

一种自适应的分布式深度神经网络推理框架

吴宾宾, 杨桂松*

上海理工大学光电信息与计算机工程学院, 上海

收稿日期: 2024年6月17日; 录用日期: 2024年7月10日; 发布日期: 2024年7月17日

摘要

近年来, 随着深度学习的发展, 深度神经网络(Deep Neural Network, DNN)模型变得越来越复杂, 所需的内存和数据传输量也随之增大, 这不仅降低了DNN的训练和推理速度, 也限制了DNN在一些内存较小、计算能力较差的物联网(Internet of Things, IoT)设备上的部署。现有研究将基于云-边-端协同的分布式计算框架与深度神经网络相结合, 组成了分布式深度神经网络(Distributed Deep Neural Network, DDNN)框架, 该框架在IoT应用场景下有着显著的优势。然而, DDNN框架存在设备的计算能力有限、以及设备之间的传输成本较高等问题。针对上述问题, 本文提出了自适应的分布式深度神经网络(Adaptive Distributed Deep Neural Network, ADA-DDNN)推理框架。ADA-DDNN框架采用了多个边缘出口, 这些边缘出口允许ADA-DDNN框架中的模型在不同的深度层次上进行自适应地推理, 以适应不同的任务需求和数据特性。此外, 该框架增加了额外的边缘处理模块, 边缘处理模块可以在边缘端进行特征融合之前, 判断每个终端模块的输出结果是否可信, 若可信, 则直接输出分类结果, 无需进行特征融合和后续计算。这大大增加了样本的边缘出口概率, 减少了后续的计算成本。本文在开放的CIFAR-10数据集上进行验证, 实验结果表明, ADA-DDNN框架在保证云端测试精度的前提下, 显著提升了边缘测试精度。

关键词

深度神经网络, 物联网, 分布式深度神经网络, 特征融合

An Adaptive Distributed Deep Neural Network Inference Framework

Binbin Wu, Guisong Yang*

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Jun. 17th, 2024; accepted: Jul. 10th, 2024; published: Jul. 17th, 2024

*通讯作者。

文章引用: 吴宾宾, 杨桂松. 一种自适应的分布式深度神经网络推理框架[J]. 建模与仿真, 2024, 13(4): 4449-4459.
DOI: 10.12677/mos.2024.134402

Abstract

In recent years, the development of deep learning has led to an increase in the complexity of deep neural network (DNN) models, accompanied by a proportional increase in the amount of memory and data transfer required. This has a detrimental effect on the training and inference speed of DNNs, as well as limiting their deployment on some Internet of Things (IoT) devices with limited memory and computational capabilities. Existing research combines a distributed computing framework based on cloud-edge-end collaboration with deep neural networks to form the Distributed Deep Neural Network (DDNN) framework, which has significant advantages in IoT application scenarios. However, the DDNN framework suffers from the problems of limited computational power of devices, as well as high transmission cost between devices. To address these issues, this paper proposes the Adaptive Distributed Deep Neural Network (ADA-DDNN) inference framework. The ADA-DDNN framework employs multiple edge exits, which allow the models in the ADA-DDNN framework to perform different depth levels of adaptive reasoning to accommodate different task requirements and data characteristics. Furthermore, the framework incorporates an additional edge processing module, which is responsible for evaluating the credibility of the output results produced by each terminal module. If the results are deemed credible, the classification results are directly outputted without feature fusion and subsequent computation. This significantly increases the probability of the sample being outputted at the edge and reduces the subsequent computation cost. The paper validates the ADD-DDNN framework on the open CIFAR-10 dataset. The experimental results demonstrate that the framework significantly improves edge testing accuracy while maintaining the same level of testing accuracy in the cloud.

Keywords

Deep Neural Network, Internet of Things, Distributed Deep Neural Network, Feature Fusion

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

深度神经网络(Deep Neural Network, DNN)已经成为了人工智能(Artificial intelligence, AI)的基础性技术。DNN 通过从大量数据中自动学习复杂的特征表示,已经在计算机视觉[1]-[3]、自然语言处理[4]、语音识别[5]-[7]等多个领域表现出独特的优势。与此同时,包括物联网(Internet of Things, IoT)设备在内的终端设备数量急剧增加。这些设备是机器学习应用程序的诱人目标,因为它们通常直接连接到传感器,这些传感器以流方式捕获大量输入数据。然而,随着 DNN 模型结构也越来越复杂,所需的内存和数据传输量也随之增大,这不仅降低了 DNN 的训练和推理速度,也限制了 DNN 在一些内存较小、计算能力较差的物联网(Internet of Things, IoT)设备上的部署。例如,将 IoT 设备收集的数据传输到云端的大型神经网络模型进行处理,不仅会产生通信成本,还可能导致高延迟和隐私泄露问题。为了在 IoT 设备上实现高效的人工智能任务,可以采用决策树、支持向量机等简单机器学习模型。这些模型对设备的内存和计算能力要求较低,易于部署在边缘设备上。然而,这种方法可能会牺牲系统的精度,其次,直接在 IoT 设备上部署日益复杂的 DNN 模型也会受到内存和算力的限制。为了解决这一矛盾,可以采用分布式深度神经网络(Distributed Deep Neural Network, DDNN)框架。在 DDNN 框架中,DNN 的不同部分被部署在不同

层次的计算设备上。例如, 原始数据可以在 IoT 设备上进行处理或特征提取, 然后再上传到边缘服务器或云服务器进行深度学习模型的推理。这种框架对于多终端的协作任务具有显著优势, 包括减少时延、保护用户数据隐私和提高系统的可扩展性。然而, 现有的 DDNN 框架存在设备的计算能力有限、以及设备之间的传输成本较高等问题。针对上述问题, 本文提出了自适应的分布式深度神经网络(Adaptive Distributed Deep Neural Network, ADA-DDNN)推理框架。ADA-DDNN 框架采用了多个边缘出口, 这些边缘出口允许 ADA-DDNN 框架中的模型在不同的深度层次上进行自适应地推理, 以适应不同的任务需求和数据特性。大大增加了样本的边缘出口概率, 减少了后续的计算成本。通过 ADA-DDNN 模型, 可以在保持高精度的同时, 有效利用边缘计算和云计算资源, 从而对 IoT 设备收集的海量数据进行快速处理和智能分析。

2. 相关工作

随着深度学习的不断发展, DNN 模型结构也越来越复杂, 性能较低的 IoT 设备很难对规模庞大的深度学习模型进行训练和计算推理, 针对这一难题, 已经出现了大量的研究工作。其中的重点是提高深度神经网络训练和推理的效率[8]。例如, 为了在边缘节点进行快速和低功耗的网络计算推理, 文献[9]提出了深度神经网络压缩和结构优化等方法, 其目的是减少神经网络模型的规模和计算量, 以便在有限的资源下实现更快的训练和推理速度。有研究显示, 可以在网络的推理阶段, 制定合理的策略动态的选择要执行的深度神经网络模型的层数, 在使用尽量少的神经网络模块进行推理计算的同时, 保持模型较高预测精度[10][11]。2017 年, Leroux 等人提出了一种新的神经网络架构, 称为级联神经网络[12], 该架构通过在除了最后一个隐藏层之外的隐藏层边添加额外的输出层, 实现了输出结果能够提前退出的推理机制。这种设计允许网络在任一层级的推理中, 一旦输出符合分类阈值, 即可输出结果, 从而减少了终端设备的通信成本和模型的平均推理时间。

此外, 可以将分布式技术和 DNN 进行融合[13][14]。DNN 结合分布式架构, 将神经网络进行了划分, 分层映射到不同的网络节点中[15]-[17]。2017 年, 哈佛大学 Teerapittayanon 等人提出了 BranchyNet [18]。BranchyNet 在级联分布类器的基础上增加了多个出口。与级联神经网络相比, 它能够在不同出口分支之前根据需求部署更多的卷积层, 以进行更深层次的特征提取。在 BranchyNet 的基础上, Teerapittayanon 等人进一步提出了 DDNN 框架[19]。该框架采用了多层计算结构, 由云端、边缘端和终端组成, 其中云端作为计算框架的最高层, 终端设备作为最低层, 可以根据实际的计算需求增加数量。在云端和终端设备之间, 还可以根据需要增加边缘端设备, 以满足不同的计算需求。

现有的 DDNN 框架如图 1 所示。DDNN 将训练好的 DNN 映射到分布在终端、边缘端和云端的异构物理设备上。DDNN 可以由多个终端设备、边缘端和云端设备组成。这些终端设备可能分布不同的地理位置上, 通过协同工作以做出分类决策。每个终端设备执行本地计算, 但不做分类输出, 它们需要将输出特征统一上传至边缘端, 进行特征融合, 然后对融合后的特征进行分类计算, 如果边缘端对终端设备融合后的特征的分类结果足够自信, 即可从边缘端出口输出结果。否则, 将融合后的特征上传至云端, 云端进行深层的 DNN 推理, 并做出最终的分类决策。

尽管将不同终端所采集的特征上传至边缘端进行融合, 可以形成一个具有更加丰富信息的特征向量, 为边缘端模型提供了更全面的信息。但由于将所有终端的输出特征进行融合, 也会增加模型的输入维度, 需要更多的计算资源, 而且由于部分终端位置较差, 所采集的特征可能掩盖核心位置的终端设备所提取的特征, 从而对模型的输出结果进行干扰。针对上述问题, 本文提出了自适应的分布式深度神经网络(Adaptive Distributed Deep Neural Network, ADA-DDNN)推理框架。该框架通过增加额外的边缘处理和边缘出口, 该出口可以在边缘端进行特征融合之前, 判断每个终端模块的输出是否可信, 如果可信, 则直

接输出分类结果, 无需进行特征融合和后续计算。这大大增加了边缘出口概率, 减少了后续的计算成本和通信成本。

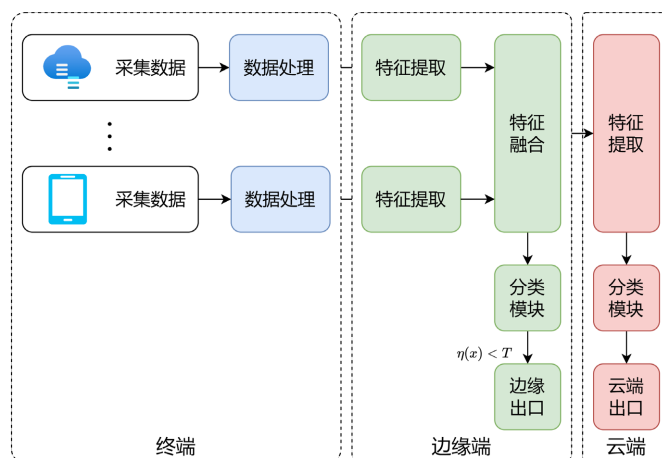


Figure 1. The Existing Inference Framework of DDNN

图 1. 现有的 DDNN 推理框架

3. 系统框架

3.1. ADA-DDNN 中的模型训练

虽然 ADA-DDNN 推理分布在分布式计算层次结构上, 但 ADA-DDNN 框架中所使用的模型可以在单个功能强大的服务器或云中进行训练。与传统的集中式深度神经网络不同, ADA-DDNN 框架采用了多个退出点, 这些退出点允许模型在不同的深度层次上进行自适应的推理, 在反向传播时, 通过组合计算不同出口的损失, 以达到联合优化整个网络, 实现对整个模型的协同训练。本文以 Softmax 交叉熵作为损失函数来介绍 ADA-DDNN 中模型的训练过程。每个退出点的损失函数如式(1)所示:

$$L(y', y; \theta) = -\frac{1}{|C|} \sum_{c \in C} y_c \log y'_c \quad (1)$$

其中真实标签向量 y 通过独热编码表示, 输入样本为 x , 所有可能标签的集合为 C 。为了完成分布式深度神经网络的训练, 需要构建一个联合优化问题, 其目标是 minimized 网络中所有设备出口点的损失函数的加权和。这个问题可以通过联合考虑网络中所有子网络的损失来实现, 从而协调整个网络的训练过程。具体计算如式(2)所示:

$$\mathcal{L}_{\text{total}}(y', y; \theta) = \sum_{m=1}^M w_m L(y'_{\text{exit}_m}, y; \theta_m) \quad (2)$$

其中, M 表示网络的划分层数, 即网络中退出点的总数, w_m 代表位于第 m 个退出节点前的子网络的参数权重。通过这种方式, 分布式深度神经网络可以同时优化所有子网络的性能, 每个子网络的损失函数都乘以相应的权重, 以反映其对整体网络性能的贡献。这种方法允许网络在不同的层次上进行自适应的调整, 以适应不同的任务需求和数据特性。联合损失目标函数的构建, 确保了在反向传播过程中, 每个子网络的梯度都被计算并用于更新网络参数, 从而实现了整个网络的端到端训练。

3.2. ADA-DDNN 推理框架

本文提出的 ADA-DDNN 推理框架如图 2 所示, 该框架可划分为三个主要部分: 终端、边缘端以及

云端。

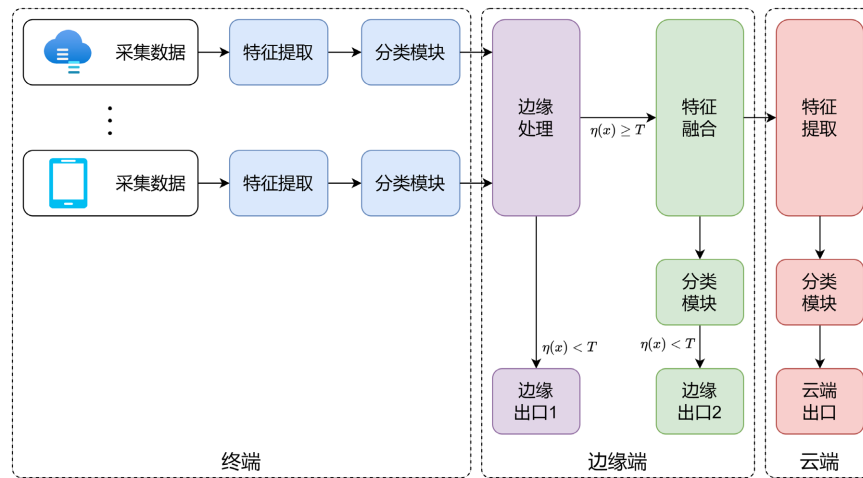


Figure 2. The Inference Framework of ADA-DDNN

图 2. ADA-DDNN 的推理框架

终端：在终端部分，部署了 n 台终端设备，它们承担着从各自视角的特征提取及分类的任务。每个终端设备由两个模块组成，即特征提取模块和分类模块。特征提取模块主要负责当前视角采集的原始图片的浅层特征，提取的特征一般更加简洁。分类模块则负责对这些特征进行分类，并进一步计算分类结果的归一化熵值。随后，终端设备将计算得到的熵值、分类结果以及特征提取模块输出的特征一并传输至边缘端。

边缘端：边缘端由五个核心组件构成：边缘处理模块、边缘出口 1、特征融合模块、分类模块以及边缘出口 2。与现有的 DDNN 框架相比，ADA-DDNN 框架采用了多个边缘出口，这些边缘出口允许 ADA-DDNN 框架中的模型在不同的深度层次上进行自适应的推理，以适应不同的任务需求和数据特性。此外，该框架增加了额外的边缘处理模块，边缘处理模块负责聚合终端设备上传的熵值、分类结果以及终端部分上传的特征。边缘处理模块计算所有终端上传的熵值的最小值，并将该熵值与退出点阈值 T 进行比较，如果该熵值小于退出点阈值，则将该熵值对应的分类结果通过本文所设计的边缘出口 1 输出，否则边缘处理模块将每个终端设备上传的输出特征进行特征融合以及后续计算。特征融合模块负责将融合后的特征进行更深层次的特征提取，生成更加复杂的特征。更深层次的特征为后续的分类提供了更为精准的输入。分类模块负责对深层次特征进行分类。如果分类结果的置信度足够高，即归一化熵值小于退出点阈值，那么样本分类结果将会从边缘出口 2 输出。如果分类的置信度不够高，那么特征融合器会将融合的特征上传到云端做进一步的处理。

云端：云端由三个模块构成，即特征提取模块、分类模块和云端出口模块。其中，特征提取模块负责对边缘端融合的特征做进一步的特征提取以获得具有更强表达能力的深层特征。深层特征通过分类模块进行最终分类，并从云端出口输出最终分类结果。

3.3. ADA-DDNN 推理流程

ADA-DDNN 中的详细推理分多个阶段执行，每个阶段退出点有一个退出阈值 T 作为样本预测的置信度量，当样本预测的置信度小于退出阈值 T 时，推理过程提前退出。定义 T 的一种方法是在测试集上搜索 T 的范围，并选择精度最高的一个。本文使用归一化熵阈值作为置信标准，确定是否在特定出口点

对样本进行分类(退出)。归一化熵定义为:

$$\eta(\mathbf{X}) = -\sum_{i=1}^{|C|} \frac{x_i \log x_i}{\log |C|} \quad (3)$$

其中 C 是所有可能标签的集合, $\mathbf{x}$ 是一个概率向量。这个归一化熵的值在 0 到 1 之间, 这使得解释和搜索相应的阈值 T 更容易。例如, $\eta(\mathbf{X})$ 接近 0 意味着 ADA-DDNN 对样本的预测有信心; $\eta(\mathbf{X})$ 接近 1 表示不可信。在每个出口点, 计算 $\eta(\mathbf{X})$ 并与 T 比较, 以确定样品是否应在该点出口。在给定的出口点, 如果 $\eta(\mathbf{X}) > T$, 系统会进一步进行特征处理, 并将分类结果上传到更高的出口点, 直到到达云端最后一个出口, 该出口始终执行分类。

4. 实验评估

4.1. 实验环境

在本文中, 所有实验均在单一服务器上进行, 具体的硬件配置详见表 1。在软件方面, 实验采用了 Python 编程语言, 版本为 3.8.0, 确保代码的兼容性。

Table 1. Experimental setup

表 1. 实验设置

参数	参数取值
操作系统	Ubuntu 20.04
开发语言	Python 3.8.0
CUDA	11.3
CPU	Xeon(R) Platinum 8358P
GPU	RTX 3090(24G)

4.2. 参数设置

本实验采用 CIFAR-10 数据集作为实验数据集, CIFAR-10 是一个包含日常物品的彩色图像数据集, 由 Hinton 的学生 Alex Krizhevsky 和 Ilya Sutskever 整理, 旨在识别广泛的物体类别。整个数据集由 50000 张训练图像和 10000 张测试图像构成[20]。在实验设计阶段, 鉴于设备的内存和计算能力有限, 本文所设计的特征提取模块较为简单, 主要由四个残差层(Reslayer)构成, 每个 Reslayer 层均使用了 3×3 的卷积核, 步长和边缘填充均设置为 1。这样的设计旨在通过适当增加网络的深度来提取更高层次的特征, 同时保持模型的轻量化。本实验的其它训练参数如表 2 所示。

Table 2. Training parameters

表 2. 训练参数

名称	设置
优化器	随机梯度下降
初始学习率	0.1
学习率调度	余弦退火
批大小	200
训练轮数	180

4.3. 结果分析

4.3.1. 训练损失

本文主要从三个方面评估 ADA-DDNN 框架的性能, 分别是 ADA-DDNN 框架中模型训练损失、测试精度以及框架的容错性。训练损失衡量的是模型预测结果与实际结果之间的差异。理想情况下, 随着训练过程的进行, 损失值应该逐渐减小, 这意味着模型正在逐步学习和适应训练数据, 提高其分类准确性。训练损失分为边缘训练损失和云端训练损失, 本文以 Softmax 交叉熵损失函数作为优化目标。其中边缘训练损失是网络中所有边缘出口训练损失的加权和, 具体计算如式(4)所示。

$$\mathcal{L}_b(y', y; \theta) = \sum_{b=1}^B w_b L(y'_{\text{exit}_b}, y; \theta_b) \quad (4)$$

其中, B 表示边缘端网络的划分层数, 即边缘端退出点的总数, w_b 代表位于边缘端第 b 个退出节点前的子网络的参数权重。通过这种方式, 可以评估所有边缘端模型的收敛性能。云端出口的训练损失是网络中所有云端出口训练损失的加权和, 具体计算如式(5)所示。其中, D 表示云端退出点的总数, w_d 代表位于云端第 d 个退出节点前的子网络的参数权重。

$$\mathcal{L}_d(y', y; \theta) = \sum_{d=1}^D w_d L(y'_{\text{exit}_d}, y; \theta_d) \quad (5)$$

模型的训练损失如图 3 所示。由图可见, 随着训练的深入, ADA-DDNN 边缘出口的训练损失稳步下降, 最终接近 0.45; 而云端出口的损失也在逐渐下降, 最终趋近于 0.17。算法 1 的边缘出口的训练损失稳步下降, 最终趋近于 1.01; 云端出口的训练损失同样呈现下降趋势, 最终接近 0.18。这种云端与边缘出口之间的损失差异较为显著, 造成这个结果主要是由于云端具有深层的特征提取模块, 提取了数据的深层特征, 提高了模型的表达能力。其次, ADA-DDNN 框架的边缘训练损失明显低于算法 1 的边缘训练损失, 而 ADA-DDNN 和算法 1 的云端训练损失没有明显差异, 其主要原因是 ADA-DDNN 采用了额外的边缘处理模块和边缘出口 1, 增强了边缘端模型的处理能力。

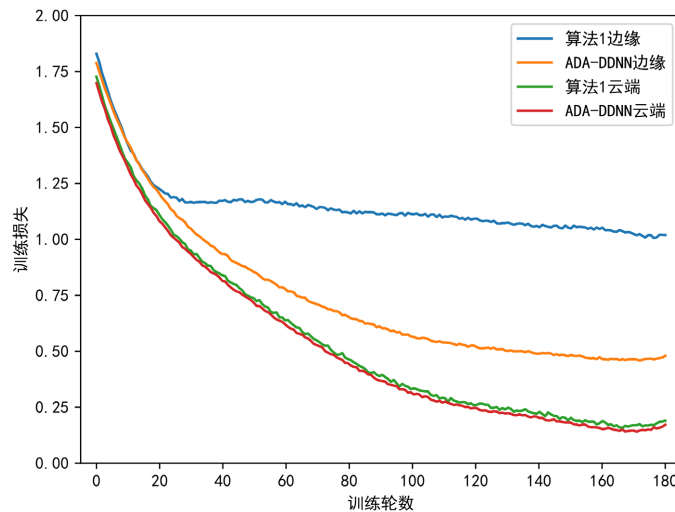


Figure 3. Training loss
图 3. 训练损失

4.3.2. 测试精度

为了验证本文算法的优越性, 进一步在相同的测试集上评估了 ADA-DDNN 和算法 1 的测试精度。

测试精度分为边缘测试精度和云端测试精度。其中边缘测试精度具体计算如式(6)所示。

$$ACC_b = \frac{P_b}{N_b} \quad (6)$$

其中 P_b 表示从边缘输出的所有正确分类的样本数, N_b 表示边缘端输出的总样本数。云端测试精度如式(7)所示。

$$ACC_d = \frac{P_d}{N_d} \quad (7)$$

其中 P_d 表示从云端输出的所有正确分类的样本数, N_d 表示云端输出的总样本数。

测试精度曲线如图 4。由图可见, ADA-DDNN 边缘测试精度稳步上升, 最终接近 84.70%, 云端测试精度也在逐渐上升, 最终趋近于 86.62%。算法 1 的边缘测试精度最终趋近于 67.41%; 云端测试精度最终趋近于 86.57%。因此, ADA-DDNN 的边缘测试精度明显高于算法 1 的测试精度。进一步验证了 ADA-DDNN 的增加边缘处理模块和边缘出口 1 的优越性。

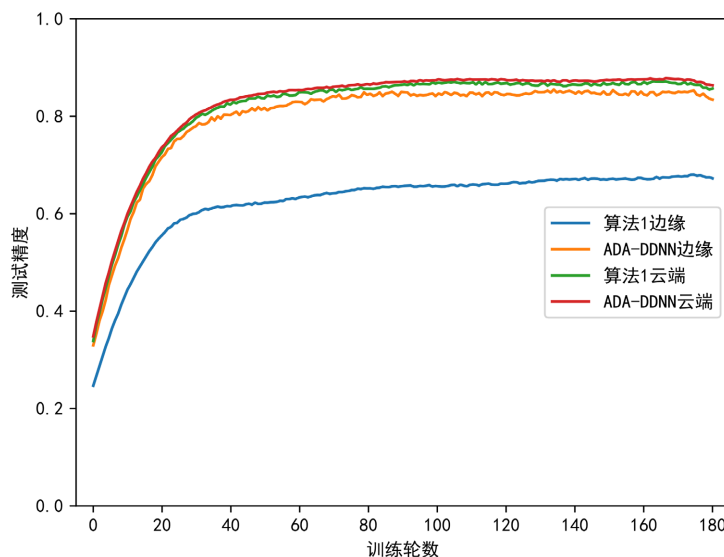


Figure 4. Test accuracy

图 4. 测试精度

4.3.3. 容错性分析

分布式深度神经网络相较于传统的集中式神经网络, 在鲁棒性方面展现出显著的优势。这种优势主要得益于其分布式架构, 该架构允许网络在部分终端设备发生故障或停止工作时, 依然能够维持较高的准确率。为了评估 ADA-DDNN 框架的容错能力, 本文通过模拟实验来观察六个终端设备分别停止工作以及当多个终端设备无法工作时, ADA-DDNN 的边缘测试精度和云端测试精度变化情况。

图 5 和图 6 分别展示了 ADA-DDNN 框架在单一设备缺失和多个设备缺失情况下的准确率。如图 5 所示, 展示的是当六个终端设备有一个发生缺失时, 测试精度的变化情况。横坐标表示的是缺失设备的编号, 从 1 到 6。纵坐标表示的是在相应设备丢失时的测试精度。根据图中的数据, 可以注意到, 当一个终端设备缺失时, 边缘测试精度和云端测试精度均没有太大影响。这表明 ADA-DDNN 框架具备相对较好的单台设备缺失的容错能力。

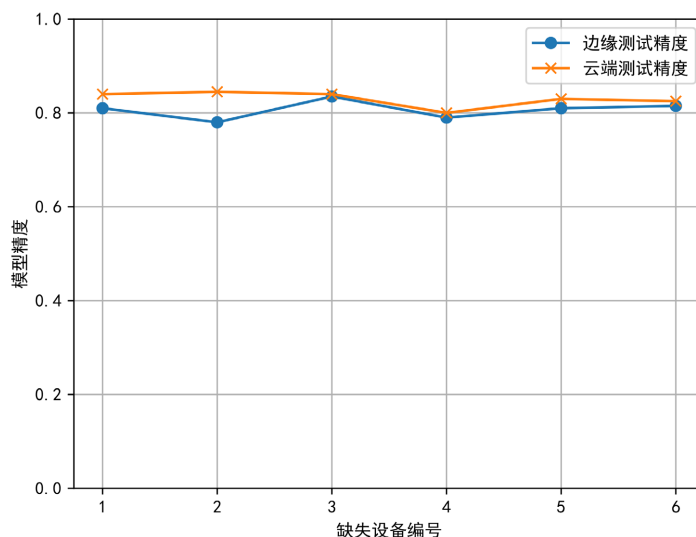


Figure 5. The test accuracy after the loss of a device

图 5. 一台设备缺失后的测试精度

如图 6 所示, 模拟的是多台终端设备缺失时, ADA-DDNN 测试精度的变化趋势。需要说明的是, 缺失的设备是随机的。其中, 横坐标代表的是缺失设备总数量, 从 1 到 6。纵坐标代表的是这些设备缺失时的测试精度。当网络中缺失两台设备以内时, 准确率稍有下降。并且当缺失更多设备时, 准确率会显著下降。实验表明 ADA-DDNN 框架具备一定程度的容错能力, 能够在接受范围内的设备故障情况下保持较高的准确率。但是, 为了保证 ADA-DDNN 的精度不发生较大幅度下降, 需要尽量减少终端设备的缺失。

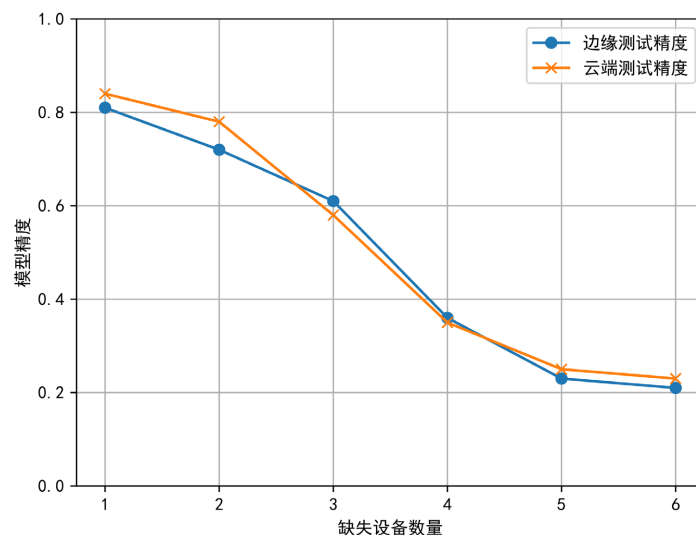


Figure 6. The test accuracy after the loss of multiple devices

图 6. 多台设备缺失后的测试精度

5. 总结

本文提出了 ADA-DDNN 推理框架。与现有的 DDNN 框架相比, 该框架增加了额外的边缘处理模块

和边缘出口, 边缘处理模块通过聚合所有终端上传的分类结果和输入特征, 并在边缘端进行特征融合之前, 判断每个终端模块的输出是否可信, 如果可信, 则直接从本文增加的边缘出口输出分类结果, 无需进行特征融合和后续计算。这大大增加了边缘出口概率, 减少了后续的计算成本。本文在开放的 CIFAR-10 数据集上进行验证, 实验结果表明, ADA-DDNN 在保证云端测试精度的前提下, 显著提升了边缘测试精度。最后, 通过模拟终端设备缺失情况, 验证了 ADA-DDNN 框架的高容错性。尽管 ADA-DDNN 框架已经在多个方面显示出优越性, 但本文认为在未来的工作中仍有进一步探索和优化的空间。例如, 如何根据设备的计算资源, 更加精细化地分配任务, 以及如何对设备之间传输的信息进行压缩, 进一步降低通信开销, 都是未来值得深入研究的方向。

基金项目

国家自然科学基金资助项目(61602305, 61802257); 上海市自然科学基金资助项目(18ZR1426000, 19ZR1477600)。

参考文献

- [1] Khan, A., Sohail, A., Zahoor, U. and Qureshi, A.S. (2020) A Survey of the Recent Architectures of Deep Convolutional Neural Networks. *Artificial Intelligence Review*, **53**, 5455-5516. <https://doi.org/10.1007/s10462-020-09825-6>
- [2] 郑远攀, 李广阳, 李晔. 深度学习在图像识别中的应用研究综述[J]. 计算机工程与应用, 2019, 55(12): 20-36.
- [3] 圣文顺, 孙艳文. 卷积神经网络在图像识别中的应用[J]. 软件工程, 2019, 22(2): 13-16.
- [4] Lauriola, I., Lavelli, A. and Aiolli, F. (2022) An Introduction to Deep Learning in Natural Language Processing: Models, Techniques, and Tools. *Neurocomputing*, **470**, 443-456. <https://doi.org/10.1016/j.neucom.2021.05.103>
- [5] Hema, C. and Garcia Marquez, F.P. (2023) Emotional Speech Recognition Using CNN and Deep Learning Techniques. *Applied Acoustics*, **211**, Article ID: 109492. <https://doi.org/10.1016/j.apacoust.2023.109492>
- [6] Li, J. (2022) Recent Advances in End-to-End Automatic Speech Recognition. *APSIPA Transactions on Signal and Information Processing*, **11**, e8. <https://doi.org/10.1561/116.00000050>
- [7] 张瑞珍, 韩跃平, 张晓通. 基于深度 LSTM 的端到端的语音识别[J]. 中北大学学报: 自然科学版, 2020, 41(3): 244-248.
- [8] Qi, C., Shen, S., Li, R., Zhao, Z., Liu, Q., Liang, J., et al. (2021) An Efficient Pruning Scheme of Deep Neural Networks for Internet of Things Applications. *EURASIP Journal on Advances in Signal Processing*, **2021**, Article No. 21. <https://doi.org/10.1186/s13634-021-00744-4>
- [9] Xu, H., Ho, C.Y., Abdelmoniem, A.M., et al. (2020) Compressed Communication for Distributed Deep Learning: Survey and Quantitative Evaluation. <http://hdl.handle.net/10754/662495>
- [10] Ren, J., He, Y., Yu, G. and Li, G.Y. (2019) Joint Communication and Computation Resource Allocation for Cloud-Edge Collaborative System. 2019 *IEEE Wireless Communications and Networking Conference (WCNC)*, Marrakesh, 15-18 April 2019, 1-6. <https://doi.org/10.1109/wcnc.2019.8885877>
- [11] Heigold, G., Vanhoucke, V., Senior, A., Nguyen, P., Ranzato, M., Devin, M., et al. (2013) Multilingual Acoustic Models Using Distributed Deep Neural Networks. 2013 *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, 26-31 May 2013, 8619-8623. <https://doi.org/10.1109/icassp.2013.6639348>
- [12] Leroux, S., Bohez, S., De Coninck, E., Verbelen, T., Vankeirsbilck, B., Simoens, P., et al. (2017) The Cascading Neural Network: Building the Internet of Smart Things. *Knowledge and Information Systems*, **52**, 791-814. <https://doi.org/10.1007/s10115-017-1029-1>
- [13] Ding, C., Zhou, A., Liu, Y., Chang, R.N., Hsu, C. and Wang, S. (2022) A Cloud-Edge Collaboration Framework for Cognitive Service. *IEEE Transactions on Cloud Computing*, **10**, 1489-1499. <https://doi.org/10.1109/tcc.2020.2997008>
- [14] Chen, J. and Ran, X. (2019) Deep Learning with Edge Computing: A Review. *Proceedings of the IEEE*, **107**, 1655-1674. <https://doi.org/10.1109/jproc.2019.2921977>
- [15] Ali, M., Anjum, A., Yaseen, M.U., Zamani, A.R., Balouek-Thomert, D., Rana, O., et al. (2018) Edge Enhanced Deep Learning System for Large-Scale Video Stream Analytics. 2018 *IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)*, Washington DC, 1-3 May 2018, 1-10. <https://doi.org/10.1109/cfec.2018.8358733>
- [16] Ongati, F. and Muchemi, D.E. (2019) Big Data Intelligence Using Distributed Deep Neural Networks. arXiv:

1909.02873.

- [17] Yang, S., Zhang, Z., Zhao, C., Song, X., Guo, S. and Li, H. (2022) CNNPC: End-Edge-Cloud Collaborative CNN Inference with Joint Model Partition and Compression. *IEEE Transactions on Parallel and Distributed Systems*, **33**, 4039-4056. <https://doi.org/10.1109/tpds.2022.3177782>
- [18] Teerapittayanon, S., McDanel, B. and Kung, H.T. (2016) BranchyNet: Fast Inference via Early Exiting from Deep Neural Networks. 2016 *23rd International Conference on Pattern Recognition (ICPR)*, Cancun, 4-8 December 2016, 2464-2469. <https://doi.org/10.1109/icpr.2016.7900006>
- [19] Teerapittayanon, S., McDanel, B. and Kung, H.T. (2017) Distributed Deep Neural Networks over the Cloud, the Edge and End Devices. 2017 *IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, Atlanta, 5-8 June 2017, 328-339. <https://doi.org/10.1109/icdcs.2017.226>
- [20] Krizhevsky, A. and Hinton, G. (2009) Learning Multiple Layers of Features from Tiny Images. *Handbook of Systemic Autoimmune Diseases*, 1. <https://api.semanticscholar.org/CorpusID:18268744>