基于节能计算的边缘智能框架设计

杨 清[1] 石昌鑫[1] 冯继凡[1]
YANG Qing SHI Changxin FENG Jifan

摘 要 随着人工智能物联网（AIOT）的发展，网络节点数据不断增多，其数据采集和处理量不断增大，越来越多的数据选择在边缘智能端进行处理。但是目前在智能边缘领域仍缺乏利用智能计算优化 AIOT 能效的边缘智能框架，以减少边缘节点的计算压力，降低节点能耗。因此，重点研究了边缘设备和云服务在处理 AIOT 任务时的能耗，针对边缘智能节点和云任务的能效问题，提出了一种基于节能计算的边缘智能框架。通过实验测试了典型智能边缘和云服务的能耗，结果表明，所提出的优化方法能耗更低，效率更高，明显优于其他方法，具有较高的能效。

关键词 人工智能物联网（AIOT）；能耗；边缘智能；资源调度；节能计算

doi：10.3969/j.issn.1672-9528.2024.05.045

0 引言

作为物联网（IOT）和人工智能（AI）的连接，人工智能物联网（AIOT）不仅能够感知和处理数据 [1]，还可以与其他设备进行通信。这些物联网设备通常配备大量传感器，并在网络边缘实时产生海量数据。然而有限的网络带宽，尤其是在航空航天系统中，数据传输处理的实时性受到很大的限制，无法将这些数据不计成本、花费巨大功耗进行处理。因此，边缘计算通过采用相关人工智能方法将计算能力从集中节点推向网络的逻辑极限边缘，在更靠近数据源的地方实现数据的处理 [2]。

由于 AI 任务计算复杂，通过高性能云服务器采用云辅助的方法，使用 AI 模型处理物联网设备收集的数据，构建 AIOT 服务是目前常见且多数使用的一般方法。然而，由于物联网中传感设备的数量激增，云数据中心与物联网设备之间的距离过远，无论在边缘还是在云端，及时感知处理这些巨大数据的能力差距也在不断加大，严重影响了 AIOT 应用的体验质量（Quality of Experience, QoE）[3]。例如，在监控应用中，传统传感器（如地面雷达和摄像头）、非传统传感器（如物联网）和非有机机载平台（如无人机系统）的数量可以增加探测、跟踪和识别目标以及应对威胁的机会。但是这些系统却缺乏高效和有效的处理能力，其复杂的组织架构和调度策略使得在边缘端很难实现数据的实时处理，导致收集的数据失去了时间敏感性，从而失去利用价值。于是，边缘智能计算通过在边缘部署 AI 加速器，将 AI 过程从云端转移到 AIOT 设备附近 [4]。但是边缘设备在处理人工智能任务时性能和能耗因不同的硬件和任务类型而异，复杂的架构和调度策略使得整体系统的能耗不受控制 [5]，尤其是对于无人机系统，严重影响了其存在意义和相关价值。因此，有必要合理编排从边缘端到云的所有资源，为 AIOT 构建一个节能的智能边缘计算框架。

基于 AIOT 构建节能计算框架主要存在以下挑战：（1）边缘设备处理大量 AI 任务时的可扩展性有限；（2）不同计算硬件（包括边缘设备和云）之间缺乏合作 [6-7]。根据最新的边缘设备性能研究，其计算能力足以处理大多数 AI 任务。然而由于大多数设备都是主从架构，单个设备的可伸缩性非常有限 [8]，很少有产品支持云环境的虚拟化并发处理技术 [9]。所以边缘计算框架无法保证在峰谷负载时间内获得更好的能效。另外，由于边缘设备之间或者边缘设备和云之间的网络通信能力较低，如何在多个边缘设备之间并行或分布式处理人工智能任务成为难题。因此，在智能边缘计算框架中，有必要考虑在处理不同规模的人工智能任务时所涉及的能耗问题 [10]。

在本文中，基于上述挑战，提出了一个多层智能边缘架构。该架构协调轻量级边缘设备、高性能边缘设备和云，以高能效处理不同负载和不同规模的 AI 任务。

1 相关研究

现有的 AIOT 应用主要分为以下四类：（1）智能医疗；（2）智能家居；（3）智慧交通；（4）智慧工业。在智能医疗中，AIOT 已被应用于健康监测和疾病分析。通过部署在相关可穿戴设备中采集日常生活的相关信息，实现医疗诊断 [11]。智能家居主要涉及室内定位、智能控制等相关技术，

1. 航空工业西安航空计算技术研究所 陕西西安 710065

通过无线信号监控或者图像采集，实现对相关物联网设备的操控[12]。在智慧交通中，可以通过大量的监控实时采集交通信息，预测交通流量，给出相关监管建议，从而保证城市交通的通畅。在智慧工业中，自动化制造通常将计算机视觉和传感器相结合，对生产制造进行数据记录，通过对数据进行分析，给出预测结果[13]。

AIOT 已在许多关键领域得到应用，因此有必要寻找一种低功耗、高能效的方式来构建 AIOT 基础框架。而边缘智能计算是未来人工智能物联网应用构建一个稳定、具有可扩展性基础框架的最大机遇。边缘智能计算的第一阶段是将人工智能任务从云端卸载到边缘。然而早期的边缘硬件性能太低，无法直接执行传统的计算型人工智能任务。最简单的方式便是通过对模型进行修剪来降低模型的复杂度，可实现在大幅度降低网络复杂度的情况下依旧保持良好的性能。然而固定的剪枝方法难以适应不同的硬件，根据不同的物联网性能，无法实现自适应调整。

另一种实现边缘智能计算的方法是通过模型分割，将 AI 模型分割成不同部分，并将分割后的部分模型推送到边缘设备进行实时推理。该方法的好处是在不修改模型的情况下减少边缘设备的计算负荷。Li 等人提出一种垂直分区方式来支持卷积运算在低规格边缘设备上运行。然而该方法需要在边缘和云之间进行大量的数据传输，以实现模型的不断更迭，其网络带宽和能耗成为制约效率的最大瓶颈。

为了满足不同边缘设备的功耗和计算能力设定，通过设定多种输出，将不同的人工智能模型用于不同性能的边缘设备。对应低功耗设备，人工智能任务通过输出处理，减少计算负荷。对于高功耗设备，人工智能任务将整个模型进行处理，以获得最佳精度。Teerapittayanon 等人首次提出为本地设备、边缘设备和云服务器提供不同输出的人工智能模型，整体网络推理通过云端进行处理，边缘设备和本地设备只需要单独处理节点数据，最后将处理结果上传到云端进行融合，实现边缘智能计算。但是该方法需要针对不同环境设计不同的人工智能模型，增加了在前期的模型设计成本，无法满足所有的边缘计算情况，普适性较差。因此，需要设计一种低功耗且可满足不同计算能力的物联网智能架构。

2 基于节能计算的边缘智能框架设计

本文通过使用一个典型的边缘网络结构说明在一般网络环境下的多层边缘计算，其边缘网络结构如图 1 所示。首先将网络划分为几个层次，包括 AIOT 设备层、客户边缘层、网络边缘层、接入层和核心网层。基于共识，边缘计算普遍指前四层所涉及的计算。本文将在这四个层面上讨论边缘智能计算的局限性和潜力。

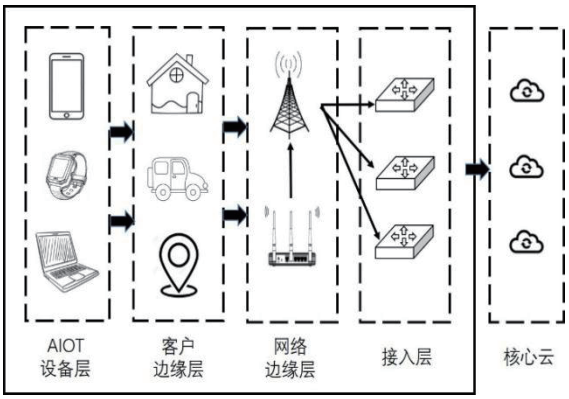

图 1 物联网四层架构图

首先，在边缘计算中存在两个限制：网络连接和硬件规格。在边缘设备进行处理 AI 任务时，其有限的无线连接带宽需承载大多数物联网设备的通信要求。例如，ZigBee 协议能耗低，可连接的设备数量多，因此被用于连接物联网设备。同时，为了降低成本和能耗，大多数边缘节点设备的硬件规格也非常低，并且物联网设备和上层之间没有数据传输，因此可以在物联网设备中支持轻量级 AIoT 应用程序。网络边缘层是指公共网络服务的接入点，在这一级中由于边缘服务器的物理空间和功耗仍然有限，所以依旧无法部署大型服务器。接入层是运营商提供作为网络边缘层和核心云服务之间的连接层，在这一层面客户可以租用不同量级的计算资源来处理一些复杂的人工智能任务，并且将其作为一个小型的数据处理中心。但是由于接入层更加贴近核心云网，距离底层边缘节点较远，因此会受到数据传输的影响，其传输延时和任务处理延时将会影响处理 AI 任务的 QoE。

根据边缘智能计算的四个层次，本文设计了一个节能框架，在降低能耗的同时保证处理 AI 任务的 QoE。由图 2 所示，边缘智能框架由边缘节点调度程序、任务管理器、人工智能处理接口和管理接口组成。

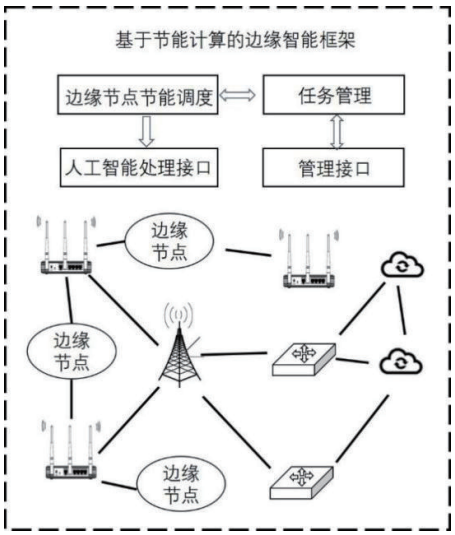

图 2 基于节能计算的边缘智能框架

该框架管理从客户边缘层到接入层的不同边缘节点。使用边缘节点调度器记录网络中所有的边缘节点信息和状态，包括工作负载、硬件处理能力和软件设置。同时执行资源调度策略，寻找最优边缘节点处理即将到来的人工智能任务。

任务管理器模块监控所有边缘节点上的任务，包括执行时间、边缘节点 ID、任务 ID、资源消耗等。边缘节点调度程序可以根据任务管理器模块提供信息调度 AI 任务。同时，任务管理器模块控制任务和边缘节点的状态，使任务或边缘节点从一种状态切换到另一种状态。

AI 处理接口模块为 AI 模型存储提供统一的命令集。由于不同的边缘节点和 AI 平台处理 AI 任务的命令不同，因此 AI 处理接口模块可以将与设备相关或与平台相关的接口转移到单个模块的统一接口。

与 AI 处理接口模块类似，管理接口模块也为上层模块提供统一的边缘节点控制接口。任务管理器模块将边缘节点 ID 或任务 ID 和操作发送给管理接口模块后，所有操作将被转移到指定边缘节点的直接命令中。

同时，本文还在框架中添加了 AI 模型管理，以提高 AIoT 应用程序部署的 QoE。由于边缘硬件的类型很多，客户需要针对不同的接口开发特定于硬件的模型。在提出的框架中，增加了一个模块来存储边缘环境中不同硬件的预训练 AI 模型。同时，在框架中还部署了模型转换器，将用户模型转换为特定于硬件的模型，以提高能效。

通过本文所提出的边缘智能框架设计，将物联网进行合理划分，并根据其不同作用进行不同功耗的限定，解耦了边缘智能计算中的彼此依赖，提升了框架内不同边缘节点之间的扩展性，协调轻量级边缘设备、高性能边缘设备和云，以高能效处理不同负载和不同规模的 AI 任务。

3 边缘智能框架设计性能的实验评估

首先介绍用于测试边缘设备功耗的实验测试平台。本文所使用的测试平台由以下几种边缘设备组成。在基于节能计算的边缘智能框架中，Jetson AGX Xavier 设备部署在接入网层，Jetson Xavier NX 设备部署在网络边缘层，Jetson Nano 设备部署在客户边缘层。本文使用 NVIDIA RTX 4090 显卡作为服务器来显示云端的性能和能耗。

本文测试了 YOLOV3 和 VGG16 两种深度学习模型的功耗和平均处理时间。在这些测试中，边缘设备从 COCO 2017 数据集中随机选取 50 幅图像，记录上述两个模型的模型加载和处理时间，还使用 YOLOV3 测试了所有的边缘设备对视频处理的速度。视频数据为 80 s 视频，分辨率为 1280×720。整个边缘智能框架所开发使用的编程环境是 Python3.8.1，其中模型转换使用 TensorRT 7.1.3 库和 CUDA 10.2 版本。

如图 3 所示，测试了四种边缘设备在处理人工智能任务和空闲状态下的功耗。其中网络边缘层的功耗和客户边缘层的功耗相接近，但是处理能力方面，前者比后者更加强大。从实验结果发现，其服务器的功耗远远大于所有的边缘节点。同理，服务器拥有的处理能力也远大于所有的边缘节点。

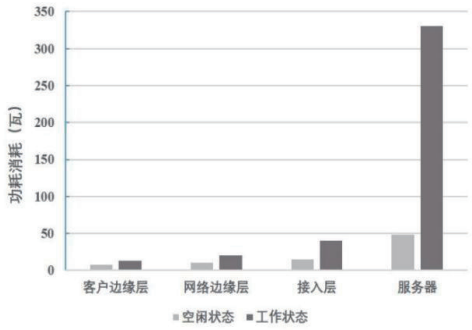

图 3 不同设备在不同状态下的功耗表现

图 4 是边缘智能框架处理模型的推理延时结果。在处理边缘设备中的人工智能任务之前，需要将预训练好的或者通过 TensorRT 转换后的固化模型加载到边缘节点。从图 4 所示的结果来看，加载两个 AI 模型的处理延迟较为接近。由于服务器的显卡计算能力最高，所以其加载人工智能模型的速度最快，与边缘设备相比用时最短。

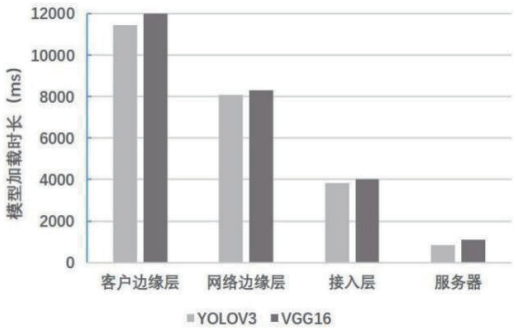

图 4 不同设备模型加载时长

从图 5 可知，模型处理时长与模型加载时间互不相同。作为边缘设备中处理能力最高的接入层设备，其 YOLOv3 TensorRT 模型的处理时间约为 225 ms。经过 TensorRT 优化后，接入层和网络边缘层之间的性能差异很小。而客户边缘层的性能优化表现也十分优秀，YOLOv3 在边缘节点上处理所需的时间接近接入层中模型的 3 倍，VGG16 的处理时长也同上述类似。从测试平台的功耗和性能测试来看，处理同样的任务，由于 RTX 服务器的功耗是边缘客户端节点设备功耗的 30 多倍，而其处理任务所需的时长却相差 5 倍，可以发现边缘设备的能效远远高于云端硬件。由此可知，当智能边缘计算框架的性能足够好时，其能耗与云平台相比大幅度下降。边缘节点进行分布式处理后，其性能远大于云端硬件，而功耗却远低于云平台。

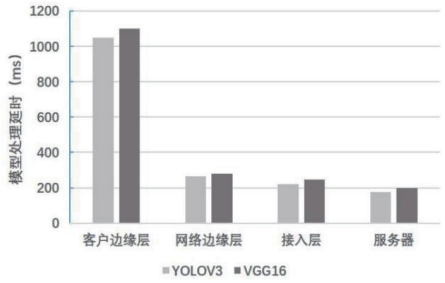

图 5 模型处理延时

图 6 展示了演示系统的视频处理速度结果。RTX 服务器以近 130 帧 /s 的速度超越了其他设备，而接入层的轻量级边缘设备节点也可以在 1 s 内处理 100 多个视频帧，同样可满足其视频处理性能要求，表现出本文所提出的边缘智能框架的性能的优越性。

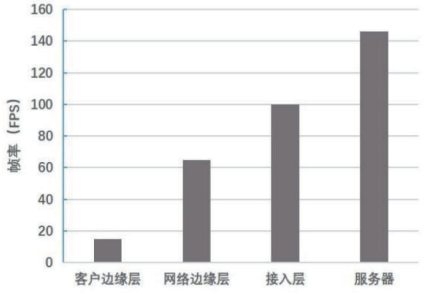

图 6 视频处理速度

4 结语

本文提出了一个基于节能计算的边缘智能计算框架，该框架利用了边缘节点的计算能力，通过优化计算分布，降低节点能耗。通过对物联网进行合理划分，根据边缘节点所处的层级进行功耗的限定，解耦了边缘智能计算中的彼此依赖，提升了框架内边缘节点之间的扩展性，协调轻量级边缘设备、高性能边缘设备和云，提高了系统的可用性和可扩展性。通过实验验证，所提出的框架可在满足实际任务处理的前提下减少功耗，在大多数商用边缘设备上处理人工智能任务时可展现良好的性能，具有较高的商用价值。

参考文献：

[1]CHIU T, SHIH Y, PANG A, et al. Semisupervised distributed learning with non-IID data for AIoT service platform[J].IEEE internet things, 2020,7(10):9266-9277.

[2]MUNIR K P, KHAN S.IFCIoT: integrated fog cloud IoT: a novel architectural paradigm for the future internet of things[J].IEEE consum. electron, 2017,6(3):74-82.

[3]XU J, OTA K, DONG M. Saving energy on the edge: in-memory caching for multi-tier heterogeneous networks[J]. IEEE commun., 2018,56(5):102-107.

[4]LI H, OTA K, DONG M.Deep reinforcement scheduling for mobile crowdsensing in fog computing[J]. ACM trans. internet technol, 2019,19(2):1-18.

[5]MOHAMMAD M, AL-FUQAHA A, SOROU S, et al. Deep learning for IoT big data and streaming analytics: a survey[J]. IEEE commun. surveys tuts, 2018,20(4):2923-2960.

[6]CUI H, ZHANG H, GANGER G, et al.GeePS: scalable deep learning on distributed GPUs with a GPU-specialized parameter server[C]// 11th Eur. Conf. Comput. Syst.New York:ACM, 2016:1-16.

[7]BLASCH E, CRUISE R, NATARAJAN S, et al. Control diffusion of information collection for situation understanding using boosting MLNs[C]// 21st Int. Conf. Inf. Fusion. Piscataway:IEEE, 2018:1407-1414.

[8]GUPTA K, STUART J, OWENS J.A study of persistent threads style GPU programming for GPGPU workloads[C]//2012 Innovative parallel computing.Piscataway:IEEE, 2012:1-14.

[9]HONG C, SPENCE I, NIKOLOPOULOS D. GPU virtualization and scheduling methods: a comprehensive survey[J].ACM Comput. Surveys, 2018,50(3):1-37.

[10]CHETLUR C, WOOLLEY C, VANDERMERSCH P, et al.CuDNN: efficient primitives for deep learning[EB/OL]. (2014-10-03)[2024-01-25].https://arxiv.org/abs/1410.0759.

[11]ZHU J, PANDE A, MOHAPATRA P, et al.Using deep learning for energy expenditure estimation with wearable sensors[C]// 2015 17th International conference on e-health networking, application & services. Piscataway:IEEE,2015:501-506.

[12]HANNUN A, RAJPURKAR P, HAGHPANAHI M, et al.Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network[J].Nature medicine, 2019, 25(1): 65-69.

[13]EROL B, MAJUMDAR A, LWOWSKI J, et al.Improved deep neural network object tracking system for applications in home robotics[C]// Computational Intelligence for Pattern Recognition. Cham:Springer,2018:369-395.

【作者简介】

杨清（1998—），男，陕西榆林人，硕士，助理工程师，研究方向：轻量级物联网设计。

石昌鑫（1998—），男，陕西渭南人，硕士，助理工程师，研究方向：人工智能。

冯继凡（1999—），男，陕西咸阳人，硕士，助理工程师，研究方向：车载物联网设计。

（收稿日期：2024-02-27）