

1

Deep Reinforcement Learning and Its Applications

1.1 Wireless Networks and Emerging Challenges

Over the past few years, communication technologies have been rapidly developing to support various aspects of our daily lives, from smart cities and healthcare to logistics and transportation. This will be the backbone for the future's data-centric society. Nevertheless, these new applications generate a tremendous amount of workload and require high-reliability and ultrahigh-capacity wireless communications. In the latest report [1], Cisco projected the number of connected devices that will be around 29.3 billion by 2023, with more than 45% equipped with mobile connections. The fastest-growing mobile connection type is likely machine-to-machine (M2M), as Internet-of-Things (IoT) services play a significant role in consumer and business environments. This poses several challenges in future wireless communication systems:

- Emerging services (e.g. augmented reality [AR] and virtual reality [VR]) require high-reliability and ultrahigh capacity wireless communications. However, existing communication systems, designed and optimized based on conventional communication theories, significantly prevent further performance improvements for these services.
- Wireless networks are becoming increasingly ad hoc and decentralized, in which mobile devices and sensors are required to make independent actions such as channel selections and base station associations to meet the system's requirements, e.g. energy efficiency and throughput maximization. Nonetheless, the dynamics and uncertainty of the systems prevent them from obtaining optimal decisions.
- Another crucial component of future network systems is network traffic control. Network control can dramatically improve resource usage and the efficiency of information transmission through monitoring, checking, and controlling data flows. Unfortunately, the proliferation of smart IoT devices and ultradense

radio networks has greatly expanded the network size with extremely dynamic topologies. In addition, the explosive growing data traffic imposes considerable pressure on Internet management. As a result, existing network control approaches may not effectively handle these complex and dynamic networks.

- Mobile edge computing (MEC) has been recently proposed to provide computing and caching capabilities at the edge of cellular networks. In this way, popular contents can be cached at the network edge, such as base station, end-user devices, and gateways to avoid duplicate transmissions of the same content, resulting in better energy and spectrum usage [2, 3]. One major challenge in future communication systems is the straggling problems at both edge nodes and wireless links, which can significantly increase the computation delay of the system. Additionally, the huge data demands of mobile users and the limited storage and processing capacities are critical issues that need to be addressed.

Conventional approaches to addressing the new challenges and demands of modern communication systems have several limitations. First, the rapid growth in the number of devices, the expansion of network scale, and the diversity of services in the new era of communications are expected to significantly increase the amount of data generated by applications, users, and networks [1]. However, traditional solutions may be unable to process and utilize this data effectively to improve system performance. Second, existing algorithms are not well-suited to handle the dynamic and uncertain nature of network environments, resulting in poor performance [4]. Finally, traditional optimization solutions often require complete information about the system to be effective, but this information may not be readily available in practice, limiting the applicability of these approaches. Deep reinforcement learning (DRL) has the potential to overcome these limitations and provide promising solutions to these challenges.

DRL leverages the benefits of deep neural networks (DNNs), which have proven effective in tackling complex, large-scale engines, speech recognition, medical diagnosis, and computer vision. This makes DRL well suited for managing the increasing complexity and scale of future communication networks. Additionally, DRL's online deployment allows it to effectively handle the dynamics and unpredictable nature of wireless communication environments.

1.2 Machine Learning Techniques and Development of DRL

1.2.1 Machine Learning

Machine learning (ML) is a problem-solving paradigm where a machine learns a particular task (e.g. image classification, document text classification, speech recognition, medical diagnosis, robot control, and resource allocation in



Figure 1.1 A data-driven ML architecture.

communication networks) and performance metric (e.g. classification accuracy and performance loss) using experiences or data [5]. The task generally involves a function that maps well-defined inputs to well-defined outputs. The essence of data-driven ML is that there is a pattern in the task inputs and the outcome which cannot be pinned down mathematically. Thus, the solution to the task, which may involve making a decision or predicting an output, cannot be programmed explicitly. If the set of rules connecting the task inputs and output(s) were known, a program could be written based on those rules (e.g. if-then-else codes) to solve the problem. Instead, an ML algorithm learns from the input data set, which specifies the correct output for a given input; that is, an ML method will result in a program that uses the data samples to solve the problem. A data-driven ML architecture for the classification problem is shown in Figure 1.1. The training module is responsible for optimizing the classifier from the training data samples and providing the classification module with a trained classifier. The classification module determines the output based on the input data. The training and classification modules can work independently. The training procedure generally takes a long time. However, the training module is activated only periodically. Also, the training procedure can be performed in the background, while the classification module operates as usual.

There are three categories of ML techniques, including supervised, unsupervised, and reinforcement learning.

- *Supervised learning*: Given a data set $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\} \subseteq \mathbb{R}^n \times C$, a supervised learning algorithm predicts y that generalizes the input–output mapping in $\mathbf{D}$ to inputs $\mathbf{x}$ outside $\mathbf{D}$. Here, $\mathbb{R}^n$ is the n -dimensional feature space C , $\mathbf{x}_i$ is the input vector of the i th sample, y_i is the label of the i th sample, and C is the label space. For binary classification problems (e.g. spam filtering), $C = \{0, 1\}$ or $C = \{-1, 1\}$. For multiclass classification (e.g. face classification), $C = \{1, 2, \dots, K\} (K \geq 2)$. On the other hand, for regression problems (e.g. predicting temperature), $C = \mathbb{R}$. The data points $(\mathbf{x}_i, y_i)$ are drawn from a (unknown) distribution $\mathcal{P}(X, Y)$. The learning process involves learning a function h such that for a new pair $(\mathbf{x}, y) \sim \mathcal{P}$, we have $h(\mathbf{x}) = y$ with high probability (or $h(\mathbf{x}) \approx y$). A loss function (or risk function), such as the mean squared

error function, evaluates the error between the predicted probabilities/values returned by the function $h(\mathbf{x}_i)$ and the labels y_i on the training data.

For supervised learning, the data set $\mathbf{D}$ is usually split into three subsets: $\mathbf{D}_{\text{TR}}$ as the training data, $\mathbf{D}_{\text{VA}}$ as the validation data, and $\mathbf{D}_{\text{TE}}$ as the test data. The function $h(\cdot)$ is validated on $\mathbf{D}_{\text{VA}}$: if the loss is too significant, $h(\cdot)$ will be revised based on $\mathbf{D}_{\text{TR}}$ and validated again on $\mathbf{D}_{\text{VA}}$. This process will keep going back and forth until it gives a low loss on $\mathbf{D}_{\text{VA}}$. The standard supervised learning techniques include the following: Bayesian classification, logistic regression, K -nearest neighbor (KNN), neural network (NN), support vector machine (SVM), decision tree (DT) classification, and recommender system. Note that supervised learning techniques require the availability of labeled data sets.

- *Unsupervised learning* techniques are used to create an internal representation of the input, e.g. to form clusters, extract features, reduce dimensionality, estimate density. Unlike supervised learning, these techniques can deal with unlabeled data sets.
- *Reinforcement learning (RL)* techniques do not require a prior dataset. With RL, an agent learns from interactions with an external environment. The idea of learning by interacting with a domain is an imitation of humans' natural learning process. For example, at the point when a newborn child plays, e.g. waves his arms or kicks a ball, his/her brain has a direct sensorimotor connection with its surroundings. Repeating this process produces essential information about the impact of actions, causes and effects, and what to do to reach the goals.

Deep learning (DL), a subset of ML, has gained popularity thanks to its DNN architectures to overcome the limitations of ML. DL models are able to extract the key features of data without relying on the data's structure. The "deep" in deep learning refers to the number of layers in the DNN architecture, with more layers leading to a deeper network. DL has been successfully applied in various fields, including face and voice recognition, text translation, and intelligent driver assistance systems. It has several advantages over traditional algorithms as follows [6]:

- *No need for system modeling*: The system must be well modeled in traditional optimization approaches to obtain the optimal solution. Nevertheless, all information about the system must be available to formulate the optimization problem. In practice, this may not be feasible, especially in future wireless networks where users' behaviors and network states are diverse and may randomly occur. Even if the optimization problem is well defined, solving it is usually challenging due to nonconvexity and high-dimensional problems. DL can efficiently address all these issues by allowing us to be data-driven. In particular, it obtains the optimal solution by training the DNN with sufficient data.
- *Supports parallel and distributed algorithms*: In many complex systems, DL may require a large volume of labeled data to train its DNN to achieve good training

performance. DL can be implemented in parallel and distributed to accelerate the training process. Specifically, instead of training with single computing hardware (e.g. graphics processing unit [GPU]), we can simultaneously leverage the computing power of multiple computers/systems for the training process. There are two types of parallelism in DL: (i) model parallelism and (ii) data parallelism. For the former, different layers in the deep learning model can be trained in parallel on other computing devices. The latter uses the same model for every execution unit but trains the model with different training samples.

- *Reusable*: The trained model can be reused in other systems/problems effectively with DL. Using well-trained models built by experts can significantly reduce the training time and related costs. For example, AlexNet can be reused in new recognition tasks with minimal configurations [6]. Moreover, the trained model can be transferred to a different but related system to improve its training using the transfer learning technique. The transfer learning technique can obtain a good training accuracy for the target system with a few training samples as it can leverage the gained knowledge in the source system. This is very helpful as collecting training samples is costly and requires human intervention.

There are several types of DNNs, such as artificial neural networks (ANNs) (i.e. feed-forward neural networks), convolutional neural networks (CNNs), and recurrent neural networks (RNNs). However, they consist of the same components: (i) neurons, (ii) weights, (iii) biases, and (iv) functions. Typically, layers are interconnected via nodes (i.e. neurons) in a DNN. Each neuron has an activation function to compute the output given the weighted inputs, i.e. synapses and bias [7]. During the training, neural network parameters are updated by calculating the gradient of the loss function.

1.2.2 Artificial Neural Network

An ANN is a typical neural network known as a feed-forward neural network. In particular, ANN consists of nonlinear processing layers, including an input layer, several hidden layers, and an output layer, as illustrated in Figure 1.2. A hidden layer uses the outputs of its previous layer as the input. In other words, ANN passes information in one direction from the input layer to the output layer. In general, ANN can learn any nonlinear function; thus, it is often referred to as a universal function approximator. The essential component of this universal approximation is activation functions. Specifically, these activation functions introduce nonlinear properties to the network and thus help it to learn complex relationships between input data and their outputs. In practice, there are three main activation functions widely adopted in DL applications: (i) sigmoid, (ii) tanh, and (iii) relu [6, 8]. Due to its effectiveness and simplicity, ANN is the most popular neural network used in DL applications.

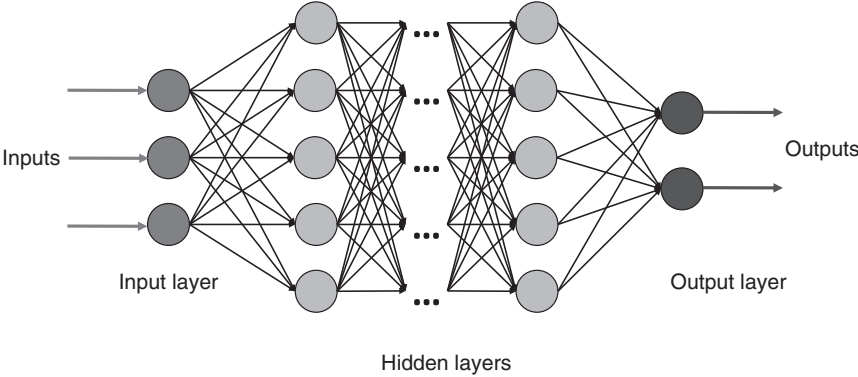


Figure 1.2 Artificial neural network architecture.

1.2.3 Convolutional Neural Network

Another type of deep neural network is CNN, designed mainly to handle image data. To do that, CNN introduces new layers, including convolution, Rectified Linear unit (Relu), and pooling layers, as shown in Figure 1.3.

- *Convolution layer* deploys a set of convolutional filters, each of which handles certain features from the images.
- *Relu layer* can map negative values to zero and maintain positive values during training, and thus it enables faster and more effective training.
- *Pooling layer* is designed to reduce the number of parameters that the network needs to learn by performing down-sampling operations.

It is worth noting that a CNN can contain tens or hundreds of layers depending on the given problem. The filters can learn simple features such as brightness and edges and then move to complex properties that uniquely belong to the object. In general, CNN performs much better than ANN in handling image data. The main reason is that CNN does not need to convert images to one-dimensional vectors before training the model, which increases the number of trainable parameters and cannot capture the spatial features of images. In contrast, CNN uses convolutional layers to learn the features of images directly. As a result, it can effectively

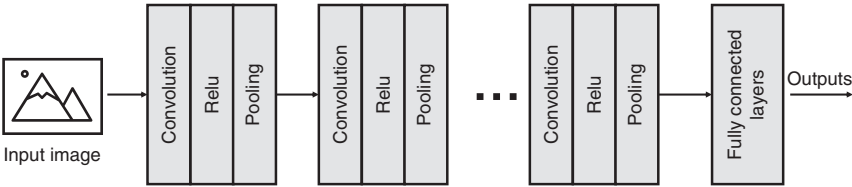


Figure 1.3 Convolutional neural network architecture.

learn all the features of input images. In the area of wireless communications, CNN is a promising technique to handle network data in the form of images, e.g. spectrum analysis [9–11], modulation classification [12, 13], and wireless channel feature extraction [14].

1.2.4 Recurrent Neural Network

An RNN is a DL network structure that leverages previous information to improve the learning process for the current and future input data. To do that, RNN is equipped with loops and hidden states. As illustrated in Figure 1.4, by using the loops, RNN can store previous information in the hidden state and operate in sequence. In particular, the output of the RNN cell at time $t - 1$ will be stored in the hidden state h_t and will be used to improve the training process of the input at time t . This unique property makes RNN suitable for dealing with sequential data such as natural language processing and video analysis.

In practice, RNN may not perform well with learning long-term dependencies as it can encounter the “vanishing” or “exploding” gradient problem caused by the backpropagation operation. Long short-term memory (LSTM) is proposed to deal with this issue. As illustrated in Figure 1.5, LSTM uses additional gates to decide

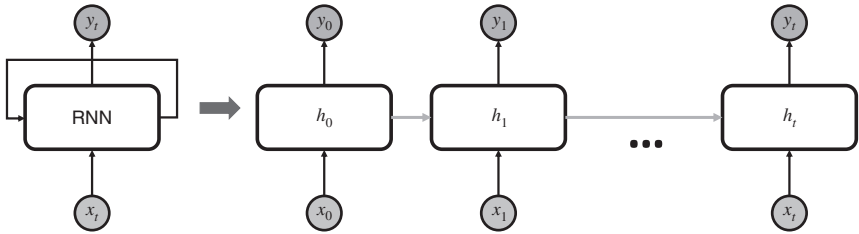


Figure 1.4 Recurrent neural network architecture.

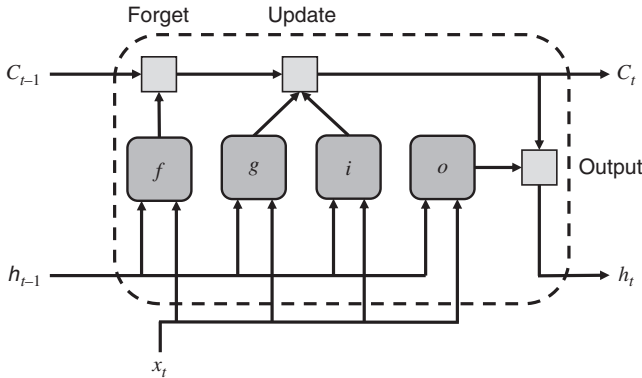


Figure 1.5 LSTM network architecture.

the proportion of previous information in the hidden state for the output and the next hidden state. Recently, RNN, especially LSTM, has emerged as a prominent structure for signal classification tasks in wireless communications [15–18]. The reason is that signals are naturally sequential data since they are usually collected over time.

1.2.5 Development of Deep Reinforcement Learning

In the last 20 years, RL [19] has become one of the most important lines of research in ML and has had an impact on the development of artificial intelligence (AI). An RL process entails an agent making regular decisions, observing the results, and then automatically adjusting their strategy to achieve the optimal policy to maximize the system performance, as illustrated in Figure 1.6. In particular, given a state, the agent makes an action and observes the immediate reward and next state of the environment. Then, this experience (i.e. current state, current action, immediate reward, and next state) will be used to update the Q-table using the Bellman equation to obtain the optimal policy. By interacting with the environment, RL can effectively deal with the dynamics and uncertainty of the environment. When the environment changes, the algorithm can adapt to the new properties and obtain a new optimal policy. While traditional RL techniques are limited to low-dimensional problems, DRL techniques are very effective to handle large-dimensional issues. In particular, DRL is a combination of RL and a DNN, where the DNN generates an action as the output given the

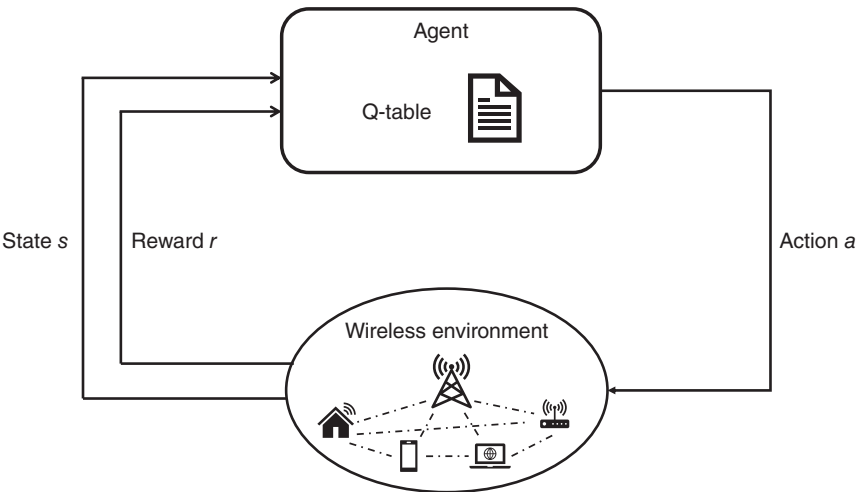


Figure 1.6 An illustration of a reinforcement learning process.

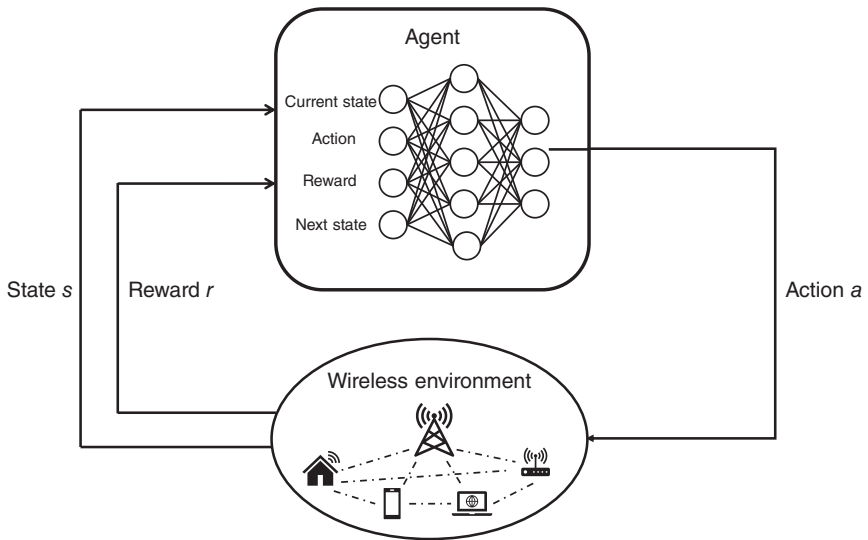


Figure 1.7 An illustration of a DRL process.

current state of the system as the input, as illustrated in Figure 1.7. This unique feature of DRL is beneficial in future wireless communication systems in which users and network devices are diverse and dynamic.

1.3 Potentials and Applications of DRL

The applications of DRL have the potential to attract significant attention from the research community and industry. The reasons for this can be explained based on the following observations.

1.3.1 Benefits of DRL in Human Lives

According to the McKinsey report, AI techniques, including DL and RL, can create between US\$ 3.5 trillion and US\$ 5.8 trillion annually across nine business functions in 19 industries. Experts believe DRL is at the cutting edge and has finally been applied practically to real-world applications. They also believe that the development of DRL will significantly impact AI advancement and can eventually bring us closer to artificial general intelligence (AGI).

1.3.2 Features and Advantages of DRL Techniques

DRL inherits outstanding advantages of both RL and DL techniques, thereby offering distinctive features which are expected to address and overcome diverse technical challenges existing for many years [7, 20].

- *Ability to solve sophisticated optimizations*: DRL can solve complex optimization problems by simply learning from its interactions with the system. As a result, it allows network controllers (e.g. gateways, base stations, and service providers) to solve nonconvex optimization problems to maximize the system performance in certain aspects without requiring complete and accurate network properties in advance.
- *Intelligent learning*: By interacting with the communication and networking environment, DRL can intelligently adapt its optimal policy corresponding to the system's conditions and properties. Thus, by using DRL, network controllers can efficiently perform optimal actions (e.g. beam associations, handover decisions, and caching and offloading decisions) in a real-time manner.
- *Enable fully autonomous systems*: With DRL, network entities can automatically and intelligently make optimal control decisions with minimal or without information exchange by interacting with the environment (e.g. wireless connections, users, and physical infrastructures). For that, DRL can greatly reduce communication overhead as well as enhance the system's security and robustness.
- *Overcome limitations of conventional ML techniques*: Compared with conventional RL approaches, DRL has a much faster convergence speed, especially in large-scale problems with large action and state spaces (e.g. IoT networks that contain a massive number of devices with different configurations). As a result, with DRL, network controllers can quickly adapt with the new conditions of the network and adjust their policies (e.g. user association, transmit power, and spectrum access) to maximize the long-term performance of the system.
- *Novel solutions to several conventional optimization problems*: Various problems in communications and networking, such as data offloading, interference control, and cyber-security, can be modeled as optimization problems. Unfortunately, conventional approaches cannot efficiently solve these problems due to their nonconvexity and the lack of global information. DRL, on the other hand, can effectively address these issues by simply interacting with the system to learn and obtain the optimal solutions, thanks to the power of DNNs.

1.3.3 Academic Research Activities

A substantial amount of research activities related to DRL in communications and wireless networks have been initiated. These days, the major flagship IEEE conferences (e.g. IEEE International Conference on Communications [ICC], IEEE

Global Communications Conference [GLOBECOM], IEEE Wireless Communications and Networking Conference [WCNC], and IEEE Vehicular Technology Conference [VTC]) have special sessions on the DRL techniques. Some recent IEEE magazines and journals have had their special issues on this topic, e.g. IEEE Transactions on Cognitive Communications and Networking's special issue on "Deep Reinforcement Learning for Future Wireless Communication Networks" in 2019 and IEEE IoT Journals' special issue on "Deep Reinforcement Learning for Emerging IoT Systems" in 2019. Recently, there are quite many tutorials on this topic presented in IEEE flagship conferences, such as IEEE GLOBECOM, IEEE ICC, and IEEE WF-IoT. Clearly, the research on the DRL in wireless networks is emerging and has already received significant attention from researchers worldwide.

1.3.4 Applications of DRL Techniques

As mentioned, DRL can significantly improve the learning process of reinforcement learning algorithms by using DNNs. As a result, there are many practical applications of DRL in various areas such as robotics, manufacturing, and healthcare as follows [7, 20]:

- **Video games:** In complex interactive video games, DRL is utilized to enable the agent to adapt its behavior based on its learning from the game in order to maximize the score. A wide range of PC games, such as StarCraft, Chess, Go, and Atari, utilize DRL to enable enemies to adapt their moves and tactics based on the human player's performance, as illustrated in Figure 1.8.

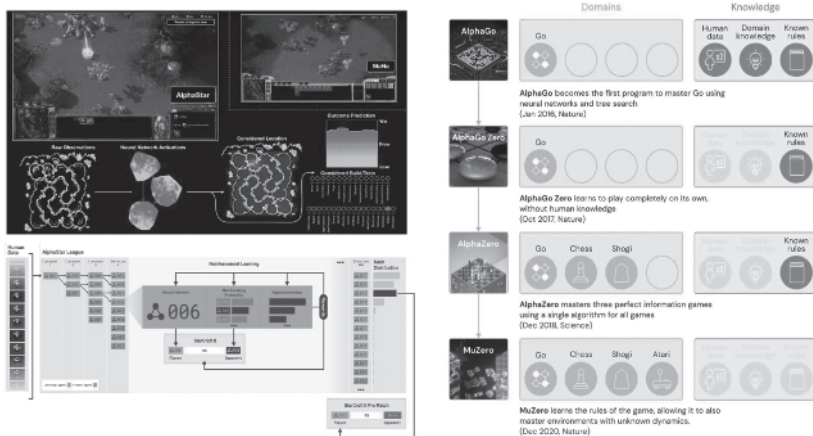


Figure 1.8 Google DeepMind's DRL applications in playing games. Adapted from [21, 22].

- **Chemistry:** DRL has been applied to optimize chemical reactions by using an agent to predict the actions that will lead to the most desirable chemical reaction at each stage of the experiment. In many cases, DRL has been found to outperform traditional algorithms used for this purpose.
- **Manufacturing:** DRL is used by many manufacturing companies, such as Fanuc, to assist their robots in picking up objects from one box and placing them in a container with high speed and accuracy [23]. These DRL-assisted machines are able to learn and memorize the objects they handle, allowing them to perform tasks efficiently and effectively [24–26]. In warehouses and e-commerce facilities, these intelligent robots are utilized to sort and deliver products to customers. For example, Tesla’s factories used DRL to reduce the risk of human error by performing a significant portion of the work on vehicles.
- **Robotics:** Google developed the Soft actor-Critic algorithm, which allows robots to learn real-world tasks using DRL efficiently and safely, without the need for numerous attempts, as depicted in Figure 1.9. The algorithm has been successful in quickly training an insect-like robot to walk and a robot hand to perform simple tasks. It helps protect the robot from taking actions that could potentially cause harm.
- **Healthcare:** DRL can be used on historical medical data to determine the most effective treatments and to predict the best treatment options for current patients. For instance, DRL has been applied to predict drug doses for sepsis patients, identify optimal cycles for chemotherapy, and select dynamic treatment regimens combining hundreds of medications based on medical registry data [28–30].

In addition, many other real-world applications can be summarized in Figure 1.10.

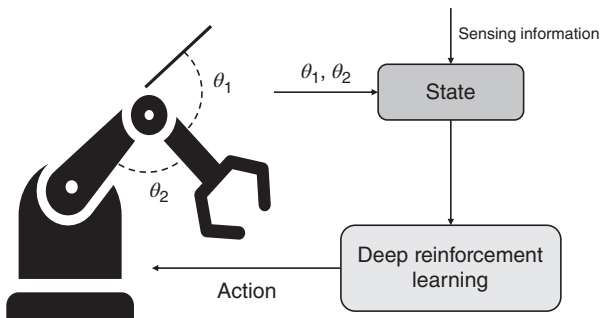


Figure 1.9 Applications of DRL in robotics. Adapted from Pilarski et al. [27].

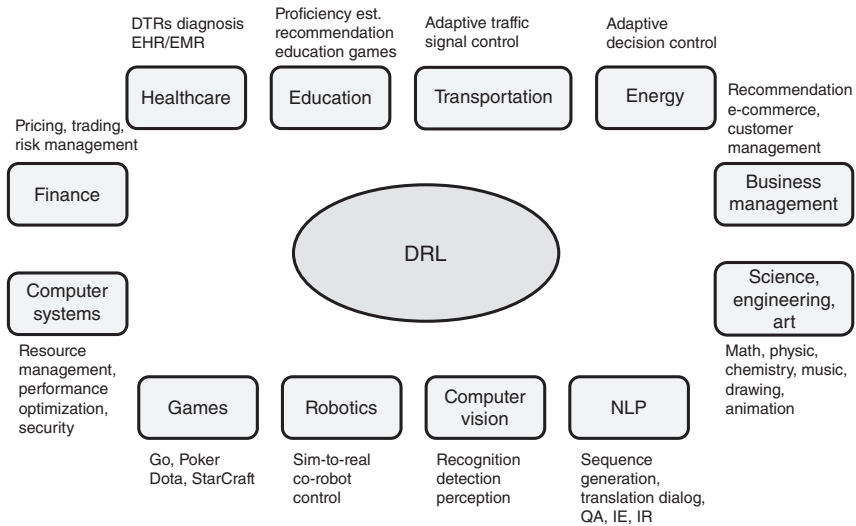


Figure 1.10 Real-word applications of DRL. EHR; electronic health record, EMR; electronic medical record, NLP; natural language processing, QA; question answering, IE; information extraction, IR; information retrieval.

1.3.5 Applications of DRL Techniques in Wireless Networks

With many applications of DRL in practice, especially for wireless and mobile networks, many patents for DRL techniques have been issued in recent years. KR102240442B1 patent [31] claims a novel idea of using DRL for proactive caching in millimeter-wave vehicular networks. CN109474980B invention [32] provides a wireless network resource allocation method based on DRL, which can improve the energy efficiency in a time-varying channel environment to the maximum extent with lower complexity. CN110488861B invention [33] discloses an unmanned aerial vehicle (UAV) track optimization method and device based on DRL and an UAV. More importantly, many applications of DRL have been deployed in our real lives, such as applications of DRL for self-driving cars and industrial automation. AWS DeepRacer is an example of an autonomous racing car that utilizes DRL to navigate a physical track using cameras and an RL model to control its throttle and direction. Similarly, Wayve.ai has recently used DRL to train a car to drive within a single day by using an algorithm to complete a lane following task using a deep network with four convolutional layers and three fully connected layers. The example in Figure 1.11 demonstrates the lane following task from the perspective of the driver. Obviously, the development of DRL will be the next big thing for future wireless networks.



Figure 1.11 DRL applications in self-driving cars. Mwiti [34].

1.4 Structure of this Book and Target Readership

1.4.1 Motivations and Structure of this Book

DRL has been shown to be a promising approach for addressing high-dimensional and continuous control problems using deep neural networks as powerful function approximators. The integration of DRL into future wireless networks will transform the traditional model-based network optimization approach to a model-free approach and enable the network to meet a variety of application requirements. By interacting with the environment, DRL provides a mechanism for network entities to make autonomous decisions and solve complex, model-free problems such as spectrum access, handover, scheduling, caching, data offloading, and resource allocation [35–37]. This can not only reduce communication overhead but also improve network security and reliability. While DRL has demonstrated significant potential for addressing emerging issues in complex wireless networks, there are still domain-specific challenges that require further investigation, such as designing appropriate DNN architectures for future network optimization problems, addressing state explosion in dense networks, handling multiagent learning in dynamic networks, dealing with limited training data and exploration space in practical networks, and balancing the trade-off between information quality and learning performance.

There are five primary objectives of this book.

- Introduce an emerging research topic together with promising applications of DRL in wireless networks.
- Provide fundamental knowledge, including comprehending theory, building system models, formulating optimization problems, and designing appropriate algorithms to address practical problems in wireless communications and networks using DRL.
- Provide a short tutorial to help the readers learn and practice programming DRL under a specific scenario.

- Provide a comprehensive review of the state-of-the-art research and development covering different aspects of DRL in wireless communications and networks.
- Introduce emerging applications of DRL in wireless communications and networks and highlight their challenges and open issues.

To achieve the objectives above, the book includes three main parts as follows:

Part I: Fundamentals of Deep Reinforcement Learning

This part presents an overview of the development of DRL and provides fundamental knowledge about theories, formulation, design, learning models, algorithms, and implementation of DRL, together with a particular case study to practice.

Chapter 1: The first chapter provides an overview of DRL, its development, and potential applications. In particular, the chapter starts with the development of wireless networks and the emerging challenges that researchers and practitioners face. We then present the remarkable development of ML and its significant impacts on all aspects of our lives. After that, we introduce recent breakthroughs in ML, mainly focusing on DRL, and discuss more details about DRL's outstanding features and advantages to address future wireless networks' challenges.

Chapter 2: In the second chapter, we provide fundamental background and theory of the Markov decision process (MDP), a critical mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision-maker. Specifically, essential components of an MDP and some typical extension models are presented. After that, specific solutions to address MDP problems, e.g. linear programming, value iteration, policy iteration, and reinforcement learning, are reviewed.

Chapter 3: Then, we will discuss DRL, a combination of RL and DL to address the current drawbacks of RL. In particular, we discuss more details how different DL models can be integrated into RL algorithms to speed up the learning processes. Many advanced DRL models are reviewed to provide a comprehensive perspective for the readers.

Chapter 4: In this chapter, we provide a particular scenario with detailed implementation to help the readers have a deeper understanding of step-by-step how to design, analyze, formulate, and solve an MDP optimization problem with DRL codes using conventional programming tools, e.g. TensorFlow. In addition, many simulation results are provided to discuss different aspects of implementing DRL and evaluate the impacts of parameters on learning processes.

Part II: Applications of DRL in Wireless Communications and Networking

This part focuses on studying diverse applications of DRL to address various problems in wireless networks, such as caching, offloading, resource sharing, and

security. We show example problems at the physical, media access control (MAC), network, and application layers and potential applications of DRL techniques to address them. Comparisons and detailed discussions are also provided to help the readers to have a comprehensive view of the advantages and limitations of using DRL to solve different problems in wireless networks.

Chapter 5 – DRL at the Physical Layer: The need for high-reliability and ultra-high capacity wireless communication has driven significant research into 5G communication systems. However, traditional techniques used for the design and optimization of these systems often struggle with the complexity and high dimensionality of the problems. In recent years, DRL has been recognized as a promising tool for addressing these complicated design and optimization problems. In this chapter, we will examine the potential applications of DRL in addressing three key issues at the physical layer of communication systems: beamforming, signal detection and channel estimation, and channel coding.

Chapter 6 – DRL at the MAC Layer: In modern networks like the IoT, sensors and mobile users often need to make independent decisions, such as selecting channels and base stations, in order to achieve their goals, such as maximizing throughput. However, this can be difficult due to the dynamic and uncertain nature of these networks. DRL algorithms like deep Q-learning (DQL) can help network entities learn and understand the states of the network, allowing them to make optimal decisions. In this chapter, we will review applications of DRL for addressing three issues at the MAC layer of these networks: resource allocation, channel access, and heterogeneous MAC protocols.

Chapter 7 – Network Layer: Network traffic control, which involves monitoring, inspecting, and regulating data flows, is a vital aspect of network systems that can improve information delivery efficiency and resource utilization. However, the proliferation of smart mobile devices in the IoT era and the future ultradense radio networks has greatly increased the scale of these networks and introduced highly dynamic topologies, leading to a rapid growth in data traffic and significant challenges in managing the Internet. In this chapter, we will discuss the potential applications of DRL to address three critical issues at the network layer: traffic routing, network slicing, and network intrusion detection.

Chapter 8 – Application and Service Layer: MEC has the potential to support computation-intensive applications, such as autonomous and cooperative driving, and reduce end-to-end latency in wireless and mobile networks. It also enables the development and deployment of intelligent systems and applications, such as intelligent transportation systems and smart city applications. However, there are several challenges that have yet to be addressed in the use of MEC, such as how to design and implement an efficient network framework, how to place MEC

servers to serve a large number of users, and how to design offloading, caching, and communication strategies to improve system performance. In this chapter, we will review the applications of DRL frameworks in addressing three issues at the application and service layer: content caching, computation offloading, and data/computing task offloading.

Part III: Challenges, Approaches, Open Issues, and Emerging Research Topics

This part is dedicated to discussing some technical challenges along with open issues and introduces advanced approaches of DRL to address emerging issues in networking and communications.

Chapter 9: Although DRL has many advantages and has been developing rapidly, some technical limitations still need to be resolved. This chapter reviews some challenges in implementing DRL approaches in wireless networks, e.g. the curse of dimensionality, online learning, offline training processes, and quick adaptation to the uncertainty and dynamics of wireless environments. Several open issues in deploying DRL in wireless networks, e.g. inheriting and transferring knowledge among intelligent network entities, are discussed.

Chapter 10: In the last chapter, we introduce some emerging research topics in the context of applications of DRL in future wireless networks. These topics include joint radar and data communications, ambient backscatter communications, intelligent reflecting surface-aided communications, and rate splitting communications, as well as several advanced DRL models such as deep reinforcement transfer learning, generative adversarial networks, and meta-reinforcement learning.

Figure 1.12 presents the structure of the book.

1.4.2 Target Readership

The target readers for the proposed book consist of

- Researchers and communications engineers who are interested in studying applications of ML, especially RL and DRL for future wireless networks.
- Researchers and communications engineers who are interested in the state-of-the-art research on ML together with emerging problems for future autonomous communication systems.
- Developers and entrepreneurs who are interested in developing DRL applications in autonomous systems, e.g. smart homes, intelligent vehicles, and UAVs.
- Graduate students who are interested in learning and obtaining comprehensive knowledge on the design, analysis, and implementation of DRL to address practical issues in wireless networks.

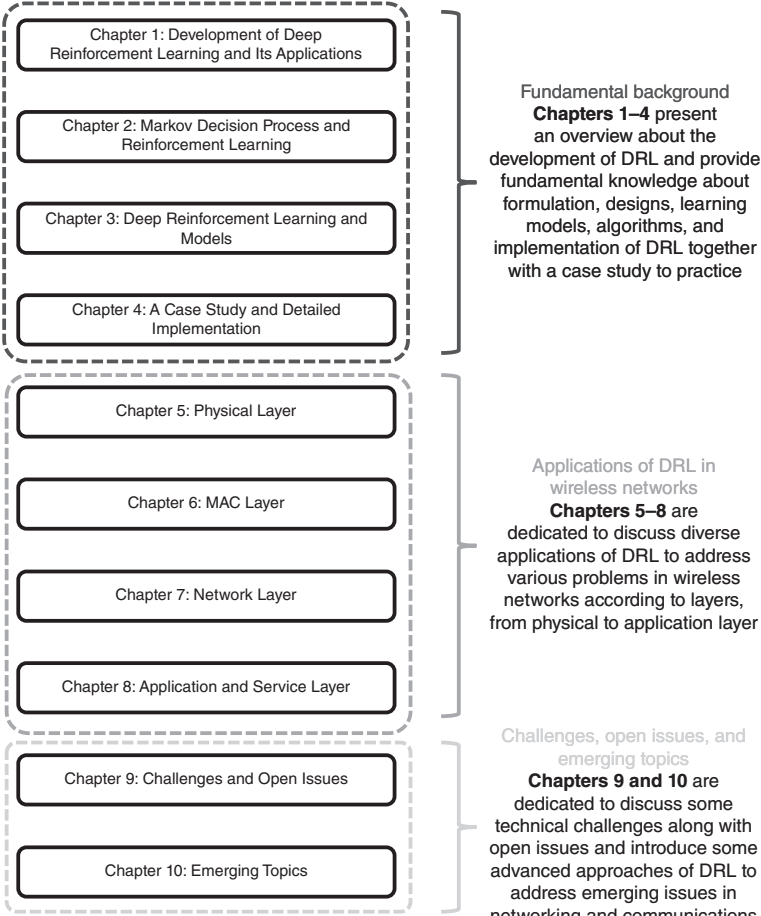


Figure 1.12 Structure of the book.

1.5 Chapter Summary

We have first discussed the development of wireless communications and the emerging challenges. Then, the fundamental of ML and DL as well as their development have been highlighted. After that, we have introduced some basic concepts of RL and DRL with their outstanding features in dealing with the emerging issues of future wireless communication systems. Finally, the structure of this book and target readership have been discussed.

References

- 1 Cisco, “Cisco annual internet report (2018–2023) white paper.” <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>, 2018. (Accessed on 10/01/2022).
- 2 X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, “In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning,” *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019.
- 3 Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, “A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art,” *IEEE Access*, vol. 8, pp. 116974–117017, 2020.
- 4 Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, “Application of machine learning in wireless networks: Key techniques and open issues,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3072–3108, 2019.
- 5 Y. C. Eldar, A. Goldsmith, D. Gündüz, and H. V. Poor (eds.), *Machine learning and wireless communications*, Cambridge University Press, Aug. 2022.
- 6 MATLAB, “Introducing deep learning with matlab.” <https://www.mathworks.com/campaigns/offers/next/deep-learning-ebook.html>. (Accessed on 10/01/2022).
- 7 N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, “Applications of deep reinforcement learning in communications and networking: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.
- 8 I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.
- 9 C. Liu, J. Wang, X. Liu, and Y.-C. Liang, “Deep cm-cnn for spectrum sensing in cognitive radio,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2306–2321, 2019.
- 10 J. Xie, J. Fang, C. Liu, and X. Li, “Deep learning-based spectrum sensing in cognitive radio: A CNN-LSTM approach,” *IEEE Communications Letters*, vol. 24, no. 10, pp. 2196–2200, 2020.
- 11 D. Han, G. C. Sobabe, C. Zhang, X. Bai, Z. Wang, S. Liu, and B. Guo, “Spectrum sensing for cognitive radio based on convolution neural network,” in *2017 10th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI)*, pp. 1–6, IEEE, 2017.
- 12 A. P. Hermawan, R. R. Ginanjar, D.-S. Kim, and J.-M. Lee, “Cnn-based automatic modulation classification for beyond 5g communications,” *IEEE Communications Letters*, vol. 24, no. 5, pp. 1038–1041, 2020.

- 13 S. Zheng, P. Qi, S. Chen, and X. Yang, "Fusion methods for cnn-based automatic modulation classification," *IEEE Access*, vol. 7, pp. 66496–66504, 2019.
- 14 H. Li, Y. Li, S. Zhou, and J. Wang, "Wireless channel feature extraction via gmm and cnn in the tomographic channel model," *Journal of communications and information networks*, vol. 2, no. 1, pp. 41–51, 2017.
- 15 D. Hong, Z. Zhang, and X. Xu, "Automatic modulation classification using recurrent neural networks," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, pp. 695–700, IEEE, 2017.
- 16 B. Bhargava, A. Deshmukh, M. V. Rupa, R. P. Sirigina, S. K. Vankayala, and A. Narasimhadhan, "Deep learning approach for wireless signal and modulation classification," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, pp. 1–6, IEEE, 2021.
- 17 N. Van Huynh, D. N. Nguyen, D. T. Hoang, T. X. Vu, E. Dutkiewicz, and S. Chatzinotas, "Defeating super-reactive jammers with deception strategy: Modeling, signal detection, and performance analysis," *arXiv preprint arXiv:2105.01308*, 2021.
- 18 Ö. Yildirim, "A novel wavelet sequence based on deep bidirectional LSTM network model for ecg signal classification," *Computers in biology and medicine*, vol. 96, pp. 189–202, 2018.
- 19 R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 2018.
- 20 A. Feriani and E. Hossain, "Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial," *IEEE Communications Surveys & Tutorials*, 2021.
- 21 "Alphastar: Mastering the real-time strategy game starcraft ii."
- 22 "Muzero: Mastering go, chess, shogi and atari without rules."
- 23 "Automating pick and pack operations - the benefits."
- 24 I.-B. Park, J. Huh, J. Kim, and J. Park, "A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1420–1431, 2019.
- 25 L. Zhou, L. Zhang, and B. K. Horn, "Deep reinforcement learning-based dynamic scheduling in smart manufacturing," *Procedia Cirp*, vol. 93, pp. 383–388, 2020.
- 26 K. Xia, C. Sacco, M. Kirkpatrick, C. Saidy, L. Nguyen, A. Kircaliali, and R. Harik, "A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence," *Journal of Manufacturing Systems*, vol. 58, pp. 210–230, 2021.
- 27 P. M. Pilarski, M. R. Dawson, T. Degris, F. Fahimi, J. P. Carey, and R. S. Sutton, "Online human training of a myoelectric prosthesis controller

- via actor-critic reinforcement learning,” in *2011 IEEE international conference on rehabilitation robotics*, pp. 1–7, IEEE, 2011.
- 28 C. Yu, J. Liu, S. Nemati, and G. Yin, “Reinforcement learning in healthcare: A survey,” *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–36, 2021.
 - 29 A. Coronato, M. Naeem, G. De Pietro, and G. Paragliola, “Reinforcement learning for intelligent healthcare applications: A survey,” *Artificial Intelligence in Medicine*, vol. 109, p. 101964, 2020.
 - 30 M. Baucum, A. Khojandi, and R. Vasudevan, “Improving deep reinforcement learning with transitional variational autoencoders: A healthcare application,” *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 6, pp. 2273–2280, 2020.
 - 31 “Kr102240442b1 - quality-aware deep reinforcement learning for proactive caching in millimeter-wave vehicular networks and system using the same.”
 - 32 “Cn109474980b - wireless network resource allocation method based on deep reinforcement learning.”
 - 33 “Cn110488861b - unmanned aerial vehicle track optimization method and device based on deep reinforcement learning and unmanned aerial vehicle.”
 - 34 D. Mwit, “10 real-life applications of reinforcement learning,” Nov 2021.
 - 35 X. Di, K. Xiong, P. Fan, H.-C. Yang, and K. B. Letaief, “Optimal resource allocation in wireless powered communication networks with user cooperation,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 7936–7949, 2017.
 - 36 M. Chu, H. Li, X. Liao, and S. Cui, “Reinforcement learning-based multiaccess control and battery prediction with energy harvesting in iot systems,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2009–2020, 2018.
 - 37 J. Zhu, Y. Song, D. Jiang, and H. Song, “A new deep-q-learning-based transmission scheduling mechanism for the cognitive internet of things,” *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2375–2385, 2017.

