的自适应调度等, 并介绍了我们在这方面开展的研究探索. 目前, 智能物联网 AIoT 方兴未艾, 在智慧城市、智能制造等国家重大需求背景下, 人工智能与终端应用的结合将越来越紧密. 本文探索的深度学习模型终端环境自适应这一新兴方向面向 AI 落地实践具体问题, 对于推动人机物深度融合, 促进自适应、可成长智慧系统空间的构建具有重要意义. 同时, 该领域还面临着很多新的研究挑战和机遇, 需要广大研究人员去进一步探索和解决.

参考文献

1 Iresearch. White Paper on China's Artificial Intelligence Internet of Things (AIoT) in 2020. Iresearch Series of Research Reports, 2020, 2: 36–80 [上海艾瑞市场咨询有限公司. 2020 年中国智能物联网 (AIoT) 白皮书. 艾瑞咨询系列研究报告, 2020, 2: 36–80]

2 Zhou Z, Chen X, Li E, et al. Edge intelligence: paving the last mile of artificial intelligence with edge computing. Proc IEEE, 2019, 107: 1738–1762

3 Liu S C, Lin Y Y, Zhou Z M, et al. On-demand deep model compression for mobile devices: a usage-driven model selection framework. In: Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, 2018. 389–400

4 Teerapittayanon S, McDanel B, Kung H T. BranchyNet: fast inference via early exiting from deep neural networks. In: Proceedings of 2016 23rd International Conference on Pattern Recognition (ICPR), 2016. 2464–2469

5 Zhao Z, Barijough K M, Gerstlauer A. DeepThings: distributed adaptive deep learning inference on resource-constrained IoT edge clusters. IEEE Trans Comput-Aided Des Integr Circ Syst, 2018, 37: 2348–2359

6 Yang Q, Zhang Y, Dai W, et al. Transfer Learning. Cambridge: Cambridge University Press, 2020

7 Wang L, Guo B, Yang Q. Smart city development with transfer learning. Computer, 2018, 51: 32–41

8 Jordan M I, Mitchell T M. Machine learning: trends, perspectives, and prospects. Science, 2015, 349: 255–260

9 Chen Z, Liu B. Lifelong machine learning. In: Synthesis Lectures on Artificial Intelligence and Machine Learning, 2018, 12: 1–207

10 Mitchell T, Cohen W, Hruschka E, et al. Never-ending learning. Commun ACM, 2018, 61: 103–115

11 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv: 1409.1556

12 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778

13 Han S, Mao H, Dally W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding. 2015. ArXiv: 1510.00149

14 He Y, Lin J, Liu Z, et al. AMC: automl for model compression and acceleration on mobile devices. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 784–800

15 Cai H, Gan C, Wang T, et al. Once-for-all: train one network and specialize it for efficient deployment. 2019. ArXiv: 1908.09791

16 Chen T, Moreau T, Jiang Z, et al. TVM: an automated end-to-end optimizing compiler for deep learning. In: Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation, 2018. 578–594

17 Ma X, Guo F M, Niu W, et al. PCONV: the missing but desirable sparsity in DNN weight pruning for real-time execution on mobile devices. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020. 5117–5124

18 Kang Y, Hauswald J, Gao C, et al. Neurosurgeon: collaborative intelligence between the cloud and mobile edge. SIGARCH Comput Archit News, 2017, 45: 615–629

19 Li E, Zhou Z, Chen X. Edge intelligence: on-demand deep learning model co-inference with device-edge synergy. In: Proceedings of the 2018 Workshop on Mobile Edge Communications, 2018. 31–36

20 Ganin Y, Ustinova E, Ajakan H, et al. Domain-adversarial training of neural networks. J Machine Learn Res, 2016, 17: 1–35

21 Wang X, Li L, Ye W, et al. Transferable attention for domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2019. 33: 5345–5352

22 Nagabandi A, Clavera I, Liu S, et al. Learning to adapt in dynamic, real-world environments through metareinforcement learning. 2018. ArXiv: 1803.11347

23 Kirkpatrick J, Pascanu R, Rabinowitz N, et al. Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci USA, 2017, 114: 3521–3526

24 Wen W, Wu C, Wang Y, et al. Learning structured sparsity in deep neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2016. 2074–2082

25 Luo J H, Wu J. An entropy-based pruning method for cnn compression. 2017. ArXiv: 1706.05791

26 Guo Y, Yao A, Chen Y. Dynamic network surgery for efficient dnns. In: Proceedings of Advances in Neural Information Processing Systems, 2016. 1379–1387

27 Han S, Pool J, Tran J, et al. Learning both weights and connections for efficient neural network. In: Proceedings of Advances in Neural Information Processing Systems, 2015. 1135–1143

28 He Y, Zhang X, Sun J. Channel pruning for accelerating very deep neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 1389–1397

29 Zhang T, Ye S, Zhang K, et al. A systematic DNN weight pruning framework using alternating direction method of multipliers. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 184–199

30 Zhang T, Zhang K, Ye S, et al. ADAM-ADMM: a unified, systematic framework of structured weight pruning for DNNs. 2018. ArXiv: 1807.11091

31 LeCun Y, Denker J S, Solla S A. Optimal brain damage. In: Proceedings of Advances in Neural Information Processing Systems, 1990. 598–605

32 Luo J H, Wu J, Lin W. ThiNet: a filter level pruning method for deep neural network compression. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 5058–5066

33 Polyak A, Wolf L. Channel-level acceleration of deep face representations. IEEE Access, 2015, 3: 2163–2175

34 Liu N, Ma X, Xu Z, et al. AutoCompress: an automatic DNN structured pruning framework for ultra-high compression rates. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020. 4876–4883

35 Yao S, Zhao Y, Zhang A, et al. DeepIoT: compressing deep neural network structures for sensing systems with a compressor-critic framework. In: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, 2017. 1–14

36 Dai X, Zhang P, Wu B, et al. ChamNet: towards efficient network design through platform-aware model adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 11398–11407

37 Julie B. The 'Google Brain' is a real thing but very few people have seen it. Business Insider, 2016. https://www.businessinsider.com.au/what-is-google-brain-2016-9

38 Norm J. Google supercharges machine learning tasks with TPU custom chip. AI Machine Learning, 2016. https://cloud.google.com/blog/products/gcp/google-supercharges-machine-learning-tasks-with-custom-chip

39 Siegler M G. Apple's Massive New Data Center Set To Host Nuance Tech. TC Sessions, 2011. https://techcrunch.com/2011/05/09/apple-nuance-data-center-deal

40 Ben L. Apple moves to third-generation Siri back-end, built on opensourceMesos platform. 9To5Mac, 2015. http://9to5mac.com/2015/04/27/siri-backend-mesos

41 Osia S A, Shamsabadi A S, Sajadmanesh S, et al. A hybrid deep learning architecture for privacy-preserving mobile analytics. IEEE Internet Things J, 2020, 7: 4505–4518

42 Mao Y, Yi S, Li Q, et al. A privacy-preserving deep learning approach for face recognition with edge computing. In: Proceedings of USENIX Workshop Hot Topics Edge Comput (HotEdge), 2018. 1–6

43 Ko J H, Na T, Amir M F, et al. Edge-host partitioning of deep neural networks with feature space encoding for resource-constrained Internet-of-Things platforms. In: Proceedings of 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2018. 1–6

44 Li H, Hu C, Jiang J, et al. JALAD: joint accuracy-and latency-aware deep structure decoupling for edge-cloud execution. In: Proceedings of 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), 2018. 671–678

45 Xu M W, Qian F, Zhu M Z, et al. DeepWear: adaptive local offloading for on-wearable deep learning. IEEE Trans Mobile Comput, 2020, 19: 314–330

46 Xiao D, Huang Y, Zhao L, et al. Domain adaptive motor fault diagnosis using deep transfer learning. IEEE Access, 2019, 7: 80937–80949

47 Xu G, Liu M, Jiang Z, et al. Online fault diagnosis method based on transfer convolutional neural networks. IEEE Trans Instrum Meas, 2020, 69: 509–520

48 Li X, Zhang W, Ding Q. Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks. IEEE Trans Ind Electron, 2019, 66: 5525–5534

49 Shao S, McAleer S, Yan R, et al. Highly accurate machine fault diagnosis using deep transfer learning. IEEE Trans Ind Inf, 2019, 15: 2446–2455

50 Liu Z H, Lu B L, Wei H L, et al. Deep adversarial domain adaptation model for bearing fault diagnosis. IEEE Trans Syst Man Cybern Syst, 2020. doi: 10.1109/TSMC.2019.2932000

51 Fang J, Sun Y, Peng K, et al. Fast neural network adaptation via parameter remapping and architecture search. 2020. ArXiv: 2001.02525

52 Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. FNT Mach Learn, 2010, 3: 1–122

53 Ye S, Xu K, Liu S, et al. Adversarial robustness vs. model compression, or both. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019. 111–120

54 Li J, Guo B, Wang Z, et al. Where to place the next outlet? Harnessing cross-space urban data for multi-scale chain store recommendation. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, 2016. 149–152

55 Li N, Guo B, Liu Y, et al. Commercial site recommendation based on neural collaborative filtering. In: Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers, 2018. 138–141

56 Liu Y, Guo B, Li N, et al. DeepStore: an interaction-aware wide&deep model for store site recommendation with attentional spatial embeddings. IEEE Internet Things J, 2019, 6: 7319–7333

57 Guo B, Li J, Zheng V W, et al. CityTransfer: transferring inter-and intra-city knowledge for chain store site recommendation based on multi-source urban data. Proc ACM Interact Mob Wearable Ubiquitous Technol, 2018, 1: 1–23

Context-aware adaptation of deep learning models for IoT devices

Bin GUO*, Yungang WU, Hongli WANG, Hao WANG, Sicong LIU, Jiaqi LIU,
Zhiwen YU & Xingshe ZHOU

School of Computer, Northwestern Polytechnical University, Xi'an 710072*, China*
* Corresponding author. E-mail: guob@nwpu.edu.cn

Abstract The rapid development of both Artificial Intelligence (AI) and the Internet of Things (IoT), has cultivated the new research area: the Artificial Intelligence of Things (AIoT). AIoT is used to deploy many different deep learning models on a variety of local IoT terminals including smartphones, wearables, and other embedded devices. Adapting to these dynamic and varied AIoT application scenarios, and the IoT platform resources (e.g., computation and storage resources) available in each diverse, requires a novel scheme for improving on device environmental adaptability. Deep learning models aim to dynamically adjust either the model structure, the calculation scheme, or both, of them specifically to adapt to the environment context. They must reduce costs and improve computational efficiency while creating negligible performance degradation. Specifically, an environmental adaptation evolution framework must actively and continuously assess the constantly changing environmental context including factors, such as application data, knowledge base, task-related performance requirements, and platform-imposed resource constraints. Then it must adopt on-demand model compression, model segmentation, and domain adaptation techniques to achieve a appropriate balance between the model's performance and the environment's budget. This paper focuses on making deep learning models for context-aware adaptation. We discuss the system architecture and core technologies solving this problem requires. We address research challenges in this area, and introduce our pilot research practice in this field.

Keywords AIoT, context-aware adaptation, model evolution, deep learning model compression, edge-based model partition, domain adaptation

Bin GUO was born in 1980. He is a Ph.D. professor and Ph.D. supervisor. He is a senior member of China Computer Federation. His main research interests include ubiquitous computing, social and community intelligence, urban big data mining, mobile crowd sensing, and human-computer interaction.

Yungang WU was born in 1997. He is a master's degree candidate at Northwestern Polytechnical University (NWPU), China. His main research interest includes context-aware adaptation of deep learning models.

Hongli WANG was born in 1999. She received her B.E. degree in computer science and technology from NWPU in 2020. She is currently working toward a master's degree at NWPU. Her research area is context-aware adaptation of deep learning models.

Hao WANG was born in 1996. He received his B.E. degree in computer science and technology from NWPU in 2019. He is currently working toward a Ph.D. degree at NWPU. His current research interests include natural language processing and dialog systems.