利用记忆单元改进 DQN 的 Web 服务组合优化方法

杨 波[1] 胡国兵[2]

[1](南京信息职业技术学院信息服务学院 江苏 南京 210023)

[2](金陵科技学院电子信息工程学院 江苏 南京 211169)

摘 要 针对面向高可扩展性、复杂性和异构性服务环境的 Web 服务组合难以进行优化的问题,提出一种利用长短期记忆单元改进深度 Q 神经网络(Long Short-Term Memory Deep Q-Network,LSTM-DQN)的 Web 服务组合优化方法。利用 Markov 对 Web 服务组合优化问题进行建模,分析各变量参数之间的关系,同时加入强化学习的组合优化模型,简化了组合优化过程;引入记忆单元对深度 Q 网络算法进行优化,提出 LSTM-DQN 方法,提升了 DQN 算法的全局寻优能力;将 LSTM-DQN 应用于大规模服务环境下的 Web 服务组合优化问题,对所建立的马尔可夫决策(Markov Decision Process,MDP)模型进行优化,以提升 Web 服务组合的处理效率。实验结果表明,该方法相对于传统方法在大规模服务环境下对 Web 服务组合优化所消耗时间更短,服务组合成功率更高,具有更强的处理能力和处理效率。

关键词 记忆单元 LSTM-DQN MDP 简化建模 Web 服务组合

中图分类号 TP391 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2020.11.002

A WEB SERVICE COMPOSITION OPTIMIZATION APPROACH BASED ON MEMORY UNITS AND IMPROVED DQN

Yang Bo[1] Hu Guobing[2]

[1](*School of Information Service, Nanjing Vocational College of Information Technology, Nanjing 210023, Jiangsu, China*)

[2](*School of Electronic and Information Engineering, Jinling Institute of Technology, Nanjing 211169, Jiangsu, China*)

Abstract In order to solve the problem that it is difficult to optimize Web service composition for high scalability, complexity and heterogeneous service environment, an improved deep Q neural network (LSTM-DQN) based on long-term and short-term memory units is proposed to optimize the composition of Web services. Markov was used to model the Web service composition optimization problem, and the relationship between the parameters of each variable was analyzed. The combinatorial optimization model of reinforcement learning was added to simplify the combinatorial optimization process. The memory unit was introduced to optimize the deep Q network algorithm, and the LSTM-DQN method was proposed to improve the global optimization ability of DQN algorithm. The LSTM-DQN was applied to the optimization of Web service composition in large-scale service environment, and the established Markov decision process (MDP) model was optimized to improve the processing efficiency of Web service composition. The experimental results show that the proposed method consumes shorter time, has higher success rate of service composition, and has stronger processing ability and efficiency than the traditional method in large-scale service environment.

Keywords Memory unit LSTM-DQN MDP simplified modeling Web service composition

---

收稿日期:2019-11-07。江苏省自然科学基金项目(BK20161104)。**杨波**,副教授,主研领域:Web 服务,企业信息化,数据挖掘。**胡国兵**,教授。

0 引 言

随着网络技术的发展,面向复杂环境的服务组合在现实生活中的应用越来越多,而面向复杂环境的 Web 服务组合的进一步应用会面临难以优化的问题。面向服务计算(Service-Oriented Computing,SOC)的基本目标是实现在各种平台上运行的不同软件和数据应用程序之间的互操作性[1]。因此只有当存在众多服务提供商和服务消费者彼此协作时,才能发挥 SOC 的全部潜力[2]。由于面向服务的环境中固有的动态性和复杂性,良好的服务组合解决方案需要适应这些动态服务环境的变化和波动。此外,功能等同服务数量的爆炸性增长迫切需要能够处理此类任务的有效服务组合算法[3-4]。由于一个服务不能满足所有用户要求,因此需要将组件服务转换成组合服务。在这方面,服务组合成为实现面向服务的体系结构(System of Archtecture,SOA)的最有效技术。

为了对面向复杂环境的 Web 服务组合进行优化,学者进行了大量的研究,在复杂环境下,从大量的满足功能需求但是非功能需求的 Web 服务中,选择出满足用户需求的组合服务所采取的服务选择策略可以有多种,研究文献可主要分为三种:群智能算法、人工智能(Artificial Intelligence,AI)方法和其他混合方法。群智能算法如文献[5]提出的多目标服务组合优化推荐方法(Multi-Objective Service Combination Optimization Recommendation Method,MO-SCORM),具有较好的可适应性和自组织性,通过将细粒度的 Web 服务粗粒化实现了面向个性化客户的体系结构,其协作性和健壮性较好,但效率较低;文献[6]从功能等同的具体服务集合中选择了满足消费者强加的服务质量(Quality of Service,QoS),对最佳服务集合进行约束,提高了服务组合的服务质量,但在服务效率方面还需进一步提高。在大数据背景下为改善算法效率,智能算法得到了广泛应用:文献[7]提出了一种结合顺序决策过程的强化学习方法(Sequential Decision-Making Process-Reinforce Learning,SDMP-RL)并将其应用在 Web 服务组合的优化问题中,使代理与环境交互,通过反复实验提高了 Web 组合服务的学习最优解,但该方法在面对可扩展性的服务环境时表现较差;文献[8]采用深度 Q 学习(Deep Q-Learning,DQL)解决了 Web 服务组合效率低的问题,但是没能兼顾 Web 组合服务对复杂环境的适用性和可扩展性。现有的 Web 服务组合方法在面对复杂的服务环境时,难以同时在服务环境的适应性、可扩展性和动态性这几个指标上获得良好的综合性能。针对面向高可扩展性、复杂性和异构性服务环境的 Web 服务组合难以进行优化的问题,本文提出一种利用记忆单元改进 DQN 的 Web 服务组合优化方法。其创新点主要体现为以下两点:

(1) 引入 LSTM-DQN 方法进行优化,提升了 DQN 算法的全局寻优能力;

(2) 利用强化学习的组合优化模型简化组合优化过程,并将 LSTM-DQN 方法应用于 Web 服务组合优化问题,提升了 Web 服务组合的处理效率。

1 基于 Markov 的组合优化模型

组合优化模型采用马尔可夫决策过程 MDP 作为一般方案,以描述动态环境中的服务组合和适应过程。MDP 是离散时间随机控制过程,用于对不确定域中的顺序决策进行建模。MDP 的关键组成部分正式定义如下[9]:

定义 1 Markov 决策过程(MDP)。一个 MDP 可以定义为一个五元组 $MDP = < S,A,P,R,\gamma >$,其中:S 是一组有限状态;$A(s)$ 是一个有限的动作集合,取决于当前状态 $s \in S$;P 是一个概率值,也就是当动作 $a \in A$ 被执行时,发生一个从当前状态 s 到结果状态 s' 的一个状态转移,其转移概率分布为 $P(s' \mid s,a)$;R 是奖励函数。类似地,当执行操作 a 时,从状态 s 转移到 s',将收到一个实际值的奖励 r,其预期值为 $r = E(R(s' \mid s,a))$;$\gamma \in [0,1]$ 是区分未来奖励和即时奖励重要性的折扣因素。

MDP 的解决方案是一个决策策略,通常决策策略 π 是从状态到操作的概率分布的映射,定义为 $\pi: S \rightarrow A$ 如果 MDP 是偶发性的,即状态在长度 t 的每一场景后重置,则一个场景中的状态、行动和奖励序列构成策略的轨迹或推出[10]。策略的每个推出都会从环境中累积一个奖励,从而返回 R。解决方案中算法的目标是找到一个最佳策略,该策略累积了所有状态的最大预期回报。

2 基于强化学习的组合优化模型

RL 的目的是设计算法,通过这些算法,代理可以学习在某些环境中的自主操作,从他们与该环境的交互或从环境中收集的观测值中学习。此环境通常作为 MDP 制定。与传统的动态编程技术不同,强化学习算法不需要有关 MDP 的知识,它们针对的是精确方法变得不可行的大型 MMDP。在此背景下,RL 旨在根据它

们与环境的交互来确定最佳控制策略[11]。此策略可以通过基于一组四元组 (s_t,a_t,r_t,s_{t+1}) 近似所谓的 Q 函数来实现。其中:s_t 表示 t 时刻的环境状态;a_t 表示所执行的控制操作;r_t 表示所获得的瞬时奖励;s_{t+1} 表示环境的后续状态,并通过 Q 函数决定控制策略。

最适合 RL 的问题类型是复杂的控制问题,其中似乎没有明显或易于编程的解决方案。因此,在开放和动态环境中使用 RL 进行自适应服务组合具有明显的优势。通过在组合模型中使用 RL,它可以以自适应的方法学习最佳服务选择策略[11]。基于 MDP 的动态服务组合中使用的关键概念定义如下:

定义 2　基于 MDP 的 Web 服务组合(MDP-WSC)。一个 MDP-WSC 可以定义为一个六元组 $MDP\text{-}WSC = <S^i,s_0^i,s_r^i,A^i(s),P^i,R^i>$,其中:$S^i$ 是代理 i 观察到的一组有限的状态/抽象服务;$s_0^i \in S$ 是初始状态,服务组合的任何执行通常从此状态开始;$s_r^i \subset S$ 是一组终端状态,到达其中一个状态后,服务组合的执行终止;$A^i(s)$ 是一组可以在状态 $s \in S^i$ 中执行的 Web 服务,只有当 s 满足先决条件 ws^P,Web 服务 ws 属于 A^i;P^i 是代理 i 从当前状态 s 转换到结果状态 s' 时,调用 Web 服务 $ws \in A^i(s)$ 的概率,对于每个状态 s,这种转变发生的概率为 $P^i(s' \mid s,ws)$;R^i 是一个奖励函数,当一个 Web 服务 $ws \in A^i(s)$ 被调用时,代理 i 从 s 向 s' 转换,服务消费者得到一个即时回报 r^i,其期望值为 $E(R^i(s' \mid s,ws))$。

考虑到前面介绍的 MDP-WSC 模型,学习代理的任务将变为区分提供最高累积回报的最佳工作流。对于每个代理 i,让 W^i 成为候选工作流的集。让 R_{ws}^i 成为与每个 Web 服务调用 ws_i 相关的奖励,用于某些工作流 w。让 N 成为每个候选工作流中固定的最大调用数。产生最大预期回报的工作流 w^*,称为最佳工作流。

奖励值 r 可以使用操作值函数计算:

$$Q^i(s,a) \leftarrow Q^i(s,a) + \alpha[r + \gamma \max_{a'} Q^i(s',a') - Q^i(s,a)] \tag{1}$$

式中:s 表示状态空间(即抽象服务),表示代理 i 遍历所有可能的工作流时全部状态的集合;α 是学习率,它控制收敛;当代理 i 选择一个 Web 服务 ws 时,代理 i 收到一个奖励,这是一个聚合值的 QoS 的 ws 属性。此奖励值可以根据下式计算:

$$r = \sum \omega \times \frac{Q_n - Q_n^{\min}}{Q_n^{\max} - Q_n^{\min}} \tag{2}$$

式中:Q_n 表示 Web 服务 ws 的第 n 个质量属性的观测值;$Q_n^{\max}$ 和 $Q_n^{\min}$ 分别表示 Web 服务 ws 的第 n 个质量属性的最大值和最小值;ω 是一个加权因子。

在此模型中,采用 ε-贪婪策略,使学习代理能够在选择过去尝试过的 Web 服务(即利用)和随机选择可能提供更好的结果的新 Web 服务之间进行权衡(即探索)。对于代理 i,给定状态和一组可用的 Web 服务 $A^i(s)$,代理 i 选择下一个 Web 服务 j 的概率为:

$$P^i(a_j \mid s) = \begin{cases} 1-\varepsilon & a_j = \operatorname{argmax}_a Q[s,a] \\ \varepsilon & \text{其他} \end{cases} \tag{3}$$

式中:ε 是单个 Web 服务的概率分布,[·]表示对中括号内的内容进行记分。代理 i 根据 ε-贪婪策略的概率$(1-\varepsilon)$选择最佳的 Web 服务,否则以概率 ε 选择一个统一随机的 Web 服务[12]。

3　LSTM-DQN 的 Web 服务组合优化方法

3.1　深度 Q 神经网络算法

深度 Q 神经网络(Deep Q-Network,DQN)是基于深度学习和强化学习思想而提出的无监督学习方法。在高维度状态或动作空间中,深度强化学习存在难以估计每个大型状态和操作空间所对应的 Q 值问题。为了解决该问题,引入了深度 Q 神经网络,使深度强化学习代理的各个组成部分利用梯度下降对参数进行训练,以尽量减少一些不必要的损失函数[13]。DQN 算法原理如图 1 所示。

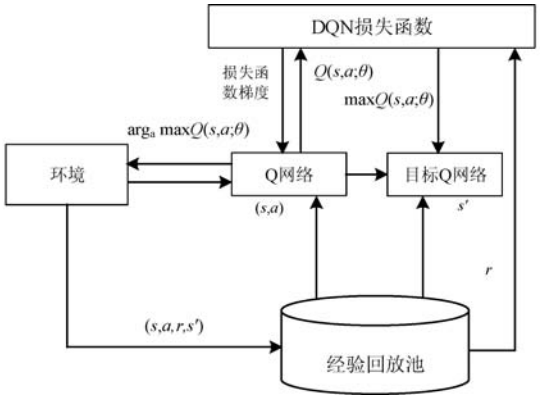

图 1　DQN 算法原理

在深度学习计算 Q 值的过程中,DQN 算法通过权重为 θ 的神经网络近似器来估计 Q 值,该神经网络的输入为状态,经过卷积、池化、全连接等操作,输出该状态下每个动作的 Q 的估计值。智能体的目标是找到一种未来反馈值最大的动作选择方式,利用该动作选择方式与环境进行交互。因此定义最优动作选择函数为 $Q(s,a)$,其算式为[14]:

$$\begin{aligned} Q(s,a) &= \max_\pi E[R_t \mid s_t = s, a_t = a, \mid \pi] \\ R_t &= \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \end{aligned} \tag{4}$$

式中：s 为状态；a 为该状态下执行的动作；π 为动作和状态映射；s_t 为 t 时间步时的状态；a_t 为 s_t 状态下执行的动作；R_t 为状态时 s 执行动作 a 得到的反馈值；T 为总时间步；t' 为求和过程变量；r_t 为第 t 个时间步的反馈值[15]。

综上所述，根据深度学习和强化学习计算的 Q 值，通过对损失函数 $L_e(\theta_e)$ 使用梯度下降法更新权重 θ，来优化 Q 神经网络：

$$\begin{aligned} L_e(\theta_e) &= E_{s,a\sim p(\cdot)}[(y_e - Q(s,a;\theta_e))^2] \\ y_e &= E_{s'}[r + \gamma \max_{a'} Q_e(s',a';\theta_e) \mid s,a] \end{aligned} \tag{5}$$

式中：$Q(s,a;\theta_e)$ 为 $Q(s,a)$ 的估计值；e 为迭代次数；s 为当前状态；a 为当前动作；s' 为下一个状态；a' 为下一个动作；$p(s,a)$ 为状态 s 和动作 a 的概率分布。

3.2 LSTM 改进的深度 Q 神经网络算法

为了在大型服务环境中启用自适应服务组合，提出一种基于改进 DQN 的模型，该模型包括生成器 ϕR 和动作记分器 ϕA，模型的整体架构如图 2 所示。

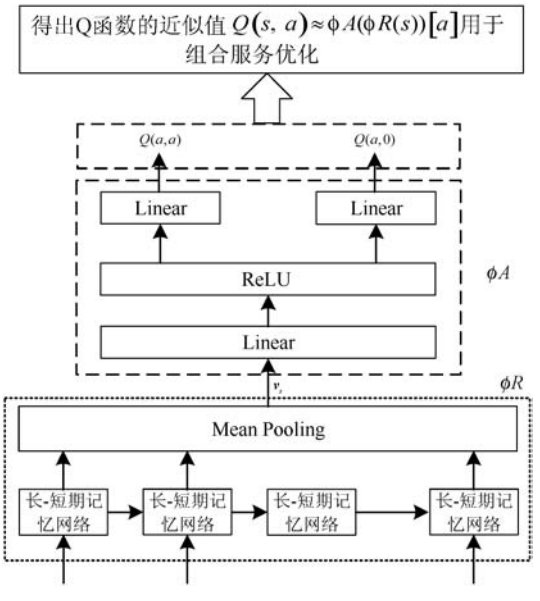

图 2 改进 DQN 模型框架

为了更好地描述模型，采用长-短期记忆网络（Long Short-Term Memory Network，LSTM）表示生成器。LSTM 是递归神经网络，能够连接和识别输入向量之间的长程模式，在一定程度上可捕捉潜在信息。所提模型中，LSTM 网络将输入向量嵌入潜在因子 w_k，并在每个步骤生成输出向量 $\boldsymbol{x}_k$。为了得到最终的状态 $\boldsymbol{v}_s$，添加了一个平均池层，用于计算输出向量 $\boldsymbol{x}_k$ 上的元素平均值[16]：

$$\boldsymbol{v}_s = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{x}_k \tag{6}$$

生成器的输出向量作为动作记分器的输入，即 $\boldsymbol{v}_s = \phi R(s)$，输出是所有动作的分数。同时预测所有动作的分数，这比分别对每个状态动作进行分数计算更有效。通过生成器和动作记分器，可得到 Q 函数的近似值 $Q(s,a) \approx \phi A(\phi R(s))[a]$。

由于计算上的限制，所提 LSTM-DQN 方法将命令视为由一个操作和一个参数对象组成[17]。考虑到所有可能的动作和对象，使用同一个网络对每个状态进行预测。该法使用随机梯度下降和 RMSprop 学习表示生成器的参数 r 和动作记分器的参数 a，完整的训练过程如算法 1 所示。在每次迭代 e 中，更新参数以减少当前状态 $Q(s_t,a_t;\theta_e)(\theta_e = [\theta_R;\theta_A]_e)$ 的预测值与给定奖励 r_t 的预期 q 值和下一状态 $\max_a Q(s_{t+1},a;\theta_{e-1})$ 的值之间的差异[18]。

所提方法不使用当前事件的转换对 Q 值进行更新，而是从 D 采样随机转换 $(\hat{s},\hat{a},s',r)$，以避免在使用同一事件的转换时由于强相关性而产生的问题[19]。结合式(5)，参数更新按 $L_e(\theta_e)$ 的如下梯度执行：

$$\begin{aligned} \nabla_{\theta_e} L_e(\theta_e) = E_{\hat{s},\hat{a}}[&2(y_e - Q(s,a;\theta_e)) \\ &\nabla_{\theta_e} Q(s,a;\theta_e)] \end{aligned} \tag{7}$$

式中：∇_{θ_e} 表示对 θ_e 求导。

算法 1 LSTM-DQN 训练程序

1. 输入经验记忆 D
2. 初始化表示生成器 ϕR 和动作记分器 ϕA 的随机的初始化参数
3. for $j=1$；最大迭代次数 M do
4. 初始化游戏并获得开始状态描述 s_1
5. for $t=1$；T do
6. 用 ϕR 转换 st 以表示 v_{st}
7. if 随机数 random( ) < 迭代次数 e then
8. 选择随机一个动作 a_t
9. else
10. 对所有动作，通过 $\phi A(v_{st})$ 计算 $Q(s_t;a)$
11. 选择 $a_t = \text{argmax}\ Q(s_t;a)$
12. 执行动作 a_t 并获得奖励 r_t 和新状态 s_{t+1}
13. if $r_t > 0$，设置优先权 $p_t = 1$，else $p_t = 0$
14. 存储转移量 $(s_t;a_t;r_t;s_{t+1};p_t)$ 到 D 中
15. 从 D 中随机采样小数量的转换 $(s_j;a_j;r_j;s_{j+1};p_j)$
16. 设置

$$y_j = \begin{cases} r_j & s_{ij+1}\text{是末期的} \\ r_j + \gamma \max a' Q(s_t+1;a';\theta) & s_{ij+1}\text{是非末期的} \end{cases}$$

17. 对损失执行梯度下降步骤 $L(\theta) = y_i - Q(s_j,a_j;\theta)^2$
18. 输出 Web 服务组合优化方案

4 实 验

4.1 实验设置

本文提出的方法在连续迭代循环中运行，直到达

到收敛点。一旦学习代理收到了若干个连续时间的累积奖励的相同值,代理则将收敛到最优策略。这些累积奖励按节进行比较,差异根据阈值进行预测。所有模拟实验都在六核心 Intel Xeon 3.2 GHz iMac Pro 上进行,具有 32 GB 的 RAM 和 8 MB 的 GPU;采用 Windows 系统运行 MATLAB 仿真软件,利用 MATLAB 语言进行程序编写,阈值设置为 0.001,迭代次数设置为 1 000。

在以下实验中,基于 QWS 数据集考虑了三个 QoS 属性,即可用性、可靠性和响应时间。通过使用表 1 聚合其成员 Web 服务的 QoS 向量,计算每个工作流的平均累积奖励 r。

表 1　聚合参数

QoS 参数	聚合函数
可用性	$\sum_{i=1}^{n}(\log(availability(ws_i)))$
响应时间	$\sum_{i=1}^{n}(responsetime(ws_i))$
可靠性	$\sum_{i=1}^{n}(\log(reliability(ws_i)))$

在学习质量、选择策略和消耗时间性能方面,将本文方法与文献[5]提出的 MO-SCORM 方法、文献[7]提出的 SDMP-RL 方法和文献[8]提出的 DQL 方法进行比较。学习参数是根据文献[13]中的第一次经验模拟建立的,具体设置如表 2 所示。

表 2　参数设置

参数	意义	值
α	学习率	1.0
γ	折扣系数	0.8
ε	探索能力	0.7
ν	启发权重因子	0.5
ω	QoS 加权因子	0.3

4.2　学习质量

本节主要验证所提方法在大型环境中寻找高质量服务组合的能力。当解决方案收敛到最佳服务选择策略时,使用学习代理获得的平均累积奖励来衡量方法能力,此奖励值表示最佳工作流的聚合 QoS。

测试分成两次。测试 1 中,测试环境抽象任务数量固定为 150 个和 250 个,其可用的具体 Web 服务的数量范围为 600 到 900。在此环境下,运行四种方法并将结果统计于图 3 中。

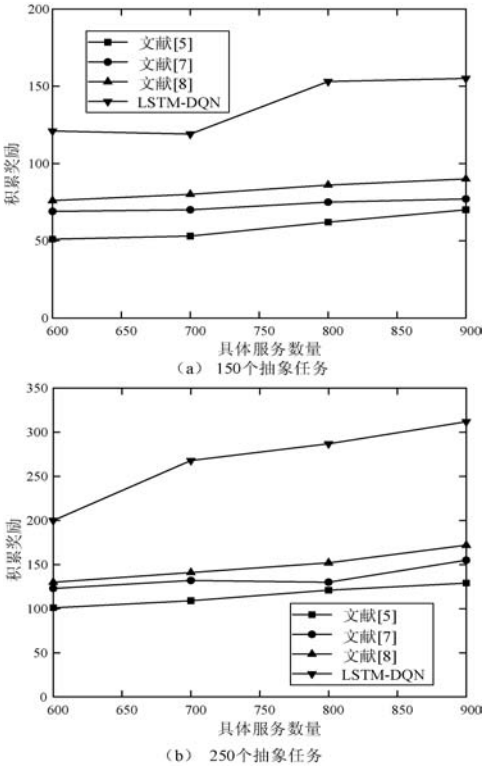

图 3　不同抽象服务任务数量下的积累奖励对比

可以看出,虽然环境规模较大,但本文提出的 LSTM-DQN 方法的运行结果优于文献[5]、文献[7]和文献[8]方法,LSTM-DQN 方法显然在整个学习过程中获得更高的累积回报,并带来更高质量的解决方案。

测试 2 中,将具体服务的数量固定为 700 和 900,并将抽象任务服务的数量范围设置为 100 到 400,实验结果如图 4 所示。

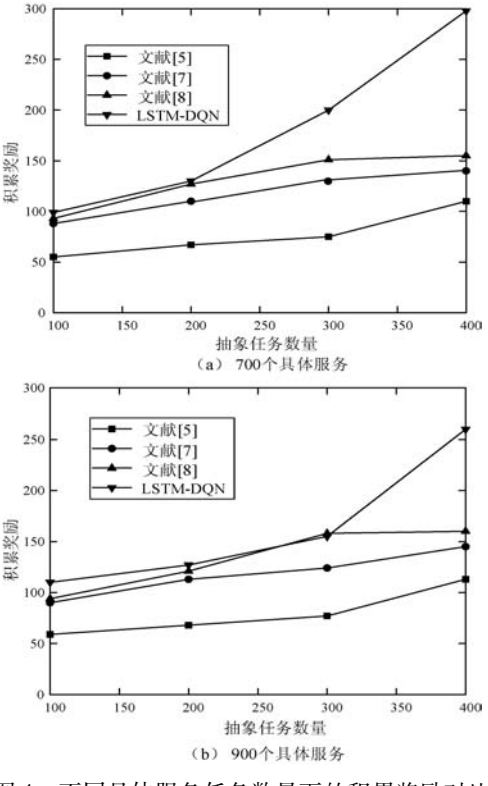

图 4　不同具体服务任务数量下的积累奖励对比

无论每个工作流的抽象任务的数量大小为多少，LSTM-DQN 服务组合方法的结果均优于文献[5]、文献[7]及文献[8]方法。随着抽象任务服务数量的增加，方法性能差距会越来越大，由此验证了 LSTM-DQN 方法可以找到更好的服务组合、在大型环境中的可扩展性及其查找高质量服务的能力。

4.3 最佳服务选择策略

本节验证所提出的学习方法在动态服务环境中找到最佳服务选择策略的能力，该能力由获得的累积奖励来衡量的。服务环境中的动态更改取决于参与者具体服务的 QoS 值变化。QoS 值动态变化会影响学习代理收到的奖励值 r。本实验中用两个因素来衡量服务环境的动态变化，即更改的规模和变化的频率。

为验证变化规模这一因素的影响，考虑一个每个任务包含 200 个抽象任务服务和 700 个具体服务的工作流，改变参与者具体服务的 QoS 值，变化百分比分别为 1%、5% 和 10%。实验结果如图 5 所示。其中：x 轴表示参与者具体服务的 QoS 值的更改百分比；y 轴表示学习代理在收敛到最佳值之前所获得的累积奖励。

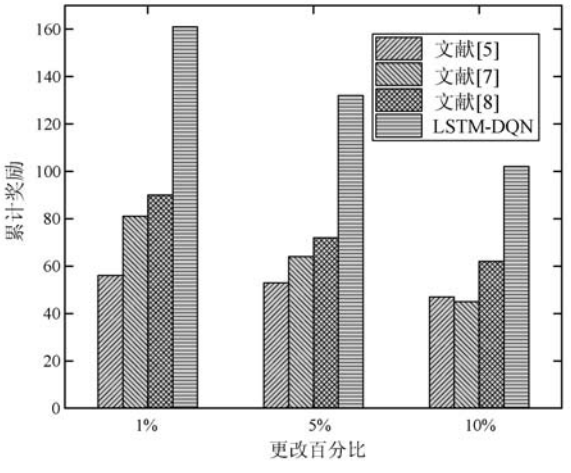

图 5 变化尺度对比 1

可以看出，LSTM-DQN 方法在服务环境中汇聚到最佳策略之前，分别累积了 162 和 111 个单位的奖励，而服务环境在其参与者的具体 QoS 值中经历 1% 和 5% 的周期性变化服务。与 DQN 和 RL 方法在同一环境下分别获得的 85 和 77 个单位的奖励，以及 64 和 56 个单位的奖励相比，在复杂动态的环境中学习最佳服务选择策略时，LSTM-DQN 方法的效率不高。

为验证更改频率这一因素，本文考虑一个每个任务包含 200 个抽象任务服务和 700 个具体服务的工作流，参与者具体服务的 5% 的 QoS 值每 1 000、500 和 250 段落按顺序定期变化。结果如图 6 所示。其中：x 轴表示段落数；y 轴表示学习代理在收敛到最佳值之前所获得的累积奖励。

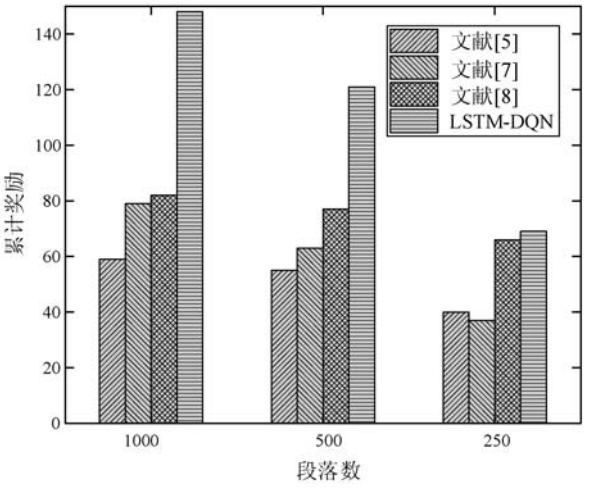

图 6 变换频率对比

可以看出，LSTM-DQN 方法在汇合到服务环境中的最佳策略，每个 1 000 段和 500 段的参与者服务的 QoS 值分别经历 5% 的周期性更改。这与 DQN 方法和 RL 方法在相同服务环境中分别获得的 80 和 67 个奖励单位以及 57 和 53 个单位的奖励相比更好。

4.4 服务组合成功率

本文方法在大规模服务数量下效果明显，为了验证这一点，采用不同的任务数对四种方法进行对比实验，实验结果如图 7 所示。

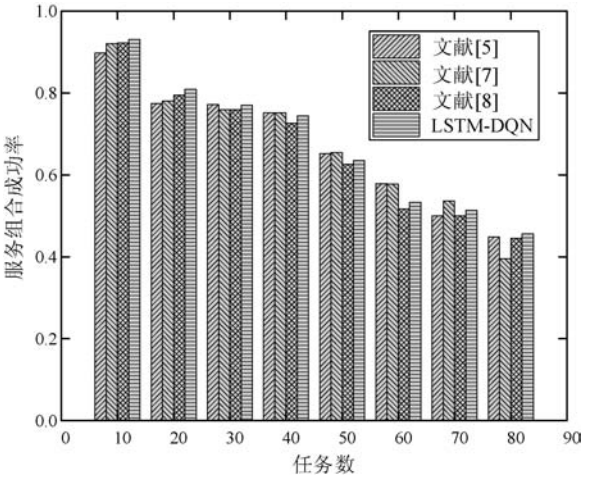

图 7 变化尺度对比 2

可以看出，使用 LSTM-DQN 方法能够得到相对高的服务组合成功率。并且服务组合时，组合成功率与任务数有关系，随着任务数的增加，组合成功率会降低，但使用 LSTM-DQN 方法组合成功率变化缓慢，说明候选服务数越多，系统的性能越稳定，但系统的代价也会增大，这与实际是相符合的。

4.5 消耗时间

对 20 个服务组合要求进行 20 次取样，将 LSTM-DQN 方法与文献[5]、文献[7]、文献[8]方法进行实验对比，服务组合消耗的时间如图 8 所示。

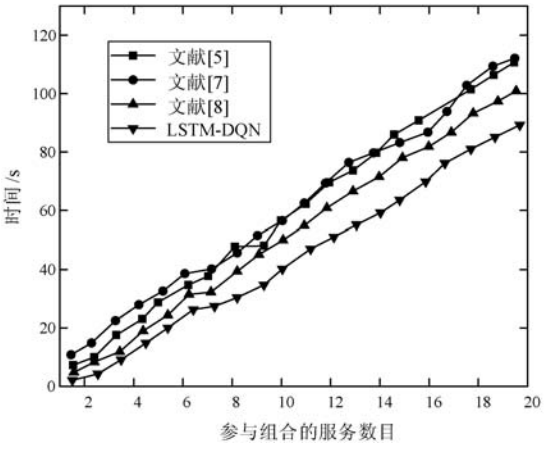

图 8　服务组合消耗的时间对比

可以看出,随着参与组合服务数目的增加,在网络环境经常变动的情形下,服务组合所耗费时间明显都在增加,但是 LSTM-DQN 方法与其他两种方法相比,其消耗的时间相对较少,服务组合的数目越多,其优势越明显。

5　结　语

本文提出一种利用记忆单元和改进 DQN 的 Web 服务组合优化方法,该法利用 Markov 对 Web 服务组合优化问题进行建模,并引入了强化学习的组合优化模型,简化了组合优化过程。并且基于记忆单元对深度 Q 网络算法进行优化,提出 LSTM-DQN 方法,极大地提升了 DQN 算法的全局寻优能力。为验证所提方法的性能,将其基于 QWS 数据集与 DQN 和 RL 方法进行对比分析,结果表明,本文方法相对于其他两种方法在大规模服务环境下对 Web 服务组合优化所消耗时间更短,服务组合成功率更高,具有更强的处理能力和处理效率。

本文方法只考虑一个代理的 Web 服务数量,在未来的工作中,可以考虑将 LSTM-DQN 方法扩展到多代理设置,并针对学习代理数量与服务环境规模之间的权衡问题作进一步的研究。

参　考　文　献

[1] Yong J,Fortino G,Shen W,et al. Special issue on service-oriented collaborative computing and applications[J]. IEEE Transactions on Services Computing,2018,11(2):277 – 278.

[2] Wang H,Gu M,Yu Q,et al. Adaptive and large-scale service composition based on deep reinforcement learning[J]. Knowledge-Based Systems,2019,180:75 – 90.

[3] 丁志军,周泽霞. Web 服务组合测试综述[J]. 软件学报,2018,29(2):299 – 319.

[4] 郭星,陈姗姗,张以文,等. 烟花粒子群优化算法在 Web 服务组合上的应用[J]. 小型微型计算机系统,2018,39(6):1312 – 1316.

[5] Liu Z,Guo S,Wang L,et al. A multi-objective service composition recommendation method for individualized customer: Hybrid MPA-GSO-DNN model[J]. Computers & Industrial Engineering,2019,128:122 – 134.

[6] Traore B B,Foguem B K,Tangara F,et al. Service-Oriented computing for intelligent train maintenance[J]. Enterprise Information Systems,2019,13(1):63 – 86.

[7] Mutanu L,Kotonya G. State of runtime adaptation in service-oriented systems: What,where,when,how and right[J]. IET Software,2019,13(1):14 – 24.

[8] Moustafa A,Ito T. A deep reinforcement learning approach for large-scale service composition[C]//International Conference on Principles and Practice of Multi-Agent Systems,2018:296 – 311.

[9] Quan L,Wang Z,Liu X. A Real-Time Subtask-Assistance strategy for adaptive services composition[J]. IEICE TRANSACTIONS on Information and Systems,2018,101(5):1361 – 1369.

[10] 柴雪霞,马学森,周雷,等. 基于 SMDP 模型的 Web 服务组合优化方法[J]. 合肥工业大学学报(自然科学版),2011,34(10):1496 – 1500.

[11] 牛作东,李捍东. 基于 Python 与 flask 工具搭建可高效开发的实用型 MVC 框架[J]. 计算机应用与软件,2019,36(7):21 – 25.

[12] 高岭,任杰,王海,等. 基于支持向量机的移动 Web 浏览性能优化研究[J]. 计算机学报,2018,41(9):2077 – 2088.

[13] Nawaz F,Asadabadi M R,Janjua N K,et al. An MCDM method for cloud service selection using a Markov chain and the best-worst method[J]. Knowledge-Based Systems,2018,159:120 – 131.

[14] Nastase H,Núñez C. Deriving three-dimensional bosonization and the duality web[J]. Physics Letters B,2018,776:145 – 149.

[15] 白琮,黄玲,陈佳楠,等. 面向大规模图像分类的深度卷积神经网络优化[J]. 软件学报,2018,29(4):1029 – 1038.

[16] Li Y,Hu J,Wu Z,et al. Research on QoS service composition based on coevolutionary genetic algorithm[J]. Soft Computing,2018,22(23):7865 – 7874.

[17] Liu Z,Guo S,Wang L,et al. A multi-objective service composition recommendation method for individualized customer: Hybrid MPA-GSO-DNN model[J]. Computers & Industrial Engineering,2019,128:122 – 134.

[18] 曾棕根. 便携式高性能 WNMP 架构的研制[J]. 计算机应用与软件,2018,35(4):168 – 171.

[19] 刘川莉,蔡乐才,高祥,等. 基于期望值函数的离策略深度 Q 神经网络算法[J]. 四川理工学院学报(自然科学版),2019,32(1):52 – 60.