吴宾宾,杨桂松. 基于注意力机制融合特征的分布式深度神经网络推理框架[J]. 智能计算机与应用,2026,16(3):206-213.
DOI:10.20169/j.issn.2095-2163.24051601

基于注意力机制融合特征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 200093)

摘 要: 在现有的分布式深度神经网络(Distributed Deep Neural Network, DDNN)框架中,设备之间的传输成本较高,并且没有考虑不同终端设备所提取的特征对全局任务的重要程度,使得终端设备将提取的所有特征上传至边缘端进行计算,从而增加了不必要的通信开销和计算成本。针对这些问题,本文将基于注意力机制的特征融合方法应用于 DDNN 框架中,提出了基于注意力机制的 DDNN 框架(ATT-DDNN)。该框架采用了多个边缘出口,这些边缘出口允许 ATT-DDNN 框架中的深度学习(Deep Learning, DL)模型在不同的深度层次上进行自适应推理,以适应不同的任务需求和数据特性。并通过注意力机制为不同终端计算特征的重要程度,显式地建模通道之间的关系,同时自适应地重新标定特征图中的通道,提高了模型对重要特征的响应能力,抑制负面特征的响应能力,从而降低了通信成本和计算成本。实验结果表明,与现有的 DDNN 相比,ATT-DDNN 在保证了较高准确度的前提下,显著降低了设备之间的通信消耗。

关键词: 分布式深度神经网络; 注意力机制; 特征融合; 深度学习
中图分类号: TP393 **文献标志码**: A **文章编号**: 2095-2163(2026)03-0206-08

A distributed deep neural network inference framework based on attention for feature fusion

WU Binbin, YANG Guisong
(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: In the existing Distributed Deep Neural Network (DDNN) frameworks, the cost of communication between devices is relatively high, and the importance of the features extracted by different end devices for the global task is not considered. This leads to end devices uploading all the extracted features to the edge for computation, thereby increasing unnecessary communication overhead and computational costs. To address these issues, this paper applies an attention-based feature fusion method to the DDNN framework and proposes an ATTention-based Distributed Deep Neural Network (ATT-DDNN) framework. This framework employs multiple edge exits, allowing the Deep Learning (DL) models within the ATT-DDNN framework to perform adaptive inference at different depth levels to meet various task requirements and data characteristics. By leveraging the attention mechanism, it explicitly models the relationships between channels and adaptively re-scales the channels in the feature map, enhancing the model's response to important features and suppressing the response to negative features. This approach reduces communication and computation costs. The experimental results demonstrate that, in comparison to the existing DDNN, the ATT-DDNN significantly reduces the communication consumption between devices while maintaining a high level of accuracy.

Key words: distributed deep neural detwork; attention; feature fusion; deep learning

0 引 言

近年来,深度学习(Deep Learning, DL)在计算机视觉[1-3]、自然语言处理[4]和语音识别[5-7]等领域得到了广泛应用。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任务目标,因其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,随着深度学习的不断发展,深度神经网络(Deep Neural Network, 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

作者简介: 吴宾宾(1996—),男,硕士研究生,主要研究方向:深度学习,分布式计算。Email:973571898@qq.com;杨桂松(1982—),男,博士,副教授,硕士生导师,主要研究方向:物联网,边缘计算等。

收稿日期: 2024-05-16

题，陆续开展了一系列的研究工作。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，其目的是减少深度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地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 等学者[12]提出了一种新的神经网络架构，称为级联神经网络，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。

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]。2016 年，Teerapittayanon 等学者[18]提出了 BranchyNet。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，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 等学者[19]进一步提出了 DDNN 框架（Distributed Deep Neural Network，DDNN）。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。此外，Li 等学者[20]提出了一种自调整神经网络量化框架，用于协同推理。该框架能够自动调整以帮助开发人员确定最合适的神经网络分区。采用云端协同推理，可以减少移动设备上的推理存储需求，并在保持较高的准确度的同时，保护个人隐私信息。Mao 等学者[21]提出了 MoDNN，这是一个分布式移动计算系统，通过将预训练的深度神经网络模型分别部署到多个移动终端设备上，以此来加速深度神经网络的运算，降低运算成本和通信开销。Zhao 等学者[22]提出了 Deep Things，这是一个轻量级的框架，用于自适应分布式推理系统。Deep Things 能够在内存和计算能力受限的终端和边缘设备之间进行自适应的卷积神经网络的部署和推理，优化了资源使用效率。

现有的 DDNN 框架如图 1 所示。DDNN 将训练好的 DNN 模型映射到分布在终端、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在不同的地理位置上，通过协同工作以做出分类决策。

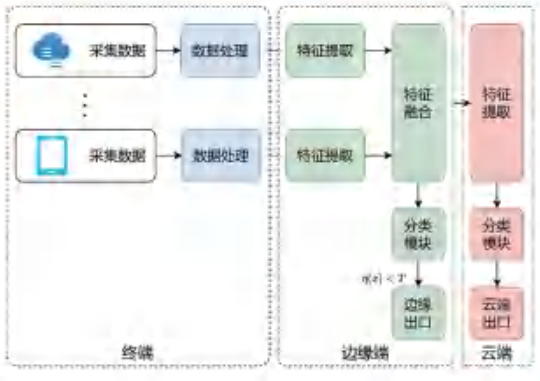

图 1　现有的 DDNN 推理框架

Fig. 1　The existing inference framework of DDNN

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，由于不同终端的部署位置和视觉角度不同，导致对同一图像或物体的输入信息重要性存在差异。现有的 DDNN 框架在执行特征融合时，并没有区分特征对全局任务的重要程度。这导致不必要的通信开销和计算成本。针对上述问题，本文提出了基于注意力机制的分布式深度神经网络（ATTention-based Distributed Deep Neural Network based on Attention，ATT-DDNN）推理框架。该框架通过增加额外的边缘处理和边缘出口，提升了边缘出口概率，减少了后续的计算成本和通信成本。此外，将基于注意力机制的特征融合方法应用于 DDNN 框架中，通过注意力机制为不同终端计算特征的重要程度，显式地建模通道之间的关系，并自适应地重新标定特征图中的通道，提高了终端模型对重要特征的响应能力，抑制负面特征的响应能力，从而进一步降低了通信成本和计算成本。

1　系统框架

1.1　ATT-DDNN 中的模型训练

虽然 ATT-DDNN 推理分布在分布式计算层次结构上，但 ATT-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，ATT-DDNN 采用了多个退出点，这些退出点允许网络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,实现对整个网络的协同训练[23]。本文以 Softmax 交叉熵损失函数来介绍 ATT-DDNN 的训练过程。目标损失函数定义为:

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

其中,

$$y = \mathrm{Softmax}(z) = \frac{\exp(z)}{\sum_{c\in C}\exp(z_c)} \tag{2}$$

此外可得:

$$z = f_{\mathrm{exit}_n}(x;\theta) \tag{3}$$

其中,f_{exit_n} 表示表示神经网络各层从入口点到第 n 个出口分支的计算的函数,θ 表示各层的权重和偏差等网络参数。为了训练 ATT-DDNN,本文形成了一个联合优化问题,即最小化每个出口损失函数的加权和,由此可得:

$$L_{\mathrm{total}}(y',y;\theta) = \sum_{m=1}^{M} w_m L(y'_{\mathrm{exit}_m},y;\theta) \tag{4}$$

其中,m 表示网络的划分层数、即网络中退出点的总数,w_m 表示位于第 m 个退出节点前的子网络的参数权重。通过这种方式,ATT-DDNN 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。

1.2 ATT-DDNN 推理框架

DDNN 通常有多个不同的终端,必须对每个终端设备的输出进行汇总,以便进行分类。由于各个终端的部署位置和视觉角度的不同,导致对同一图像或物体的输入信息重要性存在差异。这种差异性意味着在特征提取后,不同特征对于全局推理任务的贡献程度是不同的。现有的 DDNN 在执行特征融合时,并没有区分特征对全局任务的重要程度。这导致不必要的通信开销和计算成本。为了提高效率并减少成本,本文提出的 ATT-DDNN 框架采用了 Attention 特征融合方法。该方法通过注意力机制为特征计算不同通道的重要程度,显式地建模通道之间的关系,并自适应地重新标定特征图中的通道,提高了模型对重要特征的响应能力,抑制了模型对负面特征的响应程度,从而在保证模型较高准确度前提下,显著降低设备之间的通信成本和计算成本。

1.2.1 ATT-DDNN 推理框架设计

本文提出的 ATT-DDNN 推理框架如图 2 所示,该框架主要可分为 3 个部分:终端、边缘端以及云端。

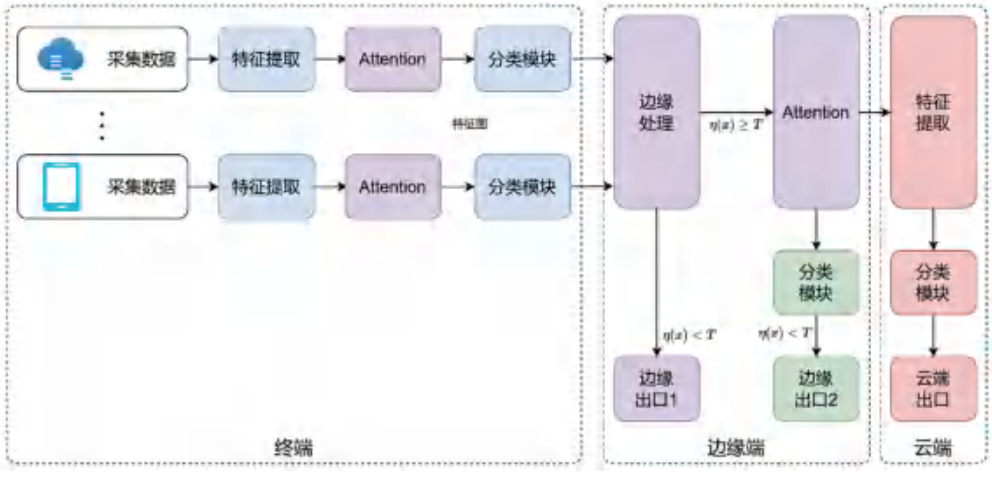

图 2 ATT-DDNN 推理框架

Fig. 2 The inference framework of ATT-DDNN

(1)终端:在终端部分,部署了 n 台终端设备,承担着从各自视角的特征提取及进行分类的任务。每个终端设备由 3 个模块组成,即特征提取模块、Attention 模块和分类模块。特征提取模块的主要任务是从当前视角采集的原始图片中提取浅层特征。这些特征具有更加简洁的表达,并且捕捉到了输入图片的基本信息。本文在终端部分新增了 Attention 模块,该模块是将每个终端的特征提取模块输出的浅层特征通过注意力机制进行特征融合。通过计算每个特征不同通道的重要程度,显式地建模了通道之间的关系,并自适应地重新标定特征图中的通道权重,从而提高模型对重要特征的响应能力。分类

模块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。该模块将特征映射到相应的类别，并衡量分类结果的置信度。随后，终端部分将计算得到的熵值、分类结果以及 Attention 模块的输入特征一起上传至边缘部分，以供后续处理和决策。

(2)边缘端：边缘端主要由 5 个核心组件构成，包括边缘处理模块、边缘出口 1、边缘出口 2、Attention 模块和分类模块。与现有的 DDNN 框架相比，ATT-DDNN 框架采用了多个边缘出口，这些边缘出口允许 ATT-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，ATT-DDNN 框架在边缘端也采用了 Attention 模块作为特征融合方法。Attention 模块通过融合不同终端视角的特征，为特征的不同通道计算重要程度，显式地建模通道之间的关系，并自适应地重新标定特征图中的通道，提高了模型对重要特征的响应能力，抑制了模型对负面特征的响应程度。该方法在保证模型较高准确度前提下，显著降低设备之间的通信成本和计算成本。分类模块负责对 Attention 模块融合后的特征进行分类。如果分类结果的置信度足够高，那么样本分类结果将会从本文所增加的边缘出口 2 输出；否则，Attention 模块会将融合的特征上传到云端做进一步处理。

(3)云端：云端主要由特征提取、分类和云端出口三个模块组成。其中，特征提取模块主要对边缘端融合的特征做深层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。

1.2.2　Attention 模块

与现有的 DDNN 框架相比，本文受 SENET 启发，提出的 ATT-DDNN 框架采用了 Attention 特征融合方法[24]。该方法在保证模型较高准确度前提下，显著降低设备之间的通信成本和计算成本。Attention 模块结构如图 3 所示，该方法融合特征的具体步骤如下。

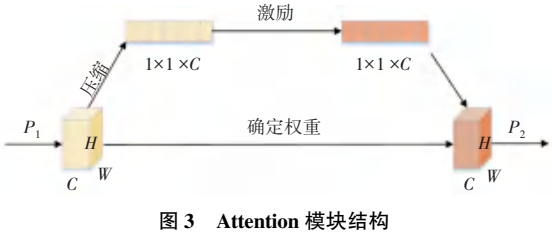

图 3　Attention 模块结构

Fig. 3　Attention module structure

(1)对输入的特征图通过最大池化法，将特征层的尺寸进行压缩，但留下通道维度的信息，压缩函数为：

$$F_{sq}(\boldsymbol{P}_C)=\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\boldsymbol{P}_C(i,j) \tag{5}$$

其中，$\boldsymbol{P}_C$ 为 P_1 的特征；压缩过程将 $H\times W\times C$ 的 P_1 特征图转化为 $1\times 1\times C$ 的特征向量 $F_{sq}(\boldsymbol{P}_C)$。

(2)在特征压缩后，对该特征进行重新标定，标定过程如下：

$$S=\sigma(g(F_{sq}(\boldsymbol{P}_C),Q))=\sigma(\boldsymbol{W}_2\delta(\boldsymbol{W}_1(F_{sq}(\boldsymbol{P}_C)))) \tag{6}$$

其中，$\boldsymbol{W}_1$ 和 $\boldsymbol{W}_2$ 分别表示与特定特征通道相关联的权重参数矩阵；$\delta(\cdot)$ 表示 ReLU 激活函数；$\sigma(\cdot)$ 表示 Sigmoid 激活函数。在激励阶段，通过 2 个连接层实现。第一个全连接层负责将特征从 C 个通道压缩至 $\frac{C}{r}$ 个通道，即式(6)中的 $\boldsymbol{W}_1(F_{sq}(\boldsymbol{P}_C))$ 部分，其中 r 表示压缩比例；其后面连接 ReLU 函数获取非线性变换，第二个全连接层则将压缩后的特征重新扩展到 C 个通道，即 $\boldsymbol{W}_2\delta(\boldsymbol{W}_1(F_{sq}(\boldsymbol{P}_C)))$ 部分，然后通过 Sigmoid 激活函数将输出权重映射到 0 和 1 之间，并将输出权重集合记为 S，权重越低，则在后续特征融合中的重要性也就越低。如果终端将这一部分的特征直接传输到边缘端中进行融合并不会对预测结果产生明显影响，反而会增加终端和边缘端之间的通信成本、以及后续的计算成本。因此，可以设定一个重要性阈值 μ，特征重要性权重超过 μ 的特征会参与后续的特征融合，特征重要性权重低于 μ 的特征不会参与后续的特征融合。通过调节 μ 的值，可以在预测精度与通信和计算成本之间做出一个合理的权衡。通过引入重要性阈值 μ，可以进一步得到参与融合的特征融合权重 S_C，融合特征权重 S_C 的计算如下：

$$S_C=\max(S-\mu,0) \tag{7}$$

(3)将得到的参与融合的 S_C 与 $\boldsymbol{P}_1$ 最原始的特征 $\boldsymbol{P}_C$ 相乘，并输出一个新的特征 $\boldsymbol{P}_2$。整个过程通过显式地建模通道之间的互相依赖关系，自适应地重新校正通道的特征响应。对此可以表示为：

$$\boldsymbol{P}_2=\boldsymbol{P}_C\cdot S_C \tag{8}$$

1.3　ATT-DDNN 推理流程

ATT-DDNN 框架的推理分多个阶段执行，每个退出点有一个退出阈值 T 作为样本预测的置信度度量。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:

$$\eta(\boldsymbol{X}) = -\sum_{i=1}^{|C|} \frac{x_i \log x_i}{\log |C|} \tag{9}$$

其中,C 表示所有可能标签的集合,$\boldsymbol{X}$ 表示一个概率向量。这个归一化熵 η 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,$\eta(\boldsymbol{X})$ 接近 0 意味着 ATT-DDNN 对样本的分类结果有信心;$\eta(\boldsymbol{X})$ 接近 1 表示不可信。在每个出口点,计算 $\eta(\boldsymbol{X})$ 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,如果分类模块对结果没有信心,系统会回落到层次结构中更高的出口点,直到到达最后一个出口,该出口始终执行分类。具体的 ATT-DDNN 推理流程如下。

输入 原始图片

输出 图片类别

1. 初始化重要性阈值 μ 和出口阈值 T
2. 终端设备采集同一目标的不同视角图片
3. 终端设备提取图片特征 x_1
4. Attention 模块计算 x_1 不同通道的特征权重 S
5. 根据权重阈值 μ 选择较高重要性的权重 S_C
6. 特征权重 S_C 与原始特征 x_1 进行加权融合得到特征 x_2
7. 分类模块输出分类结果 y、熵值 $\eta(x)$ 和融合后的特征 x_3
8. 边缘处理模块聚合所有终端的上传的特征得到集合 X、熵集合 N
9. if min(N) < T then
10. 边缘出口 1 输出分类结果 y_1
11. else
12. Attention 模块进一步对特征进行融合并计算处理得到 $\boldsymbol{X}_1$
13. 计算 Attention 模块融合后特征的熵值 n
14. if n < T
15. 边缘出口 2 输出分类结果 y_2
16. else
17. 云端输出分类结果 y_3
18. end if
19. end if

2 实验评估

2.1 数据集介绍

本文实验采用 CIFAR-10 数据集作为实验数据集[25]。CIFAR-10 是一个包含日常物品的彩色图像数据集,旨在识别广泛的物体类别。该数据集涵盖了 10 个类别,包括飞机、汽车、鸟类、猫、鹿、狗、蛙类、马、船和卡车,每个类别包含 6 000 张 32×32 像素的 RGB 彩色图像。整个数据集由 50 000 张训练图像和 10 000 张测试图像构成。具体的数据集样例如图 4 所示。

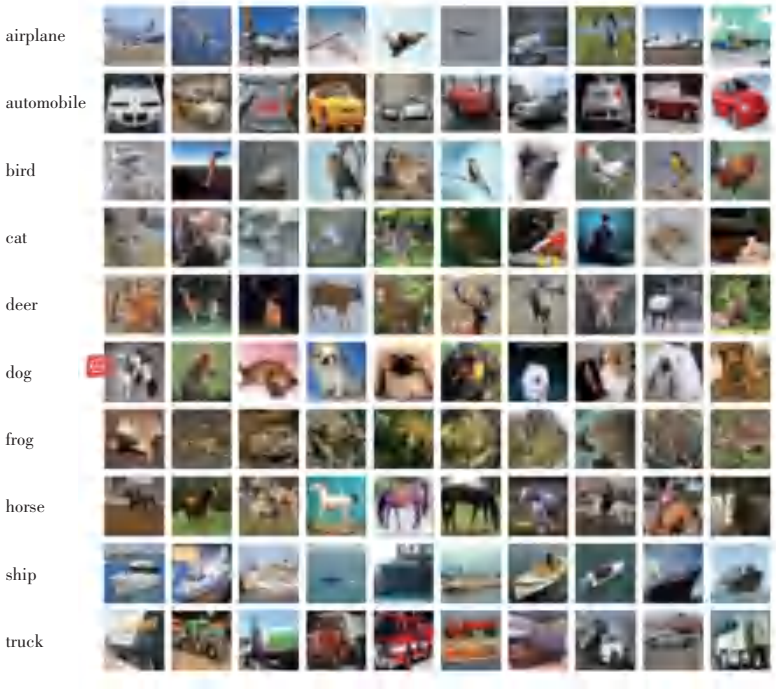

图 4 CIFAR-10 数据集

Fig. 4 CIFAR-10 datasets

2.2　实验环境

本文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。

表 1　实验环境参数

Table 1　Experiment environment parameters

名称	设置
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(24 G)
内存	80 G

2.3　参数设置

由于 CIFAR-10 数据集的训练集包含 50 000 张图片，因此可以设置每个批次训练 200 张图片，使得整个训练集可以均匀分为 250 批次。此外，训练迭代的总轮数设定为 180 轮。在训练过程中，每完成一轮训练，就会进行一轮测试集上的评估，并且会记录样本的训练损失以及测试精度。

2.4　结果分析

2.4.1　训练损失

本文主要从 3 个方面评估 ATT-DDNN 框架的性能，分别是训练损失、测试精度以及重要性阈值分析，对比算法是算法 1[19]。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，具体计算公式如下：

$$L_b(y',y;\theta)=\sum_{b=1}^{B}w_bL(y'_{\text{exit}_b},y;\theta) \tag{10}$$

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 表示位于边缘端第 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，具体计算公式为：

$$L_d(y',y;\theta)=\sum_{d=1}^{D}w_dL(y'_{\text{exit}_d},y;\theta) \tag{11}$$

其中，D 表示云端退出点的总数，w_d 表示位于云端第 d 个退出节点前的子网络的参数权重。

训练损失如图 5 所示，图 5 展示了不同算法在 180 轮训练中的训练损失，模型在训练过程中的损失分为边缘训练损失和云端训练损失。由图 5 可见，与算法 1 相同的是，云端的损失在训练过程中始终低于边缘损失，由于云端额外的 DNN 模型处理，网络模型结构更强，能够更好地提取图片的深层特征。其次，ATT-DDNN 框架的边缘训练损失和云端训练损失始终低于算法 1。经过分析，算法 1 的特征融合模块采用的最大池化方法仅将单一终端设备的特征提取结果用于分类，忽略了图片特征中的许多细节信息。而 ATT-DDNN 采用的是基于注意力机制的特征融合方法，提高了 ATT-DDNN 框架中模型对重要特征的响应能力，因此 ATT-DDNN 展现出更好的性能。

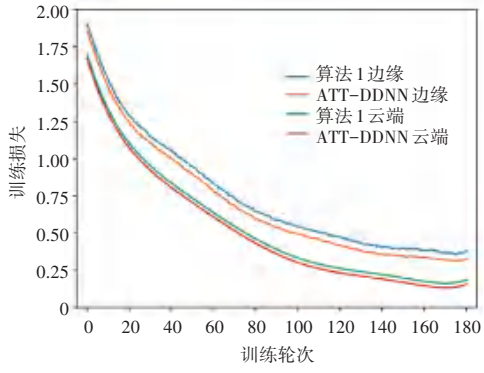

图 5　训练损失

Fig. 5　Training loss

2.4.2　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进行了验证，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公式为：

$$\text{ACC}_b=\frac{P_b}{N_b} \tag{12}$$

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。云端测试精度计算公式为：

$$\text{ACC}_d=\frac{P_d}{N_d} \tag{13}$$

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 6 所示。由图 6 可见，ATT-DDNN 的边缘测试精度和云端测试精度始终优于算法 1 的边缘测试精度和云端测试精度。进一步验证了 ATT-DDNN 的优越性。

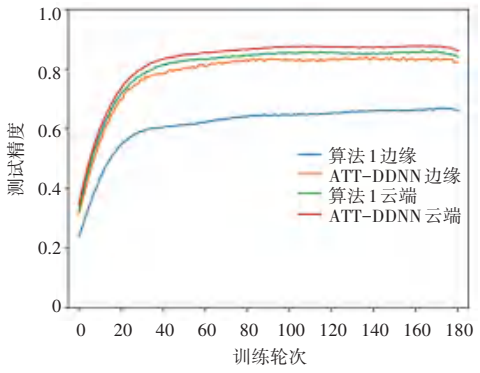

图 6 测试精度

Fig. 6 Test accuracy

2.4.3 重要性阈值分析

在 ATT-DDNN 中,重要性阈值的设定对框架性能有着很大的影响。较低的阈值设定会增加参与融合的特征数量,测试精度会上升,同时也会因为需要传输更多的特征信息而导致通信和计算成本增加。而如果提高阈值会减少用于融合的特征数量,可以减少通信量和计算成本,但会因为损失了部分特征而降低了测试精度。

因此,在使用 ATT-DDNN 时,如何设置最合适的重要性阈值至关重要。为了更加直观地观察重要性阈值对 ATT-DDNN 性能的影响,将重要性阈值从 0 开始,以 0.02 为步长逐步提高至 1.0,同时使用 CIFAR-10 数据集进行测试,观察 ATT-DDNN 的测试精度和特征的丢失率。仿真实验结果如图 7 所示。

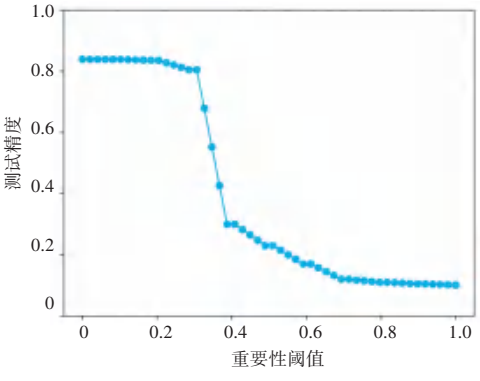

图 7 重要性阈值对测试精度的影响

Fig. 7 Influence of importance threshold on test accuracy

由图 7 分析可知,随着重要性阈值的增加,测试精度有降低的趋势,重要性阈值从 0 增加到 0.3 时,测试精度有轻微下降并保持在 80%以上。重要性阈值从 0.3 增加到 0.4 时,准确率快速下降至 30%左右。之后随着重要性阈值增加到 1.0,准确率有轻微下降,但最终趋近于 10%。这说明了,随着重要性阈值的增加,参与融合的特征逐渐降低,这导致了测试精度的降低。

重要性阈值对特征丢失率的影响如图 8 所示。随着重要性阈值的增加,特征丢失率也在不断上升,重要性阈值从 0 增加到 0.2 时,特征丢失率只有 1%的上升。随着重要性阈值增加至 0.3 时,特征丢失率稍微上升到至 20%。随着重要性阈值继续增加至 1.0,特征丢失率显著上升,达到了 95%以上。这也意味着只有 5%的特征参与了后续的特征融合与计算过程。随着重要性阈值的增加,测试精度虽然会被降低,但是特征丢失率的增加会减少设备之间的通信消耗和后续的计算成本。因此,选择合适的重要性阈值可以在保持较高精度的基础上减少通信成本。比如在本文实验中,可以将重要性阈值设为 0.3,可以在保持 84%的测试精度的同时降低 20%的通信成本。

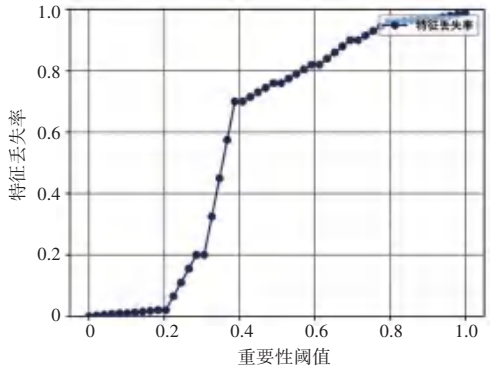

图 8 重要性阈值对特征丢失率的影响

Fig. 8 Influence of importance threshold on feature loss rate

3 结束语

本文提出了 ATT-DDNN 推理框架。与现有的 DDNN 框架相比,该框架通过增加额外的边缘处理和边缘出口,提升了边缘出口概率,减少了设备之间的通信成本和后续的计算成本。此外,通过注意力机制为不同终端计算特征的重要程度,显式地建模通道之间的关系,并自适应地重新标定特征图中的通道,提高了模型对重要特征的响应能力,抑制负面特征的响应能力,从而进一步降低了通信成本和计算成本。本文在开放的 CIFAR-10 数据集上进行验证,实验结果表明,ATT-DDNN 可以在保持较高测试精度的同时,降低 20%的通信成本。尽管 ATT-DDNN 框架已经在多个方面显示出优越性,但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。

参考文献

[1] KHAN A，SOHAIL A，ZAHOORA U，et al. A survey of the recent architectures of deep convolutional neural networks[J]. Artificial Intelligence Review，2020，53：5455-5516.

[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，AIOLLI F. An introduction to deep learning in natural language processing：Models，techniques，and tools[J]. Neurocomputing，2022，470：443-456.

[5] HEMA C，MARQUEZ F P G. Emotional speech recognition using CNN and deep learning techniques[J]. Applied Acoustics，2023，211：109492.

[6] LI Jinyu . Recent advances in end-to-end automatic speech recognition[J]. arXiv preprint arXiv，2111.01690，2021.

[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 Chen，SHEN Shibo，LI Rongpeng，et al. An efficient pruning scheme of deep neural networks for Internet of Things applications[J]. EURASIP Journal on Advances in Signal Processing，2021，2021：31.

[9] XU Hang，HO C Y，ABDELMONIEM A M，et al. GRACE：A compressed communication framework for distributed machine learning[C]// Proceedings of 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS). Piscataway，NJ：IEEE，202：561-572.

[10] REN Jinke，HE Yinghui，YU Guanding，et al. Joint communication and computation resource allocation for cloud-edge collaborative system[C]//Proceedings of 2019 IEEE Wireless Communications and Networking Conference (WCNC). Piscataway，NJ：IEEE，2019：1-6.

[11] HEIGOLD G，VANHOUCKE V，SENIOR A，et al. Multilingual acoustic models using distributed deep neural networks[C]// Proceedings of 2013 IEEE International Conference on Acoustics，Speech and Signal Processing. Piscataway，NJ：IEEE，2013：8619-8623.

[12] LEROUX S，BOHEZ S，CONICK D E，et al. The cascading neural network：Building the internet of smart things[J]. Knowledge and Information Systems，2017，52：791-814.

[13] DING Chuntao，ZHOU Ao，LIU Yunxin，et al. A cloud-edge collaboration framework for cognitive service[J]. IEEE Transactions on Cloud Computing，2020，10(3)：1489-1499.

[14] CHEN Jiasi，RAN Xukan. Deep learning with edge computing：A review[J]. Proceedings of the IEEE，2019，107(8)：1655-1674.

[15] ALI M，ANJUM A，YASEEN M U，et al. Edge enhanced deep learning system for large-scale video stream analytics[C]// Proceedings of 2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC). Piscataway，NJ：IEEE，2018：1-10.

[16] ONGATI F，MUCHEMI D E. Big data intelligence using distributed deep neural networks[J]. arXiv preprint arXiv，1909.02873，2019.

[17] YANG Shusen，ZHANG Zhanhua，ZHAO Cong，et al. CNNPC：End-edge-cloud collaborative CNN inference with joint model partition and compression[J]. IEEE Transactions on Parallel and Distributed Systems，2022，33(12)：4039-4056.

[18] TEERAPITTAYANON S，MCDANEL B，KUNG H T. Branchynet：Fast inference via early exiting from deep neural networks[C]//Proceedings of 2016 23rd International Conference on Pattern Recognition (ICPR). Piscataway，NJ：IEEE，2016：2464-2469.

[19] TEERAPITTAYANON S，MCDANEL B，KUNG H T. Distributed deep neural networks over the cloud，the edge and end devices[C]//Proceedings of 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). Piscataway，NJ：IEEE，2017：328-339.

[20] LI G，LIU L，WANG X，et al. Auto-tuning neural network quantization framework for collaborative inference between the cloud and edge[C]//Proceedings of the 27th International Conference on Artificial Neural Networks(ICANN 2018). Cham：Springer，2018：402-411.

[21] MAO Jiachen，CHEN Xiang，NIXON K W，et al. Modnn：Local distributed mobile computing system for deep neural network[C]//Proceedings of Design，Automation & Test in Europe Conference & Exhibition (DATE). Piscataway，NJ：IEEE，2017：1396-1401.

[22] ZHAO Zhuoran，BARIJOUGH K M，GERSTLAUER A. Deepthings：Distributed adaptive deep learning inference on resource-constrained iot edge clusters[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems，2018，37(11)：2348-2359.

[23] DISABATO S，ROVERI M，ALIPPI C. Distributed deep convolutional neural networks for the internet-of-things[J]. IEEE Transactions on Computers，2021，70(8)：1239-1252.

[24] HU Jie，SHEN Li，SUN Gang. Squeeze-and-excitation networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway，NJ：IEEE，2018：7132-7141.

[25] KRIZHEVSKY A，HINTON G. Learning multiple layers of features from tiny images[R]. Toronto：University of Toronto，2009.