·开发研究与工程应用·　　文章编号：1000-3428(2023)10-0313-08　　文献标志码：A　　中图分类号：TP181

基于人工智能与边缘代理的物联网框架设计

李亚国[1]，李冠良[2]，张　凯[2]，晋　涛[2]

（1.国网山西省电力公司，太原 030001；2.国网山西省电力公司电力科学研究院，太原 030001）

摘　要：随着物联网和人工智能(AI)的技术发展及产品在各业务领域的推广，将边缘计算与AI模型集成融合，实现物联网智能化与计算前置化能满足更多的应用场景，但边缘代理设备通常受到硬件资源能力、性能及安全隐私等问题限制，将AI和边缘计算有效融合集成存在较大挑战。在物联网系统中，基于AI对边缘架构进行优化调整，构建具备边缘计算及AI能力的物联网新型智能框架，有效实现将边缘计算和AI集成到物联网系统中。在边端侧AI模型引导阶段，设计私有数据和公共数据的存储策略，有效提高数据安全性；在模型部署阶段，设计可配置压缩比的云端压缩、边端解压缩的部署模式，减少模型大小和传输所需数据流量，实现模型在边端侧的轻量级部署；在模型学习阶段，设计迁移学习和增量学习互补的学习方式，增加边端侧的模型训练及实用能力，提高云-边协作水平。实验结果表明，集成在边端的AI模型在资源占用率不足云模型50%情况下，准确率达到88%，同时训练时间比云模型快5倍以上。

关键词：物联网；边缘计算；人工智能；深度学习；云边协同

开放科学(资源服务)标志码(OSID)：

中文引用格式：李亚国，李冠良，张凯，等. 基于人工智能与边缘代理的物联网框架设计[J]. 计算机工程，2023，49(10)：313-320.

英文引用格式：LI Y G，LI G L，ZHANG K，et al. Design of Internet of Things framework based on artificial intelligence and edge agent[J]. Computer Engineering，2023，49(10)：313-320.

Design of Internet of Things Framework Based on Artificial Intelligence and Edge Agents

LI Yaguo[1]，LI Guanliang[2]，ZHANG Kai[2]，JIN Tao[2]

（1.State Grid Shanxi Electric Power Company，Taiyuan 030001，China；
2.Electric Power Research Institute of State Grid Shanxi Electric Power Company，Taiyuan 030001，China）

【Abstract】This paper explores the integration of edge computing with Artificial Intelligence（AI）models to enhance the capabilities of the Internet of Things（IoT）and expand its application potential. However，challenges arise from limitations in edge devices，including hardware resource capacity，performance，and concerns related to security and privacy. This study addresses the effective fusion of AI and edge computing by presenting a novel IoT framework. This framework improves data security by developing storage strategies for private and public data. Additionally，it enhances model training and its practical applications at the edge through migration learning and incremental learning，thereby enhancing collaboration in the cloud environment. Concurrently，the framework achieves lightweight deployment of models on the edge by implementing cloud compression and edge decompression techniques. Despite the constrained hardware capabilities of edge devices，the proposed approach ensures high AI recognition accuracy and performance. Experimental results demonstrate that the resource usage of edge-integrated AI model is less than 50% compared to cloud-based models，with edge-based AI achieving an accuracy rate of up to 88%. Moreover，its training time is more than five times faster than that of cloud-based models.

【Key words】Internet of Things（IoT）；edge calculation；Artificial Intelligence（AI）；deep learning；cloud side collaboration

DOI：10. 19678/j. issn. 1000-3428. 0065868

基金项目：国网山西省电力公司科技项目(520530202002)。

作者简介：李亚国(1977—)，男，高级工程师、硕士，主研方向为配电新技术研究；李冠良、张　凯，工程师、硕士；晋　涛，高级工程师、硕士。

收稿日期：2022-09-28　　**修回日期：**2022-11-18　**E-mail：**594122581@qq.com

0 概述

随着物联网技术及其产品研究的推广，物联网已逐步应用到各类生产、制造等领域中。通过物联网技术可实现现场各数据实时采集和分析预警功能，由此对物联网提出了低延迟、安全性、可靠性和智能性等更高要求[1]。由于边缘设备比云服务器更接近数据源，能够缩短网络延迟和带宽消耗，将数据采集后进行本地处理，也称为边缘计算，是克服上述障碍的必然[2]。边缘计算目前被应用于智能生产、智能运输、智能运检等各种智能物联网应用，其核心能力是实现现场各传感器及终端设备物联感知与实时采集。由于近年来人工智能（Artificial Intelligence，AI）技术被广泛采用，有望为边缘节点提供分布式智能服务[3]。将人工智能集成到边缘代理中，实现预警分析前置化具有较大的应用场景。例如在变电站、配电站房智能运检业务中，对站内生产环境实现智能分析，将现场作业安全违章识别等AI模型集成在边缘代理中，不依赖于网络传输，告警产生后现场即时处理，提高现场安全管控和管理能力。

本文通过在边缘代理中融合与集成人工智能模型，构建新型智能的边缘物联网框架，实现物联网智能化和智能计算前置化。在边缘层面采用迁移学习和增量学习，通过边缘侧本地私有数据重新训练AI模型，并使用增量数据迭代训练模型。在此基础上，设计云压缩和边缘解压机制，以降低模型部署的网络成本，并通过配置压缩比降低网络成本。最后设计一个实验仿真的系统原型来评估该框架的性能。

1 关键问题

人工智能和基于边缘计算的物联网技术在国内外已取得较大的进展，两者融合最初是将简单机器学习算法部署到边缘节点以实现对视频流数据的识别认知，比如火灾检测、交通流分析等[4]。然而，边缘计算、人工智能和物联网技术是为特定目的而开发的，将它们组合成一个统一的解决方案仍处于早期阶段，并面临以下挑战：

1）在传统的边缘云范例中，人工智能模型由云服务器上的公共数据集进行全面训练，而边缘侧只负责推理操作。这类设计对于存在个性化的上下文中或服务于特定目的应用场景中效率低下[5]。

2）进入边缘设备的大部分数据都包含个人用户和本地现场数据，尤其是电网数据涉及高度的敏感性。因此，与云服务器共享这些数据对于云边协作构建高性能人工智能模型，会存在一定的安全隐患和风险。

3）由于边缘设备在计算、存储和通信能力方面受到限制，频繁地从云端接收大型AI模型可能会导致设备过载和网络拥塞。

本文研究如何将边缘计算和人工智能有效地集成到物联网系统中[6]。一方面，在云端训练人工智能模型的传统方法不适用于物联网应用，因为通过互联网传输大量数据可能会带来严重的延迟和信息安全风险。另一方面，训练AI模型需要很高的计算能力，在边缘设备上进行训练会受到性能约束，且高功耗较高。因此，基于人工智能对边缘架构进行优化调整，包括设计数据共享结构、模型部署方法和云-边协作策略，并由此构建一个支持AI的边缘代理物联网框架，实现将边缘计算和AI集成到物联网系统中[7]。

2 主要技术及思路

近年来，国内外研究人员对人工智能和边缘计算的整合已开展一定研究。本文关注最近边缘人工智能相关的研究成果，总体思路大致分为框架设计、模型适应和处理器加速[8]。

2.1 框架设计

基于边缘代理框架进行改进设计，在不干扰AI模型结构的情况下提高模型在边缘侧的训练和推理阶段的性能。

1）模型训练：模型训练的工作主要基于知识蒸馏的框架技术开展。基于知识蒸馏框架的设计目标是最大限度地将模型从大型和深度神经网络（Deep Neural Network，DNN）转移到小型和浅层网络[9]。该框架将模型训练学习过程分为边缘代理侧学习和服务器侧学习两个阶段。在前一个阶段对每个代理进行模型迁移，并基于代理的本地数据进行强化学习（Reinforcement Learning，RL）训练。在后一阶段，联邦学习（Federated Learning，FL）服务器会定期地从所有代理收集RL模型并重新生成联邦模型。该方法使边缘设备能够学习和改进预测模型，同时在本地保留其私有数据[10]。

2）模型推理：模型推理性能优化的工作主要通过众多节点在模型分区上进行，这对大型和复杂的学习模型特别有益。将模型参数分为多个部分，每个部分由一个单独节点保存，并由各节点聚合执行训练或推理任务。通过模型分区可避免用户私有数据暴露，开发差分隐私机制可保证数据在边缘设备上实现隔离。文献[11]设计了一个保护隐私的深度神经网络，对敏感数据和模型参数进行保护。模型分区提高了数据隐私，因每个节点只有权访问各自的数据部分并保留相应的模型参数子集。

2.2 模型适配

通过对模型进行适当的适配处理，可使预训练的AI模型更适合边缘架构，包括模型压缩、条件计算、算法异步和彻底去中心化。

1）模型压缩：文献[12]提出了一种支持智能设备个性化和提高容错能力的修剪后再训练方案，并设计了MobileNets模型，包括宽度乘数和图像分辨率参数，形成了16个具有不同延迟、精度和大小值的模型。文献[13]引入一个模型标准，可根据给定

的任务分析连接的重要性,将重要性较低的连接淘汰,以降低空间和时间的复杂度。

2)条件计算:文献[14]提出了条件计算的方法,通过高度非线性的非微分函数设计一个梯度的无偏估计器,解决了DNN耗时且计算成本高的问题,文献[15]扩展了该设计,在应用非线性时通过消除具有零值的隐藏单元来降低权重矩阵的等级,并通过打开或关闭部分网络组件来优化性能和资源利用率。

3)算法异步化:为将本地模型异步合并到联邦学习中,文献[16]引入了Gossip随机梯度下降(GoSGD)算法来异步训练模型,模型信息通过随机八卦算法在不同线程之间完全共享,并给出一个专门为边缘计算(Edge Computing,EC)平台设计的框架,减少不同EC平台之间的性能差异。

4)去中心化设计:区块链作为去中心化的技术,主要是用于比特币引入,以消除联邦学习中所用的单一服务器[17]。文献[18]提出了一种基于区块链的联邦学习架构(BlockFL),使模型更新由相应的矿工利用工作证明机制进行交换和验证,并设计了基于联邦学习和区块链的新系统,实现去中心化。

2.3　处理器加速

通过改善计算密集型操作(如乘法或累加)来优化DNN的结构,以实现对处理器加速。在处理器加速方面的研究主要集中在DNN特定指令集设计和驱动内存计算设计。

1)DNN特定指令集设计:由于DNN的快速发展,许多硬件制造商都具有针对DNN计算的特殊特性。基于容错性考虑,DNN模型倾向于以16 bit浮点(FP16)格式存储数据,而不是FP32或FP64,以节省内存并易于扩大系统。例如英特尔Knights Mill CPU(Xeon Phi 7295 SR3VD)可通过添加指令集来实现BFloat16格式——BF16计算,以支持深度学习训练任务。虽然CPU和GPU在访问数据时会消耗大量时间,但谷歌设计的张量处理单元(TPU)可实现在每周期基于收缩阵列机制处理多达65 536个8 bit乘加单元,并用于矩阵运算[19]。

2)驱动计算内存设计:可应用记忆体来模拟ISAAC架构中的原位模拟运算,以减少数据移动。文献[20]提出了一种名为radix-X卷积神经元网络横杆阵列的新型算法,通过利用记忆体横杆阵列加速CNN推理,以实现近数据处理。

本文基于上述技术开展人工智能和边缘代理的融合研究,设计一种新型的AI+边缘智能框架,并取得了以下成果:1)设计一种新颖的学习过程,包括迁移学习和增量学习,并应用到边缘代理设备中,使其具备增强的AI智能功能,可使用本地存储的个性化和增量数据实现模型动态再训练;2)设计一种数据存储策略,将边缘数据分为公共和私有,私有数据被锁定在本地边缘以避免隐私问题,而公共数据被传输到云端以进行模型训练任务,从而实现保护数据隐私的同时,提高边缘-云协作能力[21];3)设计一种轻量级部署范式,支持压缩比可配置的AI模型云压缩和边缘解压机制,实现网络消耗最小化,降低了边缘AI模型的部署成本。

3　AI+边缘智能框架设计

为解决上述问题,本文提出了AI+边缘智能框架,如图1所示,该框架对模型的处理分为引导、部署、操作及学习3个阶段。

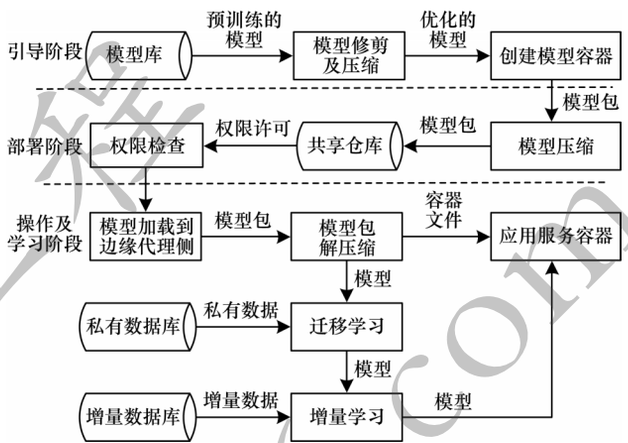


图1　AI+边缘智能框架的主要工作阶段设计

Fig.1　Main working stage design of AI + edge intelligence framework

该框架主要工作阶段相关设计说明如下:

1)引导阶段:在部署到边缘节点之前创建基于容器的微服务,并将用户选择的模型与其运行环境打包成一个docker容器,即“模型包”,以进行后续的部署。引导阶段可应用模型优化技术来减小模型,启用适用于模型和目标边缘节点的机器学习平台(如TensorFlow、PyTorch、Keras),并分配访问权限策略以控制边缘设备之间的协作[22]。

2)部署阶段:模型包被压缩并加载到共享存储库中,具有权限的边缘节点可以访问该存储库。收到部署请求后,将模型包下发到相应的边缘节点。该阶段可采用模型压缩和修剪技术降低网络成本。

3)操作及学习阶段:边缘代理首先解压模型包并使用相应的docker文件创建容器,容器在启动后使用边缘代理私有数据重新训练模型。使用本地数据重新训练可使模型更符合实际场景,并提高其推理准确性[23]。本框架具备较好的灵活扩展性,当边缘环境及场景发生较大变化时,可通过最新的数据使用增量学习来更新模型。

为提高预训练模型的准确性,需在云和边缘之间共享更多数据,但这使得边缘层面的数据存在安全及隐私风险。本文提出了一种隐私保护机制,将存储在边缘设备中的数据分为公共数据和私有数据[24],其存储策略设计如图2所示。

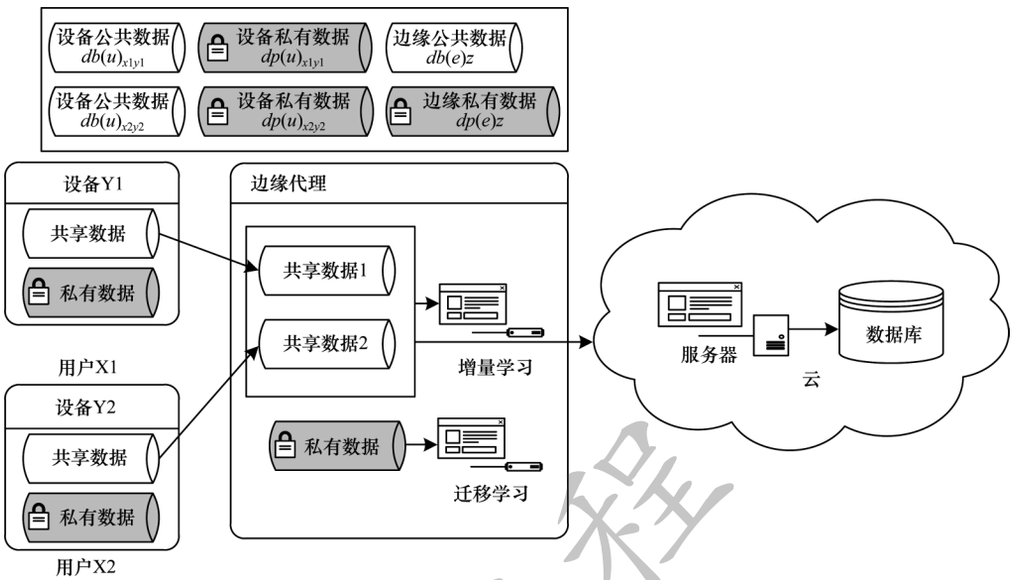


图2 AI+边缘智能框架的存储策略设计

Fig.2 Storage strategy design of AI + edge intelligence framework

框架存储策略主要设计如下：

假定某用户x，物联网设备y对该用户有数据访问权限并接入到边缘设备z，存储在y中x的数据分为公共部分$db(u)_{xy}$和私有部分$dp(u)_{xy}$。由于物联网设备存在存储资源限制，$db(u)_{xy}$数据发送到相应的边缘设备之前，在y上进行短时存储，时间为T_s，而$dp(u)_{xy}$被锁定在物联网设备中进行安全保护。在时间t时，存储在y中x的数据可定义如下：

$$D_{xy}=db(u)_{xy}\left|t-t_0\right|<T_s\cup dp(u)_{xy} \tag{1}$$

其中：t_0是数据上传至边缘设备的最新时间。同样地，将边缘设备z从IoT设备收集的数据分为公共部分$db(e)_z$和私有部分$dp(e)_z$，$db(e)_z$和$db(e)_{xy}$的组合用于增量学习，$dp(e)_z$用于迁移学习。$db(e)_z$和$db(u)_{xy}$临时存储在边缘设备的时间为T_e，之后数据传输到云端。在系统中，需配置$T_e>T_s$以降低数据传输的网络成本。此外，若系统收到数据所有者的认可，$dp(u)_{xy}$可共享给边缘设备。在上传到边缘之前，数据$dp(u)_{xy}$将在本地受到隐私及匿名保护[25]，边缘设备无法从$dp(u)_{xy}$中识别x和y。因此，存储在z中的数据可以表示如下：

$$D(e)_z=db(e)_z\cup\sum_x\sum_{y\in z}db(u)_{xy}\cup dp(e)_z\cup\sum_x\sum_{\hat{y}\in z}dp(u)_{x\hat{y}} \tag{2}$$

其中：$\hat{y}$是与边缘设备z链接并共享私有数据的物联网设备列表。

3.1 模型部署阶段

在使用云端的数据进行预训练后，模型被部署在不同的多个边缘节点上。该部署方式可能会消耗过多的网络资源，并增加系统停机时间而损害系统可靠性。为此，本文提出了一种支持AI模型压缩和解压缩的轻量级模型部署范式，在压缩之前对模型修剪以消除次要因素。部署阶段的设计如图3所示，包括模型修剪和模型压缩两个步骤。

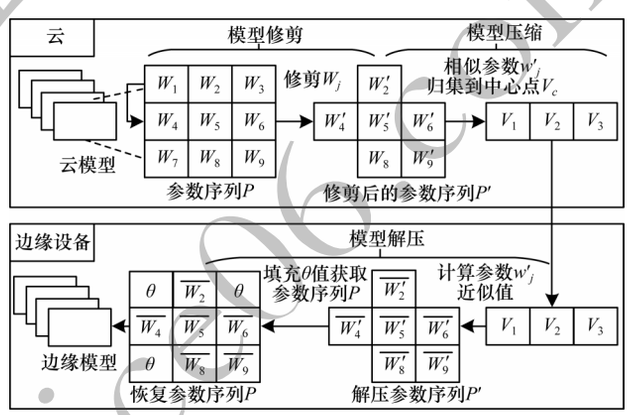


图3 部署阶段的设计

Fig.3 Design of deployment phase

模型部署过程及设计具体如下：

1）模型修剪：模型修剪旨在减少参数数量并去除层之间的冗余连接。给定N层人工智能模型M，参数序列$P=\{w_i|1\leqslant i\leqslant N\}$，压缩比$\lambda$在0~1之间，修剪过程会删除所有在阈值$K$以下的权重，并在此基础上进行修改，计算公式如下：

$$k=\frac{1}{2}\left(P_{\left\lfloor\frac{N+1}{\lambda}\right\rfloor}+P_{\left\lceil\frac{N+1}{\lambda}\right\rceil}\right) \tag{3}$$

其中：$\lfloor*\rfloor$和$\lceil*\rceil$分别表示下限和上限函数。

2）模型压缩：通过压缩模型M可减小其大小。令$P'=\left\{w'_j|1\leqslant j\leqslant N'\right\}$表示经修剪后的参数序列，其中$N'<N$。将所有相似的参数$w'_j\in P'$归集到$C$中心点$V_c$，其中$\lceil C=N/\lambda'\rceil$，这意味着一个或多个$w'_j$被映射到一个中心点$V_c$。该映射被定义为哈希函数$h_j\rightarrow c\in\{0,1\}$，如果$h_j\rightarrow c=1$，则参数$w_j$被映射到$V_c$，中心点$V_c$计算公式如下：

$$v_c=\frac{1}{m}\sum_{j=1}^{N'}w_jh_{j\rightarrow c} \tag{4}$$

其中:m是$h_j \to c=1$的数量。通过上述步骤,减少模型M的层数和维度。

3)模型解压:在模型成功部署到边缘设备后,通过从C中心点V_c恢复参数序列P来重构模型M。首先,通过计算w_i'的近似值来恢复修剪后的参数序列P',表示为$\overline{w_i}'$,其计算公式如下:

$$\overline{w_i}' = v_p h_{j \to c} \tag{5}$$

哈希函数和哈希种子需在云和边缘设备之间预先共享。

然后,通过在P'中低于阈值K的修剪后参数填充一个小的θ值来获得参数序列P,这些参数在后续的增量学习中将会被更新。

上述设计为边缘设备部署AI模型带来了较大便利:一是修剪模型可明显减少其大小,节省了模型部署的带宽消耗和边缘设备的存储空间;二是使用哈希函数压缩模型增加了模型的保密性,若没有哈希函数和哈希种子,攻击者很难恢复原始模型。

3.2　模型学习阶段

随着边缘设备计算能力的增长,模型可处理更多任务,不再仅是通过有效的学习策略进行推理。然而,仅在边缘侧训练AI模型不足以提供高精度的智能服务。为此,本文提出一种协作式边缘学习算法,如图4所示,通过在边缘侧中使用迁移学习和增量学习,通过私有和增量数据迭代地训练并更新模型,解决了边缘设备上对模型训练的性能瓶颈及其动态更新的问题。

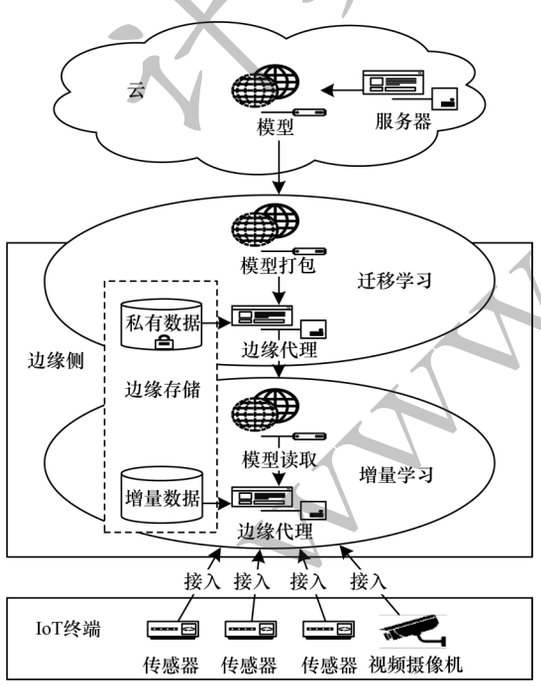


图4　学习阶段的设计

Fig.4　Design of learning phase

本文设计具备两大优点:一是提高了模型精度,满足了边缘个性化需求,仅在云端使用一般数据训练模型无法达到个性化边缘上下文的准确性要求,因此采用存储在边缘代理中的私有数据通过迁移学习来重新训练模型是必要的;二是保证了模型准确性,由于物联网设备存在移动性,从这些设备收集的上下文和数据经常变化,导致部署到边缘的模型不满足后续实际场景[26]。从云端重新训练并部署新模型可能导致设备过载和网络消耗等问题,为此本文提出采用增量学习来使用边缘设备中的传入数据更新模型。主要过程及方法如下:

给定一个具有L层和每层l个神经元的AI模型M。首先使用云上的公共数据集DS(c)对模型M进行预训练。给定学习率γ_1、期望准确度ε_1和迭代次数N_1,预训练过程表示为CT(DS(c),γ_1,ε_1,N_1)。其次使用迁移学习TL($dp(e)$,γ_2,ε_2,N_2)在边缘侧重新训练模型M,其中,$dp(e)$、γ_2、ε_2、N_2分别是边缘侧私有数据、学习率、期望精度和迭代次数。为提高准确性,微调优化被应用到最后一层$w^L=\{w_i^L \mid 1 \leqslant i \leqslant n^L\}$。假设$w^L$层的输入和输出为$x^L$和$\hat{y}^L$,$w^L$的损失函数定义为:

$$\delta L = \left(-\frac{1}{k}\right)\sum_{i=1}^{k}\left[y_i \log_a \hat{y}_i^L + (1-y_i)\log_a\left(1-\hat{y}_i^l\right)\right] \tag{6}$$

其中:y和k分别是M中的标记数据和神经元数量。计算w^L的梯度下降值计算公式如下:

$$\Delta w^L = \left(-\frac{1}{k}\right)\sum_{i=1}^{k} x^L\left(\hat{y}_i^l - y_i\right) \tag{7}$$

w^L的权重被重新定义如下:

$$w^L = w^L - y2\Delta w^L \tag{8}$$

通过使用增量学习,模型M可使用新传入的增量数据进行更新,系统只需使用小批量的最新数据来更新模型即可保持其高可用性[27]。

4　实验与仿真

为准确评估AI+边缘智能框架的准确性和性能,本文以电网配电站房环境监测智能分析和应用场景为例,构建一个仿真模型进行测试,包括终端设备、边缘代理装置和云服务器来模拟配电站房内的环境监测系统。如图5所示,配电站房环境监测智能分析需求有:通过布设传感器及视频监控实现运行状态数据实时采集及站内环境实时监测,将采集数据及视频数据流传输到边缘代理装置,通过深度学习模型自动识别站内是否存在火灾火情、异物侵入等异常情况,模型被同时部署在边缘代理和云服务器上。实验系统采用Raspberry Pi 3 Model B(1.2 GHz CPU和1 GB RAM)模拟终端设备,使用Raspberry Pi4 Model B(1.5 GHz CPU and 4 GB RAM)模拟边缘代理装置,操作系统为Ubuntu 18.04。云服务器配备英特尔酷睿i9-9900k处理器、32 GB RAM和Nvidia RTX 3080 GPU。边缘代理和云服务器通过互联网连接。

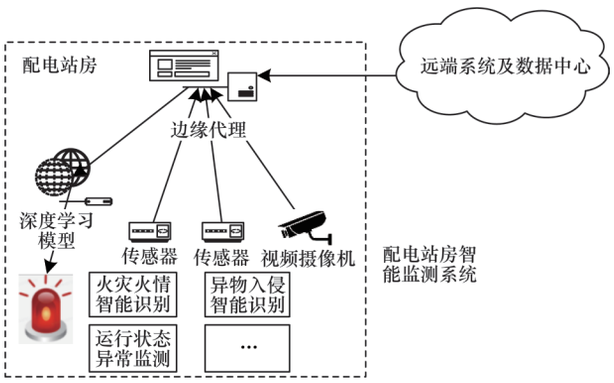


图5 配电站房智能监测系统仿真环境

Fig.5 Simulation environment of intelligent monitoring system for distribution station building

1)准确性评估:为验证边缘侧上使用私有数据集再训练的重要性,研发动物识别模型模拟异物入侵场景。模型由CIFAR-10[2]作为云端公共数据集和Cat/Dog[3]作为边缘侧私有数据集进行训练。公有数据集在云上对模型训练完后,可识别10种动物(如马、鸟、猫、狗等)。但配电站房实际场景只需针对猫和狗进行识别。因此,预训练模型被部署到边缘节点后,需使用本地猫狗图像的专用数据集进行重新训练[25]。实验通过边缘侧私有数据集和公共数据集,分别比较了云模型和边缘模型在预训练及再训练后的识别结果,如图6所示,两个模型的测试准确率在预训练阶段是相等的,准确率大约为21%。对比本地私有数据集进一步训练的结果,边缘模型的质量显着提高,其测试准确率达到88%,而云模型的测试准确率略微提高到61%。测试结果表明,相比传统云AI架构,AI+边缘智能框架在实际应用场景中的模型训练与准确率具有较好的优越性。

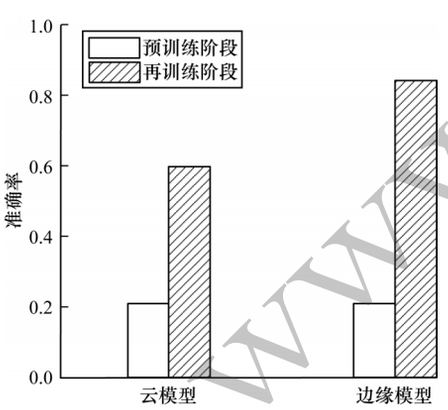


图6 边缘模型和云模型学习阶段的测试准确率比较

Fig.6 Comparison of test accuracy between edge model and cloud model in learning stage

2)延迟性能评估:在边缘和云模式上可采用不同的识别模型(SSD MobileNet V1、SSD MobileNet V2、SSD MobileDet),模型的端到端延迟可通过不同的图像尺寸(1 080p、720p和360p)进行评估。如图7所示,在边缘侧上模型识别的延迟都明显低于云模式,这主要得益于处理组件和数据源之间的距离得以缩短原因。

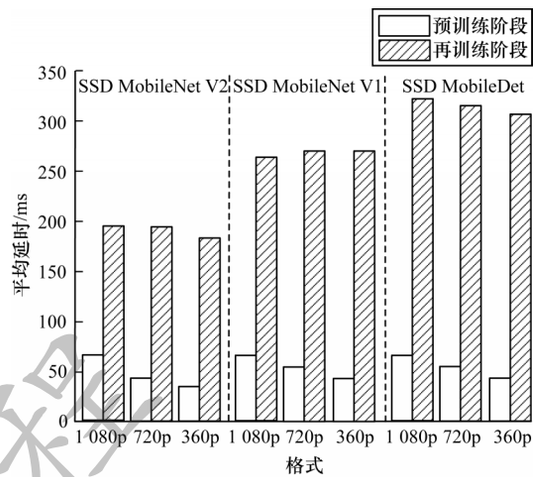


图7 边缘模型和云模型的平均延时

Fig.7 Average delays of edge model and cloud model

实验结果显示,边缘代理在延迟方面的优势对小图像尺寸较为明显,边缘代理和人工智能集成可显著降低端到端延迟,更适用于现场实际应用环境。

3)资源消耗评估:为评估AI+边缘智能框架对边缘设备硬件资源的影响,实验观察了在不同图像尺寸上执行推理任务时各种已部署的物体检测模型的RAM和CPU使用情况。实验结果如图8所示,由于推理任务涉及的计算量比存储量大,无论模型类型和图像大小如何,内存消耗都保持在25%以下。此外,AI+边缘智能框架经过高度优化,可作为微服务在后台运行。在同一个实验中,在边缘代理上处理1 080p图像的CPU消耗在所有模型上约为50%,处理360p图像只需要大约30%的计算能力。实验结果表明,本文框架可完全在硬件资源有限的边缘设备上运行。

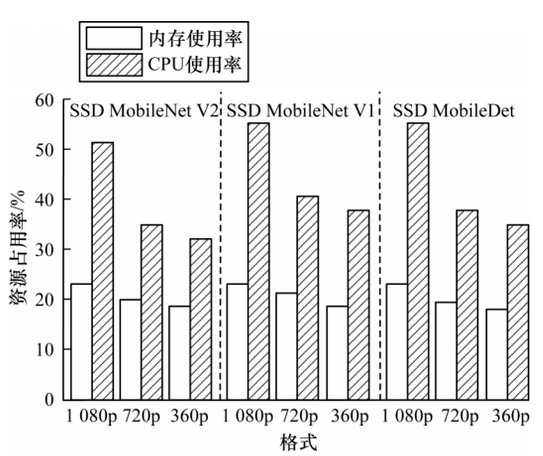


图8 AI+边缘模型和云模型的资源占用分析

Fig.8 Resource occupancy analysis of AI+edge model and cloud model

4)CPU和GPU设备训练时间对比分析:在CPU设备(以Raspberry Pi4为代表)及GPU设备(以

Nvidia Jetson Nano为代表)上训练不同的深度学习模型(MobileNet V1, MobileNet V2, Inception V3),训练周期为一个epoch。实验结果如图9所示。

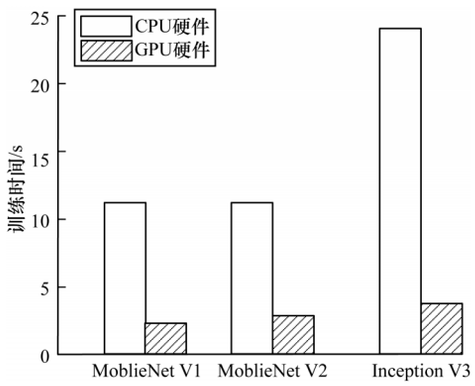
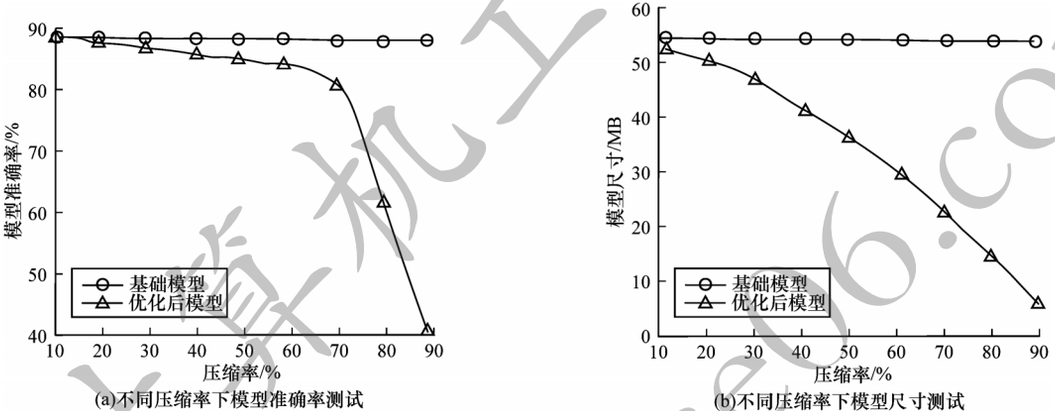


图9　CPU及GPU设备模型训练时间

Fig.9　Training time of CPU and GPU device models

使用GPU设备训练模型的速度比CPU设备快5倍,GPU设备中所有模型的训练时间都在4 s以内,而CPU设备对SSD MobileNet V1、SSD MobileNet V2和Inception V3的训练时间分别为10.2、10.4和24.8 s。实验结果表明,GPU设备可显著加快AI+边缘智能框架的模型学习阶段。

5)模型压缩对比分析:实验分析了在不同压缩比下重建模型和原始模型之间的精度和尺寸的变化。如图10所示,将压缩比设置在0.64以下,模型准确性变化不大,尺寸却明显减小。例如当压缩比为0.64时,原始模型和重建模型的准确率分别约为86%和84%,而模型大小从55 MB减小到22 MB。实验结果表明,AI+边缘智能框架在一定压缩比下能提高准确率的同时,较大地减少了模型尺寸。



(a)不同压缩率下模型准确率测试　　(b)不同压缩率下模型尺寸测试

图10　不同压缩率下的模型准确率与尺寸测试

Fig.10　Model accuracy and size test under different compression rates

5　结束语

本文介绍了AI+边缘代理智能框架,旨在将AI和边缘计算集成到物联网中。首先在边缘层采用迁移学习和增量学习,迁移学习帮助使用边缘侧本地私有数据重新训练AI模型,增量学习使用增量数据迭代训练这些模型。涉及训练过程的边缘数据分为永久存储在边缘中的私有数据和用于在云上训练模型的公共数据,从而在维护数据隐私的同时增强边缘与云的协作能力。其次设计了云压缩和边缘解压机制,以降低模型部署的网络成本,并通过配置压缩比平衡节省的网络成本和模型准确性。最后设计并实现了一个实验仿真的系统原型来评估框架的性能。实验结果表明,该框架不仅提高了AI模型的推理精度,而且降低了延迟和网络成本,实现了边缘代理与人工智能有机融合和集成。下一步将开展更多业务场景的应用研究,对该框架的实用性进行进一步验证,提高其推广价值。

参考文献

[1] 李钦豪,张勇军,陈佳琦,等. 泛在电力物联网发展形态与挑战[J]. 电力系统自动化,2020,44(1):13-22.
LI Q H, ZHANG Y J, CHEN J Q, et al. Development patterns and challe nges of ubiquitous power Internet of Things[J]. Automation of Electric Power Systems, 2020, 44(1): 13-22. (in Chinese)

[2] 施巍松,张星洲,王一帆,等. 边缘计算:现状与展望[J]. 计算机研究与发展,2019,56(1):69-89.
SHI W S, ZHANG X Z, WANG Y F, et al. Edge computing: state-of-the-art and future directions[J]. Journal of Computer Research and Development, 2019, 56(1): 69-89. (in Chinese)

[3] 周知,于帅,陈旭. 边缘智能:边缘计算与人工智能融合的新范式[J]. 大数据,2019,5(2):53-63.
ZHOU Z, YU S, CHEN X. Edge intelligence: a new nexus of edge computing and artificial intelligence[J]. Big Data Research, 2019, 5(2): 53-63. (in Chinese)

[4] WU H, HAN H T, WANG X, et al. Research on artificial intelligence enhancing Internet of things security: a survey[J]. IEEE Access, 2020, 8: 153826-153848.

[5] KHAN W Z, AHMED E, HAKAK S, et al. Edge computing: a survey[J]. Future Generation Computer Systems, 2019, 97: 219-235.

[6] SARWAR MURSHED M G, MURPHY C, HOU D Q, et al. Machine learning at the network edge: a survey[EB/OL]. [2022-08-10]. https://arxiv. org/abs/1908. 00080.

[7] 杨维永,刘苇,崔恒志,等. SG-Edge:电力物联网可信边缘计算框架关键技术[J]. 软件学报,2022,33(2):641-663.
YANG W Y, LIU W, CUI H Z, et al. SG-Edge: key technology of power Internet of things trusted edge computing framework[J]. Journal of Software, 2022, 33(2):641-663. (in Chinese)

[8] DENG S G, ZHAO H L, FANG W J, et al. Edge intelligence: the confluence of edge computing and artificial intelligence[J]. IEEE Internet of Things Journal, 2020, 7(8):7457-7469.

[9] 柏财通,崔翛龙,李爱. 基于本地蒸馏联邦学习的鲁棒语音识别技术[J]. 计算机工程,2022,48(10):103-109.
BAI C T, CUI Y L, LI A. Robust speech recognition technology based on local distillation federated learning[J]. Computer Engineering, 2022, 48(10): 103-109. (in Chinese)

[10] CERUTTI G, PRASAD R, BRUTTI A, et al. Neural network distillation on IoT platforms for sound event detection[C]//Proceedings of ISCA'19. Washington D. C., USA: IEEE Press, 2019: 3609-3613.

[11] MAO Y, YI S, LI Q, et al. A privacy-preserving deep learning approach for face recognition with edge computing[C]//Proceedings of USENIX Workshop on Hot Topics Edge Computing. Washington D. C., USA: IEEE Press, 2018: 1-6.

[12] CHANDAKKAR P S, LI Y K, DING P L K, et al. Strategies for re-training a pruned neural network in an edge computing paradigm[C]//Proceedings of IEEE International Conference on Edge Computing. Washington D. C., USA: IEEE Press, 2017: 244-247.

[13] LEE N, AJANTHAN T, TORR P H S. SNIP: single-shot network pruning based on connection sensitivity[EB/OL]. [2022-08-10]. https://arxiv. org/abs/1810. 02340.

[14] BENGIO Y, LÉONARD N, COURVILLE A. Estimating or propagating gradients through stochastic neurons for conditional computation[EB/OL]. [2022-08-10]. https://arxiv. org/abs/1308. 3432.

[15] BENGIO Y. Deep learning of representations: looking forward[M]. Berlin, Germany: Springer, 2013: 1-37.

[16] BLOT M, PICARD D, CORD M, et al. Gossip training for deep learning[EB/OL]. [2022-08-10]. https://arxiv. org/abs/1611. 09726.

[17] 殷昱煜,叶炳跃,梁婷婷,等. 边缘计算场景下的多层区块链网络模型研究[J]. 计算机学报,2022,45(1):115-134.
YIN Y Y, YE B Y, LIANG T T, et al. Research on multi-layer blockchain network model in edge computing[J]. Chinese Journal of Computers, 2022, 45(1): 115-134. (in Chinese)

[18] KIM H, PARK J, BENNIS M, et al. Blockchained on-device federated learning[J]. IEEE Communications Letters, 2020, 24(6):1279-1283.

[19] WANG Y E, WEI G Y, BROOKS D. Benchmarking TPU, GPU, and CPU platforms for deep learning[EB/OL]. [2022-08-10]. https://arxiv. org/abs/1907. 10701.

[20] LEE J, ESHRAGHIAN J K, CHO K, et al. Adaptive precision CNN accelerator using radix-X parallel connected memristor crossbars[EB/OL]. [2022-08-10]. https://arxiv. org/abs/1906. 09395.

[21] 刘雯,刘欣然,龚舒,等. 边缘计算的电力网关设计及数据传输安全研究[J]. 单片机与嵌入式系统应用,2022,22(4):42-46,51.
LIU W, LIU X R, GONG S, et al. Power gateway design and data transmission security based on edge computing[J]. Microcontrollers & Embedded Systems, 2022, 22(4): 42-46, 51. (in Chinese)

[22] 岑伯维,蔡泽祥,武志刚,等. 电力物联网边缘计算终端的微服务建模与计算资源配置方法[J]. 电力系统自动化,2022,46(5):78-91.
CEN B W, CAI Z X, WU Z G, et al. Microservice modeling and computing resource configuration method for edge computing terminal in electric Internet of things[J]. Automation of Electric Power Systems, 2022, 46(5): 78-91. (in Chinese)

[23] 赵宇峰,雷晟,张国钢,等. 基于容器技术的电力设备仿真云平台设计与开发[J]. 计算机工程,2021,47(9):171-177,184.
ZHAO Y F, LEI S, ZHANG G G, et al. Design and development of container-based cloud platform for power equipment simulation[J]. Computer Engineering, 2021, 47(9): 171-177, 184. (in Chinese)

[24] 曹芷晗,卢煜成,赖思思,等. 基于边缘计算的传感云研究进展[J]. 软件学报,2019,30(11):40-50.
CAO Z H, LU Y C, LAI S S, et al. Research progress of sensing cloud based on edge computing[J]. Journal of Software, 2019, 30(11): 40-50. (in Chinese)

[25] 沙乐天,肖甫,陈伟,等. 面向工业物联网环境下后门隐私泄露感知方法[J]. 软件学报,2018,29(7):1863-1879.
SHA L T, XIAO F, CHEN W, et al. Leakage perception method for backdoor privacy in industry Internet of things environment[J]. Journal of Software, 2018, 29(7): 1863-1879. (in Chinese)

[26] LIN J, YU W, ZHANG N, et al. A survey on Internet of things: architecture, enabling technologies, security and privacy, and applications[J]. IEEE Internet of Things Journal, 2017, 4(5): 1125-1142.

[27] 黄伟楠,朱秋煜,王越,等. 基于典型样本的卷积神经网络增量学习研究[J]. 电子测量技术,2018,41(6):76-80.
HUANG W N, ZHU Q Y, WANG Y, et al. Incremental learning in convolutional neural network based on typical samples[J]. Electronic Measurement Technology, 2018, 41(6): 76-80. (in Chinese)

编辑 索书志