International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

# Study of Deep Reinforcement Learning

*Avinash H. Hedaoo*

*Dept. of Computer Science, Prerna College of Commerce, Nagpur -44009, (M.S.) India*

**ABSTRACT**

Deep reinforcement learning is poised to revolutionise the field of AI and represents a step towards building autonomous systems with a higher level understanding of the visual world. The study of generalisation in deep Reinforcement Learning (RL) aims to produce RL algorithms whose policies generalise well to novel unseen situations at deployment time, avoiding overfitting to their training environments. Tackling this is vital if we are to deploy reinforcement learning algorithms in real world scenarios, where the environment will be diverse, dynamic and unpredictable. In this survey, we systematically categorize the deep RL algorithms and applications, and provide a detailed review over existing deep RL algorithms by dividing them into modelbased methods and model-free methods. Finally, we outline the current representative applications

Keywords: Reinforcement learning, Deep reinforcement learning, Reinforcement learning applications Generalisation, Reinforcement Learning Survey Review

## 1. Introduction

Although RL had some successes in the past (Nate Kohl and Peter Stone, 2004, Andrew Y Ng et. A.,2006, Satinder Singh et.al., 2002, Gerald Tesauro, 1995), previous approaches lacked scalability and were inherently limited to fairly low-dimensional problems. These limitations exist because RL algorithms share the same complexity issues as other algorithms: memory complexity, computational complexity, and in the case of machine learning algorithms, sample complexity (Alexander L Strehl et. al., 2006 ) . What we have witnessed in recent years - the rise of deep learning, relying on the powerful function approximation and representation learning properties of deep neural networks - has provided us with new tools to overcoming these problems. The advent of deep learning has had a significant impact on many areas in machine learning, dramatically improving the state-of-the-art in tasks such as object detection, speech recognition, and language translation (Yann LeCun et. al.,2015 ). The most important property of deep learning is that deep neural networks can automatically find compact low-dimensional representations (features) of high-dimensional data (e.g., images, text and audio). Through crafting inductive biases into neural network architectures, particularly that of hierarchical representations, machine learning practitioners have made effective progress in addressing the curse of dimensionality(Yoshua Bengio et. al., 2013). Deep learning has similarly accelerated progress in RL, with the use of deep learning algorithms within RL defining the field of "deep reinforcement learning" (DRL) (Kai Arulkumaran, et. al., 2017 AA).

Why has deep learning been helping reinforcement learning make so many and so enormous achievements? Representation learning with deep learning enables automatic feature engineering and end-to- end learning through gradient descent, so that reliance on domain knowledge is significantly reduced or even removed. Feature engineering used to be done manually and is usually time consuming, over-specified, and incomplete. Deep, distributed representations exploit the hierarchical composition of factors in data to combat the exponential challenges of the curse of dimensionality. Generality, expressiveness and flexibility of deep neural networks make some tasks easier or possible, e.g., in the breakthroughs and novel architectures and applications. Deep learning, as a specific class of machine learning, is not without limitations, e.g., as a black-box lacking interpretability, as an "alchemy" without clear and sufficient scientific principles to work with, and without human intelligence not able to competing with a baby in some tasks. However, there are lots of works to improve deep learning, machine learning, and AI in general. Deep learning and reinforcement learning, being selected as one of the MIT Technology Review 10 Breakthrough Technologies in 2013 and 2017 respectively, will play their crucial role in achieving artificial general intelligence. David Silver, the major contributor of AlphaGo (Silver et al., 2016a; 2017), even made a formula: artificial intelligence = reinforcement learning + deep learning (Silver, 2016) ( Yuxi Li , 2018 ).

The aim of this study is to outline and critically review all significant research done to date in the context of combining reinforcement learning algorithms and deep learning methods. The research will review both supervised and unsupervised deep models that have been combined with RL methods for environments which might be partially observable MDPs or not. This study will also present recent outstanding success stories of the combined RL and deep learning paradigms, which led to the introduction of a novel research route called deep reinforcement learning, to overcome the challenges in learning control policies from high-dimensional raw input data in complex RL environment (Seyed Sajad Mousavi et. al., 2018 ).

I endeavour to provide as much relevant information as possible. For reinforcement learning experts, as well as new comers, I hope this overview would be helpful as a reference. In this overview, I mainly focus on contemporary work in recent couple of years, by no means complete . In this version, I endeavour to provide a wide coverage of fundamental and contemporary RL issues, about core elements, important mechanisms, and applications. Rest of the paper composition is as follows. Section II defines the background of RL . Section III describes the reinforcement learning algorithms, section IV illustrates applications and section V concludes the paper.

## 2. Background

### *2.1 Reinforcement Learning*

Reinforcement learning (RL) is one of the machine learning areas in which an agent has to interact with its environment in order to achieve a goal. RL based on the structure of Markov Decision Processes (MDPs); a reliable structure for the agent learning while interacting with its environment in order to receive rewards and drawbacks. The essential elements of RL are the states, actions and reinforcements (Maia, T. V. 2009). Via the agent's sensors, the agent recognizes the environment and implements actions (according to a policy) in which leads to changes in the environment. According to these changes, the agent obtains rewards based on the taken actions. RL improves strategy through the learning via trial and error by interacting with the environment and recognize the best actions at each state in order to reach the goal and gain the best rewards (Ming, G. F., et. al. 2010, Gil, P., 2013). RL tries to find the best policy that increases the total reward. (Barto, A. G, 2003) indicated that RL algorithms work on how the agent can learn to estimate an optimal strategy while to interact with its environment (Mostafa Al-Emran, 2015).

### *2.1.1. Reinforcement in Learning Classifier System*

To better understand how reinforcement learning is applied in the artificial intelligent system, we have to know how a learning classifier system works. A learning classifiers system (LCS) works by interacting with the real world from which it attains feedback in the form of mostly numerical reward (R). Learning is driven by trying to maximize the amount of reward received. Usually, the LCS consists of four components : a finite population of condition-action rules, called classifiers, that represents the current knowledge of the system; the performance component, which governs the interaction with the environment; the reinforcement component (also called credit assignment component), which distributes the reward received from the environment to the classifiers accountable for the rewards obtained; the discovery component, which is responsible for discovering better rules and improving existing ones through a genetic algorithm. Therefore, reinforcement learning is essential for capturing the diachronic behaviours of an intelligent system (Amit Kumar Mondal,2021). Conventionally used RL algorithms are Markov decision process, Q learning, Temporal difference and Monte Carlo.

### *2.1.2. Challenges with reinforcement learning*

Safety is an important parameter while considering system operations during the learning phase. In RL, due to the limited availability of data in the real world, algorithms are trained with a limited number of patterns during the learning phase. RL algorithms have many practical real-world problems with large and continuous state and action spaces. In many cases of RL direct training is not possible. In this case, an off-policy and off-line training system is used where training is done by the recent iterations of the algorithms (Pamina, J. et. al.,2019) (Surjeet Balhara et. al., 2022).

### *2.2 Hierarchical Reinforcement Learning*

A larger goal is sub-divided into a hierarchy of sub-goals. Each subtask could still be decomposed to sub- tasks, and the lowest level of tasks, typically, are primitive actions (Jose JFR et. al. , 2011). This approach shifts the focus of RL problem from being a state-to-state or action-to-action oriented towards subgoals to larger-goal oriented. HRL can be summarized as an approach that abstracts and divides the state space into key landmarks, from start to the final goal. Thus tackling large dimensions of state-space could be easier. This algorithm can be drawn as parallel to Work Breakdown Structure, which is followed in many of the software project management processes and elsewhere(Chapman JR, 2004) (N. R. Ravishankar et. Al., 2017).

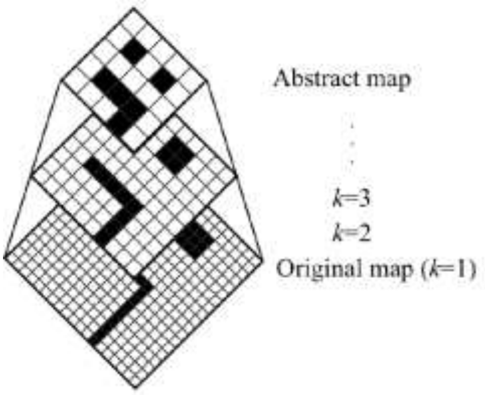

Figure 1. Multi-scale Value Functions. Courtesy19.

### *2.3 Q-Learning*

Q-learning is one of the RL algorithms that has been successfully used in many domains such as: face recognition, simple toys, web-based education and many others (Rodrigues Gomes, E., 2009). Q-learning tries to find an optimal action policy by estimating the optimal state-action function Q(s, a) where s -> state from the set of the possible states S, a -> action from the set of the possible actions A. The Q function described the maximum reward achieved when an action a is executed over the states. The Q-learning equation is described as follows:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r+\gamma \max_{a'} Q(s',a'))$$

Where α refers to the learning rate, γ refers to the discount factor and r refers to the reward of executing the action a over the states (Mostafa Al-Emran, 2015).

## 3. Reinforcement Learning Algorithms

In the following, I will distinguish between different classes of RL algorithms. Reinforcement learning algorithms are categorized as model-free and model-based reinforcement learning algorithms. “Fig. 2,” illustrates the difference between model free and model based algorithm (Eisha Akanksha, 2021).

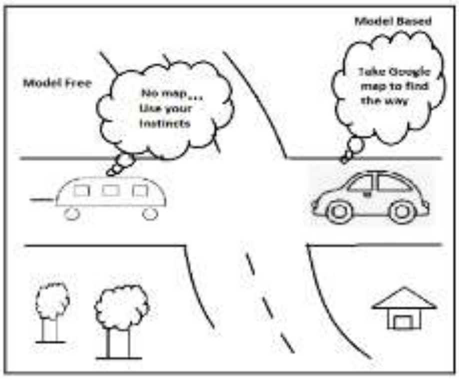

Fig.2. Illustration of model free and model based reinforcement learning algorithms

### *3.1 Model-free RL*

Model-free RLs do not have any knowledge about the environment in which the agent acts(Quentin JM, 2014).The agent only acts in any given state, but doesn’t know where it leads to. So they rely on the instantaneous reward obtained by taking an action in a state, and then evaluate the utility of that state. Utility is defined as the expected total reward from the current state to the goal-state. Once all the paths are traversed they then evaluate the utilities of each of the states experienced. As always, the action sequence that yields the maximum utility is considered as the optimal policy. So they learn the utilities by trial-and error method. Some of the most popular, legacy model-free RL algorithms are Q-learning, SARSA, Dyna-Q, and Temporal Difference (TD) (N. R. Ravishankar et. Al., 2017).

Model-free RL further divided into two scenarios: 3.1.1 RL based on the value function 3.1.2 RL based on policy gradient.

### *3.1.1 RL based on the value function*

### *3.1.1.1 Deep Q-Learning [DQN]*

Researchers in DeepMind technologies have developed an approach called Deep Q learning Network (DQN)( Mnih, V. et. al., 2013) , DQN combine a deep convolutional neural network with the simplest reinforcement learning method (Q-learning) to play several Atari 2600 computer games only by watching the screen. For the correlated states issue DQN provides approach named experience replay. In the process of learning, DQN store agent’s experience (st,at,rt,rt+1)at each time step into a date set D, where st, at and rt, respectively the state, selected action and received reward at time step t and st+1 is state at the next time step. For updating Q-values, it uses stochastic minibatch updates with uniformly random sampling from experience replay memory (previous transitions) at training time. This work break strong correlations between consecutive samples, And for instability in the policy, the network is trained with a target Q-network to obtain consistent Q-learning targets by fixing weight parameters used in Q-learning target and updating them periodically. In some games its strategy outperformed the human player and achieved state of the art performance on many Atari games with the same network architecture or hyperparameters. However, using deep neural networks need sufficient data to be fed into network to learn better representations and as a result getting good performance. Hence, applying this approach in real environment such as robotics is very challenging and difficult since performing a large number of episodes to collect samples is source consuming and even not possible (Seyed Sajad Mousavi et. al., 2018).

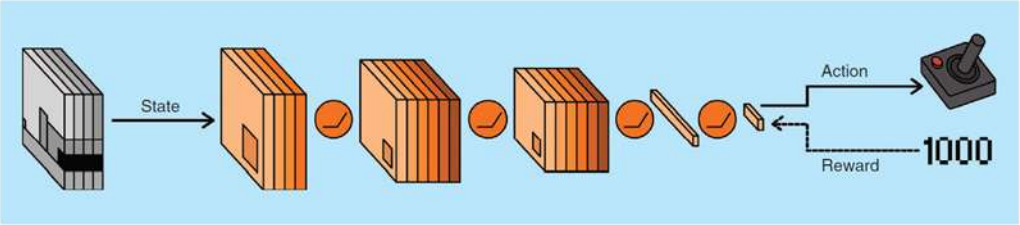

**Fig. 3**. The DQN. The network takes the state—a stack of gray-scale frames from the video game—and processes it with convolutional and fully connected layers, with ReLU nonlinearities in between each layer. At the final layer, the network outputs a discrete action, which corresponds to one of the possible control inputs for the game. Given the current state and chosen action, the game returns a new score. The DQN uses the reward—the difference between the new score and the previous one—to learn from its decision. More precisely, the reward is used to update its estimate of Q, and the error between its previous estimate and its new estimate is backpropagated through the network ( Kai Arulkumaran, et. al., 2017).

### *3.1.1.2 Temporal difference*

TD learning (Sutton, 1988) learns value function V (s) directly from experience with TD error, with bootstrapping, in a model-free, online, and fully incremental way. TD learning is a prediction problem. The update rule is V (s) <- V (s) + α[r + γ V (s') - V(s)], where α is a learning rate, and r + γ V (s') - V(s) is called TD error( Yuxi Li , 2018 ).

TD learning is a model-free RL algorithm which is the combination of both Monte Carlo (MC) algorithm and dynamic programming technology that is used for solving forecast problems in RL, which are in time series. In the TD algorithm, the learning process uses the current action and immediate state to estimate the current state. It aims to maximize the reward by adjusting the strategy continuously while interacting with the environment (Surjeet Balhara et. al., 2022).

### *3.1.1.3 Q-learning*

"Q" stands for quality. The algorithm represents how fruitful a given action would be in gaining the best future reward. The Q table is a reference matrix that makes the agent to locate the best activity for each state. It assists with augmenting the expected rewards by choosing the most ideal of all actions (state, activity) restores the normal potential compensation of that activity at that state. The Q (s, a) is iteratively refreshed utilizing the Bellman equation (Eisha Akanksha, 2021).

Q-learning is a typical type of off_policy learning that updates a target policy π using samples generated by any stochastic behaviour policy in an environment. Following the Bellman equation and temporal difference (TD) for the action-value function, the Q-learning algorithm is recursively updated using the following equation:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a' \in A} Q(s_{t+1}, a') - Q(s_t, a_t)].$$

where a' follows the target policy a' ~ π ( . | st) and α is the learning rate. While updating Q-learning, the next actions at+1 are sampled from the behaviour policy which follows an ϵ-greedy exploration strategy, and among them, the action that makes the largest Q-value, a' is selected. (Pawel Ladosz et. al., 2022)

### *3.1.1.4 State-Action-Reward-State-Action (SARSA)*

The algorithm learns the Q- value dependent on the activity performed by the current arrangement rather than the greedy approach(Eisha Akanksha, 2021).The name Sarsa comes from ($s_t$, $a_t$, $r_{t+1}$, $s_{t+1}$, $a_{t+1}$) which is in essence the description of the backup diagram of Sarsa. Sarsa uses the current action-value, the reward and the action-value belonging to the next state and action to backup the current action-value. An exploring policy has to be followed e.g. ϵ-greedy. A combination of the Bellman-equation and the incremental average is used to derive the following equation (Victor Dolk,2010).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right]$$

### *3.1.2 RL based on policy gradient*

REINFORCE (Williams, 1992) is the prototype of policy gradient (PG) algorithms. Compared with value-based RL, policy-based RL not only avoids the policy degradation caused by the value function error, but also is easier to apply in the continuous action space problem. Specifically, value based methods, such as Q-learning and SARSA, require a one-step operation to calculate the maximum value, which can hardly be found in the continuous space or high-dimensional space. In addition, value-based methods learn implicit policies but policy-based RL methods can learn stochastic policies. That is, in the value-based method, the policies obtained through policy improvement are all deterministic policies, and will encounter some problems that cannot be resolved in some tasks like Rock-Paper-Scissors. Policy-based methods also have some common shortcomings: (1) data efficiency or sample utilization is low; (2) the variance is large, which makes it difficult to converge (Hao-nan Wang et. al.,2020 ).

#### *3.1.2.1 Policy gradient*

There are two sorts of policies: deterministic and stochastic. Deterministic strategy that maps state to activity with no ambiguity. The stochastic approach yields a probability distribution over activities in a given state. It is called the Partially Observable Markov Decision Process. The software programs which are considered as learner and decision-maker are called as an agent. The agent interacts with the environment, the environment provides rewards in return and the probability distribution of the future state based on the actions of the agent. The reward can be positive or negative based on the actions. Hence the algorithm consists of the policy $\pi$ having parameter $\theta$ where the $\pi$ yields a probability distribution of activities(Zhang, J. et. al.,2021) (Eisha Akanksha, 2021).

#### *3.1.2.2 Proximal Policy Optimization (PPO)*

Proximal policy optimization (PPO) algorithm performs unconstrained optimization, requiring only first-order gradient information (Pieter Abbeel et. al. ,2016). The two main variants include an adaptive penalty on the KL divergence, and a heuristic clipped objective which is independent of the KL divergence. Being less expensive whilst retaining the performance of TRPO means that PPO (with or without GAE) is gaining popularity for a range of RL tasks ( Kai Arulkumaran, et. al., 2017).

#### *3.1.2.3 Trust Region Policy Optimization (TRPO)*

Trust region policy optimization (TRPO), has been shown to be relatively robust and applicable to domains with high-dimensional inputs . To achieve this, TRPO optimizes a surrogate objective function—specifically, it optimizes an (importance sampled) advantage estimate, constrained using a quadratic approximation of the KL divergence. The constrained optimization of TRPO requires calculating second order gradients, limiting its applicability (Kai Arulkumaran, et. al., 2017).

#### *3.1.2.4 Asynchronous Advantage Actor-Critic (A3C)*

Asynchronous Advantage Actor-Critic Algorithm is policy-based RL algorithms. Policy-based algorithms output policies rather than the q values and each policy distribution has different exploration estimations. Policy-based methods can handle continuous action spaces easily as it represents parameters of the distribution as output which is finite. In training a policy-based algorithm, instead of minimizing error and finding optimal policy, the concept of gradient is used (Deepanshu Mehta, 2019).

### *3.2 Model-based RL*

The algorithm makes a simulated model for every condition. The agent learns from the environment by taking actions and observing the outcomes. The various types of model-based reinforcement learning algorithms are:

#### *3.2.1 Imagination-Augmented Agents (I2A)*

A hybrid approach, which combines model-based elements with a model-free algorithm, is Imagination-Augmented Agents (I2A). This particular method incorporates an imagination core module used for producing possible future trajectories (i.e. rollouts) from past experience and through action-conditional next-step predictors. Imagined rollouts are then encoded using LSTM encoders . At the same time, a model-free agent is trained naturally at each time step, only to feed both their output and the concatenated model-based encoded rollouts into a policy network which undertakes the task of producing the final action to be executed. I2A solved 85% of the Sokoban puzzles (Aristotelis Lazaridis et al., 2020).

#### *3.2.2 Imagination-Augmented Agents (I2A)*

The model-based value expansion (MBVE) algorithm learns a policy $\pi$ and a critic Q, to solve some well known control tasks. The way the critic is updated is by using rollouts done with the model up to a fixed horizon H. The horizon length represents a measure of trust in the model, and controlling it helps controlling the uncertainty of the model without the need of more complex uncertainty estimation techniques (Constantin-Valentin Pal, 2020).

#### *3.2.3 Model-based policy optimization (MBPO)*

Model-based policy optimization (MBPO) uses a probabilistic model ensemble and performs a large amount of short model rollouts that start from a state distribution with states from the real environment dynamics. This state distribution contains states collected in a buffer D constructed with the previous policy in the iterative algorithm, and the rollouts are used to perform policy optimization using a model-free optimizer ((Constantin-Valentin Pal, 2020).

There is a performance gap between puremodel- based and model-free methods. Compared with model-free methods that require 10 days, model-based methods enable a complete training process using only 10 min in real time. However, model-free methods can achieve much better performance, differing by at most three orders of magnitude (Nagabandi et al., 2018) (Hao-nan Wang et. al.,2020 ). The “Fig. 4,” gives an overview of reinforcement learning agent taxonomy (Eisha Akanksha, 2021).

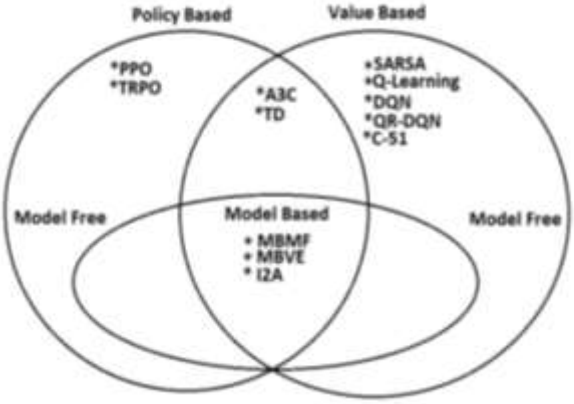

Fig.4. Illustration of RL agent taxonomy

## 4. Applications

Deep RL has achieved significant success in various fields. I outline current representative RL applications including robotics, natural language processing (NLP), and computer systems etc. in this section as follows :

### *4.1 Energy*

Our mankind is facing issues of sustainable and green energy. It is critical to consume energy efficiently. I discuss data center cooling and smart grid in the following.

### *4.1.1 Data Center Cooling*

RL is one approach to data center cooling. I discuss how to control fan speeds and water flow in air handling units (AHUs) to regulate the airflow and temperature inside server floors, using model-predictive control (MPC). In MPC, the controller (agent) learns a linear model of the data center dynamics with random, safe exploration, with little or no prior knowledge. It then optimizes the cost (reward) of a trajectory based on the predicted model, and generates actions at each step, to mitigate the effect of model error and unexpected disturbances, at the cost of extra computation. The controls or actions are variables to manipulate, including fan speed to control air flow and valve opening to regulate the amount of water. States refer to the process variables to predict and regulate, including differential air pressure (DP), cold-aisle temperature (CAT), entering air temperature(EAT) to each AHU, leaving air temperature (LAT) from each AHU.

The method is compared with a local proportional integral derivative (PID) controller and a certainty-equivalent controller, which updates parameters of the dynamics model assuming the estimated model were accurate. Experiments show that the method can achieve data center cooling in a large-scale commercial system in a safe, effective, and cost-efficient way (Yuxi Li, 2019).

### *4.1.2 Smart Grid*

A smart grid is a power grid utilizing modern information technologies to create an intelligent electricity delivery network for electricity generation, transmission, distribution, consumption, and control (Fang et al., 2012). An important aspect is adaptive control (Anderson et al., 2011). Application of RL for electric power system decision and control is reviewed. Here demand response discussed briefly. Demand response systems motivate users to dynamically adapt electrical demands in response to changes in grid signals, like electricity price, temperature, and weather, etc. With suitable electricity prices, load of peak consumption may be rescheduled/lessened, to improve efficiency, reduce costs, and reduce risks. Design of a fully automated energy management system with model-free reinforcement learning is proposed , so that it doesn’t need to specify a disutility function to model users’ dissatisfaction with job rescheduling. The authors decomposed the RL formulation over devices, so that the computational complexity grows linearly with the number of devices, and conducted simulations using Q-learning. The demand response problem is tackled with batch RL. The exogenous prices is taken as states, and the average as feature extractor to construct states is utilized. (Yuxi Li , 2018)

### *4.2 Finance*

RL is a natural solution to some finance and economics problems (Hull, 2014), like option pricing, and multi-period portfolio optimization, where value function based RL methods were used. To utilize policy search to learn to trade is proposed; it is extended with deep neural networks. Deep (reinforcement) learning would provide better solutions in some issues in risk management. The market efficiency hypothesis is fundamental in finance. However, there are well-known behavioural biases in human decision-making under uncertainty, in particular, prospect theory. A reconciliation is the adaptive markets hypothesis, which may be approached by reinforcement learning. It is nontrivial for finance and economics academia to accept blackbox methods like neural networks. However, there is a lecture in AFA 2017 annual meeting: Machine Learning and Prediction in Economics and Finance. We may also be aware that financial firms would probably hold state-of-the-art research/application results. FinTech has been attracting attention, especially after the notion of big data. FinTech employs machine learning techniques to deal with issues like fraud detection, consumer credit risk, etc. ( Yuxi Li , 2018).

### *4.3 Healthcare*

AI technologies are realistically altering and empowering the healthcare system. At present RL and DL have been extensively used to determine and discover innovative healthcare applications and services namely, medical imaging. DRL also focuses on lung cancer as much of the global population is suffering from lung tumours; and on providing solutions to computer-aided diagnosis. Value-based DL models including DQN and hierarchical DRL models are used in the treatment of lung cancer and its diagnosis (Surjeet Balhara et. al., 2022).

Dynamic treatment regimes (DTRs) or adaptive treatment strategies are sequential decision making problems. I discuss an approach to optimal treatment strategies for sepsis in intensive care. A state is constructed from the multidimensional discrete time series composed of 48 variables. An action, or a medical treatment, is defined by the total volume of intravenous fluids and maximum dose of vasopressors over each 4 hour period. A reward and a penalty is associated with survival and death, respectively, to optimize patient mortality. Experiments show that the policy learned by RL has larger value, or lower mortality, than those from human clinicians, and that the patients have the lowest mortality when they receive treatments similar to those recommended by the policy learned by RL (Yuxi Li, 2019).

Q-learning is the RL method in DTRs(Dynamic treatment regimes). Deep RL is applied to the problem of inferring patient phenotypes (Yuxi Li, 2018).

Medical image report is generated by following a hybrid retrieval-generation approach trained by RL, to integrate human prior knowledge and neural networks. A convolutional neutral network (CNN) extract visual features of a set of images of a sample, and an image encoder transforms the visual features into a context vector, then a sentence decoder generates latent topics recurrently (Yuxi Li, 2019).

### *4.4 Robotics*

Robotics is a classic area for RL. RL can implement behavioral control of complex robots in the simulation environment, thus enabling realistic responses to perturbations and environmental variation. Apart from Atari and simple agents in Mujoco (e.g., half-cheetah, ant, and spider), DeepMimic (Peng et al., 2018a) further develops challenging multiskilled agents including multiple characters (e.g., human, Atlas robot, bipedal dinosaur, and dragon) and a large variety of skills (e.g., locomotion, acrobatics, and martial arts). In addition, RL has obtained numerous research results in robot control tasks in real situations. A range of real-world tasks are contact-rich and require close coordination between vision and control, such as stacking tight-fitting Lego blocks and screwing bottle caps onto a bottle. The improvements (Levine et al., 2015) control manipulation to complete these tasks by reducing the sample count and automating parameter selection in GPS. An RNN with LSTM (Rahmatizadeh et al., 2016) helps the controller learn from virtual demonstrations and successfully performs the manipulation tasks on a physical robot By closed-loop vision-based control (Kalashnikov et al., 2018), re-grasping strategies are automatically learned, probing an object and repositioning objects to find the most effective grasps and perform other non-prehensile pre-grasp manipulations. However, training data of real robots is scarce for real scenarios. The method which combines knowledge from previous tasks with online adaptation of the dynamics model (Fu et al., 2016) helps solve a variety of complex robotic manipulation tasks in a single attempt. Multiple robots (Gu SX et al., 2017a; Yahya et al., 2017) learn collaboratively to sample and train in parallel. Manipulating the source domain (Peng et al., 2018b) narrows the gap between simulation and real physical systems. Al- Nima et al. (2019) produced suitable road tracking actions based on RL by collecting input states from forward car facing views. Based on transfer learning (Devin et al., 2017), robots can share task-specific modules across robots and robot-specific modules across all tasks. The improved methods of meta-RL (Finn et al., 2017b; Yu TH et al., 2018) enable the agents to learn rapidly from little data in new environments. Learning to adapt to dynamic real-world environments (Nagabandi et al., 2019) further alleviates the problem of missing training data and has better generalization ability in dealing with robotic manipulation tasks [WW].

DRL is applied in robotics for navigation of the mobile robot in an unfamiliar environment by avoiding the obstacles to reach the desired destination autonomously with an RGB-D camera by using a DDQN algorithm. Navigation of the mobile robot to the desired destination without using any maps is done by asynchronous deterministic policy gradients with light detection and ranging (LIDAR) and the commands will be provided to the mobile robot for avoiding obstacles by estimating the Q value from the DQN to know the depth of the image by using RGBD sensor. DRL is also used in solving flocking control problems in multi robotic systems in complex environments using an algorithm called multi-agent DDPG, which helps the multi-robot system in performing a flocking task with greater convergence speed (Surjeet Balhara et. al., 2022).

Following points are discussed in (Yuxi Li, 2018) : See Kober et al. (2013) for a survey of RL in robotics, Deisenroth et al. (2013) for a survey on policy search for robotics, and Argall et al. (2009) for a survey of robot learning from demonstration. See the journal Science Robotics. It is interesting to note that from NIPS 2016 invited talk, Boston Dynamics robots did not use machine learning. In the following, I discuss guided policy search (Levine et al., 2016a) and learn to navigate (Mirowski et al., 2017). See more recent robotics papers, e.g., Chebotar et al. (2016; 2017); Duan et al. (2017); Finn and Levine (2016); Gu et al. (2016a); Lee et al. (2017); Levine et al. (2016b); Mahler et al. (2017); P´erez-D'Arpino and Shah (2017); Popov et al. (2017); Yahya et al. (2016); Zhu et al. (2017b). I recommend Pieter Abbeel's NIPS 2017 Keynote Speech, Deep Learning for Robotics, slides at, https://www.dropbox.com /s/fdw7q8mx3x4wr0c/.

### 4.5 Transportation

Today traffic congestion is a serious issue in many metropolitan cities. DQN is used to optimize the real-time traffic control policies. To solve many real-time problems and to provide better navigation when compared with some traditional routing algorithms, DRL-based real-time navigation and vehicle routing method is proposed by using simulation of urban mobility (SUMO) for training DNN to reroute vehicles to the destination in the real-time complex environment (Surjeet Balhara et. al., 2022).

Hierarchical RL is an approach to knowledge representation with temporal abstraction at multiple levels, and to learn and plan. Transfer learning techniques can help adapt a learned policy to multiple cities. The learned policy show promising performance results w.r.t. metrics about total driver income and user experience on the platform of Didi Chuxing.

I next discuss the problem formulation for ride-sharing order dispatching. A high-fidelity simulator is helpful for RL, e.g., generating synthetic data to evaluate or optimize a policy. At the beginning of the simulation, drivers' status and order information are initialized with historical real data. After that, the simulator determines drivers' status, following an order-driver matching algorithm, with the help of an order dispatching policy learned with RL. A busy driver will fulfill an order. An idle driver follows a random walk, according to a driver movement model, and be in an online/offline mode, according to an online/offline model. These two models are learned from the historical real data (Yuxi Li, 2019).

### 4.6 Games

DRL has significant applications in games. After applying DRL to games like cart pole and mountain car, the masters have tested the gaming strategy and they found that it repeatedly won the game. In this case, the designers faced the challenges that came across during the gaming process and analysed every possible scenario of gaming by learning from wins, losses and draw over sometimes. The designer feeds the neural network with thousands of rules and scenarios. The game itself starts with random play. After each play, the system analyses the result and sets the parameters of the neural network to become the strongest player. With the help of deep neurons, it makes its move very dynamically by increasing its gaming power (Surjeet Balhara et. al., 2022). Games provide excellent testbeds for RL/AI algorithms. In the following sections I discuss achievements and applications of RL in different categories of games.

#### 4.6.1. Perfect Information Board Games

Board games like Backgammon, Go, chess, checker and Othello, are classical testbeds for RL/AI algorithms. In such games, players reveal perfect information. AlphaGo (Silver et al., 2016a and Sutton and Barto (2018), a computer Go program, won the human European Go champion, 5 games to 0, in October 2015, in March 2016, AlphaGo defeated Lee Sedol, an 18-time world champion Go player, defeated Ke Jie 3:0 in May 2017. AlphaGo Zero (Silver et al., 2017) further improved previous versions by learning a superhuman computer Go program without human knowledge. AlphaGo was built with techniques of deep convolutional neural networks, supervised learning, reinforcement learning, and Monte Carlo tree search (MCTS) (Browne et al., 2012; Gelly and Silver,

2007; Gelly et al., 2012).

#### 4.6.2 Imperfect Information Board Games

Imperfect information games, or game theory in general, have many applications, e.g., security and medical decision support. Heinrich and Silver (2016) proposed Neural Fictitious Self-Play (NFSP). NFSP was evaluated on two-player zero-sum games. In Leduc poker, NFSP approached a Nash equilibrium, while common RL methods diverged. In Limit Texas Hold'em, a real-world scale imperfect-information game, NFSP performed similarly to state-of-the-art, superhuman algorithms which are based on significant domain expertise. Heads-up Limit Hold'em Poker was essentially solved (Bowling et al., 2015) with counterfactual regret minimization (CFR).

#### 4.6.3 Video Games

Video games would be great testbeds for artificial general intelligence. A3C won the champion in Track 1 of ViZDoom Competition by a large margin. The problem of sensorimotor control in immersive environments is approached with supervised learning, and won the Full Deathmatch track of the Visual Doom AI Competition. It is also discussed how to tackle Doom. Multiagent actor-critic framework is proposed , used StarCraft as the testbed. Without human demonstration or labelled data as supervision, the proposed approach learned strategies for coordination similar to the level of experienced human players, like move without collision, hit and run, cover attack, and focus fire without overkill. Usunier et al. (2017); Justesen and Risi (2017) also studied StarCraft. Oh et al. (2016) and Tessler et al. (2017) studied Minecraft, Chen and Yi (2017); Firoiu et al. (2017) studied Super Smash Bros, and Kansky et al. (2017) proposed Schema Networks and empirically studied variants of Breakout in Atari games (Yuxi Li, 2018).

### *4.7. Natural language processing*

RL methods have broad application prospects in the domain of NLP and have been successfully applied in the fields of neural machine translation (NMT), dialog systems, and speech generation. Benefitting from the development of transfer learning and deep meta-RL, universal NMT (GuJT et al., 2018a) uses a transfer-learning approach to share lexical and sentence representations across multiple source languages into one target language. This enables the low-resource language to use the lexical and sentence representations of the higher resource languages. Further, Gu JT et al. (2018b) first extended a deep meta-RL algorithm (e.g., MAML) into low-resource NMT. The model can learn to adapt to low-resource languages based on multilingual high-resource language tasks. The task of chatbots in dialogue systems is to mimic human-human interactions with extended conversations. A modified version of the episodic REINFORCE algorithm explores and learns the policy and the posterior probability over the knowledge base entries for correct retrievals to select dialogue acts An adaptive TTS approach is presented based on MAML to highly restore the speaker's voice in new scenes using very few speech samples. Similarly, the DeepVoice model is improved by predicting the embedding with an encoding network and fitting the embedding based on a small amount of adaptation data (Hao-nan Wang et. al.,2020 ).

### *4.8. Computer Vision*

Computer vision is about how computers gain understanding from digital images or videos. In the following, after presenting background in computer vision, we discuss recognition, motion analysis, scene understanding, integration with NLP, and visual control. Reinforcement learning would be an important ingredient for interactive perception (Bohg et al., 2017), where perception and interaction with the environment would be helpful to each other, in tasks like object segmentation, articulation model estimation, etc.

#### *4.8.1 Recognition*

RL can improve efficiency for image classification by focusing only on salient parts. For visual object localization and detection, RL can improve efficiency over approaches with exhaustive spatial hypothesis search and sliding windows, and strikes a balance between sampling more regions for better accuracy and stopping the search when sufficient confidence is obtained about the target's location. Mnih et al. (2014) introduced the recurrent attention model (RAM) to focus on selected sequence of regions or locations from an image or video for image classification and object detection. Caicedo and Lazebnik (2015) proposed an active detection model for object localization with DQN. Jie et al. (2016) proposed a tree-structure RL approach to search for objects sequentially. Mathe et al. (2016) proposed to use policy search for visual object detection. Kong et al. (2017) deployed collaborative multi-agent RL with inter-agent communication for joint object search. Welleck et al. (2017) proposed a hierarchical visual architecture with an attention mechanism for multi-label image classification. Rao et al. (2017) proposed an attention-aware deep RL method for video face recognition. Krull et al. (2017) for 6D object pose estimation.

#### *4.8.2. Motion Analysis*

In tracking, an agent needs to follow a moving object. Supancic and Ramanan (2017) proposed online decision-making process for tracking, formulated it as a partially observable decision-making process (POMDP). Yun et al. (2017) also studied visual tracking with deep RL. Rhinehart and Kitani (2017) proposed Discovering Agent Rewards for K-futures Online (DARKO).

#### *4.8.3. Scene Understanding*

Wu et al. (2017b) studied the problem of scene understanding, and attempted to obtain a compact, expressive, and interpretable representation to encode scene information like objects, their categories, poses, positions, etc, in a semi-supervised way. The authors deployed a variant of REINFORCE algorithm to overcome the non-differentiability issue of graphics rendering engines. Wu et al. (2017a) proposed a paradigm with three major components, a convolutional perception module, a physics engine, and a graphics engine, to understand physical scenes without human annotations. There are recent works about physics learning, e.g., Agrawal et al. (2016); Battaglia et al. (2016); Denil et al. (2017); Watters et al. (2017); Wu et al. (2015).

#### *4.8.4. Integration With NLP*

Some are integrating computer vision with natural language processing. Xu et al. (2015) integrated attention to image captioning, trained the hard version attention with REINFORCE, and showed the effectiveness of attention on Flickr8k, Flickr30k, and MS COCO datasets. Rennie et al. (2017) introduced self-critical sequence training, using the output of test-time inference algorithm as the baseline in REINFORCE to normalize the rewards it experiences, for image captioning. Strub et al. (2017) proposed end-to-end optimization with deep RL for goal-driven and visually grounded dialogue systems for GuessWhat?! game. Das et al. (2017) proposed to learn cooperative Visual Dialog agents with deep RL. See also Kottur et al. (2017). See Pasunuru and Bansal (2017) for video captioning. See Liang et al. (2017d) for visual relationship and attribute detection.

#### *4.8.5. Visual Control*

Visual control is about deriving a policy from visual inputs, e.g., in games (Mnih et al., 2015; Silver et al., 2016a; 2017; Oh et al., 2015; Wu and Tian, 2017; Dosovitskiy and Koltun, 2017; Lample and Chaplot, 2017; Jaderberg et al., 2017), robotics (Finn and Levine, 2016; Gupta et al., 2017b; Lee et al., 2017; Levine et al., 2016a; Mirowski et al., 2017; Zhu et al., 2017b), and self-driving vehicles (Bojarski et al., 2016; Bojarski et al., 2017; Zhou and Tuzel, 2017).

#### *4.8.6. Business Management*

Reinforcement learning has many applications in business management, like ads, recommendation, customer management, and marketing. Li et al. (2010) formulated personalized news articles recommendation. Theocharous et al. (2015) formulated a personalized ads recommendation systems as a RL problem to maximize life-time value (LTV) with theoretical guarantees. Li et al. (2015) also attempted to maximize lifetime value of customers. Silver et al. (2013) proposed concurrent reinforcement learning for the customer interaction problem.

#### *4.8.7. Industry*

The era of Industry 4.0 is approaching, e.g., see O'Donovan et al. (2015), and Preuveneers and Ilie-Zudor (2017). Reinforcement learning in particular, artificial intelligence in general, will be critical enabling techniques for many aspects of Industry 4.0, e.g., predictive maintenance, realtime diagnostics, and management of manufacturing activities and processes. Robots will prevail in Industry 4.0.

### *4.9. Computer systems*

Computer systems present many challenging problems for RL, including time-varying state or action, structured data sources , and highly stochastic environments. Here, I summarize some typical RL methods used in computer systems and show that RL could provide significant real-world benefits in this domain. Tackling multi-resource cluster scheduling with a PG algorithm optimizes various objectives like average job slowdown or completion time in an online manner with dynamic job arrivals, and validates the approach via simulation. Chen L et. al. (2018) proposed a two-level system called "automatic traffic optimization (AuTO)" to solve the scalability problem in data center traffic. Motivated by prior applications, a scalable RL model with the graph embedding technique is trained by the PG algorithm to deal with the issue of continuous stochastic job arrivals. Further, inspired by much potential for RL to improve the performance, an open extensible platform Park defines the MDP formulation (e.g., state, action space, and reward function). Park connects to a suite of real-world computer systems and lowers the barrier of entry for machine learning researchers to innovate based on deep RL in computer systems (Hao-nan Wang et. al.,2020).

### *4.10. Resource Allocation*

Mao et al. (2016) studied resource management in systems and networking with deep RL. The authors proposed to tackle multi-resource cluster scheduling with policy gradient, in an online manner with dynamic job arrivals, optimizing various objectives like average job slowdown or completion time. The authors validated their proposed approach with simulation. Liu et al. (2017) proposed a hierarchical framework to tackle resource allocation and power management in cloud computing with deep RL. Google deployed machine learning for data centre power management, reducing energy consumption by 40%.

#### *4.10.1 Performance Optimization*

To optimize device placement for Tensorflow computational graphs with RL is proposed . The authors deployed a seuqence-to-sequence model to predict how to place subsets of operations in a Tensorflow graph on available devices, using the execution time of the predicted placement as reward signal for REINFORCE algorithm(Yuxi Li, 2018).

## 5. Conclusion

Over the past few years, deep RL has become increasingly powerful and important in handling complex problems. Deep learning models with great power of automatically extracting complex data representations from high-dimensional input data could outperform other state of the art of traditional machine learning methods. A major challenge in reinforcement learning is to learn optimal control policies in problems with raw visual input. Hierarchical feature extraction and learning abstracted representations of deep architectures, not only made the deep learning become a valuable tool for classification, but it has made it to be a great solution for the mentioned challenge in RL tasks as well. Despite of the significant works done to data in combining RL and DL, research on deep reinforcement learning is at its first steps and there are still many unexplored aspects of this combination. Also, their challenges in real application such as robotics, are yet unsolved and need more exploration to be done. Especially, developing those mechanisms which make the end to end learning can be practical in real world application, those which doing a large number of actions is impossible. Furthermore, an open problem that has not yet been addressed is how deep architectures can help deep reinforcement learning models to transfer knowledge (transfer learning). Indeed, how to use learned features by the deep networks for different tasks, without changing the network architectures. There is an immense scope for DRL due to its learning behaviour and hence, the possibilities of its application are immeasurable. The study also concludes that a larger skill force will be required to cater to these applications with specific knowledge in different industrial sectors, especially in healthcare, automotive, smart city and intelligent transportation.

In this study , I present a comprehensive study of deep RL algorithms. I have presented recent advances in combing reinforcement learning framework and deep leaning models for both deep supervised and unsupervised learning networks. In particular, the deep architectures that have been most used in combination with RL such as deep convolutional networks, deep autoencoders and deep recurrent networks. In addition, appropriate deep networks for the problems with partially observable MDPs (POMDPs) environment, have been discussed. First, I introduce background theory of RL then I performed a clear and novel dissection of model-free and model-based deep RL algorithms. Finally in terms of applications, I discuss deep RL in robotics, NLP, and computer systems etc.

**References**

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, Anil Anthony Bharath, A Brief Survey of Deep Reinforcement Learning, IEEE signal processing magazine, special issue on deep learning for image understanding (arxiv extended version), 2017

Nate Kohl and Peter Stone. Policy Gradient Reinforcement Learning for Fast Quadrupedal Locomotion. In ICRA, volume 3, 2004.

Andrew Y Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger, and Eric Liang. Autonomous Inverted Helicopter Flight via Reinforcement Learning. Experimental Robotics, pages 363–372, 2006.

Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. JAIR, 16:105–133, 2002.

Gerald Tesauro. Temporal Difference Learning and TD-Gammon. Communications of the ACM, 38(3):58–68, 1995

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep Learning. Nature, 521(7553):436–444, 2015.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. IEEE Trans. on Pattern Analysis and Machine Intelligence, 35(8):1798–1828, 2013.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016a). Mastering the game of go with deep neural networks and tree search. Nature, 529(7587):484–489.

Silver, D. (2016). Deep reinforcement learning, a tutorial at ICML 2016. http://icml.cc/ 2016/tutorials/deep_rl_tutorial.pdf.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. (2017). Mastering the game of go without human knowledge. Nature, 550:354–359.

Pieter Abbeel and John Schulman. Deep Reinforcement Learning through Policy Optimization, 2016. Tutorial at NIPS 2016.

Yuxi Li, Deep Reinforcement Learning: An Overview, arXiv:1701.07274v6 [cs.LG] 26 Nov 2018

Fang, X., Misra, S., Xue, G., and Yang, D. (2012). Smart grid - the new and improved power grid: A survey. IEEE Communications Surveys Tutorials, 14(4):944–980.

Anderson, R. N., Boulanger, A., Powell, W. B., and Scott, W. (2011). Adaptive stochastic control for the smart grid. Proceedings of the IEEE, 99(6):1098–1115.

Hull, J. C. (2014). Options, Futures and Other Derivatives (9th edition). Prentice Hall.

Seyed Sajad Mousavi(&), Michael Schukat, and Enda Howley, Deep Reinforcement Learning: An Overview, Springer International Publishing AG

Mostafa Al-Emran, Hierarchical Reinforcement Learning: A Survey, International Journal of Computing and Digital Systems, Int. J. Com. Dig. Sys. 4, No.2 , http://dx.doi.org/10.12785/ijcds/040207 (Apr-2015)

Maia, T. V. (2009). Reinforcement learning, conditioning, and the brain: Successes and challenges. Cognitive, Affective, & Behavioral Neuroscience,9(4), 343-364.

Surjeet Balhara, Nishu Gupta, Ahmed Alkhayyat, Isha Bharti, Rami Q. Malik, Sarmad Nozad Mahmood, Firas Abedi, A survey on deep reinforcement learning architectures, applications and emerging trends, IET Commun., IET Commun. 2022;1–16.

Ming, G. F., & Hua, S. (2010). Course-scheduling algorithm of option-based hierarchical reinforcement learning. In 2010 Second International Workshop on Education Technology and Computer Science, Vol. 1, pp. 288-291.

Gil, P., & Nunes, L. (2013, June). Hierarchical reinforcement learning using path clustering. In Information Systems and Technologies (CISTI), 2013 8th Iberian Conference on (pp. 1-6). IEEE.

Barto, A. G., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. Discrete Event Dynamic Systems, 13(4), 341-379

Pamina, J., Raja, B.: Survey on deep learning algorihms. Int. J. Emerging Technol. Innovative Eng. 5(1), 6 (2019) N. R. Ravishankar and M. V. Vijayakumar, Reinforcement Learning Algorithms: Survey and Classification, Indian Journal of Science and Technology, Vol 10(1), DOI: 10.17485/ijst/2017/v10i1/109385 January 2017

Jose JFR. Solway A, Diuk C, McGuire JT, Barto AG, Niv Y.A Neural Signature of Hierarchical Reinforcement Learning. Neuron. 2011 Jul; 71(2): 370–379.

Amit Kumar Mondal, A Survey of Reinforcement Learning Techniques: Strategies, Recent Development, and Future Directions, Reasearchgate, 2021

Chapman JR. Work Breakdown Structures, ver. 2.01, [Online]. 2004. Available: http://www.hyperthot.com/pm_wbs.htm

Quentin JM, Huysa B, Anthony C, Peggy S. Reward-Based Learning, Model-Based and Model-Free. Encyclopedia of Comput. Neurosci. 2014

Eisha Akanksha, Jyoti, Neeraj Sharma, Dr. Kamal Gulati, Review on Reinforcement Learning, Research Evolution and Scope of Application, Proceedings of the Fifth International Conference on Computing Methodologies and Communication (ICCMC 2021) DVD Part Number: CFP21K25-DVD: ISBN: 978-0-7381-1203-9

Pawel Ladosz, Lilian Weng, Minwoo Kim, Hyondong Oh, Exploration in Deep Reinforcement Learning: A Survey, arXiv:2205.00824v1 [cs.LG] 2 May 2022.

Victor Dolk, Survey Reinforcement Learning, 2010

Hao-nan Wang‡, Ning LIU, Yi-yun Zhang, Da-wei Feng, Feng Huang, Dong-sheng LI, Yi-ming Zhang, Deep reinforcement learning: a survey, Frontiers of Information Technology & Electronic Engineering www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com, 2020.

Williams RJ, 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach Learn, 8(3-4):229-256.

Nagabandi A, Kahn G, Fearing RS, et al., 2018. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. IEEE Int Conf on Robotics and Automation, p.7559-7566. https://doi.org/10.1109/ICRA.2018.8463189

Gu JT, Hassan H, Devlin J, et al., 2018a. Universal neural machine translation for extremely low resource languages. Proc 16th Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.344-354. https://doi.org/10.18653/v1/N18-1032

Gu JT, Wang Y, Chen Y, et al., 2018b. Meta-learning for low-resource neural machine translation. Proc Conf on Empirical Methods in Natural Language Processing, p.3622-3631.https://doi.org/10.18653/v1/D18-1398

Chen L, Lingys J, Chen K, et al., 2018. AuTO: scaling deep reinforcement learning for datacenter-scale automatic traffic optimization. Proc Conf of the ACM Special Interest Group on Data Communication, p.191-205. https://doi.org/10.1145/3230543.3230551

Deepanshu Mehta ,State-of-the-Art Reinforcement Learning Algorithms, International Journal of Engineering Research & Technology, http://www.ijert.org ISSN: 2278-0181, Vol. 8 Issue 12, December-2019

Aristotelis Lazaridis et al., Deep Reinforcement Learning: A State-of-the-Art Walkthrough, Journal of Artificial Intelligence Research (2020)

Constantin-Valentin Pal, Florin Leon, A Brief Survey of Model-Based Reinforcement Learning Techniques, 24th International Conference on System Theory, Control and Computing, 2020.