边缘智能：边缘计算与人工智能融合的新范式

周知，于帅，陈旭
中山大学数据科学与计算机学院，广东 广州 510006

摘要
边缘计算与人工智能这两种高速发展的新技术之间存在着彼此赋能的巨大潜力。通过3个研究案例，展示协同边缘计算和人工智能这两种技术如何促进各自的进一步发展。从边缘计算赋能人工智能的维度，针对深度学习模型在网络边缘侧的部署，提出了基于边端协同的深度学习按需加速框架，通过协同优化模型分割和模型精简策略，实现时延约束下的高精度模型推理。从人工智能赋能边缘计算的维度，针对边缘计算服务的放置问题，提出了基于在线学习的自适应边缘服务放置机制和基于因子图模型的预测性边缘服务迁移方法。

关键词
边缘计算；人工智能；边缘智能；服务迁移

中图分类号：TP393 文献标识码：A doi: 10.11959/j.issn.2096-0271.2019013

Edge intelligence: a new nexus of edge computing and artificial intelligence

ZHOU Zhi, YU Shuai, CHEN Xu
School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China

Abstract
Artificial intelligence (AI) and edge computing (EC) represent two of today's most popular technologies. There is a great potential to coordinate these two emerging techniques to facilitate the further advent of both sides. Through three research cases, the profound benefits were demonstrated when AI and EC synergize. Specifically, from the perspective of EC for AI, to efficiently run deep learning at the network edge, a collaborative and on-demand deep neural network (DNN) co-inference framework with device-edge synergy was proposed. By applying DNN partitioning and right-sizing, it minimizes the inference latency under target accuracy. On the other hand, from the perspective of AI for EC, for the dynamical placement of edge computing services, two methods were proposed: an online-learning based adaptive service migration strategy and a factor graph model driven predictive service migration technique.

Key words
edge computing, artificial intelligence, edge intelligence, service migration

1 引言

近年来，随着全球范围内社会经济与科学技术的高速发展，信息通信技术（information communications technology，ICT）产业不断地涌现出了许多新兴技术。其中，有两种代表性技术被广泛认为对人类经济社会产生了巨大的推动力与深远的影响力。其一，作为人工智能领域代表性技术的深度学习，受益于算法、算力和数据集等方面的进步，近年来得到了突飞猛进的发展，并在无人驾驶、电子商务、智能家居和智慧金融等领域大展拳脚，深刻改变了人们的生活方式，提高了生产效率[1]。而另外一种技术则为从传统的云计算技术演化发展而来的边缘计算技术[2]，相比于云计算，边缘计算将强计算资源和高效服务下沉到网络边缘端，从而拥有更低的时延、更低的带宽占用、更高的能效和更好的隐私保护性。

然而，鲜为人知的是，人工智能和边缘计算这两种具有划时代意义的新技术目前正面临着各自进一步发展的瓶颈。一方面，对于深度学习技术而言，由于其需要进行高密度的计算，因此目前基于深度学习的智能算法通常运行于具有强大计算能力的云计算数据中心。考虑到当下移动终端设备的高度普及，如何将深度学习模型高效地部署在资源受限的终端设备，从而使智能更加贴近用户与物端，解决人工智能落地的“最后一公里”这一问题已经引起了学术界与工业界的高度关注[3-7]。另一方面，对于边缘计算而言，随着计算资源与服务的下沉与分散化，边缘计算节点将被广泛部署于网络边缘的接入点（如蜂窝基站、网关、无线接入点等）。边缘计算节点的高密度部署也给计算服务的部署带来了新的挑战：用户通常具有移动性，因此当用户在不同节点的覆盖范围间频繁移动时，计算服务是否应该随着用户的移动轨迹而迁移？显然，这是一个两难的问题，因为服务迁移虽然能够降低时延从而提升用户体验，但其会带来额外的成本开销（例如带宽占用和能源消耗）。

幸运的是，人工智能和边缘计算各自面临的发展瓶颈可以通过它们二者之间的协同得到缓解。一方面，对于深度学习而言，运行深度学习应用的移动设备将部分模型推理任务卸载到邻近的边缘计算节点进行运算，从而协同终端设备与边缘服务器，整合二者的计算本地性与强计算能力的互补性优势。在这种方式下，由于大量计算在与移动设备邻近的具有较强算力的边缘计算节点上执行，因此移动设备自身的资源与能源消耗以及任务推理的时延都能被显著降低，从而保证良好的用户体验。另一方面，针对边缘计算服务的动态迁移与放置问题，人工智能技术同样大有可为。具体而言，基于高维历史数据，人工智能技术可以自动抽取最优迁移决策与高维输入间的映射关系，从而当给定新的用户位置时，对应的机器学习模型即可迅速将其映射到最优迁移决策。此外，基于用户的历史轨迹数据，人工智能技术还可以高效地预测用户未来短期内的运动轨迹，从而实现预测性边缘服务迁移决策，进一步提升系统的服务性能。总体而言，边缘计算和人工智能彼此赋能，将催生“边缘智能”的崭新范式，从而产生大量创新研究机会。

本文将简要介绍笔者研究团队在融合边缘计算与人工智能两个方向上的初步探索。首先，从边缘计算赋能人工智能的维度出发，为了在网络边缘侧高效地运行深度学习应用，本文提出了一个基于边端协同的按需加速深度学习模型推理的优化框架——Edgent。为了实现降低模型推理时间，并按需保障模型精确率的双重目标，Edgent采取以下两种优化策略：深度学习模

型分割，自适应地划分移动设备与边缘服务器之间的深度神经网络模型计算量，以便在较小的传输时延代价下将较多的计算卸载到边缘服务器，从而降低整体端到端时延；深度学习模型精简，通过在适当的深度神经网络的中间层提前退出，进一步减小计算时延。其次，从人工智能赋能边缘计算的维度出发，针对边缘计算服务的动态迁移与放置问题，本文首先提出了一种用户自适应管理的在线服务放置机制。该机制能够通过在线学习（online learning）人工智能技术自适应复杂的用户行为和多变的边缘网络环境，从而辅助用户做出高效的服务迁移决策。最后，本文还将展示如何使用因子图模型（factor graph model）这一新兴人工智能技术实现用户位置预测，从而改善边缘服务动态迁移决策的质量。

2 边缘计算驱动实时深度学习

作为人工智能领域的主流技术之一，深度学习近年来得到了学术界与产业界的大力追捧。由于深度学习模型需要进行大量的计算，因此基于深度学习的智能算法通常存在于具有强大计算能力的云计算数据中心。随着移动终端和物联网设备的高速发展与普及，如何突破终端设备资源限制，从而将深度学习模型高效地运行在资源受限的终端设备这一问题已经引发了大量关注[3-7]。为解决这一难题，可考虑边缘计算赋能人工智能的思路，利用边缘计算就近实时计算的特性，降低深度学习模型推理的时延与能耗。

2.1 问题描述

常见的深度学习模型（如深度卷积神经网络）是由多层神经网络相互叠加而成的[1]。不同网络层的计算资源需求以及输出数据量都具有显著的差异性，那么一个直观的想法是将整个深度学习模型（即神经网络）切分成两部分[6]，其中计算量大的一部分卸载到边缘端服务器进行计算，而计算量小的一部分则保留在终端设备进行本地计算。显然，终端设备与边缘服务器协同计算的方法能有效降低深度学习模型的计算时延。但是，选择不同的模型切分点将导致不同的计算时间，需要选择最佳的模型切分点，从而最大限度地发挥终端与边缘协同的优势。

除了对模型进行切分外，加速深度学习模型推断的另一手段为模型精简，即选择完成时间更快的“小模型”，而不是选择对资源需求更高的“大模型”[8]。对于任意的深度学习任务，可以离线训练生成具有多个退出点的分支网络，退出点越靠后，模型越“大”，准确率也越高，但相应地推断时延也越大。因此，当深度学习任务的完成时间比较紧迫时，可以选择适当地牺牲模型的精确度，以换取更优的性能（即时延）。值得注意的是，此时需要谨慎权衡性能与精度之间的折中关系。

综合运用上述模型切分和模型精简两种方法调整深度学习模型推断时间的优化手段，并小心权衡由此引发的性能与精度之间的折中关系，本文定义如下研究问题：对于给定时延需求的深度学习任务，如何联合优化模型切分和模型精简这种方法，使得在不违反时延需求的同时最大化深度学习模型的精确度。

2.2 优化框架

针对上述问题，笔者团队提出了基于边缘与终端协同的深度学习模型推断优化框架Edgent。如**图1**所示，Edgent的优化逻辑分为3个阶段：离线训练阶段、在线优化阶段

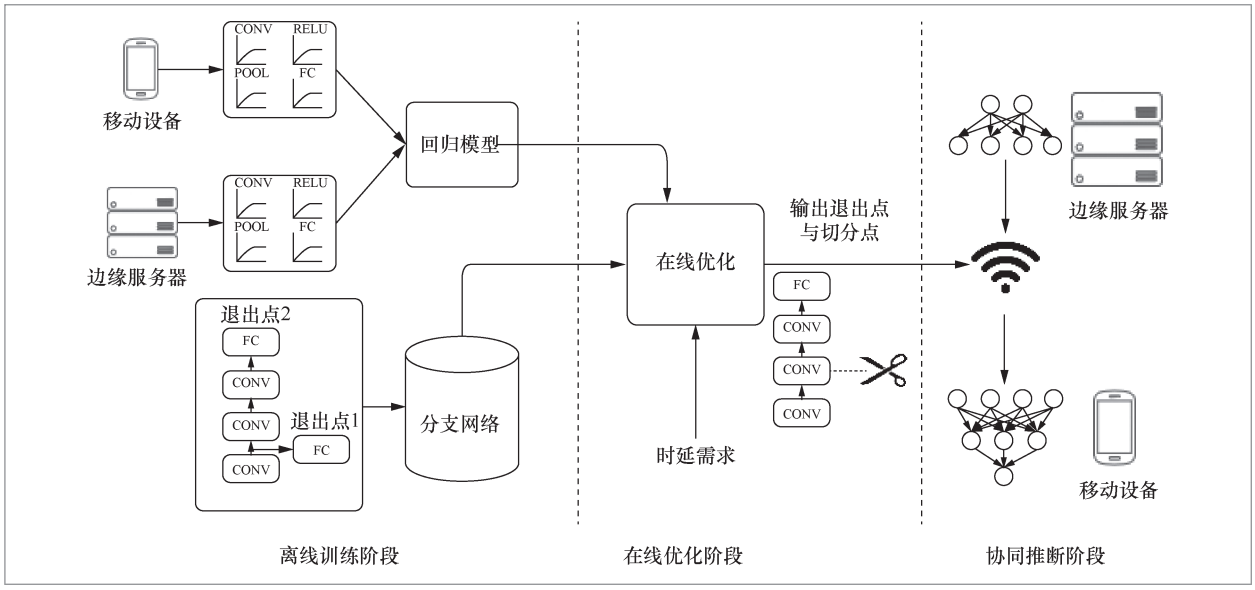

图1 基于边缘与终端协同的深度学习模型运行推断优化框架 Edgent

逻辑分为3个阶段：离线训练阶段、在线优化阶段以及协同推断阶段。

上述基于边缘服务器与终端设备协同的深度学习模型推断框架的设计思路为：在离线阶段，训练好满足任务需求的分支网络，同时为分支网络中的不同神经网络层训练回归模型，以此估算神经网络层在边缘服务器与在终端设备上的运行时延；在在线优化阶段，回归模型将被用于寻找符合任务时延需求的退出点以及模型切分点；在协同推断阶段，边缘服务器和终端设备将按照得出的方案运行深度学习模型。

（1）离线训练阶段

在离线训练阶段，Edgent需要执行以下两个初始化操作：分析边缘服务器与终端设备的性能，针对不同类型的深度学习模型网络层（如卷积层、池化层等）生成基于回归模型的时延估算模型。在估算网络层的运行时延时，Edgent会对每层网络层进行建模，而不是对整个深度学习模型进行建模，不同网络层的时延是由各自的自变量（如输入数据的大小、输出数据的大小）决定的，基于每层的自变量，可以建立回归模型，估算每层网络层的时延；训练带有多个退出点的分支网络模型，从而实现模型精简。这里笔者采用BranchyNet分支网络结构，在BranchyNet的结构下，可以设计并训练生成带有多个退出点的分支网络。需要注意的是，性能分析取决于设备（例如手机、VR头盔和智能手表等不同设备在运行同一深度学习模型时的性能表现不同），而深度学习模型是取决于应用的（例如物体识别和物体分类等不同计算机视觉应用对应的深度学习模型不同），因此在给定深度学习应用以及设备（即限定边缘服务器与终端设备）的情况下，以上两个初始化操作在离线阶段只需要完成一次。

（2）在线优化阶段

这个阶段的主要工作是利用离线训练的回归模型在分支网络中找出符合时延需求的退出点以及模型分割点。因为要最大化地给出方案的准确率，所以在该阶段中，笔者从最高准确率的分支开始，迭代地找出符合需求的退出点和切分点。在这个

务器之间链路的网络带宽，以便估算移动终端与边缘服务器间的数据传输时延。紧接着，Edgent沿着尺寸从大到小的网络分支，依次遍历每个网络分支上不同的分割点，并基于当前网络带宽和不同网络层计算时间估算所选分支网络与分割点对应的端到端时延与模型精确度。在遍历完所有的分支网络与切分点后，Edgent输出满足时延需求的所有网络分支与切分点组合中具有最大精确度的一个组合。

(3) 协同推断阶段

在协同推断阶段，根据在线优化阶段输出的最优网络分支与切分点组合，边缘服务器与移动终端对深度学习模型进行协同推断。实验表明，Edgent在提升深度学习应用实时性能方面表现优异，能在不同的计算时延需求下，实现高精度的模型推理（具体实验结果请见参考文献[9]）。

3 基于在线学习的自适应边缘服务放置机制

基于人工智能赋能边缘计算的思路，针对边缘计算服务的动态迁移与放置问题，笔者接下来介绍一种用户自适应管理的在线服务放置机制。该机制利用人工智能的在线学习技术，针对复杂的用户行为和多变的边缘网络环境进行自适应学习，从而辅助用户做出高效的服务迁移决策。

3.1 问题描述

尽管边缘计算能解决用户资源受限和云计算中时延过大的问题，但由于边缘节点服务覆盖范围较小，用户的移动将对服务质量产生极大的影响[10]。如**图2**所示，当用户从一个边缘节点服务区域移动至其他的节点服务区域时，需考虑是否进行服务迁移，以保证满意的服务质量[11]。一方面，用户可以选择继续让服务在原边缘节点进行处理，通过边缘节点之间的数据传输保证服务的连续性。另一方面，用户可以选择服务迁移，以此减小端到端时延。前者可能由于过长的网络距离导致较大的传输时延，后者则引入了服务迁移导致的额外开销。同时由于用户的个性化需求，不同的服务种类和迁移开销也增加了服务放置的难度，例如轻量级的应用用户更倾向于本地处理，计算密集型应用用户更倾向于云服务器处理，计算密集型且时延敏感型的应用用户更倾向于边缘服务器处理。为此，笔者提出了一种基于在线学习的自适应边缘服务放置机制，该机制可以有效地权衡时延与迁移代价之间的折中关系[12]。

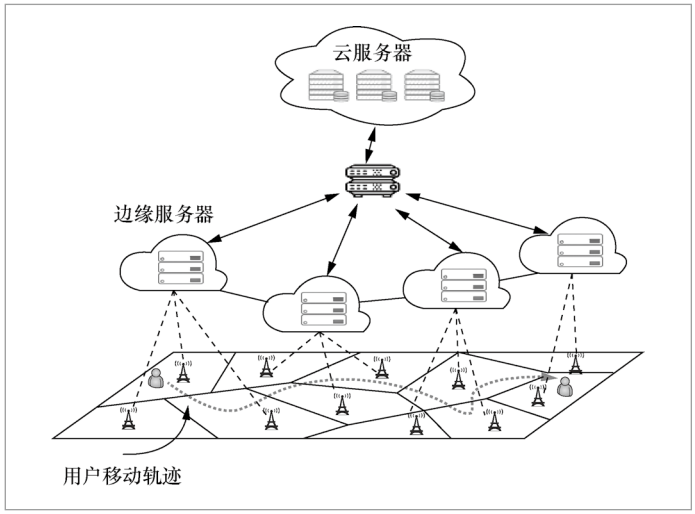

图 2 边缘网络中的用户移动示意

3.2 基于在线学习的自适应边缘服务放置算法

服务放置策略的制定通常与用户行为特性和网络环境有关。其中，行为特性主要包括用户的移动模型、请求服务的种类以及个人偏好等因素，网络环境主要包括边缘

网络可用的资源、节点之间的传输带宽等因素。对于用户来说，其当前时刻的行为特性是已知的，而网络环境是透明的。用户自适应管理机制通过在线学习的方式，根据用户的历史决策数据对动态网络环境的相关参数进行评估。随着环境探索次数的增加，用户对网络参数的评估也会愈发精准，进而优化其服务放置策略。为了方便刻画用户移动过程中的一系列决策行为，笔者将连续的时间段划分为一系列小的离散片段，在每个离散的时间片段中，用户确定当前的服务放置策略，同时用户行为和动态网络环境在该片段内保持不变。

为了有效地衡量边缘网络中的用户服务质量，笔者分别从用户可感知时延和服务迁移代价两方面进行研究。用户可感知时延从所需的计算时延和通信时延两部分进行考虑。计算时延主要取决于所需处理的请求服务数据量大小和放置节点的可用计算资源，如CPU的运算速度。通信时延主要由用户当前的接入时延和传输时延两部分构成。其中，接入时延主要取决于当前用户的位置和边缘路由器的状态。通常来说，为了减少通信能耗，用户将选择与最近的边缘路由器进行连接。而传输时延与网络状态（如边缘节点之间的带宽）、网络距离有关。服务迁移产生的额外运行代价（如带宽资源的占用等）主要取决于上一时刻和当前时刻的节点选择。由此，可以通过计算式（1）刻画用户的服务质量：

$$\omega_1 D_1(\pi_t) + \omega_2 D_2(\pi_t, y_t) + \omega_3 S(\pi_{t-1}, \pi_t) \tag{1}$$

其中，π_{t-1}、π_t分别表示上一时刻和当前时刻的服务放置决策（即在每一时刻t动态决策当前服务应当放置在云端服务器或某个边缘服务器），y_t表示用户当前的位置，ω_1、ω_2、$\omega_3(\geq 0)$分别为计算时延D_1、通信时延D_2和迁移代价S的非负权重参数，权重参数的设定与用户的个人偏好和服务种类有关：权重参数越大，表明对应时延或迁移代价的重要性越高。

如果可对未来长期时间T的用户行为特征和网络状态进行精准预测，那么就可以通过动态规划的方法求得长期服务的线下最优策略。然而在实际环境中，难以对以上用户和网络信息进行精准预测，同时对于每个决策时刻而言，由于缺少对网络环境参数的了解，用户收集系统信息将消耗额外的通信代价。

基于给定的优化目标，即用户服务质量，如何在未知后期用户行为和网络环境的情形下优化一系列服务放置决策，从而最小化可感知时延和服务迁移代价呢？针对此问题，笔者提出的自适应边缘服务放置机制将通过在线学习方法——上下文多臂老虎机（contextual multi-armed bandit）[13]不断探索动态系统，进而根据累计的历史数据分析评估整体的动态网络环境，为后续的放置策略制定提供更准确的评估。在每个离散时间片段的开始，用户将收集服务请求的行为特征，并用向量$b(t)=[\omega_1\lambda_t, \omega_2 L_t, \omega_3 \pi_{t-1}]$表示，其中包含当前用户请求服务处理量的大小$\lambda_t$、用户实时位置$L_t$以及上一时刻服务放置策略$\pi_{t-1}$。之后用户结合动态网络中可行放置策略$\pi_t$的网络估计参数$\mu_{\pi_t}$与行为特征$b(t)$对放置策略$\pi_t$产生的代价$c_{\pi_t}=b(t)^{\mathrm{T}}\mu_{\pi_t}$进行汤普森采样（Thompson sampling）[14]评估，从中选取代价最小的策略作为最终的放置策略。在该时间片段的末尾，用户将获得放置策略π_t的服务质量表现c_{π_t}，同时自适应管理机制将利用时间片段内的所有信息，更新用户行为特征和放置策略与服务质量表现之间的潜在关系，即服务放置策略的网络参数，具体算法如下。

算法1：基于在线学习的自适应服务放置算法。

初始化网络中计算节点i相应的用户特

征$B_i=I$、网络估计参数$\mu_i=0$以及其累积的上下文代价$f_i=0$。

对于每一个时间片段$t=1,2,\cdots,T$，执行以下操作。

● 根据计算节点i的网络参数μ_i，再结合当前的用户特征$b(t)$，依据汤普森采样评估其相应的代价ϕ_{i_t}。

● 选择其中最小估计代价$\phi_{i_t}^{\min}$的计算节点i_t^*作为当前的服务放置策略，并在时间片段的末尾收到相应的服务质量表现$c_{i_t^*}$。

● 更新选择计算节点i_t^*相应的用户特征$B_{i_t^*}=B_{i_t^*}+b(t)b(t)^{\mathrm{T}}$、$f_{i_t^*}=f_{i_t^*}+b(t)c_{i_t^*}$以及其网络估计参数$\mu_{i_t^*}=B_{i_t^*}^{-1}f_{i_t^*}$。

笔者对车辆在边缘网络中随机行驶的场景进行了模拟，以此对所提框架的有效性进行验证[12]。**图3**记录了基于在线学习的自适应边缘服务放置机制在不同时间片段下的服务质量表现，并同时与已知长期所有信息下的理论最优放置策略进行了比较。从结果可以看出，随着时间的增加，所提机制不断逼近最优的放置策略，这表明通过在线学习的方式能有效地优化策略制定。

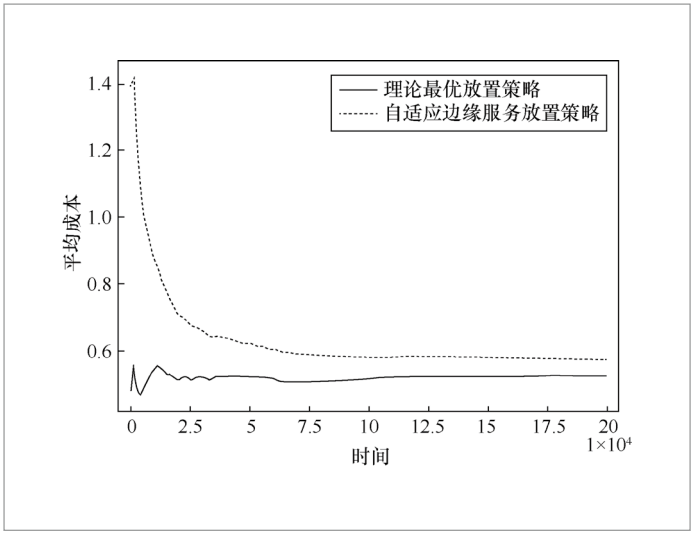

图 3 自适应服务放置策略与理论最优放置策略对比

4 基于因子图模型的预测性边缘服务迁移

前文介绍的基于在线学习的自适应边缘服务放置机制主要利用用户的统计性历史信息，接下来介绍另一种利用用户的预测性信息进行边缘服务放置优化的方法[15]。具体而言，笔者首先利用因子图模型对用户的位置进行预测，进而设计基于位置预测的边缘服务迁移算法。

4.1 基于因子图模型的用户位置预测

为了进行精准的位置预测，笔者从用户的历史信息中提取了多个维度的特征，如不同时刻的时间、位置、网络状态以及社交频率特征等，并使用因子图（factor graph，FG）模型[16]将其有效地整合到一个统一的框架之中。用户在t时刻的时间特征、社交频率特征以及网络状态特征分别用x_h^t、x_s^t和x_n^t表示，位置信息用y^t表示。其中，时间特征为从时间戳中得到的用户的时间信息（比如具体时刻、工作日或者周末等），社交频率特征则是对用户在该时段内各种社交行为特征（比如发布推文、刷新推文、转发推文、预览多媒体视频、观看多媒体视频等）分别进行频率计算得到的数据。网络状态特征是指用户当前所处的网络状态（Wi-Fi、蜂窝网络或者无网络连接）。空间位置信息即用户所处的具体位置。如**图4**所示，在同一时刻（例如t时刻），用户不同类型的特征与位置之间的关联可以用$f(x_i^t,y^t)$进行刻画，其中$i\in\{h,s,n\}$，h、s、n分别代表时间特征、社交频率特征以及网络状态特征。相邻时刻之间的位置关联用$w(y^{t-1},y^t)$表征，y^{t-1}和y^t分别代表用户在t-1和t时刻的位置信息。因子图模型可以很好地捕捉不同类型特征之间的关联，通过使用置信传播算法对该模型进行

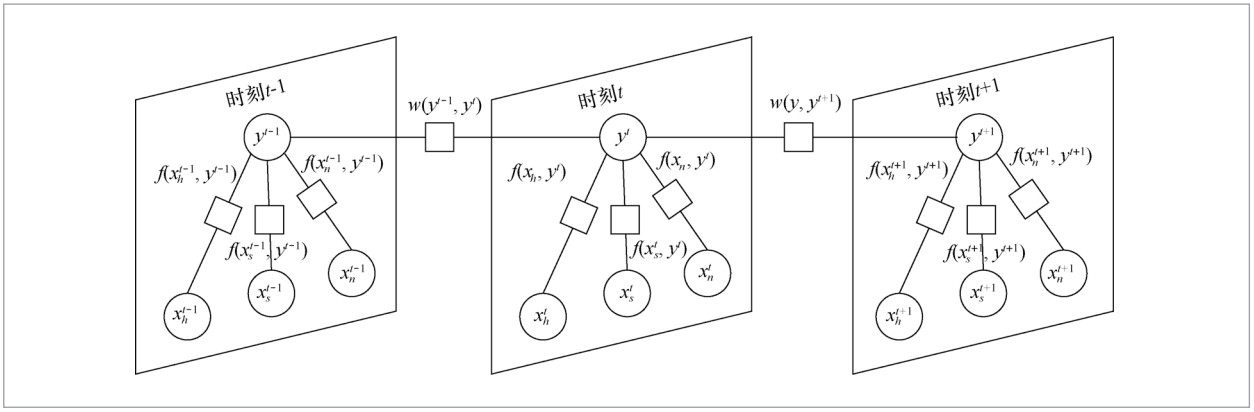

图 4 面向用户位置预测的因子图模型

训练，最终，可以通过最大化后验概率的方法获得用户下一时刻最有可能出现的位置。

同时，由于不同用户的有效特征具有较强的差异性，笔者利用ReliefF (RF) [14]算法对提取的特征进行了用户个性化筛选，以取得更好的预测效果。

4.2 基于位置预测的边缘服务迁移方法

在保证较低的服务时延的要求下，服务应该根据用户的移动性在不同的基站之间进行动态的迁移，然而频繁的服务迁移会带来巨大的代价（如额外的网络带宽使用和电力资源消耗）。因此，要考虑在给定长期成本预算C的情况下（T个时刻的预算）如何进行服务迁移决策，才能最小化用户总时延。该问题可以表示为：

$$\min \sum_{t=1}^{T} s(p^t, y^t) \tag{2}$$

$$\text{s.t.} \sum_{t=1}^{T} c(p^{t-1}, p^t) d^t \leqslant C \tag{3}$$

$$d^t \in \{0, 1\} \tag{4}$$

其中，p^t和y^t分别表示服务和用户在t时刻的位置，d^t表示在t时刻是否进行服务迁移的决策。$c(p^{t-1}, p^t)$为服务迁移所需的代价，这和带宽以及p^{t-1}和p^t之间的地理距离有关。用户在t时刻感知到的时延用$s(p^t, y^t)$表示。这一问题的关键是每一时刻下的迁移决策d^t。

一个具体的例子是，若用户在t时刻从位置a移动到位置b，在t+1时刻又返回到位置a，那么在t时刻是否需要服务迁移呢？如果无法预知用户下一时刻的位置，一种最简单直接的方法就是让服务一直跟随用户进行迁移，这样虽然能够保证用户的服务质量，但是会带来巨大的服务迁移代价。当能够预知用户下一时刻的位置时，考虑到长期预测的限制，便能够在t时刻进行更加明智的决策，以避免服务反复迁移带来的代价。这一例子充分体现了位置预测在迁移决策过程中的重要性，因此笔者考虑了基于位置预测的个性化服务迁移。

4.3 预测性延迟迁移算法

首先，笔者构思了延迟迁移 (lazy migration, LM) 算法。该算法通过“向后看”，对当前时刻进行服务迁移所需的成本和从上一迁移时刻到目前累积的非迁移时延进行权衡，进而决定是否迁移。如果累积时延不超过当前的迁移代价，那么将一直延迟服务的迁移。进一步地，利用预测的用户下一时刻的位置信息，笔者对算法进行了改进。在

进行累积时延和迁移代价的权衡之前，笔者引入了一种“向前看”的机制。考虑到用户未来可能的时延比过去的时延更具有决策的指导意义，笔者首先通过“向前看”估计用户当前时刻和下一时刻可能会感知到的时延，并和当前迁移代价进行权衡。如果预知到在不迁移情况下，用户接下来两个时间片的时延很大，则直接进行服务的迁移，否则就使用延迟迁移算法进行决策，这就是预测性延迟迁移 (predictive lazy migration, PLM) 算法，见算法2。

算法2：预测性延迟迁移算法。

输入：用户的位置集合$Y=\{y^1,y^2,\cdots,y^T\}$，预测的下一位置集合$Z=\{z^1,z^2,\cdots,z^T\}$，向前看参数μ_1和向后看参数μ_2。

输出：用户访问总时延和总迁移代价。

算法过程：

初始化$t\leftarrow1,m\leftarrow y^1,S\leftarrow0,C\leftarrow0$

$\tau\leftarrow1$./*上次迁移的时间*/

for $t = 1, 2, 3, \cdots, T$ do

  if $y^t\neq m$ then

   if $c(m,y^t)<\mu_1(s(m,y^t)+s(m,z^t)$ then

    $m\leftarrow y^t$;

    $\tau=t$;

   $C\leftarrow C+c(m,y^t)$;

   else

    if $c(m,y^t)<\mu_2\sum_{v=\tau}^{t-1}s(m,y^v)$ then

     $m\leftarrow y^t$

     $\tau=t$;

     $C\leftarrow C+c(m,y^t)$;

    else

     $S\leftarrow S+s(m,y^t)$;

    end if

   end if

  end if

end for

图5展示了在不同预算限制下，笔者提出的算法相较于不迁移情况下的时延减少率。延迟迁移算法 (LM—非预测性和PLM—预测性) 由于考虑了非迁移情况下的服务时延和迁移情况下的服务代价之间的权衡，相比于已有参考文献的非预测性分期偿还 (original amortization, OA) 算法和预测性分期偿还 (predictive amortization, PA) 算法，能够显著降低服务总时延[15]。同时，PLM算法相较于LM算法降低了13%的服务时延。该实验结果表明，利用精准的位置预测，笔者能够进行更加明智的服务迁移决策，从而减少用户感知到的服务时延，提升用户的服务质量。

5 结束语

本文初步探讨了如何通过协同边缘计算和人工智能这两种高速发展的新技术解决它们各自发展面临的难题。具体而言，从边缘计算赋能人工智能的视角，针对深度学习在终端设备高效部署的“最后一公里”难题，本文介绍了基于边端协同的按需加速深度学习模型推理的优化框架，通过协同优化模型分割和模型精简这两种不同的策略，实现时延约束下的高精度模型推理。

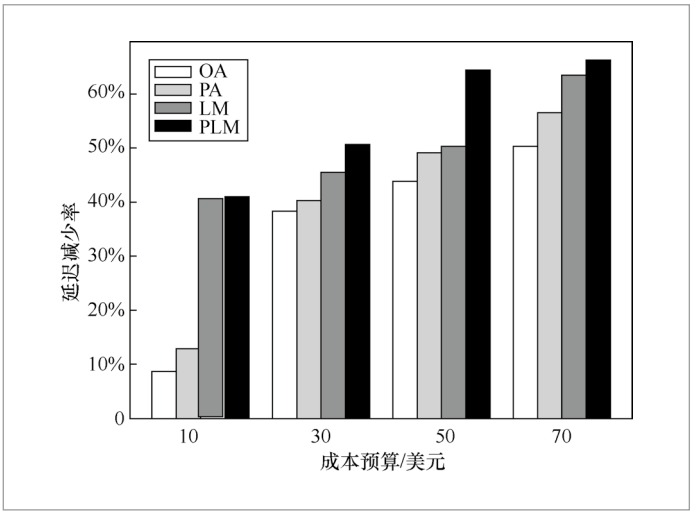

图 5 不同迁移代价约束下的延迟减少率

另外，从人工智能赋能边缘计算的视角，针对边缘节点大规模高密度部署带来的服务动态迁移问题，本文分别介绍了基于在线学习的自适应边缘服务放置机制以及基于位置预测的边缘服务迁移方法。未来，边缘智能作为边缘计算和人工智能彼此赋能的崭新范式，将催生大量的创新研究机会，在智能物联网、智能制造、智慧城市等众多领域具有广泛的应用前景。

参考文献：

[1] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2015, 521(7553): 436.

[2] SHI W S, CAO J, ZHANG Q Y, et al. Edge computing: vision and challenges[J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646.

[3] IANDOLA F N, MOSKEWICZ M W, ASHRAF K, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size[J]. Computer Science, 2016, arXiv:1602.07360.

[4] WU J X, CONG L, WANG Y H, et al. Quantized convolutional neural networks for mobile devices[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 26-July 1, 2016, Las Vegas, USA. Piscataway: IEEE Press, 2016: 4820-4828.

[5] HAN S, MAO H Z, DALLY W J. Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding[J]. Computer Science, 2015, arXiv:1510.00149.

[6] KANG Y P, HAUSWALD J, GAO C, et al. Neurosurgeon: collaborative intelligence between the cloud and mobile edge[C]//The 22nd International Conference on Architectural Support for Programming Languages and Operating Systems, April l8-12, 2017, Xi'an, China. New York: ACM Press, 2017: 615-629.

[7] LANE N, BHATTACHARYA S, MATHUR A, et al. DXTK: enabling resource-efficient deep learning on mobile and embedded devices with the DeepX toolkit[C]//The 8th EAI International Conference on Mobile Computing, Applications and Services, November 30-December 1, 2016, Cambridge, UK. New York: ACM Press, 2017: 98-107.

[8] TEERAPITTAYANON S, MCDANEL B, KUNG H T. BranchyNet: fast inference via early exiting from deep neural networks[J]. Computer Science, 2017, arXiv:1709.01686.

[9] LI E, ZHOU Z, CHEN X. Edge intelligence: on-demand deep learning model co-inference with device-edge synergy[C]//ACM Workshop on Mobile Edge Communications, August 20, 2018, Budapest, Hungary. New York: ACM Press, 2018: 31-36.

[10] MACH P AND BECVAR Z. Mobile edge computing: a survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 3(19): 1628-1656.

[11] WANG S, XU J, ZHANG N, et al. A survey on service migration in mobile edge computing[J]. IEEE Access, 2018, 6(24): 511-528.

[12] OUYANG T, LI R, CHEN X, et al. Adaptive user-managed service placement for mobile edge computing: an online learning approach[C]//IEEE International Conference on Computer Communications, April 28-May 2, 2019, Paris, France. Piscataway: IEEE Press, 2019.

[13] AGRAWAL S, GOYAL N. Thompson sampling for contextual bandits with linear payoffs[C]//The 30th International Conference on International Conference on Machine Learning, June 16-21, 2013, Atlanta, USA. [S.l.:s.n.], 2013: 1220-1228.

[14] RUSSO D J, ROY B V, KAZEROUNI A, et al. A tutorial on thompson sampling[J]. Foundations and Trends in Machine Learning, 2018, 1(11): 1-96.

[15] WU Q, CHEN X, ZHOU Z, et al. Mobile social data learning for user-centric location prediction with application in mobile edge service migration[J]. IEEE Internet of Things Journal, 2019: accepted.

[16] KSCHISCHANG F R, FREY B J, LOELIGER

H A. Factor graphs and the sum-product algorithm[J]. IEEE Transactions on Information Theory, 2002, 2(47): 498-519.

作者简介

周知 (1993-), 男, 中山大学数据科学与计算机学院特聘研究员, 主要研究方向为云计算、边缘计算和分布式系统。

于帅 (1986-), 男, 博士, 中山大学数据科学与计算机学院在站博士后, 主要研究方向为无线通信、移动计算、机器学习等。

陈旭 (1986-), 男, 中山大学数据科学与计算机学院教授、博士生导师, 主要研究方向为边缘计算、边缘智能、智能物联网等。

收稿日期: 2019-01-18

通信作者: 陈旭, chenxu35@mail.sysu.edu.cn

基金项目: 国家重点研发计划基金资助项目 (No.2017YFB1001703) ; 国家自然科学基金资助项目 (No. U1711265, No.61802449) ; 广东省引进创新创业团队基金资助项目 (No.2017ZT07X355) ; 广东省自然科学基金资助项目 (No.2018A030313032)

Foundation Items: National Key Research and Development Program of China (No.2017YFB1001703), The National Natural Science Foundation of China (No. U1711265, No.61802449), Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No. 2017ZT07X355), Guangdong Natural Science Funds (No.2018A030313032)