IEEE Vehicular Technology Magazine, Volume 16, Issue 1, March 2021, pp. 29 - 39
DOI:10.1109/MVT.2020.3015184

Green Deep Reinforcement Learning for Radio Resource Management: Architecture, Algorithm Compression, and Challenges

Zhiyong Du, Yansha Deng, Weisi Guo, Arumugam Nallanathan *IEEE Fellow*, Qihui Wu

***Abstract*—Artificial intelligence heralds a step-change in wireless networks. However, it may also cause irreversible environmental damage due to their high energy consumption. Here, we address this challenge in the context of 5G and beyond, where there is a complexity explosion in radio resource management (RRM). For high dimensional RRM problems in a dynamic environment, deep reinforcement learning (DRL) provides a powerful tool for scalable optimization, but it consumes a high amount of energy over time and risk compromising progress made in green radio research. This paper reviews and analyzes how to achieve green DRL for RRM via both architecture and algorithm innovations. Architecturally, a "cloud based training and distributed decision-making" DRL scheme is proposed, where RRM entities can make lightweight deep local decisions whilst assisted by on-cloud training and updating. At the algorithm level, compression approaches are introduced for both deep neural networks and the underlying Markov Decision Processes, enabling accurate low-dimensional representations of challenges. To scale learning across geographic areas, a spatial transfer learning scheme is proposed to further promote the learning efficiency of distributed DRL entities by exploiting the traffic demand correlations. Together, our proposed architecture and algorithms provide a vision for green and on-demand DRL capability.**

***Index Terms*—energy efficiency; machine learning; deep reinforcement learning; radio resource management; model compression; spatial transfer learning.**

I. Introduction

Future artificial intelligence (AI) driven automation of wireless networks and other critical infrastructures will bring a step change in their ability to create efficient, resilient, and also user-centric services. However, it may also cause irreversible environmental damage due to their high energy consumption and lead to serious global sustainability issues.

The wireless communication industry is one of the fastest growing carbon emission industries, and is expected to deploy millions of base stations and billions of smart phones worldwide. To meet the rapidly increasing traffic volume and demand diversity across network slices, 5G and beyond mobile networks are expected to introduce a number of fundamental innovations apart from physical layer technology enhancements. This brings the need to evolve beyond cognitive radio towards an AI driven radio resource management (RRM) ecosystem to support more fine-grained user-centric service provision (see 3GPP Release 16 TR37.816). This becomes more challenging in highly dynamic environments involving 3D heterogeneous channels. As a result, RRM is becoming increasingly complex, and high dimensional parameter optimization could be a concern.

Zhiyong Du is with National University of Defense Technology, Changsha, China. Yansha Deng is with Kings College London, London, United Kingdom. Weisi Guo is with Cranfield University, Bedford, United Kingdom and Alan Turing Institute, London, United Kingdom. Arumugam Nallanathan is with Queen Mary University of London, London, United Kingdom. Qihui Wu is with Nanjing University of Aeronautics and Astronautics, Nanjing, China. *Corresponding Author: wguo@turing.ac.uk.

A. Deep Reinforcement Learning

The growing complexity in wireless RRM cannot be solved in a scalable manner by the traditional optimization approaches, such as dynamic programming, convex optimization, *etc.*, as they predominantly work on the premise of known optimization model. Recently, the success of deep reinforcement learning (DRL) has opened new pathways to scalable optimization for high dimensional problems. DRL retains the model-free optimization capability of traditional reinforcement learning (RL), suitable for dynamic and online RRM. Meanwhile, in DRL, deep neural network (DNN) is used to approximate policy or value functions for large-scale RL problem, overcoming the intrinsic scalability issue of traditional tabular RL approaches. Specifically, the powerful function approximation and representation learning properties [1] of DNN empower RL with robust and high efficient learning. The application of DRL in 5G and beyond [2] shows great promise and is receiving increased attention in the community. Most existing DRL solutions applied in RRM use off-the-shelf algorithms with little consideration on the RRM feature set. Different from supervised DL applications, where a large amount of samples are available in advance for training, the training samples in DRL can only be generated from the interaction between the RL agent and the wireless network environment. As one interaction iteration in RRM commonly involves parameter configuration and feedback acquisition, the time penalty is not negligible.

B. Demand for Sustainable AI

A growing concern in the machine learning community is the high energy consumption in DRL. A common DNN consists of several stacked layers of neurons with tens or up to hundreds or millions of weights. Such a large number of parameters will generate high computational burden and memory access processes during both training and inference stages. Even sufficient computation capability is provided, the

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

resulting energy consumption is unacceptable for reinforcement learning, especially in battery-constrained devices and areas that do not have access to green electricity supply. For example, smartphones nowadays cannot run object classification with AlexNet in real-time for more than an hour [3].

C. Contribution & Organisation

In view of these challenges, this paper studies how to achieve green DRL for RRM. The contribution is two-fold. On one hand, we briefly reviewed current DRL based RRM and proposed the largely neglected energy consumption challenge of DRL. On the other hand, we proposed some interesting green DRL based RRM solutions and ideas, which mainly include a flexible cloud based learning architecture for mobile devices, DRL algorithm compression and spatial transfer learning schemes with a special consideration on RRM characteristics. The rest of this paper is organized as follows. First, we review the state-of-the-art in Section II. In Section III, we envision an efficient DRL architecture for RRM and outline both architectural design improvements and algorithmic methodological advances. To provide RRM entities with affordable and on-demand DRL capability, a "cloud based training and distributed decision-making" architecture is proposed. In Section IV, to reduce the computation and energy consumption in DRL algorithms, several algorithm compression approaches are introduced, including deep neural network compression, Markov decision process (MDP) model compression and spatial transfer learning scheme. Finally, some challenges are analyzed in Section V.

II. State-of-the-Art: DRL based RRM

A. RRM Formulation

To apply DRL to RRM, it is necessary to map the considered problem into an appropriate DRL model. The core elements of DRL are: (i) state, (ii) action, (iii) reward, (iv) a model of environment dynamics, (v) policy and (vii) DNN implementation [1]. The former four elements define the underling MDP of RL. Specifically, a learner or agent interacts with environment at discrete time epochs. Each time the agent observes some state from the environment and selects an action from an action set. At the beginning of the next time epoch, the agent receives a delayed numerical reward and the environment state updates accordingly.

The *goal* is to find a policy that maps states to probabilities of selecting each possible action in order to maximize the expected sum of the discounted rewards. In absence of the information on the environment dynamics, which is common in wireless network applications, traditional dynamic programming is intractable in solving MDP. Alternately, RL algorithms are proposed to learn the optimal policy from interaction without a model of the environment dynamics. The DNN is used to approximate the optimal policy or value function for large-scale RL problems.

To support the ambitious goal of future mobile networks, the general RRM problem can be seen as "*realizing context-aware optimization to maximize expected accumulative key performance indicator (KPI) such as system quality of service (QoS) or user quality of experience (QoE)*". From this perspective, the mapping between DRL and RRM optimization can be constructed as follows. The communication contexts that specify the situation of user and networks corresponds to states, which may include user profile, spectrum environment, link state, network state, application information and configuration parameters. The target configuration parameters in the considered problem are the actions. The user QoE and/or system KPI achieved in each decision epoch is the instant reward. In some scenarios, it is impractical to observe complete state information. For example, the instant quality state of all channels or traffic generation dynamics of all terminals are hard to acquire. This results in a more challenging RL problem under the partially observable Markov decision process (POMDP) model.

B. DRL Design

Different variations of DRL algorithms have been introduced for RRM, where the RL mechanism and the architecture of DNN largely determine the characteristics of DRL.

1) Reinforcement Learning Mechanism: The RL mechanisms in DRL can be broadly classified into three types [1]: value function based method, policy based method, and actor-critic method.

In *value function* based methods, the action-value function defined by the long-term return when starting in some state with some action and following a given policy subsequently, are estimated and the optimal action(s) for each state correspond to the one(s) with the largest action-value. In *policy based* DRL, a DNN is used to directly derive the stochastic policy, *i.e.*, mapping input state vector to selection probability distribution over all actions. Searching the optimal policy can be gradient-free or gradient-based methods. Note that one advantage of policy based method is that it can handle continuous action, which may be preferred for possible continuous parameters, such as power control, location and distance optimization. In this context, the output are the mean and standard deviations of Gaussian distribution. Finally, the *actor-critic* method is a combination of the above two methods: the state value function is introduced to generate feedback for policy gradient. There are different deep actor-critic algorithm variations, one of most widely used is the asynchronous advantage actor-critic (A3C) that could even run on parallel and asynchronously.

2) Neural Network Architecture for RRM: DNNs are used to exploit the potential correlation of states, actions and policies for efficient approximation of high-dimensional RL problem. Three typical deep neural network (DNN) architectures are widely used.

The first type is *full connection*, where each neuron in hidden layers is connected with all neurons in the previous and next layers. It is a deep version of traditional multi-layer perceptron and is commonly used in combination with feature detection layers.

The second type is *convolution layer*. A neuron in a convolution layer is connected to local patches in the feature maps of the previous layer sharing the same weight. The

TABLE I
RELATED WORK IN USING DRL FOR RESOURCE MANAGEMENT.

Related Work	State	Action	DNN	RL Mechanism
Random Access Control [4]	historical number of idle, collided and successful channels	number of allocated RACH, preambles and repetition values	full connection	value function
Power Control [5], [6]	require load and interference values; RSSs of wireless sensors	power allocation (continuous); power allocation (discrete)	convolution layer; full connection	policy based; actor-critic
Spectrum Sharing [7], [8]	selected channel, capacity and ACK; selected channels and conditions	selected channel	recursion connection; full connection	value function
Coverage & Connectivity [9]	active/sleep state and traffic demand of radio heads	on/sleep of radio head	full connection	value function
Mobility Management [10]	RSRQs of all cells and serving cell	selected cell	recursion connection	actor-critic
Slice Management [11], [12]	number of arriving packets in slices; number of slices of each class	allocated bandwidth to slices; access control of slices	full connection	value function

operation of convolution can be seen as a filter to local groups, which is suitable for array data processing and has the advantage of detecting spatial correlation of states in resource management problems. For example, exploring geography-dependent correlated shadowing, channel gains or complex spectrum interference patterns to enhance spectrum allocation, and extracting temporal-spatial traffic demand distribution for on-demand schedule. Instead of using complex convolutional neural network (CNN), the application of convolution architecture in DRL is relatively flexible, *i.e.*, it is only used as a lightweight feature extraction layer.

The third type is *recursive connection* specialized for processing sequential data. The recursion means neurons also take other neurons' outputs at previous time steps as inputs, to store history information for predicting future output of the sequential data. The popular recurrent neural network (RNN) can predict user behaviors in wireless communication. For example, the content or application request data can be trained to predict service request to achieve personalized service provisioning. In addition, RNN can predict user mobility pattern to improve small cell handover in mobility management.

There are also some RNN variations with memory network architecture that is suitable for longer memory requirement cases, such as long short term memory (LSTM) network.

C. An Illustrative Example: RACH Access for Massive IoT

In the following, a case study in random access is briefly illustrated. In cellular networks, base stations can observe the transmission receptions of both Random Access CHannel (RACH) and data transmission at the end of each time slot, which can be used to predict the traffic and facilitate the performance optimization of future time slots. The complexity of the problem is compounded by the lack of a prior knowledge at the base station regarding the stochastic traffic and unobservable channel statistics. RL based on tabular-Q is not feasible for multi-parameter multi-group dynamic optimization in Narrow-Band IoT (NB-IoT) networks as shown in our work [4] due to the large memory and high computation complexity required for the state-action value table, and the difficulty for the agent to repeatedly experience every state to achieve convergence within a limited time.

This motivates the application of RL based on Linear Approximation (LA-Q) and DRL based on deep neural network (DQN) at the base station with guaranteed convergence capability within largely reduced training time. With the target of maximizing the long-term average number of devices that successfully transmit data, our results in Fig. 1 show that both the DQN and LA-Q approaches outperform the conventional heuristic approaches based on load estimation (LE-URC) in current literature. More importantly, our proposed DQN approach outperforms LA-Q approach with much less training time.

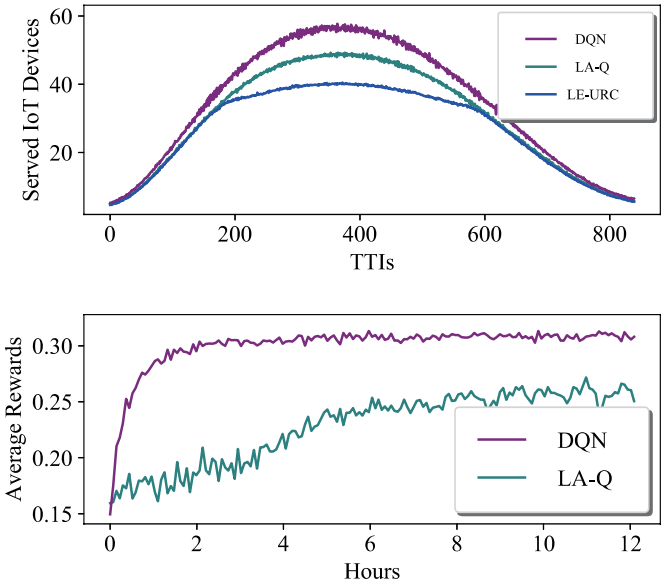


Fig. 1. The number of successfully served devices and the convergence speed.

The detailed mapping of DRL on RRM depends on specific problems and scenarios. A brief summary covering typical RRM issues is presented in Table I. As we have mentioned in Section I, the intensive computation complexity and energy consumption could hinder the application of DRL in future wireless networks. In addition, the features of RRM and networks are not fully investigated.

D. Energy Consumption Concern

Although DRL based RRM is becoming popular, there is little study on the associated energy issue or green machine learning based solutions. Model training in machine learning

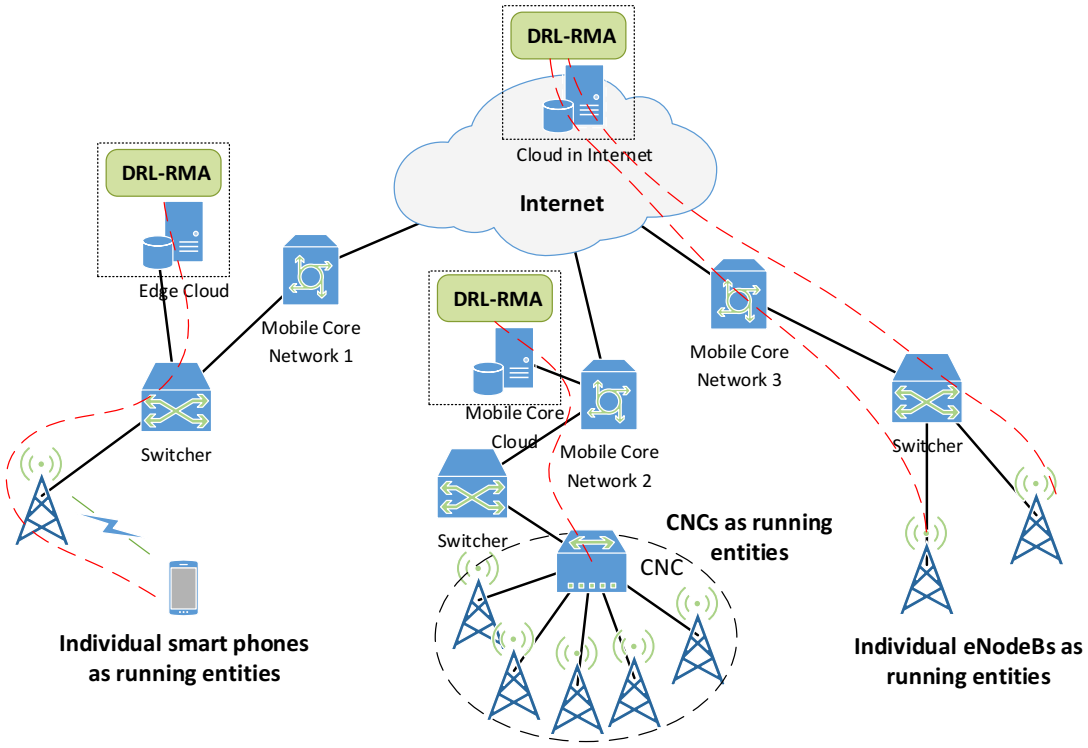


Fig. 2. DRL based RRM providing "DRL as a service" via: (1) service provider, (2) consumer entity, and (3) running entity. A third-party service provider runs and maintains DRL based resource management agent (DRL-RMA) on cloud, which could be either edge cloud in proximity of radio nodes, core cloud or remote cloud in Internet.

often involves a number of processing and memory units (CPU, DRAM, and GPU) that consume energy over time t. Broadly speaking, the energy consumption can be linearly combined as:

$$E_{\text{total}} = \alpha(P_{\text{CPU}} + P_{\text{DRAM}} + P_{\text{GPU}}) \times t \qquad (1)$$

where α is a scaling factor, P_{CPU}, P_{DRAM} and P_{GPU} are the power of CPU, DRAM and GPU during training. The carbon foot print impact depends on the location of training and updating (e.g. renewable rich countries have a low carbon footprint; US estimate is 0.954 pounds per kWh). For example, training the BERT baseline algorithm for language contextual representation consumes 1500kWh, comparable to a medium-haul air journey (4 hour flight per passenger) or a month of human life (US average). Similar metrics will be used to test the potential of some of the proposed Green machine learning techniques in later sections of this paper.

III. Efficient DRL Architecture for RRM

To make DRL based RRM green, we envision a flexible cloud based DRL architecture for mobile networks.

A. Cloud-based On-Demand DRL

For traditional DL applications such as computer vision, DNN is trained and run on centralized hardware resource (such as GPU and TPU). For RRM tasks, the running entities of bases stations or terminal devices could hardly afford the fee for sufficient computation resource deployment nor the associated energy consumption. Moreover, different from offline supervised training, training samples in RRM can be only generated from the interaction between RL agent and the environment and the instant reward feedback could be delayed and even could not be explicitly derived. For this reason, centralized and computation-intense DRL running architecture is not suitable for RRM. Instead, we proposed to decouple the DRL task and run it in a distributed and online manner to improve its efficiency and flexibility.

In order to benefit various types of devices including those that can not afford the computing capability on their own, we envision a "DRL as a service" approach by exploiting the benefits from cloud computing resources, as shown in Fig. 2. Three roles are involved: (i) service provider, (ii) consumer entity, and (iii) running entity. A third-party service provider runs and maintains DRL based resource management agent (DRL-RMA) on cloud, which could be either edge cloud in proximity of radio nodes, core cloud or remote cloud in Internet. To improve efficiency, more sophisticated schemes could be introduced to schedule DRL processing among edge and central clouds according to computation load and communication bandwidth situation. The service provider leases resource management service to different types of consumers by providing on-demand service via DRL-RMA. The introduction of virtualization to 5G has led to three different actors in networks: infrastructure provider, tenant, and the end-user. Infrastructure providers own and manage their physical networks and lease virtualized resources to tenants, and tenants offer network services to users using virtualized resources. Ac-

cordingly, infrastructure providers, tenants and users can be the DRL resource management consumers, depending on network deployments, business models (C2B, B2B), data-generation and processing pipelines, and demands. The running entities are devices that actually run RRM guided by DRL-RMA. For infrastructure providers and tenants, running entities can be a base station, CNC of dense cells, or even user equipments (UEs). While for users, running entities are their UEs, such as smart phones or tablets.

B. Information Flow & Learning Process

The general information flow of DRL service can be described as follows. When a consumer sends a service request for resource management to the DRL-RMA, a DRL optimization process (DRL-OP) will be initiated on the cloud with some necessary negotiation process. This process is responsible for ascertaining the requirements of the consumers, the problem mapping method, targeted running entities, DRL algorithm and other related parameters and requirements. After that, the DRL-RMA will allocate appropriate storage and computing resources and configure DRL algorithm for the DRL-OP. Finally, the DRL-OP will establish a connection with the specified running entities and guide their resource management.

Different from the agent-environment interactive loop of the DRL, the introduction of cloud results in two loops as shown in Fig. 3. The inner loop is the running entity-environment interaction similar to that of DRL running locally. The additional outer loop is the message exchange between running entity and DRL-OP. Notably, as the training, optimization of neural networks and other computing-intensive operations are on the cloud, there is a special requirement on the optimization of DRL: mini-batch gradient descent rather than stochastic gradient descent is used to train deep neural networks. The reason is that stochastic gradient descent generally update gradient in a sample by sample manner, thus it may incur excessive message exchange cost between running entity and DRL-OP, especial for resource management problems with high decision update frequency. On the contrary, mini-batch gradient descent allows for accumulating a set of samples for each update, which greatly reduces the message exchange cost. More specifically, in each iteration of the outer loop, the DRL-OP sends a copy of latest DNN parameters (*e.g.*, weight vectors) to the running entity; the running entity follows the recommended policy, *i.e.*, updating a local neural network with the received parameters, and makes decisions with forward propagation computation for multiple interactions of inner loop; at the end of each outer loop iteration, the running entity sends the accumulated samples of "state-action-reward" pairs to the DRL-OP for training. Obviously, one iteration of outer loop corresponds to multiple inner loop iterations.

Different from DRL in computer games where learning samples could be easily generated, the samples are collected from the practical interaction between RRM entities and the wireless and network environment under the control of RL, which incurs more time and energy costs [13]. This indicates that we actually have to employ incremental learning rather

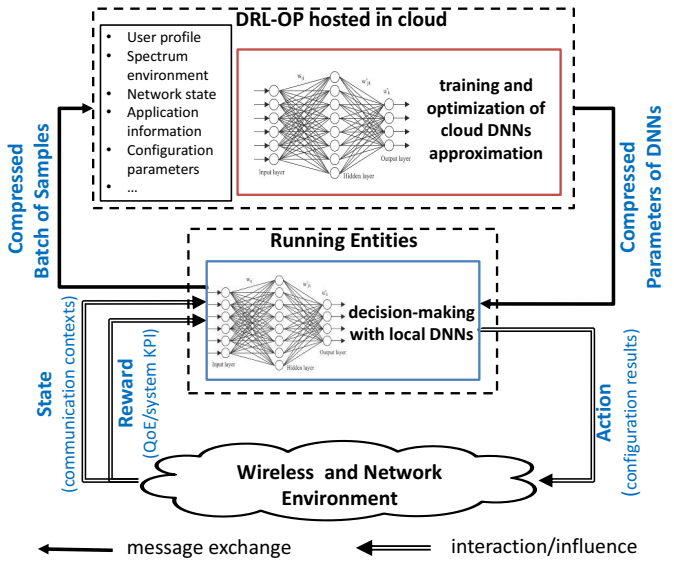


Fig. 3. The interactive DRL loops.

than one-shot training in DL. This is the reason why mini-batch of data samples are sent to the cloud periodically. To alleviate communication cost and protect privacy, both the sample batch and parameters of DNN can be compressed and encrypted before transmission.

This "DRL as a service" approach has several advantages. First, it provides flexible and on-demand resource management service for different consumers. Second, it offers devices with limited computation and battery capability a powerful optimization tool for parameter configuration. Third, it has limited influence on the network and traffic, as the service can be deployed in existing legacy cloud without additional infrastructure changes. The proposed cloud training architecture can be seen as a simplified digital twin of RRM. It tries to build an RRM model on the cloud and guide the RRM policy for real networks. In the future, it is possible to build a more complex network and radio resource management upon a digital twin of real networks. The disadvantages mainly relates to the time delay between air interface to cloud ($\approx$20ms) and its relative magnitude compared to the environment and demand changes - an area for future research.

IV. Reducing DRL Complexity & Energy via Algorithm Compression

In the previous section, we outlined an architecture that enables flexible on-demand RRM, but the computation complexity and energy consumption for DRL remain open challenges that are central to this paper. The size of DRL model largely determines the numbers of required operations and data access, which eventually determines the computation complexity and associated energy consumption. Thus, reducing the DRL algorithm size is crucial for cutting down computation complexity and energy consumption. In other words, we can compress the learning model to achieve a similar performance with significantly reduced parameters and energy consumption.

DRL can be treated as a combination of RL and DNN, where the former is responsible for the trade-off between

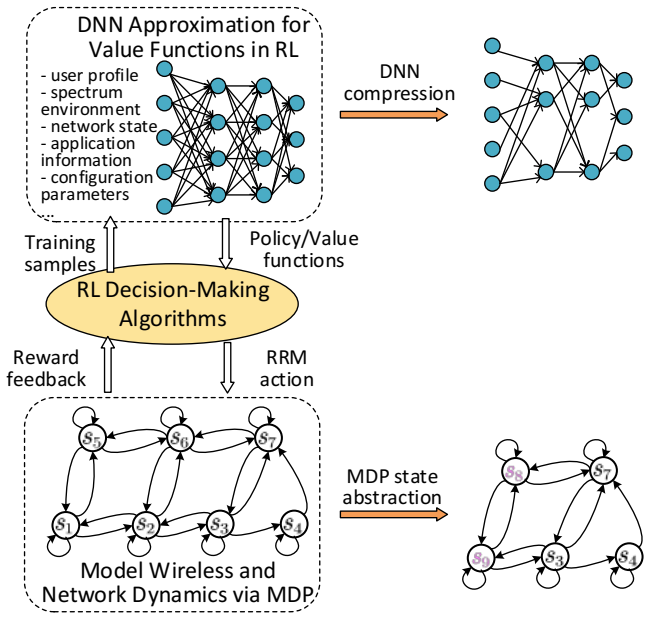


Fig. 4. DNN and MDP compression.

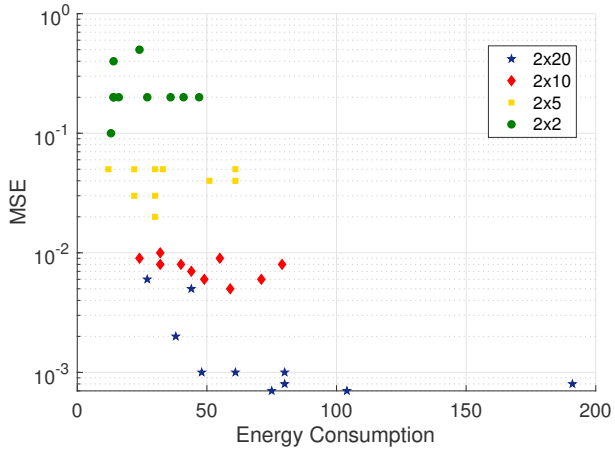


Fig. 5. MSE and energy consumption of different neural networks in power allocation.

exploitation and exploration in online RRM optimization and the latter is responsible for approximating policy or value functions for the former. The involving of RL loop in DNN indicates that the overall algorithm complexity is jointly determined by DNN and the underlying MDP. Here, to achieve lightweight and energy-efficient DRL, we introduce to compress algorithm from three aspects: DNN model, MDP model and learning process.

A. DNN Compression

The required operations and data access overhead in both training and inference of DNN are highly related with the numbers of neurons and the associated weights in it, *i.e.*, a larger model size leads to higher energy consumption. Due to the lack of theoretical results on the optimal DNN architecture, current DNN in DRL application is generally designed based on experience, commonly resulting in a large model size and complexity than necessary. Nevertheless, previous studies have revealed that neural networks are typically over-parameterized, and there is significant redundancy that can be exploited [14]. Therefore, it is possible to achieve similar function approximation performance by removing redundant network architecture (pruning the network as shown in Fig. 4) and only retaining useful parts with greatly reduced model size.

There are several typical ways on compressing DNN by exploiting sparsity in neural networks. One method is reducing the number of parameters. This could be achieved by removing the number of connections/weights, *e.g.*, weights smaller than some predefined threshold are removed, or pruning filters, *i.e.*, removing redundant neurons and connections simultaneously. The second method is architectural innovations, such as replacing fully-connected layers with convolutional layers that is relatively more compact. Another method is weight quantization, *i.e.*, reducing the precision of weights. For example, we can use 8-bit width integer rather than conventional 16-bit or 32-bit width floating-point number to store weights. Already, some of the aforementioned DNN compression practices have emerged in recent mobile deep learning applications.

In Fig. 5, we give an example of multi-channel OFDMA power allocation simulated with a 2-layer neural network with a flexible number of neurons per layer. The training time to convergence and energy consumption are plotted against the mean-square-error (MSE) in loss function for neural-network power allocation compared to optimal solutions. The results show that a 2-3x energy reduction can be achieved when we compress the DNN from 20 to 2 or 5 neurons per layer, at the cost of a 2 fold increase in MSE. Depending on the tolerance of accuracy in power allocation (especially given the lack of accuracy of non-linear power amplifiers), it maybe a good trade-off to achieve an MSE of 10^{-1}, whilst reducing mean energy consumption down from 100 to below 50.

B. MDP Compression

RL is commonly formulated under the framework of MDP or POMDP. The size of MDP is directly determined by the state and action spaces, which grow super-polynomially with the number of variables that characterize the domain. To support fine-grained RRM, we have to adopt high-resolution communication context to accommodate context-aware optimization, which often results in a large-scale MDP model. On the other hand, a smaller model is always desirable for improving energy-efficiency: a small state space will lessen the data storage space of samples and memory access cost in DNN training, and incur less exploration cost in terms of energy and time as sufficiently sampling the states is the intrinsic requirement of RL. Therefore, balancing the context characterization performance and model complexity is needed. For POMDP, in order to reduce the high dimensionality of the problem, hierarchical action space methods can be used to approximate the POMDP problem, achieving a scalable compression.

Considering that most DRL applications use discrete states and actions in DNN, we can compress MDP model in two

stages. At the beginning of MDP modelling, we can appropriately choose the definitions of state and/or action to adjust their resolution. For example, when RSS is one dimension of state or transmit power constitutes action space, we could use a limited number of discretized levels to approximate their dynamic range with controlled performance loss. Besides, during the learning process, the size of MDP model can be further reduced by aggregating identical or similar states as shown in Fig. 4, allowing us to reduce learning complexity with bounded loss of optimality [15]. The similarity of states can be measured in terms of optimal Q function, reward and state transitions, Boltzmann distributions on Q values and *etc.*.

C. Improving Joint Efficiency via Spatial Transfer Learning

Different from supervise learning tasks, accumulating training samples in DRL could be time-consuming and cause delays. While existing DRL based RRM generally use a simulated environment to generate training samples, another approach to compress the low-efficient learning process is transferring knowledge from similar RRM tasks. Transfer learning has already been successful in the widely used *experience replay*, which is transfer learning in the time dimension. Here, we propose it can be done in the spatiotemporal dimension between base stations.

In cellular systems, hyper-dense deployment of network [8] will yield spatially correlated traffic demand patterns amongst neighbouring base stations. Here, we propose that one exploits this phenomenon to achieve joint energy savings via spatial transfer learning. Many actions of multiple base stations need coordination, such as meeting user demand in a large event, offload traffic to each other, and sleep mode / cell expansion. Their RRM policies will have commonalities, which is represented by similar DNN parameters and/or RL policies. To exploit the spatial correlation, spatial transfer learning can be used between adjacent base stations. We can use a spatial kernel that relates to the urban traffic correlation to transfer DRL function parameters among a set of spatially neighbor wireless nodes, for example, the neighbors distributed within a predefined radius of the target RRM entity. This is accomplished by first modelling dynamic spatial correlations between base stations using a flexible framework used commonly in disease and ecology modelling - stochastic integral-difference equation (SIDE). A SIDE function can be parameterized based on the traffic model correlation and then can be used to determine how much information to share between the DRL entities, by relating it to the correlation between traffic demand. The combined spatial DRL process can allow individual DRLs to learn faster by leveraging on the successful results of others, as shown in Fig. 6.

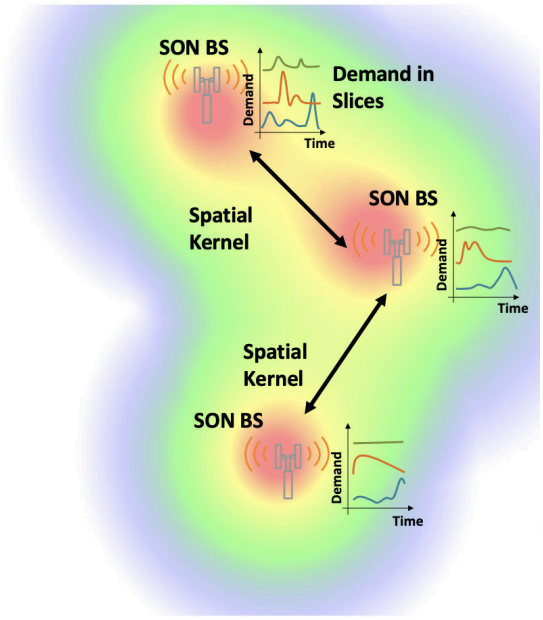


Fig. 6. Spatial transfer learning among self-organization base stations.

V. Open Challenges & Conclusions

Security and privacy are important issues in the proposed open and flexible learning solution. In the cloud based learning architecture, malicious attack during the transmission of DNN parameters trained in the cloud could seriously reduce RRM efficiency and incur excessive training cost. On the other hand, there is a risk of privacy disclosure in the uploaded data samples by RRM entities, as such data carries location dependent traffic and hidden personal activity information. Similarly, the information exchange in spatial transfer learning should also consider the privacy disclosure issue.

High-efficiency learning algorithms are needed to tackle RRM challenges. Most existing DRL algorithms rely on accurate reward feedback and large-scale samples. In wireless networks, no prior samples are available in advance and performance feedback is prone to be perturbed by random factors such as noise and interference. The evolution of underlying traffic distribution may show non-stationary changes due to temporary events, which require RRM policy to adapt quickly. These characteristics call for high efficiency in both learning processing and environmental adaptability.

Current works on DNN compression mainly focus on reducing energy consumption in the inference stage of supervised learning algorithms but neglect the training stage that is more computation-intensive. In addition, in order to compress the inference process, the training process could be even more complex than the standard training without compression. The main reason is that many existing compression approaches rely on iterative training stages to discover the sparsity information and only reduce the DNN size progressively. Thus, reducing the complexity of DNN training stage is appealing. The challenge is that DNN compression is problem specific. Unless the compressed DNN architecture information is given, we have to explore it during the training process.

One drawback of the above mentioned MDP abstraction approaches is that they generally require to know the optimal solution of the MDP. This contradicts the motivation of using DRL to solve the underlying MDP in RRM context. Although the recent abstraction approach has relaxed this condition, the key parameters such as state transition and action value are needed, which is still challenging for practical model-free RRM. Compressing MDP states without prior information is a

desirable and interesting topic. One possible way is employing online abstraction, that is, as the progress of learning, newly learnt model information is used to abstract state progressively.

In conclusion, future AI driven automation of wireless networks and other critical infrastructures will bring about a step change in their ability to create efficient, resilient, and also user-centric services. However, the very same algorithms may also cause irreversible environmental damage due to their high energy consumption and lead to serious global sustainability issues. To achieve our goal of green AI for wireless networking, we have proposed several innovations and linked them to existing literature. On the running architecture, a cloud based on-demand DRL service model is proposed to provide computation capability and battery constrained devices with intelligent RRM. To reduce the computation and energy consumption in DRL, model compression approaches for reducing the sizes of DNN and MDP model are introduced. Finally, by exploiting the correlation feature of RRM tasks among nearby RRM entities, spatial transfer learning is further proposed to promote learning efficiency.

REFERENCES

[1] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, Nov 2017.

[2] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.

[3] T.-J. Yang, Y.-H. Chen, and V. Sze, "Designing energy-efficient convolutional neural networks using energy-aware pruning," in *CVPR*, Jun 2017, pp. 5687–5695.

[4] N. Jiang, Y. Deng, and et al, "Deep reinforcement learning for real-time optimization in nb-iot networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1424–1440, 2019.

[5] Y. Lu, H. Lu, L. Cao, F. Wu, and D. Zhu, "Learning deterministic policy with target for power control in wireless networks," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec 2018, pp. 1–7.

[6] H. Zhang, N. Yang, W. Huangfu, K. Long, and V. C. M. Leung, "Power control based on deep reinforcement learning for spectrum sharing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4209–4219, 2020.

[7] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, Jan 2019.

[8] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1277–1290, June 2019.

[9] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud rans," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[10] Z. Wang, L. Li, Y. Xu, H. Tian, and S. Cui, "Handover control in wireless systems via asynchronous multiuser deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4296–4307, Dec 2018.

[11] R. Li, Z. Zhao, Q. Sun, C. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74 429–74 441, 2018.

[12] N. Van Huynh, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "Optimal and fast real-time resource slicing with deep dueling neural networks," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2019.

[13] H. Zhang, M. Feng, K. Long, G. K. Karagiannidis, and A. Nallanathan, "Artificial intelligence-based resource allocation in ultradense networks: Applying event-triggered q-learning algorithms," *IEEE Vehicular Technology Magazine*, vol. 14, no. 4, pp. 56–63, 2019.

[14] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *NIPS*, Dec 2016, pp. 2082–2090.

[15] D. Abel, D. Hershkowitz, and M. Littman, "Near optimal behavior via approximate state abstraction," in *ICML*, Jun 2016, pp. 2915–2923.

Acknowledgement: This paper is partly funded by EC H2020 grant 778305 (Data Aware Wireless Networks for IoE), 892221 (Green Machine Learning for 5G), and NSF of China grant 61601490.

Zhiyong Du (S'13-M'16) He is currently an Associate Professor with the National University of Defense Technology, China. He received the Marie Skłodowska-Curie Individual Fellowship in 2020. His research interests include quality of experience, learning theory and game theory in wireless communication and networks.

Yansha Deng (S'13–M'18) She is currently a Lecturer (Assistant Professor) with the Department of Informatics, King's College London. Her research interests include molecular communication, Internet of Things, and 5G wireless networks. She was a recipient of the Best Paper Awards from ICC 2016 and GLOBECOM 2017 as the first author.

Weisi Guo (S'07–M'11–SM'17) He is Chair Professor of human machine intelligence at Cranfield University, United Kingdom and a Turing Fellow with The Alan Turing Institute. He is a fellow of the Royal Statistical Society and Senior Member of the IEEE. His research interests are: machine learning for 5G, molecular communications, and complex networks. He won IET Innovation Award and was shortlisted for the Bell Labs Prize three times.

Arumugam Nallanathan (S'97–M'00–SM'05–F'17) He is IEEE Fellow and Professor of wireless communications and the Head of the Communication Systems Research (CSR) Group, Queen Mary University of London. His research interests include 5G wireless networks, Internet of Things (IoT), and molecular communications.

Qihui Wu (M'08, SM'13) Since 2016, he has been a Full Professor with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, China. His current research interests include system design of software defined radio, cognitive radio, and smart radio.



https://dspace.org/

Green deep reinforcement learning for radio resource management: architecture, algorithm compression, and challenges

Du, Zhiyong; Deng, Yansha; Guo, Weisi; Nallanathan, Arumugam; Wu, Qihui

2020-09-24
IEEE
https://doi.org/10.1109/MVT.2020.3015184
Du Z, Deng Y, Guo W, et al., (2020) Green deep reinforcement learning for radio resource management: architecture, algorithm compression, and challenges. IEEE Vehicular Technology Magazine, Volume 16, Issue 1, March 2021, pp. 29-39

Downloaded from a DSpace repository. https://dspace.org/