Wireless Parallel Reinforcement Learning: An Actor-Critic Approach

Ke Xing[1,2], Xinyue Ma[2], and Yanjie Dong[2(✉)]

[1] Lanzhou University, Lanzhou, China
xingk21@lzu.edu.cn
[2] Shenzhen MSU-BIT University, Shenzhen, China
ydong@smbu.edu.cn

Abstract. In this study, we introduced a novel wireless actor-critic method. Leveraging federated learning, wireless terminals could train models while ensuring data privacy, eliminating the requirement to upload raw data to a central server. The recently proposed parallel reinforcement learning framework allowed wireless terminals to maintain multiple instances of the same environment for parallel data generation. To overcome the challenges posed by the double near-far effect during model exchange, we exploit the superposition property of wireless channels. We conducted experiments on a practical environment to validate our approach and assessed its performance by adjusting threshold and power parameters. The experimental results demonstrated that our method could maintain stable signal transmission under specific noise conditions. The wireless actor-critic method presented a valuable solution for wireless machine learning model training, with potential applications in diverse domains. Future work would focus on further optimization, expansion, and practical validation in diverse real-world scenarios.

Keywords: Actor-critic · parallel reinforcement learning · wireless reinforcement learning

1 Introduction

Nowadays, wireless terminals (such as tablets, wireless sensors, internet-of-thing devices, etc.) are producing ever-increasing volume of data [6,8]. The sheer-volume data is of great value for training machine learning models. However, due to privacy concerns, the mobile devices may not be preferred to upload the local data to cloud centers for such model training. Thanks to the recent advancement of edge computing chips (e.g. RISC-V chips and NVIDIA Pascal), the emerging federated learning framework is proposed by allowing the wireless terminals and a central parameter server to orchestrate model training without sharing the raw data [5,15]. More specifically, the wireless terminals only need

This work was supported by the National Nature Science Foundation (NSF) of China: Grant 62102266.

V. C. M. Leung et al. (Eds.): Qshine 2023, LNICST 574, pp. 375–384, 2024.
https://doi.org/10.1007/978-3-031-65123-6_27

to exchange the local models and/or gradients with the parameter server in the federated learning framework [11,14]. The aforementioned merits have inspire the federated learning to be widely utilized in a large variety of intelligent services, such as, keyword prediction [2,4], ubiquitous-health [1,3], and semantic learning [13,16].

When the federated learning framework is used for reinforcement learning tasks, a recent parallel reinforcement learning (PRL) framework is proposed by allowing the wireless terminals to maintain multiple instances of the same environment [7,9]. Then, each wireless terminal can interact with the local instance of environment for parallel generation of training data. Besides, recent advances in reinforcement learning have successfully leveraged the artificial neural networks to parameterized policy and value function. In the context of PRL framework, the parameters of policy and value networks are exchanged between the central server and multiple wireless terminals.

The ever-increasing scale of neural networks demands for high-speed connections to exchange model updates between the wireless terminals and the server. Over-the-air computing leverages the superposition property of wireless channels to achieve the high-speed model exchange. However, the double near-far effect during model exchange can weaken the received signal strength. In this work, we investigate the uplink communication for the wireless PRL framework. More specifically, we develop a wireless actor-critic method that can overcome the double near-far effect. Our salient contributions are summarized as follows.

- We develop an uplink communication protocol that allows the wireless terminals to upload the local model updates to the server per several local recursions. Moreover, the wireless terminals are allowed to upload local model updates via orthogonal channels and uncoded pulse amplitude modulation to overcome the double near-far effect of the over-the-air computing.
- We combine actor network and critic network into one neural network to further reduce the communication overhead during the uplink transmission.

Numerical experiments are used to verify the performance of our proposed wireless actor-critic method.

Organization. The remaining work is organized as follows. The system model is provided in Sect. 2, and the wireless actor-critic method is proposed in Sect 3. Numerical results and concluding remarks are respectively presented in Sect 4 and Sect 5.

2 Preliminary: Markov Decision Process

An MDP is denoted by a quintuple as $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $\mathcal{P} = \{[p_a^{s,s'}] \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|} | s, s' \in \mathcal{S}, a \in \mathcal{A}\}$ collects all action-dependent transition probabilities, $R(S_t, A_t)$ denotes the instantaneous reward function at step t and is assumed bounded by $\bar{r}$ (i.e., $|R(S_t, A_t)| \leq \bar{r}$), and γ is the discount factor.

Let $\pi(a|s)$ denote the policy that determines the probability of taking action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$. The quality of a policy π is measured by the expected accumulated reward given an initial state $s \in \mathcal{S}$, while following policy π to take future actions—that is, the state-value function

$$V^{\pi}(s_t) = \mathbb{E}\Big[\sum_{\tau=0}^{T-t-1} \gamma^{\tau} R(S_{t+\tau}, A_{t+\tau}) \Big| S_t = s_t\Big] \tag{1}$$

where $A_{t+\tau} \sim \pi(\cdot|S_{t+\tau})$ and $s_t \in \mathcal{S}$ with $0 \le t \le T-1$ with T as total steps per episode.

Moreover, the action-value function (a.k.a., Q-function) as

$$Q^{\pi}(s_t, a_t) = \mathbb{E}\Big[\sum_{\tau=0}^{T-t-1} \gamma^{\tau} R(S_{t+\tau}, A_{t+\tau}) \Big| S_t = s_t, A_t = a_t\Big]. \tag{2}$$

Based on (1) and (2), we obtain

$$V^{\pi}(s_t) = \sum_{a_t \in \mathcal{A}} \pi(a_t|s_t) Q^{\pi}(s_t, a_t). \tag{3}$$

When the policy is parameterized by θ^p, the state-value function is denoted by $J(\theta^p) = V^{\pi}(s_0)$. The objective of the considered MDP is obtain an optimal policy θ^p_* that maximizes $J(\theta^p)$ as

$$\theta^p_* = \arg\max_{\theta} J(\theta^p). \tag{4}$$

Based on the policy gradient theorem [12], the gradient of $J(\theta)$ is derived as

$$\nabla J(\theta) = \mathbb{E}\Big[\sum_{a \in \mathcal{A}} \nabla \pi(a|S_t; \theta^p) Q^{\pi}(S_t, a)\Big] \tag{5a}$$

$$= \mathbb{E}\Big[\nabla \log \pi(A_t|S_t; \theta^p) Q^{\pi}(S_t, a)\Big] \tag{5b}$$

$$= \mathbb{E}\Big[\sum_{\tau=0}^{T-t-1} \gamma^{\tau} R(S_{t+\tau}, A_{t+\tau}) \nabla \log \pi(A_t|S_t; \theta^p)\Big] \tag{5c}$$

where (5b) is based on $\nabla \pi(a|S_t; \theta^p) = \pi(a|S_t; \theta^p) \nabla \log \pi(a|S_t; \theta^p)$ with randomness of policy absorbed into the expectation operator.

Note that the term $\sum_{\tau=0}^{\infty} \gamma^{\tau} R(S_{t+\tau}, A_{t+\tau}) \nabla \log \pi(A_t|S_t; \theta^p)$ in (5) is an unbiased estimator of policy gradient $\nabla J(\theta^p)$. However, it is reported that such gradient estimator $\sum_{\tau=0}^{\infty} \gamma^{\tau} R(S_{t+\tau}, A_{t+\tau}) \nabla \log \pi(A_t|S_t; \theta^p)$ experiences an increasing variance with the time horizon. Therefore, the value function $V^{\pi}(S_t)$ is used for variance reduction of the policy gradient estimator [10]. Since the expectation $\mathbb{E}[\sum_{a \in \mathcal{A}} \nabla \pi(a|S_t; \theta^p) V^{\pi}(S_t)] = 0$, we obtain an unbiased variance-reduced estimator of policy gradient as

$$\nabla J(\theta^p) = \mathbb{E}\Big[\Big[\sum_{\tau=0}^{T-t-1} R(S_{t+\tau}, A_{t+\tau}) - V^{\pi}(S_t)\Big] \nabla \log \pi(A_t|S_t; \theta^p)\Big]. \tag{6}$$

Since the value of $V^{\pi}(S_t)$ in (6) is challenging to obtain, we leverage the neural network to approximate the value as $V^{\pi}(S_t) \approx \hat{V}^{\pi}(S_t;\theta^v)$ where θ^v is the model parameter of the value network. Hence, the estimator of policy gradient is obtained as

$$\nabla J(\theta) = \Big[\sum_{\tau=0}^{T-t-1} R(S_{t+\tau}, A_{t+\tau}) - \hat{V}^{\pi}(S_t;\theta^v)\Big] \nabla \log \pi(A_t|S_t;\theta^p) \tag{7}$$

where the model parameter ψ is obtained via least-square minimization as

$$\min_{\psi} \frac{1}{2} \sum_{t=0}^{T-1} \Big[\sum_{\tau=0}^{T-t-1} R(S_{t+\tau}, A_{t+\tau}) - \hat{V}^{\pi}(S_t;\theta^v)\Big]^2. \tag{8}$$

Remark 1. Several different methods can be used to estimate the value function [10]. Here, we choose a straightforward approach by using the nonlinear function approximation as in (8).

3 System Model and Problem Description

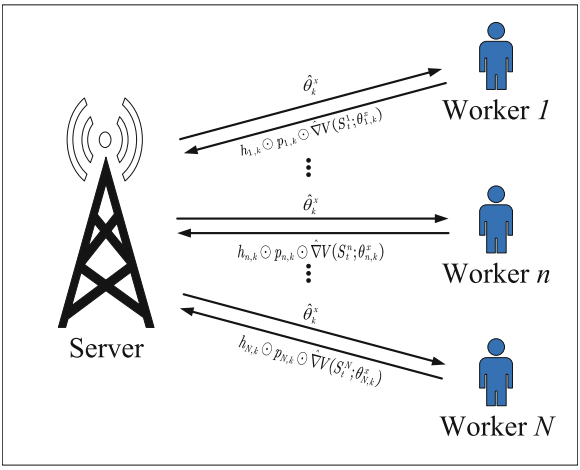


Fig. 1. An illustration of WPRL system.

3.1 Problem Description

We consider a wireless parallel reinforcement learning (WPRL) system that consists of a server and N workers. The workers maintain different instances of the identical episodic MDP described in Sect. 2. Nevertheless, the workers upload the

local model parameters (i.e., policy network θ and value network ψ) in the uplink and receive the global model parameters in the downlink over wireless channels as shown in Fig. 1. The objective of the WPRL system is to allow workers to coordinately learn a common policy network θ and value network ψ.

Define the state-action-reward trajectory of worker n per iteration k as $\{S^n_{t,k}, A^n_{t,k}, R^n_{t+1,k}\}_{t=0}^{L-1}$. Based on the state-action-reward trajectory of worker n, each worker n estimates the local policy gradient and local value gradient per iteration k, respectively, as

$$\hat{\nabla}^p_{n,k} = \sum_{t=0}^{T-1} \Big[\sum_{\tau=0}^{T-t-1} R(S^n_{t+\tau}, A^n_{t+\tau}) - \hat{V}^\pi(S^n_t; \theta^v_k) \Big] \nabla \log \pi(A^n_t | S^n_t; \theta^p_k) \tag{9}$$

and

$$\hat{\nabla}^v_{n,k} = \sum_{t=0}^{T-1} \Big[\sum_{\tau=0}^{T-t-1} R(S^n_{t+\tau}, A^n_{t+\tau}) - \hat{V}^\pi(S^n_t; \theta^v_k) \Big] \nabla \hat{V}^\pi(S^n_t; \theta^v_k). \tag{10}$$

Based on (9) and (10), each worker n can recursively update the model parameters as

$$\theta^p_{k+1}, \theta^v_{k+1} = f(\hat{\nabla}^p_{n,k}, \hat{\nabla}^v_{n,k}, \theta^p_k, \theta^v_k) \tag{11}$$

where the function $f(\cdot)$ is one of the off-the-shelf solver, e.g., Adam and SGD.

3.2 Signal Model

In the downlink, we assume that the server can broadcast the policy model θ^p_k and value model θ^v_k over reliable channels. As shown in Fig. 1, the received signal of the server in the uplink is

$$\sum_{n=1}^{N} h_{n,k} \odot p_{n,k} \odot \frac{\theta^x_k}{\|\theta^x_k\|} + z_{n,k} \tag{12}$$

where $x \in \{p, v\}$, $h_{n,k} = [h_{n,k}[i]]_{i=1}^{d_1}$, $p_{n,k} = [p_{n,k}[i]]_{i=1}^{d_1}$, and $z_{n,k} = [z_{n,k}[i]]_{i=1}^{d_1}$ respectively denote the vector of channel coefficients, the vector of power control variables, and the vector of the real-part additive white Gaussian noise (AWGN) with mean zero and covariance matrix $\sigma^2 I$. More specifically, the terms $h_{n,k}[i]$, $p_{n,k}[i]$, and $z_{n,k}[i]$ are respectively the ith element of $h_{n,k}$, the ith element of $p_{n,k}$, and the ith element of $z_{n,k}$. When Rayleigh fading is considered, each $h_{n,k}[i]$ follows an independent and identically distributed (i.i.d.) circularly symmetric complex Gaussian distribution $\mathcal{CN}(0, \xi^{-\alpha}_{n,k})$ with the propagation distance $\xi_{n,k}$ and the pathloss exponent α. The operator $\odot$ is the Hadamard product of two matrices.

Using uncoded pulse amplitude modulation, the local policy gradients can be detected from the amplitude of received signals at the server. Our objective is to quantify the required energy to obtain a certain accuracy in the over-the-air

computing; therefore, we introduce a long-term energy cost constraint for all workers as

$$\lim_{K\to\infty}\frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N}||p_{n,k}\odot\frac{\theta_k^x}{\|\theta_k^x\|}||^2\leq p_0 \tag{13}$$

where p_0 is the upper bound of energy for training.

Note that each channel can experience deep fading that requires significant large amount of transmit power to guarantee the reliable detection at the server. More specifically, when the amplitude $|h_{n,k}[i]|$ is below a predetermined threshold h_0 (i.e., $|h_{n,k}[i]|<h_0$), the channel i of worker n is assumed to experience deep fading and is not used for information transmission. Denote the channel scheduling vector of worker n per iteration k by $\mathbb{1}_{n,k}=[\mathbb{1}_{n,k}[i]]_{i=1}^{d_1}$, where each term $\mathbb{1}_{n,k}[i]$ equals to 1 when the channel i of worker n is used at iteration k, and 0 otherwise. Therefore, the expectation of channel scheduling indicator $\mathbb{1}_{n,k}[i]$ is obtained as $\mathbb{E}[\mathbb{1}_{n,k}[i]]=\mathbf{Pr}\{|h_{n,k}[i]|\geq h_0\}=\exp(-\xi_{n,k}^{\alpha}h_0^2)$. The terms of θ_k^x of each worker n are uploaded via different radio resource elements[1] In the context of over-the-air computing, different workers can experience different channel conditions for the same channel. Therefore, each worker n needs to factor in the channel scheduling probability to align the terms of $\theta_{n,k}^x$ with a same factor as

$$p_{n,k}[i]=\rho_{n,k}c_{n,k}\mathbb{1}_{n,k}[i]\frac{h_{n,k}^*[i]}{|h_{n,k}[i]|^2} \tag{14}$$

where ρ_k is the channel alignment factor per iteration k, $c_{n,k}:=\exp(\xi_{n,k}^{\alpha}h_0^2)$ is the inverse of expected channel scheduling indicator of worker n per iteration k, and $h_{n,k}^*[i]$ is the conjugate of $h_{n,k}[i]$.

Substituting (14) into (12) and scaling with $1/\rho_{n,k}$, the estimator to the global policy gradient is obtained as

$$\hat{\theta}_k^x=\frac{1}{N}\sum_{n=1}^{N}c_{n,k}\mathbb{1}_{n,k}\odot\theta_{n,k}^x+z_{n,k} \tag{15}$$

where $x\in\{p,v\}$ and $z_k=\frac{1}{N}\sum_{n=1}^{N}z_{n,k}\|\theta_{n,k}^x\|/\rho_{n,k}$.

Taking expectation over both sides of (15), we observe that $\hat{\theta}_k^x$ is an unbiased estimator to corresponding model parameters θ_k^x with $x\in\{p,v\}$. After obtaining (15), the server broadcasts the model parameters (θ_k^p and θ_k^v) to all workers over reliable channels.

3.3 Wireless Actor-Critic Algorithm

In Sect. 3, we have discussed the signal model for information exchange. Since the vanilla actor-critic method is sensitive to the disturbance of channel noise, we

[1] The resource elements denote different time-frequency blocks that are orthogonal to each other. In the WPRL system, the workers upload the local policy gradients over the set of resource elements during the uplink transmission.

propose to leverage the federated average aggregation where the workers perform several local recursions before upload the model parameters to the server over wireless channels. The detailed procedures are summarized in the following table.

Algorithm 1. Wireless Actor-Critic Algorithm

1: **for** $k = 1, \ldots, \infty$ **do**
2: The server broadcasts the model parameters $\hat{\theta}_k^p$ and $\hat{\theta}_k^v$ to all workers
3: Each worker n estimates the channel coefficient vector $h_{n,k}$ and the inverse of channel scheduling probability $c_{n,k}$
4: Each worker n sets the channel alignment factor as

$$\frac{1}{\rho_{n,k}^2} = \frac{c_{n,k}^2}{p_0} \sum_{i=1}^{d_1} \frac{|\mathbb{1}_{n,k}[i]\theta_{n,k}^x|^2}{|h_{n,k}[i]|^2 \|\theta_{n,k}^x\|^2} \tag{16}$$

where $x \in \{p, v\}$
5: Each worker n performs the channel alignment in (14)
6: Each worker n uploads to the server the local normalized gradient $\theta_{n,k}^x / \|\theta_{n,k}^x\|$
7: The server performs gradient alignment by using $[\|\theta_{n,k}^x\| / \rho_{n,k}]_{n=1}^N$ and updates model via (11)
8: **end for**

4 Numerical Results

In this section, we designed robust experiments to confirm the function of the algorithm and verify the effectiveness of the proposed algorithm.

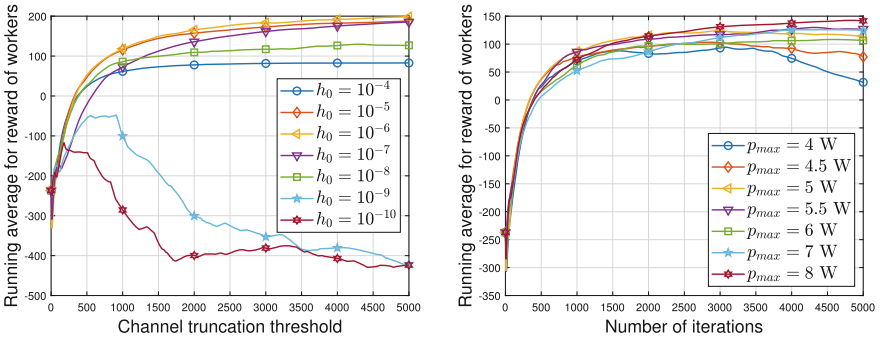
4.1 Experimental Settings

Simulation Environment. We design experiments on the LunarLander-v2 environment, the goal of which is to control a lunar lander and successfully land it on the lunar surface without crashing or running out of fuel. The lander is subject to gravity and has a limited amount of fuel. The state space of LunarLander-v2 typically includes the position, velocity, direction, angular velocity of the lander, and whether the landing legs are in contact with the ground, etc. The action space consists of discrete actions to control the lander, such as firing the main engine, firing the left or right engine, or taking no action.

Parameter Setup. We set the discount factor to be 0.99, the number of iterations to be $5,000$, the number of workers to be 100, the pathloss exponent to be 2.2, the noise to be 1×10^{-10} mW, the number of nodes in the hidden layer is 256, and the location of the server is 30 meters to 50 meters. Our neural network is trained with the Adam optimizer with learning rate is set to 3×10^{-2} and the momentum parameters are set as $(0.9, 0.999)$.

4.2 Results

We validate the effectiveness of our method on LunarLander-v2 by tuning the threshold and power parameters.



(a) Different values of h_0. (b) Different values of $p_{\max}$.

Fig. 2. Convergence behaviors of the investigated wireless actor-critic algorithm.

Under the experimental environment settings in Sect. 4.1, we adjusted the threshold parameter h_0 in the communication process, and carried out reinforcement learning on LunarLander-v2. Figure 2(a) showed that when h_0 was greater than 10^{-9} and less than 10^{-4}, the algorithm can maintain effective convergence characteristics. When the threshold is less than or equal to 10^{-9}, the algorithm diverges. At the same time, if the threshold is too large, such as 10^{-4}, It may cause the model to converge prematurely, the average return value is low.

In addition, we choose $h_0 = 10^{-8}$ under the same parameter environment to study the effect of power variation on the performance of the method. The experimental results are shown in Fig. 2(b). From the experimental results, it can be seen that when $p_{max} = 4$ W, the network shows a divergent trend in the later stage of training. When p_{max} is greater than or equal to 5 W, the average reward curve tends to converge stably during operation. On the whole, the greater the power, the more stable the performance of the algorithm, and the more stable the convergence of the results.

Figure 3 shows the results of the obtained reward under different threshold h_0. It can be seen that when the threshold is greater than 10^{-8} and less than 10^{-4}, the reward has a clear trend of change, and within this range, there is an approximate optimal threshold (i.e., 10^{-6}), the reward reaches the peak.

The experimental results prove that the uplink communication protocol we designed can successfully overcome the double near-far effect of the over-the-air computing, and the network we designed, that is, the network model that combines the actor network and the critic network, successfully reduces the communication overhead during the uplink transmission, and achieves the best performance when the power value is 5.5 W and the threshold is 10^{-6}. This shows that our method can find an optimal h_0 under the above experimental settings.

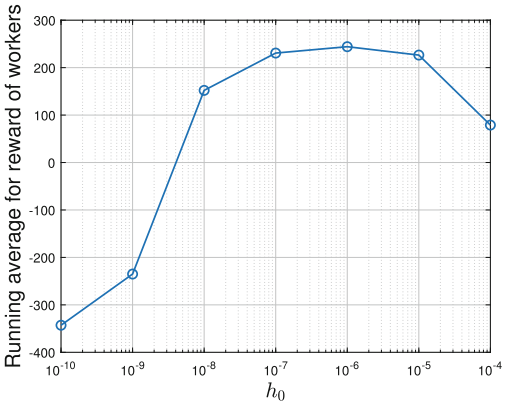


Fig. 3. Obtained reward under different threshold h_0.

5 Concluding Remarks

In this paper, we mainly study the uplink communication protocol of the wireless PRL framework, and we propose a novel actor-critic approach, which uses orthogonal channels and non-coded pulse amplitude modulation to upload local model updates, and successfully overcomes the double near-far effect of the over-the-air computing. At the same time, we combine actor network and critic network into one neural network to further reduce the communication overhead during the uplink transmission. Extensive experiments show that our method can achieve excellent performance in the LunarLander simulation environment, which proves that our method can maintain stable signal transmission under certain noise conditions, thereby improving the reliability of wireless parallel reinforcement learning. In the future, we will continue to explore on this basis to improve the practicability and performance of the model.

References

1. Antunes, R.S., André da Costa, C., Küderle, A., Yari, I.A., Eskofier, B.: Federated learning for healthcare: systematic review and architecture proposal. ACM Trans. Intell. Syst. Technol. (TIST) **13**(4), 1–23 (2022)
2. Diao, E., Tramel, E.W., Ding, J., Zhang, T.: Semi-supervised federated learning for keyword spotting (2023)
3. Halim, S.M., Khan, L., Hamlen, K.W., Thuraisingham, B., Hossain, M.D.: A federated approach for learning from electronic health records. In: 2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), pp. 218–223 (2022). https://doi.org/10.1109/BigDataSecurityHPSCIDS54978.2022.00049
4. Hard, A., Rao, K., Mathews, R., Beaufays, F., Ramage, D.: Federated learning for mobile keyboard prediction (2018)

5. Li, L., et al.: Energy and spectrum efficient federated learning via high-precision over-the-air computation. IEEE Trans. Wirel. Commun. (2023)
6. Li, X., Gong, Y., Huang, K., Niu, Z.: Over-the-air integrated sensing, communication, and computation in IoT networks. IEEE Wirel. Commun. **30**(1), 32–38 (2023)
7. Liu, T., Tian, B., Ai, Y., Li, L., Cao, D., Wang, F.Y.: Parallel reinforcement learning: a framework and case study. IEEE/CAA J. Automatica Sinica **5**(4), 827–835 (2018)
8. Martinez-Gost, M., Pérez-Neira, A., Lagunas, M.Á.: LoRa-based over-the-air computing for Sat-IoT. arXiv:2306.16333 (2023)
9. Nair, A., et al.: Massively parallel methods for deep reinforcement learning. arXiv:1507.04296 (2015)
10. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. arXiv:1506.02438 (2018)
11. Sudharsan, B., et al.: OTA-TinyML: over the air deployment of TinyML models and execution on IoT devices. IEEE Internet Comput. **26**(3), 69–78 (2022)
12. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. MIT Press, Cambridge (2018)
13. Tsouvalas, V., Saeed, A., Ozcelebi, T.: Federated self-training for data-efficient audio recognition. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 476–480 (2022). https://doi.org/10.1109/ICASSP43922.2022.9746356
14. Wang, H., Kaplan, Z., Niu, D., Li, B.: Optimizing federated learning on Non-IID data with reinforcement learning. In: IEEE Conference on Computer Communications, pp. 1698–1707 (2020)
15. Wang, Z., Zhou, Y., Shi, Y., Zhuang, W.: Interference management for over-the-air federated learning in multi-cell wireless networks. IEEE J. Sel. Areas Commun. **40**(8), 2361–2377 (2022)
16. Zhu, X., Wang, J., Hong, Z., Xiao, J.: Empirical studies of institutional federated learning for natural language processing. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 625–634. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.55. https://aclanthology.org/2020.findings-emnlp.55