基于深度强化学习的 SDN 服务质量智能优化算法

廖岑卉珊[1]，陈俊彦[1]，梁观平[2]，谢小兰[1]，卢小烨[1]
（1. 桂林电子科技大学计算机与信息安全学院，广西 桂林 541004；
2. 国防科技大学计算机学院，湖南 长沙 410073）

摘 要：深度强化学习具有较强的决策能力和泛化能力，常被应用于软件定义网络（SDN, software defined network）的服务质量（QoS, quality of service）优化中。但传统深度强化学习算法存在收敛速度慢和不稳定等问题。提出一种基于深度强化学习的服务质量优化算法（AQSDRL, algorithm of quality of service optimization based on deep reinforcement learning），以解决 SDN 在数据中心网络（DCN, data center network）应用中的 QoS 问题。AQSDRL 引入基于 softmax 估计的深层双确定性策略梯度（SD3, softmax deep double deterministic policy gradient）算法实现模型训练，并采用基于 SumTree 的优先级经验回放机制优化 SD3 算法，以更大的概率抽取具有更显著时序差分误差（TD-error, temporal-difference error）的样本来训练神经网络，有效提升算法的收敛速度和稳定性。实验结果表明，所提 AQSDRL 与现有的深度强化学习算法相比能够有效降低网络传输时延，且提高网络的负载均衡性能。

关键词：深度强化学习；软件定义网络；服务质量；数据中心网络；SumTree
中图分类号：TP393
文献标志码：A
doi: 10.11959/j.issn.2096−3750.2023.00316

Quality of service optimization algorithm based on deep reinforcement learning in software defined network

LIAO Cenhuishan[1], CHEN Junyan[1], LIANG Guanping[2], XIE Xiaolan[1], LU Xiaoye[1]
1. School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
2. College of Computer, National University of Defense Technology, Changsha 410073, China

Abstract: Deep reinforcement learning has strong abilities of decision-making and generalization and often applies to the quality of service (QoS) optimization in software defined network (SDN). However, traditional deep reinforcement learning algorithms have problems such as slow convergence and instability. An algorithm of quality of service optimization algorithm of based on deep reinforcement learning (AQSDRL) was proposed to solve the QoS problem of SDN in the data center network (DCN) applications. AQSDRL introduces the softmax deep double deterministic policy gradient (SD3) algorithm for model training, and a SumTree-based prioritized empirical replay mechanism was used to optimize the SD3 algorithm. The samples with more significant temporal-difference error (TD-error) were extracted with higher probability to train the neural network, effectively improving the convergence speed and stability of the algorithm. The experimental results show that the proposed AQSDRL effectively reduces the network transmission delay and improves the load balancing performance of the network than the existing deep reinforcement learning algorithms.

Key words: deep reinforcement learning, SDN, QoS, DCN, SumTree

---

收稿日期：2022−08−30；**修回日期：**2022−12−04
通信作者：陈俊彦，chenjunyan@guet.edu.cn
基金项目：广西自然科学基金资助项目（No.2020GXNSFDA238001）；广西高校中青年教师科研基础能力提升项目（No.2020KY05033）
Foundation Items: The Guangxi Natural Science Foundation (No.2020GXNSFDA238001), The Guangxi Project to Improve the Scientific Research Basic Ability of Middle Aged and Young Teachers (No.2020KY05033)

0 引言

近年来，随着科学技术的飞速发展，基于互联网的应用程序日益增多，网络会议、视频和游戏等产生的互联网流量呈指数级增长，互联网服务对人类生产生活的重要性日渐凸显。互联网流量的空前增长以及各种应用程序严格的服务质量（QoS, quality of service）要求给传统的数据中心网络带来了巨大的压力。

软件定义网络（SDN, software defined network）以其数据平面和控制平面解耦合、控制器管理全局网络和可编程等特性，成为数据中心网络的关键技术之一。SDN 控制器可以从全局角度改善网络性能，包括网络吞吐量、丢包率、传输时延和负载均衡等，这对于保障网络服务质量有着重要意义。

在软件定义网络中，传统的 QoS 优化方案大多基于启发式算法[1-3]。但由于缺乏数据学习的历史经验，启发式算法只能为特定问题建立模型。当网络发生变化时，难以确定网络参数，可扩展性有限，难以保障网络服务质量。

近年来，深度学习（DL, deep learning）以其强大的学习能力和卓越的性能优势逐渐被应用于 SDN 优化[4-8]。然而，深度学习需要大量的数据集进行模型训练，且其泛化能力较差，这使得动态网络的性能优化变得困难。与深度学习相比，强化学习（RL, reinforcement learning）采用在线学习的方式进行模型训练，通过不断地探索、学习和尝试来改变其行为以获得最佳回报。强化学习可以根据环境的状态和奖励反馈生成动作决策，具有较强的决策能力和泛化能力。因此，有研究者把强化学习应用在网络的 QoS 优化中，让智能体与网络环境进行交互，达到自适应优化网络性能的效果[9-13]。但强化学习算法的感知能力较弱，且没有存取以往的经验，因此在动态网络场景中强化学习算法的学习能力受限。深度强化学习（DRL, deep reinforcement learning）结合了深度学习的感知能力，可有效提升强化学习算法的收敛性能，因此得到许多研究者的关注，并把传统的深度强化学习算法应用于 SDN 性能优化，如深度 Q 网络（DQN, deep Q-network）[14-15]、深度确定性策略梯度（DDPG, deep deterministic policy gradient）[16-21]、双延迟深度确定性策略梯度（TD3, twin delayed deep deterministic policy gradient）[22-24]等。但是 DQN 算法只能在离散的动作空间进行决策，能力有限。而确定性策略梯度算法，如 DDPG 和 TD3，仍然存在收敛速度慢和不稳定等问题，会严重影响网络性能。

针对上述问题，本文提出了一种基于深度强化学习的服务质量优化算法（AQSDRL, algorithm of quality of service optimization based on deep reinforcement learning），以解决 SDN 在负载均衡和网络时延的 QoS 问题。AQSDRL 采用基于 softmax 估计的深层双确定性策略梯度（SD3, softmax deep double deterministic policy gradient）算法[25]实现模型训练，智能体通过接收数据平面中的网络链接信息及上一时刻的评估奖励值生成链路间权重矩阵，随后使用 Dijkstra 算法在网络主机之间找到最优路径并创建转发流表。SD3 算法可有效改善强化学习中 DDPG 算法的高估问题和 TD3 算法的低估偏差，在算法收敛后具有较好的稳定性。同时，本文进一步优化 SD3 算法的经验池采样方法，采用基于 SumTree 的优先级经验回放机制实现 SD3 的经验值回放。实验证明，本文提出的优化方案可以有效地提升强化学习算法的收敛速度和稳定性，减小强化学习模型在试错阶段造成的网络性能影响。

1 相关工作

传统的路由优化方案通常基于开放最短路径优先（OSPF, open shortest path first）[26]、等价多路径（ECMP, equal-cost multipath）[27]或启发式优化算法转发流量，OSPF 协议将所有流请求单独路由到最短路径，ECMP 协议通过同时使用多条链路增加传输带宽。但是，这些基于固定转发规则的方法容易出现链路拥塞，无法满足流量指数增长的需求。由于缺乏数据学习的历史经验，启发式算法只能为特定问题建立模型。当网络发生变化时，难以确定网络参数，可扩展性有限，难以保障网络服务质量。

基于传统网络的 QoS 策略无法处理日益复杂的任务，已不能满足 SDN 环境的需求。近年来，深度学习以其强大的学习算法和卓越的性能优势逐渐被应用于计算机网络领域。Zou 等[4]提出一种基于深度学习的方法 Deep TSQP，通过特征集成来执行时间感知的服务 QoS 预测任务。文献[5-8]利用长短期记忆（LSTM, long short-term memory）网络优化 SDN。其中，Chen 等[5]在 SDN 应用平面中利用 LSTM 网络预测流量，抵消时延的影响，以解决

SDN 负载均衡问题。然而，深度学习需要大量的数据集进行训练，且缺乏与环境交互的能力，泛化能力较差，这使得动态网络的性能优化变得困难。

与深度学习相比，强化学习采用在线学习方式进行模型训练，通过不断地探索、学习和尝试来改变其行为以获得最佳回报。因此，强化学习无须提前训练模型，可以根据环境状态和奖励反馈更改自身的动作，具有较强的决策能力和泛化能力。文献[9-10]结合 RL 与 SDN 优化 QoS。Younus 等[11]使用 *Q*-learning 算法优化网络性能，有效提升了网络收敛速度。Casas-Velasco 等[12]提出一种强化学习和软件定义网络的智能路由（RSIR, reinforcement learning and software-defined networking intelligent routing）算法，通过最小化 *Q*-learning 智能体奖励值的策略来避免选择具有高时延和高丢包的链路，在链路吞吐量、丢包率和时延方面优于 Dijkstra 算法。Al-Jawad 等[13]使用 *Q*-learning 算法实现了 QoS 与用户体验质量（QoE, quality of experience）之间的权衡。然而，*Q*-learning 算法通过 *Q* 表的形式学习最优动作价值函数，没有存取以往的经验，缺乏深度学习中的数据表达能力。

深度强化学习在强化学习中使用深度神经网络的数据表达能力实现决策，为解决复杂的网络 QoS 问题开辟了一条新途径。文献[14-15]将 DQN 和 LSTM 网络结合实现网络路由策略优化。然而，DQN 的决策必须在离散的动作空间进行，无法对高维的状态集合进行求解。DDPG 算法解决了 DQN 只能在离散的动作空间进行决策的问题。文献[16-18]将 DDPG 应用于 SDN 路由优化中，实现智能优化网络，有效降低网络时延。兰巨龙等[19]提出深度强化学习的软件定义网络 QoS 优化算法 R-DRL，将 DDPG 算法与 LSTM 网络相结合，生成满足 QoS 优化目标的动态流量调度策略。Mai 等[20]提出一种基于 DDPG 的切片优化算法 TMDDPG，SDN 控制器将 LoRa 网关的物理资源分为多个虚拟网络，以提供不同的 QoS 保证。文献[21]利用 SumTree 结构更新经验池的抽样方法，改进了 DDPG 中经验回放机制的随机抽取策略。虽然 DDPG 算法能够在连续的动作空间上有效地学习，但是它在选择动作时会盲目地选择 Q 值最大的动作，这使得 DDPG 存在高估问题。

为了解决 DDPG 算法的高估问题，Fujimoto 等[22]提出了 TD3 算法，在价值网络中参考截断的 Double *Q*-learning 算法，并使用延迟策略更新和目标策略平滑技术，在一定程度上改善了 DDPG 存在的高估和高方差。孙鹏浩等[23]提出了一种基于 TD3 的智能路由技术，确保路由策略能够动态适应网络流量的变化。文献[24,28-29]使用牵引理论，解决目前 DRL 算法应用于网络路由策略生成中普遍存在的可扩展性差、鲁棒性低问题。

虽然 TD3 算法缓解了高估问题，但它在使用取最小值方式进行价值截取时可能导致较大的低估偏差，也会影响性能。SD3 是基于 TD3 的优化算法，在连续控制中使用 Boltzmann softmax 算子进行值函数估计。相对于 TD3 算法和 DDPG 算法，SD3 算法可以有效地改善高估和低估偏差，具有较快的收敛速度。受文献[21]和 SD3 算法的启发，本文提出一种基于深度强化学习的服务质量优化算法（AQSDRL），使用 SumTree 优化 SD3 中经验回放机制的随机抽取策略，以更大的概率抽取具有更显著时序差分误差（TD-error, tempoal-difference error）的样本来训练神经网络，有效提升算法的收敛速度和稳定性，解决网络性能不稳定、服务质量差的问题。

2　系统架构

本文提出的 AQSDRL 架构主要包括数据平面、控制平面和应用平面，如图 1 所示。各个平面的具体功能如下。

1) 数据平面

数据平面由支持 SDN 的基础设备组成，负责网络数据传输。数据平面实时感知网络信息，如网络拓扑和各项性能指标，并将网络信息通过南向接口传递给控制平面。同时接收控制平面下发的策略，如 OpenFlow 流表。当有新的 OpenFlow 流表到达时，根据优先级查询流表项，对数据进行逐一匹配，并根据匹配规则和匹配结果进行相应的处理。数据包如果不匹配任何流表项，将会被丢弃或上报给控制器。

2) 控制平面

控制平面由 SDN 控制器组成，连接数据平面和应用平面。控制平面通过北向接口将从数据平面收集到的网络信息传递给应用平面，并通过南向接口传递从应用平面接收到的动作策略给数据平面。

3) 应用平面

应用平面运行 DRL 智能体，负责生成网络服务质量优化策略。智能体采用 SD3 算法，将从北向接口收集到网络信息作为状态输入，利用 SumTree

经验回放机制优化经验采样并执行动作，随后将生成的动作策略传达给控制平面。如此往复，直至学习到使得奖励值最大的动作策略。

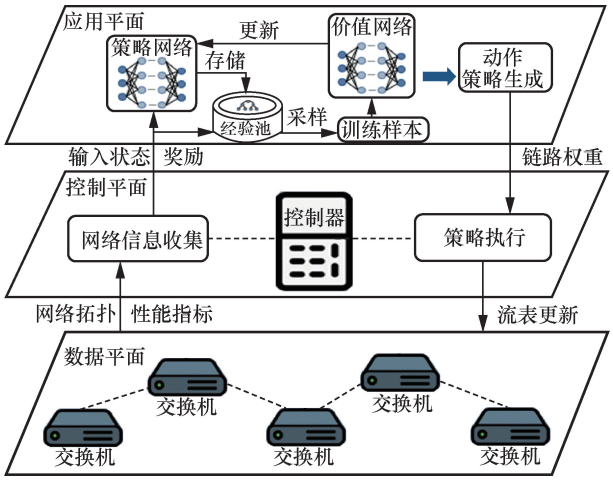


图 1　AQSDRL 架构

3　服务质量智能优化算法

3.1　基于优先级经验回放机制的 SD3 算法优化方案

本文提出的 AQSDRL 采用 SD3[25]算法实现模型训练。SD3 继承了 TD3 的优点，在价值网络中参考了截断的 Double Q-learning 算法，并采用两个目标价值网络中的最小值进行价值估计，在一定程度上缓解了高估问题；同时改进了 TD3 存在的缺点，在连续控制中使用 Boltzmann softmax 算子进行值函数估计，解决了使用最小值方法导致的低估偏差。Boltzmann softmax 函数可以用作简单但有效的动作选择策略以权衡探索和利用，平滑优化环境，从而有助于经验学习。

为了提升 SD3 算法的收敛速度和稳定性，本文提出了一种 SD3 算法的优化方案 SumTree-SD3，采用基于 SumTree 的优先级经验回放机制实现 SD3 的经验值回放，进一步优化 SD3 算法的经验池采样方法。SumTree-SD3 算法由策略网络、价值网络和经验回放机制组成，通过循环迭代找到使得奖励最大化的最优策略，基于优先级经验回放机制的 SumTree-SD3 算法如图 2 所示。各部分功能如下。

1) 策略网络

策略网络采用确定性策略函数进行策略优化，输出策略动作。它包含 4 个神经网络，即 2 个用于训练和学习的在线策略网络和 2 个用于防止训练数据被篡改的目标策略网络。策略网络通过策略梯度下降来更新参数 ϕ_i。

$$\phi_i \leftarrow \frac{1}{N}\sum_{s_t}\left[\nabla_{\phi_i}(\pi(s_t;\phi_i))\nabla_{a_t}Q_i(s_t,a_t;\theta_i)\Big|_{a_{t+1}=\pi(s_t;\phi_i)}\right] \tag{1}$$

其中，i 为策略网络的序号，$i=1,2$；N 是小批量样本的数量；s_t 表示 t 时刻的状态；a_t 表示 t 时刻的动作（具体在第 3.2 节描述）；θ_i 是对应的在线价值网络参数；$\pi(s_t;\phi_i)$ 和 $Q_i(s_t,a_t;\theta_i)$ 是用来近似策略函数和动作价值函数的神经网络。目标策略网络从经验池中通过经验值采样得到下一个状态的最

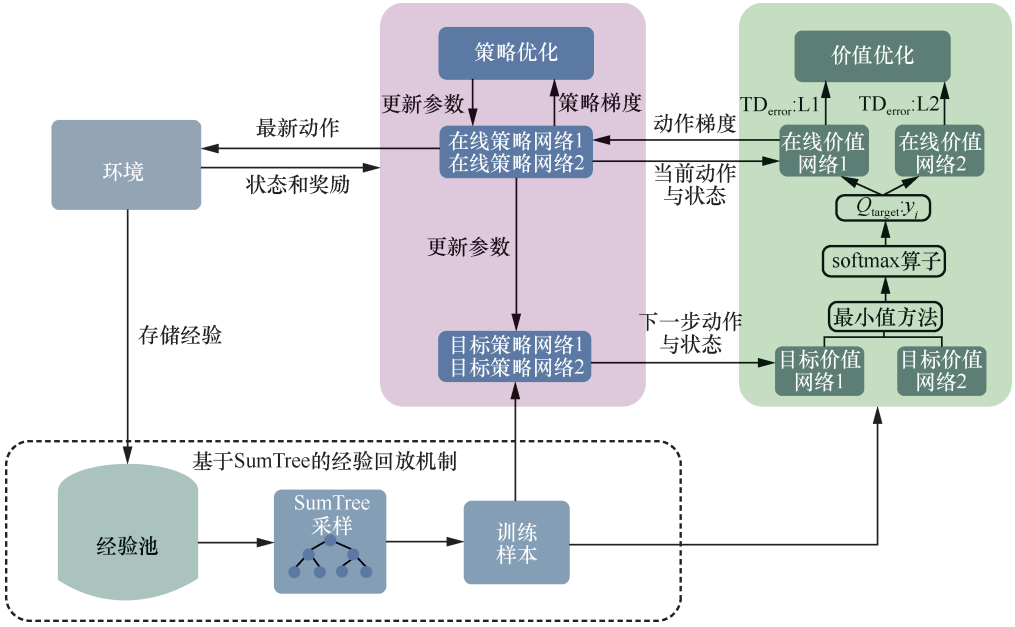


图 2　基于优先级经验回放机制的 SumTree-SD3 算法

优动作，并定期更新网络参数 ϕ_i^-。

$$\phi_i^- \leftarrow \tau\phi_i + (1-\tau)\phi_i^- \tag{2}$$

其中，τ 表示网络更新步长。

2) 价值网络

价值网络采用 Q 函数估计策略动作。它包含 4 个神经网络：2 个在线价值网络和 2 个目标价值网络。目标价值网络会产生两个 Q 值 Q_j（$j=1,2$），通过最小化截取得到较小的 Q 值

$$\hat{Q}(s_{t+1},a_{t+1}) \leftarrow \min_{j=1,2}(Q_j(s_{t+1},a_{t+1};\theta_j^-)) \tag{3}$$

其中，θ_j^- 为对应的目标价值网络参数。接着使用 Boltzmann softmax 算子对 Q 值进行处理，如式(4)所示。

$$\begin{aligned}&\text{soft}\max_\beta(\hat{Q}(s_{t+1},*)) \leftarrow \\ &\mathbb{E}_{a_{t+1}\sim p}\left[\frac{\exp(\beta\hat{Q}(s_{t+1},a_{t+1}))\hat{Q}(s_{t+1},a_{t+1})}{p(a_{t+1})}\right] / \\ &\mathbb{E}_{a_{t+1}\sim p}\left[\frac{\exp(\beta\hat{Q}(s_{t+1},a_{t+1}))}{p(a_{t+1})}\right]\end{aligned} \tag{4}$$

其中，β 是 softmax 算子的参数，$p(a_{t+1})$ 是概率密度函数。随后计算目标 Q 值 y_i，如式(5)所示。

$$y_i \leftarrow r + \gamma(1-d)\text{softmax}_\beta(\hat{Q}(s',*)) \tag{5}$$

其中，r 是奖励，γ 是奖励折扣因子，$\gamma \in [0,1]$，d 是终止符号，当迭代结束时 d=1，否则 d=0。随后将得到的 y_i 送入在线价值网络，根据贝尔曼损失更新相关的网络参数，将动作梯度送入在线策略网络更新网络参数。最后采用软更新的方式，更新目标价值网络参数 θ_i^-。

$$\theta_i^- \leftarrow \tau\theta_i + (1-\tau)\theta_i^- \tag{6}$$

3) 经验回放机制

经验池用于存储训练时产生的当前状态、动作、奖励和新状态，策略网络和价值网络通过经验回放机制处理模型获取训练样本，每次迭代都从重放缓冲区中随机抽取一小批样本来训练模型。本文采用基于 SumTree 的优先经验回放机制来优化 SD3 经验回放，以更大的概率抽取具有更显著 TD 误差的样本，将比较差的经验拿来训练，可以有效优化经验池里的经验。经验回放机制增加了显著奖励经验的抽取概率，加快了智能体学习最优策略，提高了模型的收敛速度。经验池中的样本优先级值 p_i 为 TD 误差 δ_i 的绝对值，如式(7)所示。

$$p_i = |\delta_i| \tag{7}$$

$$\delta_i = y_i - Q_i(s_t,a_t;\theta_i) \tag{8}$$

其中，TD 误差 δ_i 是目标价值网络产生的 Q 值与在线价值网络产生的 Q 值之间的差值。δ_i 越大，表明该样本具有的预测精度上升空间越大，优先级越高。

在本文提出的 AQSDRL 中，经验池数据存储包括当前状态、动作、奖励值和新状态。过渡使用优先级值作为相应的索引。本文将优先级值存储在 SumTree 叶节点中。叶节点之上的父节点存储左右子节点的优先级值之和，根节点存储所有叶节点之和。在数据采样时，SumTree 模型将优先级值的总和除以样本数，得到区间数。然后，模型在每个区间随机选择一个数字。从 SumTree 的根节点开始，按照特定的规则向下搜索。最后，模型通过前一次搜索传递的优先级值得到对应的样本数据。SumTree 数据提取算法见算法 1。

算法 1　SumTree 数据提取算法

在区间里随机抽取一个数，假设为 p；
以根节点为父节点，遍历其子节点；
for $i=1$ to N
　if 左子节点 $p_i > p$
　　将 p_i 作为父节点，并记录其子节点；
　else
　　减去左子节点的值，选择右子节点为父节点，遍历其子节点；
end for

根据上述数据提取方法，优先级值越大，提取数据的概率越大。采样概率 $P(i)$ 和重要性采样权重 ω_i 分别如式(9)和式(10)所示。

$$P(i) = \frac{p_i}{\sum_j p_j} \tag{9}$$

$$\omega_i = (N(P(i))^{-1}) / \max_j(\omega_j) \tag{10}$$

其中，N 是样本数量。本文通过添加重要性采样权重来更新策略网络参数 ϕ_i，如式(11)所示。

$$\phi_i \leftarrow \frac{1}{N}\sum_{s_t}\left[\omega_j\nabla_{\phi_i}(\pi(s_t;\phi_i))\nabla_{a_t}Q_i(s_t,a_t;\theta_i)\big|_{a_{t+1}=\pi(s_t,\phi_i)}\right] \tag{11}$$

3.2　AQSDRL

本文提出的 AQSDRL 的状态、动作和奖励的具体设计如下。

1) 状态

状态是从环境中获取的网络状态信息。在本文模型中，每一个状态对应一个流量请求矩阵（网络中的流量请求信息）。

$$s_t = [\boldsymbol{M}_t] \tag{12}$$

其中， $\boldsymbol{M}_t$ 表示 t 时刻网络中的流量请求矩阵。

2) 动作

动作是智能体根据策略网络中的状态在 t 时刻生成的所有链路权重值的集合，表示为

$$a_t = [w_1, w_2, \cdots, w_N] \tag{13}$$

其中，N 表示链路数量，w_i 表示第 i 条链路的权重值。

3) 奖励

奖励是针对前一个动作获得的收益，评估服务质量的优劣。在本文的模型中，优化目标是负载均衡下的最小化链路时延。本文将奖励定义为优化目标

$$r_t = \beta_1 \cdot \mathrm{lb}(\varphi) \cdot 10 - \beta_2 \cdot \rho \tag{14}$$

其中，$\beta_1, \beta_2 \in [0,1]$，表示奖励的权重系数。参数 φ 是所有链路时延倒数之和，用于考查网络的时延大小，如式(15)所示，其中， N 表示链路数量， d_i 表示每条链路的时延。参数 ρ 是所有链路带宽标准差之和，用于考查网络的负载均衡度，如式(16)所示。其中， u_i 表示每条链路的已用带宽， $\sigma(u_i)$ 表示已用带宽的标准差。

$$\varphi = \sum_{i=1}^{N} \frac{1}{d_i} \tag{15}$$

$$\rho = \sum_{i=1}^{N} \sigma(u_i) \tag{16}$$

AQSDRL 智能体通过上述变量（状态、动作和奖励）与 SDN 环境进行交互。首先，将 SDN 控制器收集的数据平面的网络状态和性能指标作为 AQSDRL 智能体的状态。随后，AQSDRL 根据状态确定一组链路权重。SDN 控制器使用 Dijkstra 算法生成新的路径，并根据链路权重更新流表。流表更新后，通过后续的网络测量获得奖励和下一个状态，通过迭代优化网络性能。AQSDRL 训练算法见算法 2。

算法 2　AQSDRL 训练算法

输入　奖励折扣因子 γ、目标更新率 τ、目标网络参数更新频率 F、小批量样本数 N、迭代次数 T。

随机初始化在线策略网络 1、在线策略网络 2、在线价值网络 1、在线价值网络 2，记为 π_{ϕ_1}、π_{ϕ_2}、Q_{θ_1}、Q_{θ_2}，其中 ϕ_1、ϕ_2、θ_1、θ_2 为对应网络参数；

初始化对应的目标网络参数 $\phi_1^- \leftarrow \phi_1$、$\phi_2^- \leftarrow \phi_2$、$\theta_1^- \leftarrow \theta_1$、$\theta_2^- \leftarrow \theta_2$

初始化重放缓冲区 B；

初始化 SumTree 的数据缓冲区 S，设置所有叶子节点的优先级 p_j 为 0；

为动作探索初始化一个随机噪声 ε；

从 SDN 控制器收集的信息初始化状态 s_1，获取其特征向量 $v(s_1;\phi_1)$。

for t =1 to$T$

根据在线策略网络中的状态和探测噪声生成动作 a_t；

在 SDN 控制器上执行动作 a_t；

重新计算路径，下发流表；

从 SDN 控制器获取奖励 r_t、新状态 s_{t+1} 和终止标志；

将(s_t, a_t, r_t, s_{t+1})存储在 B 中;

计算样本优先级值： $p_i = |\delta_i|$；

更新 S 的所有节点；

按照算法 1 提取样本；

计算重要性采样权重： $\omega_i = \left(N^*\left(P(i)\right)^{-1}\right) / \max_j(\omega_j)$

$a_{t+1} \leftarrow \pi_i(s_{t+1};\phi_i^-) + \mathrm{clip}(\varepsilon, -c, c)$

$\hat{Q}(s_{t+1}, a_{t+1}) \leftarrow \min_{j=1,2}(Q_j(s_{t+1}, a_{t+1};\theta_j^-))$

$\mathrm{soft\,max}_\beta\, \hat{Q}(s_{t+1}, *)) \leftarrow \mathbb{E}_{a_{t+1}\sim p}$

$$\left[\frac{\exp(\beta\hat{Q}(s_{t+1}, a_{t+1}))\hat{Q}(s_{t+1}, a_{t+1})}{p(a_{t+1})}\right] / \mathbb{E}_{a_{t+1}\sim p}\left[\frac{\exp(\beta\hat{Q}(s_{t+1}, a_{t+1}))}{p(a_{t+1})}\right]$$

$y_i = r_t + \gamma(1-d)\mathrm{softmax}_\beta(\hat{Q}(s_{t+1}, *))$

通过贝尔曼损失更新 θ_i： $\frac{1}{N}\sum_{s_t}\left[Q_i(s_t, a_t;\theta_i) - y_i\right]^2$

通过策略梯度下降更新 ϕ_i： $\frac{1}{N}\sum_{s_t}\left[\omega_j \nabla_{\phi_i}(\pi(s_t;\phi_i))\nabla_{a_t} Q_i(s_t, a_t;\theta_i)\big|_{a_{t+1}=\pi(s_t;\phi_i)}\right]$

重新计算所有样本的 TD 误差 δ_i，更新 SumTree 中样本节点的优先级 p_j

更新目标网络参数： $\theta_i^- \leftarrow \tau\theta_i + (1-\tau)\theta_i^-$、$\phi_i^- \leftarrow \tau\phi_i + (1-\tau)\phi_i^-$

end for

4　实验结果与分析

4.1　实验环境

在实验中，计算机操作系统选用 Ubuntu18.04，实验模拟环境使用 Mininet 仿真平台、RYU 控制器和 OpenFlow1.3 协议。计算机配备了基础频率为 3.60 GHz 的 AMD Ryzen 7 PRO 4750G 处理器、512 GB 固态硬盘、16 GB 内存。网络拓扑参考了美国国家科学基金会组建的主干网络 NSFNET[30]，NSFNET 包含 14 个交换机及 21 条链路，链路带宽均为 10 Mbit/s，NSFNET 拓扑如图 3 所示。每 1 台交换机都连接 1 台用于接收和发送数据的主机。本文采用 PyTorch 深度学习框架实现强化学习算法在 SDN 中的应用。

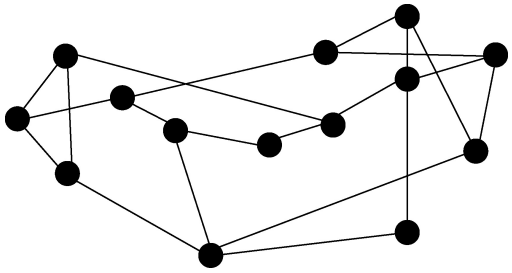


图 3　NSFNET 拓扑

4.2　参数设置

目标策略更新噪声是深度强化学习比较重要的一个参数。本文通过改变目标策略更新噪声观察 AQSDRL 的效果，通过收敛速度及稳定性确定 AQSDRL 应用到网络中的最佳数值。实验在流量强度为 100%的环境下展开，分别研究了目标策略更新噪声取值为 0.05、0.1、0.2 对收敛速度的影响。策略探索噪声训练结果如图 4 所示。

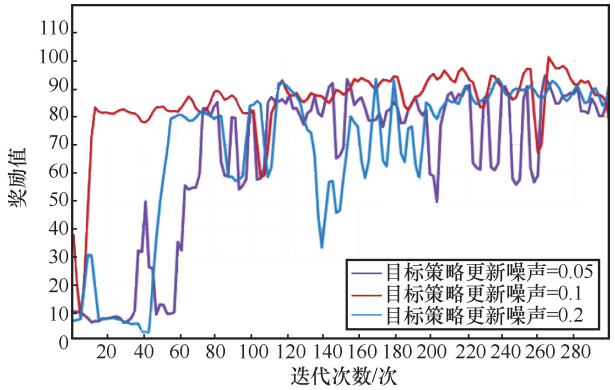


图 4　策略探索噪声训练结果

当目标策略更新噪声为 0.1 时，AQSDRL 的收敛速度最佳。当目标策略更新噪声为 0.05 或 0.2 时，AQSDRL 的收敛速度较慢且抖动较大，这是因为当更新噪声过小或过大时，AQSDRL 的学习相对不够稳定，导致无法在短时间内得到最优策略。AQSDRL 训练超参见表 1。

表 1　　AQSDRL 训练超参

超参	数值
探索噪声（expl_noise）	0.3
采样大小（batch_size）	8
奖励折扣因子（discount）	0.99
网络参数更新步长（tau）	0.005
截断目标策略噪音（noise_clip）	0.5
策略网络学习率（actor_lr）	3×10^{-4}
价值网络学习率（critic_lr）	3×10^{-4}

4.3　算法性能比较

时延和带宽是网络服务质量性能的重要指标。本文在流量强度为 50%的普通环境和 100%的高流量极端环境下展开，将 AQSDRL 的网络性能与 DDPG[18]、TD3[22]以及 SD3[25]算法进行比较，算法奖励对比如图 5 所示。

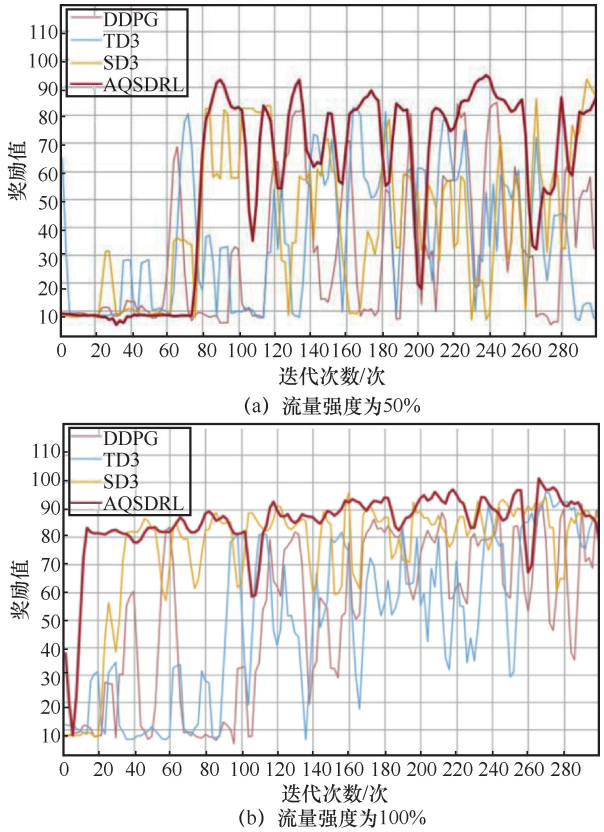
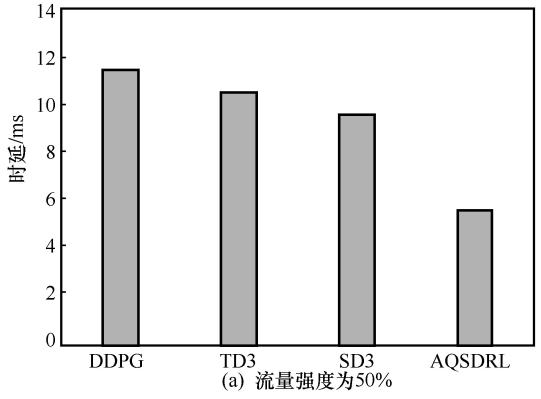


图 5　算法奖励对比

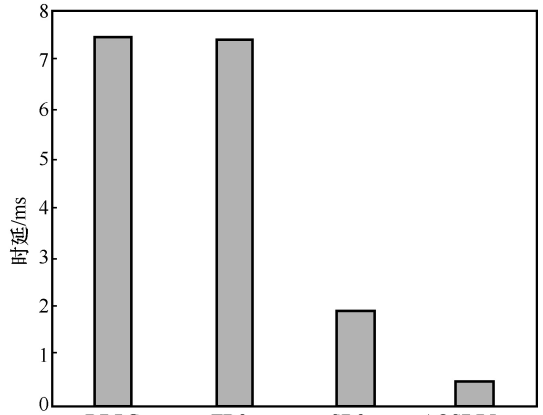
由图 5 可知，AQSDRL 相比其他算法收敛速度最快且最稳定。DDPG 由于选择动作时盲目地选择 Q 值最大的动作，存在高估问题，因此性能较差，且在高

流量强度场景下时，收敛速度较慢。TD3 对目标网络产生的两个 Q 值做最小化操作，有效解决了 DDPG 的高估问题，收敛速度相对 DDPG 来说较快，但未考虑低估问题，因此收敛值较低。SD3 是 TD3 的优化，因此性能优于 DDPG 和 TD3，但由于采样时采用随机采样的方法，仍然存在较大的波动。AQSDRL 基于 SD3 算法，采用 SumTree 采样来优化经验回放机制，当流量强度为 100%时，AQSDRL 收敛值和 SD3 相似，但收敛速度提升约 58%。值得注意的是，网络流量是动态变化的，当流量出现较大幅度的改变时，智能体需要重新探索学习，因此奖励会出现跳变的情况。AQSDRL 会出现偶尔的跳变，但很快会达到收敛，而其他算法需要经过较长时间才能收敛。实验结果表明，AQSDRL 无论是在低流量还是在高流量强度下，表现都优于 DDPG、TD3 和 SD3。

为了进一步探索以上 4 种算法的性能，本文对比了奖励设置中涉及的时延和负载均衡度的差异。时延是数据包从源节点到目标节点所需要的时间，时延越短，传输越快，这对于网络中的服务质量尤为重要。本文主要考查每条链路在每一个 episode 的平均时延，算法平均时延对比如图 6 所示。



(a) 流量强度为50%

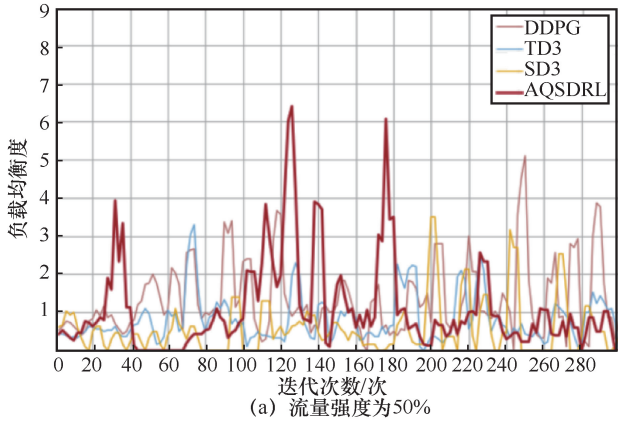


(b) 流量强度为100%

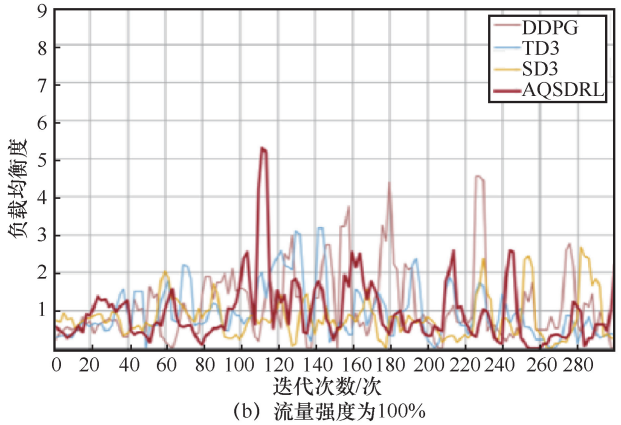
图 6　算法平均时延对比

当流量强度为 50%时，AQSDRL 的平均时延比 DDPG 低约 50%，比 TD3 低约 46%，比 SD3 低约 41%。当流量强度为 100%时，DDPG 和 TD3 的平均时延相近，原因在于当处在高流量强度场景时，算法的方差问题容易出现拥塞，导致时延增加。SD3 有效解决了 DDPG 和 TD3 的方差问题，因此在高流量强度下表现更优，时延降低了约 73%。而 AQSDRL 进一步提升了 SD3 在稳定训练控制方差方面的表现，较大幅度地提升了算法在高流量强度场景下的平均时延。当流量强度为 100%时，AQSDRL 的平均时延比 DDPG 和 TD3 低约 95%，比 SD3 低约 80%。图 6 进一步说明了 AQSDRL 在网络服务质量性能优化方面的有效性。

同时，本文还对比了各算法的负载均衡优化情况。参数 ρ 代表负载均衡度，是所有链路带宽标准差之和，参数抖动越小，负载均衡度越好。算法负载均衡度对比如图 7 所示。



(a) 流量强度为50%



(b) 流量强度为100%

图 7　算法负载均衡度对比

由图 7(a)可知，当处于低流量强度场景时，链路中的数据较少，因此在训练前期 AQSDRL 的负载均衡度不高，但随着迭代次数的增加，参数 ρ 逐

渐降低，其他算法反而升高。当在高流量强度场景时，如图 7(b)所示，AQSDRL 抖动的频率和幅度较小，展现出更优越的负载均衡性能。虽然 DDPG、TD3 和 SD3 的抖动幅度不大，但抖动频率较高于 AQSDRL。而 AQSDRL 除了在第 110 次训练附近有较大幅度的波动，其余训练时间的抖动幅度都比较小。因此，在高流量强度下更能展现本文提出的 AQSDRL 有较好的负载均衡性能。

5　结束语

本文提出了一种基于深度强化学习的服务质量优化算法 AQSDRL，以解决 SDN 在数据中心网络应用中的 QoS 问题。AQSDRL 采用 SD3 实现模型训练，同时采用基于 SumTree 结构的优先级经验回放机制优化 SD3 经验池的采样方法。实验结果表明，AQSDRL 显著提高了 SDN 的负载均衡性能，且有效降低了网络传输时延。

在未来的工作中，笔者将继续探索在大规模网络中如何提升深度强化学习算法的收敛速度，更加有效地优化网络的服务质量。

参考文献：

[1] TANHA M, SAJJADI D, RUBY R, et al. Traffic engineering enhancement by progressive migration to SDN[J]. IEEE Communications Letters, 2018, 22(3): 438-441.

[2] ONGARO F, CERQUEIRA E, FOSCHINI L, et al. Enhancing the quality level support for real-time multimedia applications in software-defined networks[C]//Proceedings of 2015 International Conference on Computing, Networking and Communications (ICNC). Piscataway: IEEE Press, 2015: 505-509.

[3] ALIZADEH M, EDSALL T, DHARMAPURIKAR S, et al. CONGA: distributed congestion-aware load balancing for datacenters[C]//Proceedings of the 2014 ACM conference on SIGCOMM. New York: ACM Press, 2014: 503-514.

[4] ZOU G B, LI T F, JIANG M, et al. Deep TSQP: temporal-aware service QoS prediction via deep neural network and feature integration[J]. Knowledge-Based Systems, 2022, 241: 108062.

[5] CHEN J Y, WANG Y, HUANG X F, et al. ALBLP: adaptive load-balancing architecture based on link-state prediction in software-defined networking[J]. Wireless Communications and Mobile Computing, 2022, 2022: 8354150.

[6] NUGRAHAB, MURTHYRN. Deep learning-based slow DDoS attack detection in SDN-based networks[C]//Proceedings of 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks. Piscataway: IEEE Press, 2020: 51-56.

[7] NOVAES M P, CARVALHO L F, LLORET J, et al. Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment[J]. IEEE Access, 2020(8): 83765-83781.

[8] BHATIA J, DAVE R, BHAYANI H, et al. SDN-based real-time urban traffic analysis in VANET environment[J]. Computer Communications, 2020, 149: 162-175.

[9] TROIA S, ALVIZU R, MAIER G. Reinforcement learning for service function chain reconfiguration in NFV-SDN metro-core optical networks[J]. IEEE Access, 2019(7): 167944-167957.

[10] LIN S C, AKYILDIZ I F, WANG P, et al. QoS-aware adaptive routing in multi-layer hierarchical software defined networks: a reinforcement learning approach[C]//Proceedings of 2016 IEEE International Conference on Services Computing. Piscataway: IEEE Press, 2016: 25-33.

[11] YOUNUS M U, KHAN M K, ANJUM M R, et al. Optimizing the lifetime of software defined wireless sensor network via reinforcement learning[J]. IEEE Access, 2020(9): 259-272.

[12] CASAS-VELASCO D M, RENDON O M C, DA FONSECA N L S. Intelligent routing based on reinforcement learning for software-defined networking[J]. IEEE Transactions on Network and Service Management, 2021, 18(1): 870-881.

[13] AL-JAWAD A, COMŞA I S, SHAH P, et al. An innovative reinforcement learning-based framework for quality of service provisioning over multimedia-based SDN environments[J]. IEEE Transactions on Broadcasting, 2021, 67(4): 851-867.

[14] XU Z Y, WU K, ZHANG W Y, et al. PnP-DRL: a plug-and-play deep reinforcement learning approach for experience-driven networking[J]. IEEE Journal on Selected Areas in Communications, 2021, 39(8): 2476-2486.

[15] LIU W X, CAI J, CHEN Q C, et al. DRL-R: deep reinforcement learning approach for intelligent routing in software-defined data-center networks[J]. Journal of Network and Computer Applications, 2021, 177: 102865.

[16] HU Y X, LI Z Y, LAN J L, et al. EARS: intelligence-driven experiential network architecture for automatic routing in software-defined networking[J]. China Communications, 2020, 17(2): 149-162.

[17] BOUZIDI E H, OUTTAGARTS A, LANGAR R, et al. Deep Q-network and traffic prediction based routing optimization in software defined networks[J]. Journal of Network and Computer Applications, 2021(192): 103181.

[18] 兰巨龙, 于倡和, 胡宇翔, 等. 基于深度增强学习的软件定义网络路由优化机制[J]. 电子与信息学报, 2019, 41(11): 2669-2674.
LAN J L, YU C H, HU Y X, et al. A SDN routing optimization mechanism based on deep reinforcement learning[J]. Journal of Electronics & Information Technology, 2019, 41(11): 2669-2674.

[19] 兰巨龙, 张学帅, 胡宇翔, 等. 基于深度强化学习的软件定义网络 QoS 优化[J]. 通信学报, 2019, 40(12): 60-67.
LAN J L, ZHANG X S, HU Y X, et al. Software-defined networking QoS optimization based on deep reinforcement learning[J]. Journal on Communications, 2019, 40(12): 60-67.

[20] MAI T L, YAO H P, ZHANG N, et al. Transfer reinforcement learning aided distributed network slicing optimization in industrial IoT[J]. IEEE Transactions on Industrial Informatics, 2022, 18(6): 4308-4316.

[21] CHEN J Y, WANG Y, OU J T, et al. ALBRL: automatic load-balancing architecture based on reinforcement learning in software-defined networking[J]. Wireless Communications and Mobile Computing, 2022, 2022: 3866143.

[22] FUJIMOTO S, VAN HOOF H, MEGER D. Addressing function approximation error in actor-critic methods[EB]. 2018.

[23] 孙鹏浩, 兰巨龙, 申涓, 等. 一种基于深度增强学习的智能路由技术[J]. 电子学报, 2020, 48(11): 2170-2177.

SUN P H, LAN J L, SHEN J, et al. An intelligent routing technology based on deep reinforcement learning[J]. Acta Electronica Sinica, 2020, 48(11): 2170-2177.

[24] 孙鹏浩, 兰巨龙, 申涓, 等. 基于牵引控制的深度强化学习路由策略生成[J]. 计算机研究与发展, 2021, 58(7): 1563-1572.
SUN P H, LAN J L, SHEN J, et al. Pinning control-based routing policy generation using deep reinforcement learning[J]. Journal of Computer Research and Development, 2021, 58(7): 1563-1572.

[25] PAN L, CAI Q P, HUANG L B. Softmax deep double deterministic policy gradients[J]. Advances in Neural Information Processing Systems, 2020(33): 11767-11777.

[26] KHAN A A, ZAFRULLAH M, HUSSAIN M, et al. Performance analysis of OSPF and hybrid networks[C]//Proceedings of 2017 International Symposium on Wireless Systems and Networks (ISWSN). Piscataway: IEEE Press, 2017: 1-4.

[27] CHIESA M, KINDLER G, SCHAPIRA M. Traffic engineering with equal-cost-multipath: an algorithmic perspective[J]. IEEE/ACM Transactions on Networking, 2017, 25(2): 779-792.

[28] SUN P H, GUO Z H, LI J F, et al. Enabling scalable routing in software-defined networks with deep reinforcement learning on critical nodes[J]. IEEE/ACM Transactions on Networking, 2022, 30(2): 629-640.

[29] SUN P H, LAN J L, LI J F, et al. A scalable deep reinforcement learning approach for traffic engineering based on link control[J]. IEEE Communications Letters, 2021, 25(1): 171-175.

[30] National Science Foundation. National science foundation network[EB]. 2022.

[作者简介]

廖岑卉珊（1999– ），女，桂林电子科技大学硕士生，主要研究方向为软件定义网络、深度强化学习。

陈俊彦（1985– ），男，博士，桂林电子科技大学高级实验师，主要研究方向为强化学习、图神经网络和软件定义网络。

梁观平（1998– ），男，国防科技大学博士生，主要研究方向为软件定义网络、流量调度与拥塞控制。

谢小兰（1999– ），女，桂林电子科技大学硕士生，主要研究方向为软件定义网络、图神经网络和深度强化学习。

卢小烨（2000– ），男，桂林电子科技大学在读，主要研究方向为软件定义网络、深度强化学习。