计算机科学
COMPUTER SCIENCE

面向边缘智能应用的多出口深度神经网络随机优化方法

李洲诚, 张毅, 孙晋

引用本文

李洲诚, 张毅, 孙晋. 面向边缘智能应用的多出口深度神经网络随机优化方法[J]. 计算机科学, 2025, 52(4): 85-93.

LI Zhoucheng, ZHANG Yi, SUN Jin. Stochastic Optimization Method for Multi-exit Deep Neural Networks for Edge Intelligence Applications [J]. Computer Science, 2025, 52(4): 85-93.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.241000097

面向边缘智能应用的多出口深度神经网络随机优化方法

李洲诚　张　毅　孙　晋
南京理工大学计算机科学与工程学院　南京 210094
(zcli@njust.edu.cn)

摘　要　边缘智能作为一种新型的智能计算范式,能够有效提升智能推理任务在嵌入式边缘设备中的响应速度。而信息年龄(AoI)作为衡量数据时效性的重要指标,对于边缘智能应用的计算资源开销和实时响应至关重要。针对多出口深度神经网络(DNN)的资源配置优化问题,考虑出口退出概率造成的 AoI 随机不确定性,引入系统 AoI 的概率约束,基于随机优化理论对出口设置进行决策,以最小化多出口 DNN 的资源开销。文中提出了一种基于布谷鸟搜索的元启发式算法对所构建的具有概率约束的随机优化问题进行求解,基于各出口的退出概率预测系统 AoI 的统计分布,根据给定的 AoI 阈值计算相应的资源消耗量并将其作为布谷鸟个体的适应度值,迭代更新布谷鸟种群并搜索得到最小计算资源开销的出口设置方案。针对多种 DNN 模型的实验结果表明,与确定性的优化方法相比,随机优化方法能够获得更佳的出口设置决策,在满足 AoI 概率约束的前提下显著降低了 DNN 的计算开销。

关键词:边缘智能;信息年龄;多出口神经网络;随机优化;概率约束;元启发式算法
中图分类号　TP301

Stochastic Optimization Method for Multi-exit Deep Neural Networks for Edge Intelligence Applications

LI Zhoucheng, ZHANG Yi and SUN Jin
School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

Abstract　As a novel intelligent computing paradigm, edge intelligence can effectively enhance the response speed of intelligent inference tasks on embedded edge devices. Age of information(AoI), an important metric for measuring data freshness, is of great significance to the computing resource overhead and real-time response of edge intelligence applications. This work studies the resource allocation optimization problem for multi-exit deep neural networks(DNNs) that takes into account the uncertainty of AoI caused by exit probabilities and introduces a probabilistic constraint on system AoI. The stochastic optimization theory is incorporated to make decision on the most appropriate exit configuration for the purpose of minimizing the resource overhead of multi-exit DNNs. A cuckoo search-based metaheuristic algorithm is proposed to solve the stochastic optimization problem with the probabilistic AoI constraint. The metaheuristic predicts the statistical distribution of system AoI based on the exit probabilities, calculates the resource consumption according to a specified AoI threshold and uses it as the fitness value of the corresponding cuckoo individual, and iteratively updates the cuckoo population to explore the exit configuration solution leading to the lowest computing resource overhead. Experimental results on various DNN models show that compared with deterministic optimization methods, the stochastic optimization approach can produce better exit configuration solutions, significantly reducing resource overhead while satisfying the probabilistic AoI constraint.

Keywords　Edge intelligence, Age of information, Multi-exit deep neural network, Stochastic optimization, Probabilistic constraint, Metaheuristic algorithm

1　引言

边缘计算与人工智能的融合以及彼此赋能催生了边缘智能的概念[1-2]。这种新型的计算范式将智能模型和算力资源部署在网络边缘的设备中,以实现本地数据的快速处理和智能推理的实时响应[3]。边缘智能使得智能应用尤其是深度神经网络(Deep Neural Network,DNN)模型能够广泛部署于嵌入式边缘设备执行。然而边缘设备通常存在计算资源有限的

到稿日期:2024-10-21　返修日期:2025-01-31
基金项目:江苏省重点研发计划(批准号:BE2022065-2);江苏省创新支撑计划(批准号:BZ2023046)
This work was supported by the Jiangsu Provincial Key Research and Development Program(Grant No. BE2022065-2) and Jiangsu Provincial Innovation Support Program(Grant No. BZ2023046).
通信作者:孙晋(sunj@njust.edu.cn)

问题[4]，而多出口 DNN 是一种有效应对该问题的解决方案。多出口 DNN 的主要特点在于模型的提前退出机制，其允许在网络的多个层次设置出口，当较低层次的出口已经能够给出较为准确的预测时，模型可以选择提前停止推理任务，以加快推理速度和减少计算资源消耗[5-7]。与模型压缩和蒸馏等模型优化方法相比，多出口 DNN 的提前退出机制在节省计算资源的同时仍然能够保持较高准确率。

信息年龄(Age of Information，AoI)是边缘智能系统中衡量数据“新鲜度”或“时效性”的重要指标[8]。在自动驾驶、智能制造等依赖实时数据进行决策的边缘智能应用中，AoI 的管理直接影响数据的新鲜程度以及决策的准确性[9-10]。在资源有限的边缘设备上，控制 AoI 有助于优化计算和通信资源的消耗。通过优先处理 AoI 较小的数据或者丢弃过时的数据，从而提高资源利用效率[11-12]。

在执行多出口 DNN 推理任务和 AoI 感知的边缘智能应用场景下，DNN 模型在各出口分支以一定概率退出推理任务，导致系统 AoI 在内的性能指标不再是一个确定值，而是服从某种统计分布[13]。这种统计上的不确定性使得多出口 DNN 的模型优化本质上成为一个随机优化问题，给边缘智能系统的 AoI 管理和计算资源配置带来了严峻挑战。一方面，对 DNN 推理任务的过度资源配置会导致硬件成本和能耗的上升；另一方面，未考虑退出概率造成的系统 AoI 不确定性而未能分配足够的计算资源，则有可能因为 AoI 的随机波动导致无法满足边缘智能应用的时效性需求而降低决策可靠性。

基于上述考虑，本文针对多出口 DNN 的资源配置优化问题展开研究。考虑出口退出概率造成的 AoI 随机不确定性，引入系统 AoI 的概率约束，基于随机优化理论对模型的出口设置进行优化决策，以最小化多出口 DNN 的资源开销为目标。本文提出了一种基于布谷鸟搜索的元启发式算法，对所构建的具有概率约束的随机优化问题进行求解。具体而言，对于每个布谷鸟个体使用区间二分类算子将其映射为一个多出口 DNN 的出口选择向量，基于各出口的退出概率预测系统 AoI 的统计分布，根据给定的 AoI 阈值计算相应的资源消耗量，将其作为布谷鸟个体的适应度值。算法根据布谷鸟的位置更新策略对布谷鸟种群进行更新，以搜索发现更高质量的布谷鸟个体。以上过程迭代执行，达到迭代轮次上限后即可得到最小计算开销的出口选择向量。针对多种 DNN 模型的实验结果表明，与确定性的优化方法相比，本文提出的随机优化方法能够获得更佳的出口设置决策，在满足 AoI 概率约束的前提下显著降低 DNN 的计算开销。

2 相关工作

在基于多出口 DNN 的模型优化方面，现有方法大多在出口位置固定的前提下提升模型的精度或效率。例如，Kaya 等[14]提出的基于置信度的前退算法和混淆分析方法，通过缓解 DNN 的过度思考来减小模型的计算开销。Laskaridis 等[15]在多出口 DNN 上设置若干个出口，以 softmax 层输出的最大值为提前退出的判断依据，允许一定比例的样本提前退出以节省计算开销。Ju 等考虑置信度与准确率的多样性，进一步提出动态退出学习算法[16]和动态退出调度算法[17]，在模型各个出口迭代进行提前退出或继续推理之间的决策优化。Dong 等[6]则充分考虑了出口位置和数量对多出口 DNN 推理性能的影响，提出了一种数据感知的多出口 DNN 出口选择机制，通过采集模型各出口的样本退出概率分布与各层计算量信息，自适应地选择最优的出口设置，最小化推理任务的计算量。

在 AoI 感知的边缘智能系统优化方面，现有方法可以分为静态和动态两类。静态方法聚焦于信息重要性的量化表征和 AoI 阈值的保障。Wang 等[18]为此构建了一个系统模型，将信息的新鲜度通过其临界水平的变化来衡量，并定义了 AoCI 概念以评估信息的重要性。为了保障源节点的最大 AoI 阈值，Li 等[19]提出了一种循环调度程序检测策略和虚构多项式映射等策略，以应对不同规模的边缘计算网络。在动态问题方面，Li 等[20]提出了一种名为 Eywa 的通用框架，为一系列 AoI 感知的优化问题构建了高性能的调度策略，重点解决最小化 AoI 的加权和以及 AoI 约束下的带宽需求最小化。Lin 等[21]使用能量调度策略来检测信道干扰，设计了低复杂度的 Juventas 快速调度方法，用于优化各传感器节点的能量和时间分配，以提升边缘智能系统的 AoI 指标和系统吞吐量。

然而，现有相关工作中尚无综合考虑多出口 DNN 的 AoI 感知的边缘智能优化方法。更重要的是，DNN 模型在各出口以一定概率退出的机制使得 AoI 指标的保证和出口设置的优化决策成为一个随机优化问题。而现有方法通常将包括 AoI 在内的性能指标假设为确定性的值(使用指标的均值或最大值作为估计)设计相应的优化方法。基于确定性假设的优化方法则会导致计算资源的浪费或无法满足 AoI 指标需求。据此，本文基于随机优化理论对多出口 DNN 的出口设置决策展开研究，通过对 AoI 指标的统计分布进行预测，引入系统 AoI 的概率约束，对资源开销最小化的出口设置方案进行求解。本文的研究工作在计算资源受限且具有实时响应需求的边缘智能应用场景下更加具有现实意义。

3 问题建模

本章构建 AoI 约束下的多出口 DNN 随机优化模型。考虑 DNN 推理任务在模型不同出口分支的退出概率，基于随机优化理论在优化问题中引入 AoI 指标的概率约束，对多出口 DNN 的出口设置进行决策，在满足 AoI 概率约束的前提下最小化多出口 DNN 的计算开销。

3.1 多出口 DNN 推理机制

多出口 DNN 的结构以传统单出口 DNN 为主干，添加若干预分类分支构造而成。图 1 为一个多出口 DNN 结构的推理模式示意图。多出口 DNN 的主干部分保留了原始 DNN 的结构，包括卷积层、池化层、全连接层等；而分支部分作为模型的预分类/识别器，包含若干池化层、全连接层与 softmax 层。每一个分支即是网络的一个提前出口，与作为主干出口的原始 DNN 出口一并构成了多出口 DNN 的出口集合。相邻提前出口在主干上间隔若干卷积层，因为 DNN 的特征提取能力一般随着卷积层的增加而提升。当样本数据前向传播至某一提前出口时，DNN 可能已经提取到足够特征，给出

推理结果，体现为对应提前出口给出的置信度大于给定阈值。此时可通过该提前出口提前退出，不必再进行之后的传播过程，如果经过 DNN 尚未能提取到样本的足够信息，则样本将继续前向传播的过程。样本在各出口提前退出的概率取决于 DNN 的训练结果与推理任务使用的数据集分布情况。对于分布特定的数据集，其在不同出口退出的概率是固定的，可通过采集输入样本数据集在各出口分支处提前完成推理任务的频率分布来确定其概率值[6]。具体而言，当 DNN 在某个出口分支处的输出达到预设的置信度阈值，即可认为该分支已经完成推断任务从而提前退出。通过多次重复实验并统计输入样本中各个出口达到此阈值的比例，即可预测出每个出口的退出概率。

以图 1 中所示 DNN 为例，当样本经过预处理后输入 DNN 模型，前向传播至主干第一个 maxpool 层后不会直接执行主干第三个卷积层的卷积操作，而是进入提前出口分支，以一定概率提前退出并返回分类结果。若未能退出，则继续执行主干第三个卷积层的计算，直至到达下一个出口重复以上过程。样本数据经过所有提前出口而无法退出最终到达主干出口时，将以 100% 的概率退出。对于分布特定的数据样本集，其在不同深度出口退出的概率确定且存在显著差异。不合理设置提前出口可能难以有效提前退出推理任务，反而因为预分类分支的额外开销降低了推理速率。因此，提高多出口 DNN 的推理性能需要对出口位置及数量执行优化策略。

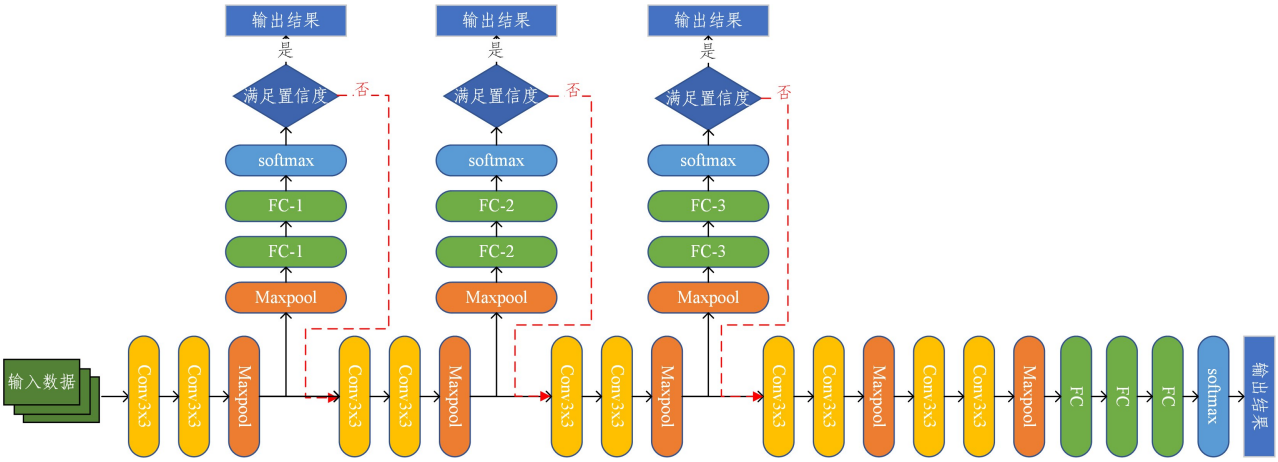


图 1 多出口 DNN 推理模式示意图[6]

Fig. 1 Schematic of multi-exit DNN inference mode[6]

3.2 AoI 系统模型

本文考虑由一个数据源与一个边缘设备上运行的 server 程序组成的 AoI 系统。数据源是数据的生产者，以一定时长为周期不断产生数据并向 server 程序传递；server 程序是数据的消费者，循环对到达的数据进行推理计算。在每一次循环中，server 程序调用预先确定的多出口 DNN 模型，将数据作为样本输入执行推理任务，得到推理结果，之后将结果输出给下一级消费者。

记数据源 z 第 j 次($j\in N$)产生的数据为 d^j，数据产生时间为 $S^j=(j-1)\times\tau$(特别地，记 $S^0=0$)，server 程序对 d^j 的推理计算过程为 G^j，忽略数据传输与预处理时间。AoI 系统的行为模式如图 2 所示。多出口 DNN(表示为 G)的执行时长 $\widetilde{ET}$ 不是一个确定值，其原因在于多出口 DNN 的执行模式：数据以一定概率在每个出口退出，这使得 $\widetilde{ET}$ 成为一个离散随机变量。其样本空间内可能的取值数量由多出口 DNN 的出口数量决定，每一个样本值为数据从对应出口退出时的推理计算用时。

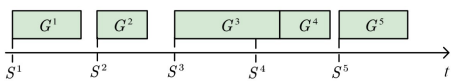


图 2 AoI 系统的行为模式

Fig. 2 Behavior pattern of AoI system

记 G^j 的完成时间为 $\widetilde{ct}^j$，系统模型中的 AoI 可定义为：

$$\widetilde{AOI}^j=\widetilde{ct}^j-S^{j-1},\forall j\in N^+ \tag{1}$$

其代表数据源 z 第 $j-1$ 次产生数据至 server 完成第 j 次推理计算 G^j 的时间间隔，度量了系统处理信息的及时程度，也反映了下一级消费者感知到的与数据源 z 的最大延迟。

通过前文的描述可知，系统 AoI 的不确定性来自多出口 DNN 推理计算时长的不确定性，而推理时长的不确定性来源于样本在不同深度出口退出的随机性。从反方向说，对多出口 DNN 出口的选择设置以及算力资源的分配决定了多出口 DNN 执行时长的概率分布，进而决定了系统 AoI 的概率分布。而在系统 AoI 的概率约束下，不合理的提前出口设置可能导致难以有效提前退出推理任务，反而因为预分类分支的额外开销降低了推理速率，需要分配更多算力资源才能满足约束。

3.3 计算任务模型

多出口 DNN 以一个 n 层单出口 DNN 为主干，根据主干结构可将该 DNN 划分为 n 个逻辑层 $L=\{L_1,L_2,\cdots,L_n\}$。每个逻辑层 $L_i\in L$ 包含了原始 DNN 的第 i 层，如果原始 DNN 第 i 层后可以设置一个提前出口分支，则 L_i 包括这一分支的全部功能层，并称 L_i 是一个候选出口。每个逻辑层 L_i 可用四元组$\langle f_i,ef_i,ex_i,p_i\rangle$描述。其中，$f_i$ 表示 L_i 中主干层的浮点计算量，单位为 s·GHz；ex_i 是一个 0～1 变量，表示 L_i 是否为候选出口，当且仅当 $ex_i=1$ 时，L_i 是一个候选

出口；ef_i 表示该提前出口分支的浮点计算量；$p_i \in [0,1]$ 为数据前向传播到 L_i 处退出的概率，当且仅当 L_i 不是出口时 p_i 为 0。最后一个逻辑层 L_n 是原始单出口 DNN 的出口，样本数据前向传播至此后必然退出，因此有 $ex_n=1$ 与 $p_n=1$。

逻辑层 L_i 的运行时长 ET_i 可表示为：

$$ET_i=\frac{f_i+ef_i \cdot x_i}{C} \tag{2}$$

其中，C 表示 server 程序分配的计算资源数量（用等效 CPU 频率表示，单位为 GHz）；x_i 是一个 0～1 决策变量，表示是否在 L_i 中设置预分类分支，即逻辑层 L_i 是否被选中作为出口。只有候选出口可以设置为出口，因此有约束 $x_i \leqslant ex_i$。此外，包含原始 DNN 出口的最后一个逻辑层 L_n 必须设置为出口，因此有约束 $x_n=1$。全部出口选择决策变量 $\{x_i \mid i \in [1,n]\}$ 构成一个 0～1 向量 $\boldsymbol{X}=[x_1,x_2,\cdots,x_n]^{\mathrm{T}}$，称为对多出口深度神经网络 G 的出口选择向量。

根据前文所述，多出口 DNN 的执行时长 $\widetilde{ET}$ 是一个离散随机变量。对于给定的 $\boldsymbol{X}$ 与 $\boldsymbol{C}$ 的值，可以用如下概率空间描述：

$$\begin{aligned}&\Omega=\{f(m) \mid m \in [1,n]\}\\&P\{\widetilde{ET}=f(m)\}=P(m)\end{aligned} \tag{3}$$

其中，Ω 表示 $\widetilde{ET}$ 的样本空间，事件 $\widetilde{ET}=f(m)$ 表示数据前向传播至逻辑层 L_m 处退出，因此样本空间大小 $|\Omega|$ 为多出口深度神经网络的逻辑层总数 n，样本值 $f(m)$ 的计算式为：

$$f(m)=\sum_{i=1}^{m} ET_i, m \in [1,n] \tag{4}$$

其中，$f(m)$ 表示 L_1 至 L_m 层的总执行时长。$f(m)$ 的概率测度 $P(m)$ 则可表示为：

$$P(m)=p_m x_m \prod_{i=1}^{m-1}(1-p_i x_i) \tag{5}$$

此概率表示在给定的出口设置 $\boldsymbol{X}$ 下，数据在 L_s 至 L_{m-1} 均未能退出而在 L_m 成功退出的概率。

记数据 d^j（$j \in N^+$）的推理计算过程 G^j 的开始时间为 $\widetilde{st^j}$，结束时间为 $\widetilde{ct^j}=\widetilde{st^j}+\widetilde{ET}$（特别地，记 $\widetilde{st^0}=0$ 与 $\widetilde{ct^0}=0$）。G^j 需要等待上一次计算 G^{j-1} 完成且本次数据 d^j 产生，本文不考虑数据传输与预处理时长，因此有：

$$\widetilde{st^j}=\max\{\widetilde{et^{j-1}},S^j\} \tag{6}$$

其概率分布可以用如下分段函数描述：

$$P\{\widetilde{st^j}=z\}=\begin{cases}0, & \text{if } z<S^j\\ P\{\widetilde{ct^{j-1}} \leqslant S^j\}, & \text{if } z=S^j\\ P\{\widetilde{ct^{j-1}}=z\}, & \text{if } z>S^j\end{cases} \tag{7}$$

其中，事件 $\widetilde{st^j}=S^j$ 表示上一次计算 G^{j-1} 在本次数据 d^j 产生前已经完成。而 $\widetilde{ct^j}=\widetilde{st^j}+\widetilde{ET}$ 的概率分布可表示为：

$$P\{\widetilde{ct^j}=z\}=P\{\widetilde{st^j}+\widetilde{ET}=z\} \tag{8}$$

随后根据式(1)推导出 G^j 的 AoI 值 $\widetilde{AOI^j}$ 的概率分布：

$$P\{\widetilde{AOI^j}=z\}=P\{\widetilde{ct^j}-S^{j-1}=z\} \tag{9}$$

进一步地，server 程序在所有推理计算任务中取得的最大 AoI $\widetilde{AOI_{\max}}=\max\{\widetilde{AOI^j}\}$ 的概率分布可表示为：

$$P\{\widetilde{AOI_{\max}}=z\}=P\{\widetilde{AOI^j} \leqslant z\}-P\{\widetilde{AOI^j}<z\} \tag{10}$$

基于 $\widetilde{AOI_{\max}}$ 的统计分布，本文引入一个针对系统最大 AoI 值的概率约束：

$$P\{\widetilde{AOI_{\max}} \leqslant dl\} \geqslant \alpha \tag{11}$$

其中，dl 表示 $\widetilde{AOI_{\max}}$ 的上限值，α 是可定义的概率约束阈值，表示 $\widetilde{AOI_{\max}} \leqslant dl$ 的概率，α 越高表示对 $\widetilde{AOI_{\max}}$ 的约束越严格。

模型优化问题的目标是在式(11)中的 AoI 概率约束下，对所有的候选出口进行决策，使得多出口 DNN 的计算资源开销最少。综上所述，多出口 DNN 随机优化模型可归纳如下：

$$\begin{aligned}&\min\ C\\&\text{s.t. } P\{\widetilde{AOI_{\max}} \leqslant dl\} \geqslant \alpha\\&\quad x \leqslant ex_i \in \{0,1\}\\&\quad x_n=1\end{aligned} \tag{12}$$

所有的 0～1 决策变量 x_i 形成一个出口选择向量，其中 $x_n=1$ 的约束表示 DNN 模型在最后一个逻辑层必然退出。优化目标 C 的值由出口选择向量以及系统 AoI 的统计分布决定：系统为 server 程序分配的计算资源数量必须满足 AoI 概率约束中的阈值要求。

4 随机优化算法

本文提出了一种基于布谷鸟搜索[22]的元启发式算法，用于求解具有 AoI 概率约束、计算资源开销最小化的随机优化问题。对于布谷鸟种群中每个个体，算法使用区间二分类规则将其映射为随机优化问题的一个解（即一个出口选择向量）。由此，算法在布谷鸟个体空间和随机优化解空间建立关联，并基于布谷鸟搜索的优化原理搜索获得随机优化问题的最佳出口设置决策。算法的核心部分是根据映射得到的出口选择向量，预测各出口退出概率导致的系统 AoI 随机分布，计算满足 AoI 概率约束的最少计算资源数量并将其作为对应的布谷鸟个体的适应度值。在获得种群中的最优个体之后，算法根据布谷鸟搜索的个体更新规则更新种群中的个体位置。通过迭代执行上述过程，不断搜索得到质量更优的个体及相应的出口选择向量。

4.1 种群初始化

算法初始化包括所有个体的初始化与初始解质量的生成。设种群集合为 $N=\{N_1,N_2,\cdots,N_\Lambda\}$，其中 Λ 表示种群规模，N_m 表示第 m 个布谷鸟个体（$m \in [1,\Lambda]$）。对任意个体 N_m，其位置表示为一个 n 维向量 $\boldsymbol{P}_m=[p_{m,1},p_{m,2},\cdots,p_{m,n}]^{\mathrm{T}}$，向量每一维度的值在 $[p_{\min},p_{\max}]$ 范围内随机生成。

4.2 适应度值评估

在布谷鸟个体的适应度值评估阶段，首先将个体的位置向量 $\boldsymbol{P}_m$ 映射至一个出口选择向量。然后向量中的出口设置结合系统 AoI 的分布信息以及概率约束阈值，计算得到必需的计算资源数量，将其作为对应布谷鸟个体的适应度值。

算法 1 个体适应度值评估

输入：个体位置向量 $\mathbf{P}_m$；$\widetilde{\mathrm{AOI}_{\max}}$ 上限 dl；约束的满足概率 α；推理任务数量 N

输出：个体适应度值即最小资源开销 C

1. 初始化出口选择向量 $\mathbf{X}=[x_1,x_2,\cdots,x_n]^T$
2. for each i∈[1,n] do
3. if $P_{m,i}\geqslant(p_{min}+p_{max})/2$ 且 $ex_i=1$then
4. 设置 $x_i=1$；
5 else
6. 设置 $x_i=0$；
7. end if
8. end foreach
9. 计算各逻辑层浮点计算量 $F=\{F_1,\cdots,F_n\}$；
10. 根据式(5)计算各层退出率 P(m)；
11. 计算最大运行时长 $ET_{max}:=dl-\tau$；
12. 计算满足 $\widetilde{ET}\leqslant ET_{max}$ 的概率 $\beta=\sqrt[N]{\alpha}$；
13. 设置 β 分位点 $F^\beta:=0$；
14. 设置累积退出率 $d:=0$；
15. for i=1:n do
16. 更新 $F^\beta:=F^\beta+F_i$；
17. 更新 $d:=d+P(i)$；
18. if $d\geqslant\beta$ then
19. break；
20. end if
21. end for
22. 更新 $C:=F^\beta/ET_{max}$；
23. return C.

算法 1 为适应度值评估的总体流程。首先根据区间二分类算子，由个体位置向量 $\boldsymbol{P}_m$ 中的各维度值得到出口选择向量 $\boldsymbol{X}$ 中的各决策变量值：若 $P_{m,i}$ 满足 $P_{m,i}\geqslant(p_{min}+p_{max})/2$ 且对应出口是一个候选出口($ex_i=1$)，则选中该出口为一个真实出口。然后，根据出口选择向量 $\boldsymbol{X}$ 计算多出口 DNN 各层的浮点运算量：

$$F_i=f_i+ef_i\cdot x_i \tag{13}$$

若要满足式(11)描述的 AoI 最大值概率约束，单次推理任务的运行时长同样需要以一定概率满足小于某一最大值的概率约束。显然，最大单次运行时长概率约束是最大 AoI 概率约束的必要条件。因此随机优化算法将 AoI 最大值的概率约束等价转化为单次推理任务运行时间的概率约束。具体而言，算法通过式(11)计算对单次推理任务运行时长的概率约束如下：

$$P\{\widetilde{ET}<ET_{max}\}\geqslant\beta \tag{14}$$

其中，$ET_{max}=dl-r$，是满足 $AoI<AoI_{max}$ 条件的推理任务最大运行时长；r 是数据源产生数据的时间间隔；$\beta=\sqrt[N]{\alpha}$ 是根据 AoI 阈值等价计算得到的推理任务运行时长的相应阈值。该步骤实现了将最大 AoI 的概率约束转化为对单次执行时长的概率约束。

在本文建立的模型中，推理所需浮点计算量 $\widetilde{F}$ 等于运行时长 $\widetilde{ET}$ 与计算资源数量 C 的乘积，即 $\widetilde{F}=\widetilde{ET}\times C$。为求得满足式(14)的最小 C 值，首先通过算法 1 第 15－21 行循环计算 $\widetilde{F}$ 的 β 分位点 F^β：

$$F^\beta=\arg\min_F\{P\{\widetilde{F}<F\}\geqslant\beta\} \tag{15}$$

算法在循环中累积各逻辑的浮点运算量 F_i 与退出概率 $P(m)$，在满足退出概率大于 β 后跳出循环。随后通过算法 1 第 22 行的除法操作 $C:=F^\beta/ET_{max}$ 计算得到为满足 AoI 概率约束而必须分配的最少算力资源。

4.3 种群更新

布谷鸟巢的更新主要包括 Lévy 飞行、抛弃部分糟糕的巢、产生新巢、保存最佳位置等步骤。具体流程如算法 2 所示。

算法 2 种群更新

输入：当代种群 N^t；发现新巢概率 pa

输出：下一代种群 N^{t+1}

1. 从 N^t 中随机选择一个个体 N_j^t；
2. 将 N_j^t 通过 Lévy 飞行产生一个解 N_{Levy}^t；
3. 评估解的质量，记为 C_{Levy}^t；
4. if $C_{Levy}^t<C_j^t$
5. $N_j^t:=N_{Levy}^t$；
6. end if
7. for i=1:m do
8. if rand() < pa then
9. 由 N_i^t 生成 N_i^{t+1}；
10. if $C_i^{t+1}>C_i^t$
11. 选择 N_i^{t+1} 进入下一代种群；
12. end if
13. else
14. 选择 N_i^t 进入下一代种群；
15. end if
16. end for
17. return N^{j+1}.

更新策略首先从当代种群中随机选中一个布谷鸟个体 N_j^t，随后执行 Lévy 飞行过程[23]进行全局搜索，由个体 N_j^t 生成一个临时个体 N_{Levy}^t。使用 Mantegna 策略[24]计算 N_{Levy}^t 的位置 P_{Levy} 在每一个维度 $i\in[1,n]$ 上的值 $p_{Levy,i}^t$，具体如下：

$$p_{Levy,i}^t=p_{j,i}^t+\theta\frac{\sigma\times u}{|v|^{\frac{1}{\lambda}}} \tag{16}$$

其中，θ 是步长缩放因子参数，u 与 v 均是符合标准正态分布 $N(0,1)$ 的随机数，$\lambda\in[1,3]$ 是 Lévy 指数值，参数 σ 的定义如下：

$$\sigma=\left[\frac{\Gamma(1+\lambda)}{\lambda\Gamma((1+\lambda)/2)}\cdot\frac{\sin(\pi\lambda/2)}{2^{(\lambda-1)/2}}\right]^{1/\lambda} \tag{17}$$

如果临时个体 N_{Levy}^t 的适应度 C_{Levy}^t 小于原个体 N_j^t 适应度 C_j^t，则将 N_j^t 替换为新个体 N_{Levy}^t。

执行 Lévy 飞行后，算法将循环遍历更新所有种群中的所有个体。对于每个个体，首先生成一个符合均匀分布 $U(0,1)$ 的随机数 rand()，如果 rand()小于寄主发现新巢的概率 pa，则寄主将抛弃该巢并建立新巢(见算法 2 第 9－11 行)，否则旧个体将直接进入下一代种群(见算法 2 第 13 行)。由 N_j^t 生成 N_i^{t+1} 即是建立新巢的过程(见算法 2 第 9 行)，算法根据计算 N_i^t 的位置向量 P_i^j 计算 P_i^{j+1}：

$$p_{i,j}^{t+1}=p_{i,j}^t+\beta(p_{k,j}^t-p_{l,j}^t) \tag{18}$$

其中，$p_{k,j}^t$ 与 $p_{l,j}^t$ 是当代种群 N^t 中随机选中的两个个体，β 是符合均匀分布 $U(0,1)$ 的随机数。如果新产生的个体适应度 C_i^{t+1} 劣于原个体适应度 C_i^t，则在下一代种群 N^{t+1} 中仍然保留旧个体 N_i^t。

4.4 算法总体流程

基于布谷鸟搜索的元启发式随机优化算法的完整流程如算法 3 所示。算法 3 在初始化阶段读取各项参数与输入数据，并随机初始化布谷鸟算法种群。然后，进入算法的迭代搜索过程，基于当前种群，通过 Lévy 飞行、以一定概率抛弃旧巢并产生新巢这两种方式产生新个体。在得到下一代种群之后，通过个体适应度评估算法进行个体质量评估。如果抛弃旧巢的动作未能触发或是产生的新个体劣于旧个体，则旧个体进入下一代种群；反之，如果产生的新个体优于旧个体，则新个体进入下一代种群。在每轮迭代后，需要遍历种群中的所有个体，更新当前最佳个体与当前最佳适应度值。

算法 3 基于布谷鸟搜索的随机优化算法

输入：DNN 各逻辑层 L；种群规模 Λ；最大迭代轮次 $iter_{max}$；发现新巢概率 pa；$\widetilde{AOI_{max}}$ 上限 dl；概率约束阈值 α；推理任务数量 N

输出：最小计算资源消耗 C_{min}

1. 读取输入数据与实验参数；
2. 初始化 $C_{min}:=+\infty$ 和 $i=1$；
3. 初始化种群 $N^1=\{N_1^1,N_2^1,\cdots,N_\Lambda^1\}$；
4. 初始化当前最优个体 N_{best}^{curr}；
5. while $i\in iter_{max}$ do
6. foreach $j\in[1,\Lambda]$ do
7. 根据算法 2 评估 N_j^{i+1} 的适应度 C_j^{i+1}；
8. if $C_j^{i+1}<C_{min}$ then
9. 更新 $N_{best}^{curr}:=N_j^{i+1}$；
10. 更新 $C_{min}i:=C_j^{i+1}$；
11. end if
12. end for
13. 根据算法 3 由 N^i 生成新种群 N^{i+1}；
14. end while
15. return C_{min}.

上述随机优化算法的计算复杂度分析如下。初始化阶段的复杂度为 $O(n\cdot\Lambda)$，其中 n 是多出口 DNN 的逻辑层数量。在每一轮迭代过程中，需要遍历种群，依次计算个体适应度以更新当前最优个体，其复杂度为 $O(n\cdot\Lambda)$。在种群更新过程中，对于任意个体的 Lévy 飞行与生成新巢的操作复杂度均为 $O(n)$。因此，算法总的计算复杂度为 $O(iter_{max}\cdot n\cdot\Lambda)$。

5 实验评估

为了验证本文算法在多出口 DNN 模型优化方面的有效性，选取了多种 DNN 模型，根据随机优化算法选取 DNN 模型的最佳出口设置。使用真实的嵌入式边缘设备执行多出口 DNN 的推理任务，并对计算资源开销和系统 AoI 等性能指标展开评估。

5.1 实验设置

本文选择 ViT，VGG，ResNet 这 3 种典型的 DNN 模型，基于 Python 编程语言和 PyTorch 深度学习框架为选取的 DNN 模型添加多出口结构。其中 ViT 模型有 24 个逻辑层和 24 个候选出口，因此共存在 2^{24} 个出口选择向量；VGG 模型共有 16 个逻辑层和 13 个候选出口；ResNet 共有 34 个逻辑层和 16 个候选出口。

本文使用 CIFAR-10 图像分类数据集对构造得到的多出口 DNN 模型完成训练后，部署在 NVIDIA AGX Orin 高性能边缘计算平台执行图像分类的推理任务。此外，基于输入数据集获取了用于多出口 DNN 随机优化所需的各项参数，包括各主干层的浮点计算量、各出口的浮点计算量与提前退出概率等。

本文的随机优化算法参数设置如下：种群规模为 100，发现新巢概率 pa 设为 0.95，最大迭代轮次设为 100。如前文所述，基于确定性假设的优化方法可能会导致计算资源的浪费或无法满足 AoI 指标需求。本文分别选取基于均值的确定性方法和基于最大值的确定性方法作为对比方法开展对比实验以验证这一结论。为公平起见，对比方法均基于布谷鸟搜索策略设计，所有参数（包括种群规模、迭代次数、发现新巢概率 pa）均采用相同设置，基于均值的确定性方法假设系统 AoI 均值服从 AoI 上限约束，基于最大值的确定性方法假设系统 AoI 最大值服从 AoI 上限约束。

5.2 评估结果与分析

图 3 给出了随机优化方法和基于均值的确定性方法在多出口 ViT 模型上的对比结果。在 AoI 概率约束阈值设为 0.95 的情况下，使用 Monte-Carlo 方法对系统 AoI 的统计分布进行仿真，并计算满足 AoI 上限约束的概率。实验结果表明，确定性方法由于未考虑出口退出概率造成的 AoI 不确定性，导致违反 AoI 上限的概率显著上升。在任务数量取 100～1000 的情况下，确定性方法的 AoI 约束满足概率仅在 50.51%～57.85%之间，而随机优化方法的 AoI 约束满足概率在不同任务数量下均能达到 95.08%以上。

以上对比结果说明了基于均值的确定性方法无法满足 AoI 上限要求，从而无法保证边缘智能应用的时效性，因此进一步将随机优化算法和基于最大值的确定性方法进行对比。图 4 给出了这两种方法在不同任务数量的情况下，在 NVIDIA AGX Orin 设备上完成推理任务所消耗的计算资源数量。图中以确定性方法的资源开销为基准，对随机优化方法的资源开销做归一化处理。实验结果表明，当任务数量在 100～1000 之间时，随机优化方法平均能够节省 37.88%的计算资源。

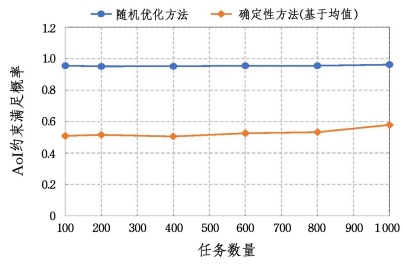


图 3 随机优化方法和基于均值的确定性方法在不同任务数量下的 AoI 约束满足概率对比

Fig. 3 Comparison of AoI constraint satisfaction probability between the stochastic method and mean-case deterministic method

此外，进一步对多出口 DNN 模型在随机优化方法的出口设置下的推理性能进行评估。实验设定任务数量为 1000。表 1 列出了 3 种 DNN 模型在使用原始结构和添加多出口结构之后的准确率对比。结果表明在经过多出口 DNN 模型优化后，不仅有效节省了计算资源开销，且推理性能并未受到

明显影响。甚至 VGG 和 ResNet 网络的准确率有所提升，因为多出口 DNN 的提前退出机制能够一定程度上抑制模型推理过程中的过度思考。

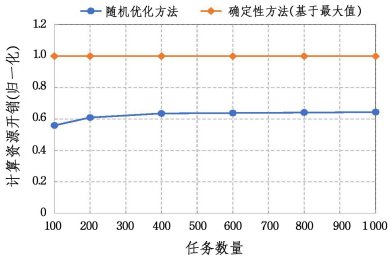


图 4 随机优化方法和基于最大值的确定性方法在不同任务数量下的计算资源开销对比

Fig. 4 Comparison of computing resource overhead between stochastic method and worst-case deterministic method

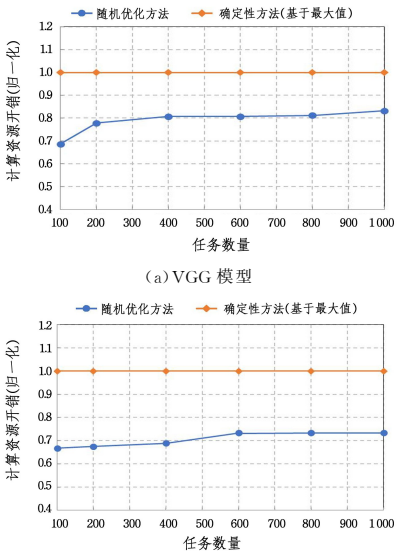
表 1 原始 DNN 和多出口 DNN 的推理准确率对比

Table 1 Comparison of inference accuracy between original DNN and multi-exit DNN

(%)

模型名称	原始网络准确率	多出口网络准确率	准确率变化
ViT	96.4	91.8	−4.6
VGG	91.8	95.9	+3.9
ResNet	95.0	96.8	+1.8

上述实验结果均为基于 ViT 模型的评估结果。图 5 进一步给出了随机优化方法和基于最大值的确定性方法在 VGG 和 ResNet 模型上的计算资源开销对比。实验中同样将 AoI 概率约束阈值设为 0.95，任务数量设为 100～1 000。从图 5 可以得到类似的结论：在 NVIDIA AGX Orin 设备上执行 VGG 推理任务时，随机优化方法相比确定性方法平均节省了 21.31%的资源开销；而执行 ResNet 推理任务时的资源消耗平均减少了 29.45%。



(a) VGG 模型

(b) ResNet 模型

图 5 随机优化方法和确定性方法在不同 DNN 模型上的计算资源开销对比

Fig. 5 Comparison of computing resource overhead between stochastic method and deterministic method on different DNN models

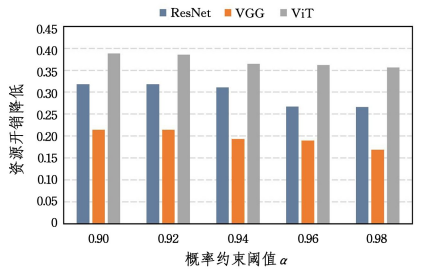


图 6 随机优化方法在取不同 AoI 概率约束阈值时节省的计算资源

Fig. 6 Saved computing resource by using the stochastic method with different threshold values for the probabilistic AoI constraint

本文随后对本文方法在不同的 AoI 概率约束阈值(即式中的 α 值)下的资源开销优化效果进行评估。如图 6 所示，当 α 取值在 0.9 至 0.98 之间，随机优化方法在执行 3 种 DNN 推理任务时均可以显著降低计算资源消耗；且当概率约束阈值的设置更加宽松时，对于计算资源的节省更加明显。以 ResNet 为例，当 α 值从 0.98 变化为 0.92 时，随机优化方法分别节省了 26.60%，26.69%，31.01%，31.75%，31.80%的资源开销。

为进一步评估本文提出的随机优化算法与其他元启发式算法的性能差异，本文选取 3 种经典元启发式算法(即遗传算法(Genetic Algorithm，GA)、粒子群优化算法(Particle Swarm Optimization，PSO)与萤火虫算法(Firefly Algorithm，FA))进行对比实验。出于公平比较的考虑，所有参与对比的元启发式算法的共有参数设置相同：种群规模均为 100，最大迭代次数均为 100。关键的算法模块(包括解的表达方式、解的初始化规则、适应度值计算方法)均保持一致。非共有参数的设置在相关文献所建议的区间内合理设置：GA 中的交叉概率设置为 0.8，变异概率设置为 0.1，精英个体数量设置为 2[25]；FA 中的光吸引系数设置为 1，吸引度衰减参数设置为 12[26]；PSO 中的初始惯性权重设置为 0.8，个体学习因子和社会学习因子分别设置为 0.5 和 2.5[27]。

本文采用方差分析(Analysis of Variance，ANOVA)方法评估元启发式算法在求解本文随机优化问题时的性能。其基本原理是通过比较组间差异和组内差异来判断不同方法的性能之间是否存在统计上的显著差异性[28]。组间差异反映了不同方法之间的均值差异，组内差异则衡量同一方法在不同测试实例上的差异性。ANOVA 方法可根据最小显著差异(Least Significant Difference，LSD)区间的形式表示方法之间性能差异的最低显著值，例如用于比较的方法的均值差异超出 LSD 区间的极值，则认为方法之间的性能差异是统计显著的。针对每一个用于评估的测试实例，本文使用参数重复运行算法求解 K 次，并量化其求解质量(Relative Percentage Deviation，RPD)：

$$RPD=\frac{1}{K}\left(\sum_{k=1}^{K}\frac{C_k-C_{\text{best}}}{C_{\text{best}}}\right)\times 100\% \tag{19}$$

其中，C_k 为当前算法第 k 次求解所获计算资源开销 C，C_{best} 为全体算法在对应数据上 K 次求解所获最小值。容易看出性能更佳的算法具有更小的 RPD 值。图 7 给出了在多个(多出口 DNN，任务数量，AoI 概率约束阈值)组合形成的测试实例下各随机优化算法的 ANOVA 分析结果，本文基于布谷鸟

算法所获 RPD 均值最小且与 3 种对比算法的 LSD 区间均无重叠，表明本文方法求解性能具有显著优势。

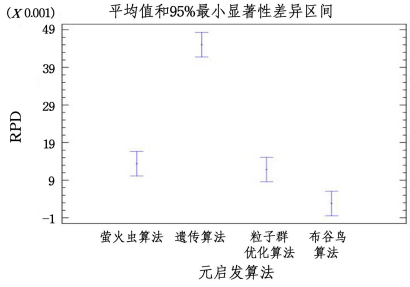


图 7 不同元启发式算法的求解质量对比

Fig. 7 Comparison of solution quality among different metaheuristic methods

最后，为定量评估不同元启发式算法求解本文的随机优化问题时的计算效率，引入标准化效率(Normalized Efficiency，NE)指标：

$$NE=\frac{T}{T_{\text{baseline}}} \tag{20}$$

其中，T 表示当前算法运行时长，T_{baseline} 表示基准算法运行时长，在实验中为本文算法运行时长，显然效率更高的算法具有更小的 NE。T 与 T_{baseline} 的值为算法在全体测试数据上运行 K 次所用总时长。如表 2 所列，本文算法的计算效率优于 3 种对比算法。

表 2 不同元启发式算法的计算效率对比

Table 2 Comparison of solution efficiency of different metaheuristic algorithms

元启发式算法	运行时间/ms	NE
本文算法	333	1.00
FA	43366	130.20
GA	754	2.26
PSO	357	1.07

结束语 本文提出了一种面向边缘智能应用的多出口 DNN 随机优化方法。考虑多出口 DNN 在不同出口一定概率提前退出所造成的系统 AoI 随机不确定性，引入系统 AoI 的概率约束，将 DNN 模型的资源开销最小化问题建模为一个随机优化问题。针对此随机优化问题，提出了一种基于布谷鸟搜索的元启发式算法，基于各出口的退出概率预测系统 AoI 的统计分布；根据给定的 AoI 阈值计算相应的资源消耗量，并以此评估布谷鸟个体的适应度值；根据布谷鸟搜索的优化原理迭代更新种群中的最优个体及相应的出口设置方案。实验结果验证了本文的随机优化方法能够在满足 AoI 概率约束的前提下显著降低 DNN 的计算开销。

未来的研究工作可围绕以下两方面展开。一方面，由于本文方法使用布谷鸟搜索的元启发式优化机制，寻优过程中存在陷入局部最优的可能，因此可在算法中引入局部搜索策略或和其他元启发式算法的优秀算子进行杂交，进一步提高算法的寻优能力。另一方面，考虑 DNN 模型的划分以及多边缘设备之间的协同，使得本文的随机优化方法能够适用于端-边协同以及边-边协同的边缘智能场景。

参 考 文 献

[1] ZHOU Z, CHEN X, LI E, et al. Edge intelligence: Paving the last mile of artificial intelligence with edge computing[J]. Proceedings of the IEEE, 2019, 107(8): 1738-1762.

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

[3] WANG X, HAN Y, WANG C, et al. In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning[J]. IEEE Network, 2019, 33(5): 156-165.

[4] SHUVO M M H, ISLAM S K, CHENG J, et al. Efficient acceleration of deep learning inference on resource-constrained edge devices: A review[J]. Proceedings of the IEEE, 2022, 111(1): 42-91.

[5] CUI W, ZHAO H, CHEN Q, et al. DVABatch: Diversity-aware multi-entry multi-exit batching for efficient processing of DNN services on GPUs[C]// USENIX Annual Technical Conference. USENIX, 2022: 183-198.

[6] DONG F, WANG H, SHEN D, et al. Multi-exit DNN inference acceleration based on multi-dimensional optimization for edge intelligence[J]. IEEE Transactions on Mobile Computing, 2022, 22(9): 5389-5405.

[7] ZHANG S, CUI W, CHEN Q, et al. PAME: Precision-aware multi-exit DNN serving for reducing latencies of batched inferences[C]// ACM International Conference on Supercomputing. ACM, 2022: 1-12.

[8] KAUL S, YATES R, GRUTESER M. Real-time status: How often should one update? [C]// International Conference on Computer Communications. IEEE, 2012: 2731-2735.

[9] XU C, XU Q, WANG J, et al. AoI-centric task scheduling for autonomous driving systems[C]// International Conference on Computer Communications. IEEE, 2022: 1019-1028.

[10] SORKHOH I, ASSI C, EBRAHIMI D, et al. Optimizing information freshness for MEC-enabled cooperative autonomous driving[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(8): 13127-13140.

[11] MA M, WANG Z, GUO S, et al. Cloud-edge framework for AoI-efficient data processing in multi-UAV-assisted sensor networks [J]. IEEE Internet of Things Journal, 2024, 11(14): 25251-25267.

[12] KADOTA I, SINHA A, MODIANO E. Scheduling algorithms for optimizing age of information in wireless networks with throughput constraints[J]. IEEE/ACM Transactions on Networking, 2019, 27(4): 1359-1372.

[13] ZHANG J, XIN W, LV D, et al. Multi-exit DNN inference acceleration for intelligent terminal with heterogeneous processors [J]. Sustainable Computing: Informatics and Systems, 2023, 40: 100906.

[14] KAYA Y, HONG S, DUMITRAS T. Shallow-deep networks: Understanding and mitigating network overthinking[C]// International Conference on Machine Learning. ACM, 2019: 3301-3310.

[15] LASKARIDIS S, VENIERIS S I, ALMEIDA M, et al. SPINN: Synergistic progressive inference of neural networks over device and cloud[C]// International Conference on Mobile Computing and Networking. ACM, 2020: 1-15.

[16] JU W,BAO W,YUAN D,et al. Learning early exit for deep neural network inference on mobile devices through multi-armed bandits[C]// International Symposium on Cluster,Cloud and Internet Computing. IEEE/ACM,2021:11-20.

[17] JU W,BAO W,GE L,et al. Dynamic early exit scheduling for deep neural network inference through contextual bandits[C]// International Conference on Information & Knowledge Management. ACM,2021:823-832.

[18] WANG X,NING Z,GUO S,et al. Minimizing the age-of-critical-information: An imitation learning-based scheduling approach under partial observations[J]. IEEE Transactions on Mobile Computing,2021,21(9):3225-3238.

[19] LI C,LIU Q,LI S,et al. Scheduling with age of information guarantee[J]. IEEE/ACM Transactions on Networking,2022,30(5):2046-2059.

[20] LI C,LI S,LIU Q,et al. Eywa:A general approach for scheduler design in AoI optimization [C]// International Conference on Computer Communications. 2023:1-9.

[21] LIN L,JU L,XUE C J,et al. Work or sleep: Freshness-aware energy scheduling for wireless powered communication networks with interference consideration[C]// Design Automation Conference. ACM/IEEE,2023:1-6.

[22] YANG X S,DEB S. Cuckoo search via Lévy flights[C]// World Congress on Nature & Biologically Inspired Computing. IEEE,2009:210-214.

[23] VISWANATHAN G M,AFANASYEV V,BULDYREV S V,et al. Lévy flights in random searches[J]. Physica A:Statistical Mechanics and its Applications,2000,282(1/2):1-12.

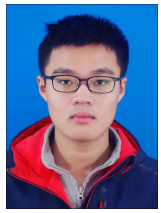
[24] MANTEGNA R N. Fast,accurate algorithm for numerical simulation of Lévy stable stochastic processes[J]. Physical Review E,1994,49(5):4677.

[25] GREFENSTETTE J J. Optimization of control parameters for genetic algorithms[J]. IEEE Transactions on Systems Man & Cybernetics,1986,16(1):122-128.

[26] MO Y B,MA Y Z,ZHENG Q Y. Optimal choice of parameters for firefly algorithm[C]// International Conference on Digital Manufacturing & Automation. 2013:887-892.

[27] WANG D F,MENG L. Performance analysis and parameter selection of PSO algorithms[J]. ACTA AUTOMATICA SINICA,2016,42(10):1552-1561.

[28] STAHLEL,WOLD S. Analysis of variance(ANOVA)[J]. Chemometrics and Intelligent Laboratory Systems,1989,6(4):259-272.



LI Zhoucheng, born in 2001, master, is a member of CCF(No. J4996G). His main research interests include edge computing and edge intelligence.

SUN Jin, born in 1983, Ph.D, professor, Ph.D supervisor, is a member of CCF (No. 43955M). His main research interests include computer architecture and embedded system.

(责任编辑:何杨)