A Survey of Deep Reinforcement Learning in Video Games (Write-up)

Kun Shao, Zhentao Tang, Yuanheng Zhu, Member, IEEE, Nannan Li, and Dongbin Zhao, Fellow, IEEE

Assignment 5 | Artificial Intelligence II
Student: Tejas Gaikwad (MT19AI021)
Department: Computer Science and Engineering
Institute: Indian Institute of Technology, Jodhpur, Rajasthan

----

The authors have surveyed the progress of Deep Reinforcement Learning (DRL) methods which include value-based, policy gradient, model-based algorithms. They have also compared the main techniques and properties of DRL.
In Reinforcement Learning, generally, the agent receives high dimensional input at each step and take actions accordingly. The game environment (virtual environment) provides an infinite supply of useful data for machine learning algorithms thus they are much faster than real-time scenarios. The artificial intelligence involves perception and decision making in-game environments but those have several challenges too like the state space of the game is very large especially, in the strategic game, learning proper policies to make decisions in the dynamic unknown environment is difficult, majority of the game AI is developed in a specified virtual environment, so transferring the AI ability to different games become a problem. So, looking at these challenges, we would need a system that can be successfully modeled with large state space, models that are data-driven to learn proper policies to\hat can take decisions in the dynamic environment, and more general learning system to help with the transferability of the AI to different games.
Game AI with deep reinforcement learning is a challenging and promising direction. Recent progress in this domain has promoted the development of artificial intelligence research. In this paper, they have also reviewed the achievements of deep reinforcement learning in video games. Different DRL methods and their successful applications are introduced. These DRL agents achieve human-level or super-human performances in various games, from 2D perfect information to 3D imperfect information, and from single-agent to multi-agent. In addition to these achievements, there are still some major problems when applying DRL methods to this field, especially in 3D imperfect information multi-agent video games. A high-level game AI requires exploring more efficient and robust DRL techniques and needs novel frameworks to be implemented in a complex environment. These challenges have not been fully investigated and could be opened for further study in the future. The details about these have been discussed further.
Challenges in Games with Deep Reinforcement Learning

1. **Deep Learning**

    - Comes from artificial Neural Networks and is used to learn data representation.
    - CNN(mainly used for images, massively used in Image Processing), RNN(User mainly in fields related to Natural Language Processing), LSTM (mainly used to learn long term dependencies).

- Deep learning architectures have been applied to many fields, and have achieved significant successes, such as speech recognition, image classification and segmentation, semantic comprehension, and machine translation.

2. Reinforcement Learning

- Its a kind of machine learning method where agents learn the optimal policy by trial and error.
- By interacting with the environment, RL can be successfully applied to sequential decision-making tasks.
- On the basis of policies, reinforcement learning can be divided into 2 classes, off-policy methods, and on-policy methods.
- Off-policy RL algorithms mean that the behavior policy used for selecting actions is different from the learning policy.
- Behavior policy is the same as the learning policy in on-policy RL algorithms.
- Reinforcement learning can also be divided into value-based and policy-based methods.
- In value-based RL, agents update the value function to learn suitable policy, while policy-based RL agents learn the policy directly.

3. Deep Reinforcement Learning

- Its the combination of Deep Learning and Reinforcement Learning
- Types:
  - Value-based methods,
  - Policy gradient methods
  - Model-based methods

- Values-Based Methods:
  - Based upon temporal difference learning (TD, SARSA, Q-learning), learn value function (or a slight generalization called the Q-function, that we will discuss shortly)
  - Deep Q-network (DQN) is the most famous DRL model which learns policies directly from high-dimensional inputs
  - Q-learning is a model-free reinforcement learning algorithm to learn a policy telling an agent what action to take under what circumstances. It does not require a model of the environment, and it can handle problems with stochastic transitions and rewards, without requiring adaptations.
  - It receives raw pixels and outputs a value function to estimate future rewards
  - DQN uses the experience replay method to break the sample correlation and stabilizes the learning process with a target Q-network.
  - DQN bridges the gap between high-dimensional visual inputs and actions

- Policy-Gradient Methods

- Policy Gradient is another popular model-free RL method but it doesn't calculate Q-value but instead uses a policy. Policy learns a mapping from every state to act, and its objective is to find which actions lead to higher rewards and increase their probability. The policy gradient observes the environment and acts in it and keeps updating its policy based on the rewards it receives. After multiple iterations, policy converges to a maximum value. Drawbacks of the policy gradient method can be that it sometimes can get stuck in local maximum and will not be able to reach the global maximum.
- Directly learn optimal policy (or try to approximate optimal policy, if true optimal policy is not attainable)

- Model-Based Methods
    - Neural networks or Gaussian processes for representing dynamics, advanced optimal control for planning. Used in DRL in video games.
    - It will converge to the correct Markov Decision Processes (MDP) and hence correct optimal value function / optimal policy) given enough samples of each state ensure that we get the "right" samples.
    - Advantages (informally) of model-based RL: make "efficient" use of data.
    - Disadvantages: requires we build the actual MDP models, solve for optimal policy (not much help if state space is too large).

4. Deep Reinforcement Learning in GAMES

- Game Research Platforms
    - Arcade Learning Environment (ALE): ALE presents both game images and signals, such as player scores, which makes it a suitable testbed.
    - Malmo is a research platform for AI experiments, which is built on top of Minecraft. It is a first-person 3D environment and can be used for multi-agent research in Microsoft Malmo collaborative AI challenge 2017 and the multi-agent RL in MalmO competition 2018.

- Atari Games
    - The visual features of the Atari games (for a frozen scene (state) of the game) are learned using Deep Neural networks having convolution layers that are rich feature learning models for images. And it so happens that for Atari games, such a feature construction is somewhat sufficient for summarizing the knowledge for state representation, in case you are concerned about the Markovness aspects of the state, for applying the Reinforcement Learning Algorithms modeled on MDP's.
    - Reinforcement Learning, then, is done, using suitable gradient-based methods, on the constructed features for the state space, based on suitable reward functions, states, and the MDP dynamics that you define or set out to learn.
    - It might be possible to incorporate gradient-based RL methods into the deep network to simultaneously learn to augment the feature models of the states based on their interpretations in the RL context.

- First Person Perspective Game
    - Agents in first-person perspective video games can only receive observations from their own perspectives, resulting from imperfect information inputs. In the RL domain, this is a POMDP problem that requires efficient exploration and memory. Example: VIZDoom, TORCS, Minecraft.

- Realtime Strategies Games
    - Realtime Strategies can be modeled as MDP. This is simply by making the state spaces more detailed. E.g., construction time of building X, positions of units, etc. are all contained in the state of the game. In general, the game engines themselves are constructed like that. However, conventional reinforcement learning will not be capable to address such a vast state+action space. There are several approaches to tackle this:
    - Generalize over state+action space; if your feature selection/model is sufficiently complex, you might get away with it or perform 'sufficient enough'.
    - Approach the model as a POMDP and keep belief states about the state of the world and accommodate for the uncertainty that comes with it.
    - Hierarchical Reinforcement learning (e.g. MAXQ). Break down the points to be addressed in a layered manner: high-level versus low-level strategy. E.g., in starcraft, one model addresses the general strategy (do I build unit X or unit Y; do I attack/defend), whereas individual units are handled by a separate model.
    - Such games can be StarCraft and MOBA and Dota2

5. Challenges in Games with Deep Reinforcement Learning

- Exploration-exploitation
    - When you keep doing what you are doing is called exploration and when you do something new is called exploitation.
    - In DRL, there is a requirement of the tradeoff between exploration and exploitation.
    - Exploring actions that have low priority of being optimal is a waste of time and resources. For this reason, it is important to use exploration methods that minimize regrets, so that the learning phase becomes faster and more efficient.

- Sample Efficiency
    - In order to reduce the exploration dimension of the environment and ease the expenditure of time on interaction, some solutions can be used for improving data efficiencies, such as hierarchy and demonstration.
    - Hierarchical reinforcement learning (HRL) allows agents to decompose the task into several simple subtasks, which can speed up training and improve sample efficiency
    - The demonstration is a proper technique to improve sample efficiency. Current approaches that learn from demonstration using supervised learning on expert data and use reinforcement learning to improve the performance

- Generalization and Transfer
    - The ability to transfer knowledge across multiple environments is considered as a critical aspect of intelligent agents.
    - More general learning system to help with the transferability of the AI to different games.

- Multi-Agent Learning
    - It’s a deep learning discipline that focuses on models that include multiple agents that learn by dynamically interacting with their environment.
    - Team learning uses a single learner to learn joint solutions in a multi-agent system, while concurrent learning uses multiple learners for each agent

- Imperfect Information
    - For the environment having imperfect information, a critical component for enabling effective learning would be the use of memory.
    - Model-free episode control learns difficult sequential decision-making tasks much faster and achieves a higher overall reward
    - The differentiable neural computer uses a neural network to read from and write to an external memory matrix. This method can solve complex, structured tasks that can not access to neural networks without external read and write memory.
    - Neural episodic control inserts recent state representations paired with corresponding value functions into the appropriate neural dictionary, and learns significantly faster than other baseline agents.

- Delayed Spare rewards
    - The sparse and delayed helps to reduce sample efficiency in reinforcement learning.
    - The agent's actions determine not only its immediate reward but also (at least probabilistically) the next state of the environment.
    - Researchers use curiosity as an intrinsic reward to encourage agents to explore the environment and learn useful skills.
    - Curiosity can be formulated as the error that the agent predicts its own actions’ consequence in a visual space.
    - Curiosity search for DRL encourages intra-life exploration by rewarding agents for visiting as many different states as possible within each episode