**A SURVEY OF DEEP RL IN VIDEO GAMES - 2019**

**Pupose of the article:**

We survey the progress of DRL methods, including value-based, policy gradient, and model-based algorithms, and compare their main techniques and properties. We also take a review of the achievements of DRL in various video games, including classical Arcade games, first-person perspective games and multi-agent real-time strategy games, from 2D to 3D, and from single-agent to multi-agent.

**Cuprins:**

Section II: we introduce the background of DL and RL

Section III: recent DRL methods, including value-based, policy gradient, and model-based DRL methods.

Section IV: brief introduction of research platforms and competitions, and present performances of DRL methods in classical single-agent Arcade games, first-person perspective games, and multi-agent real-time strategy games.

Section V: key points and research directions in this field.

Section VI: conclusion & discussion

**Intro:**

Deep reinforcement learning (DRL=DL+RL) has made great achievements since proposed. Generally, DRL agents receive high-dimensional inputs at each step, and make actions according to deep-neural-network-based policies. This learning mechanism updates the policy to maximize the return with an end-to-end method. DRL has made great progress in video games, including Atari, ViZDoom, StarCraft, Dota2, and so on.

De rezumat: Artificial intelligence (AI) in video games is a long-standing research area. It studies how to use AI technologies to achieve human-level performance when playing games. More generally, it studies the complex interactions between agents and game environments. Various games provide interesting and complex problems for agents to solve, making video games perfect environments for AI research. These virtual environments are safe and controllable. In addition, these game environments provide infinite supply of useful data for machine learning algorithms, and they are much faster than real-time. These characteristics make games the unique and favorite domain for AI research. On the other side, AI has been helping games to become better in the way we play, understand and design them.

Section II:

**Background (teorie de retinut):**

DL Deep learning is inspired by the theory of brain development.

CNN Convolutional neural network is a class of deep neural networks, which is widely applied to computer vision, inspired by biological processes, and is shift invariant based on shared-weights architecture.

RNN Recurrent Neural Network is another kind of deep nerial network, especially for natural language processing.

RL Reinforcement learning is a kind of machine learning methods 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. Considering a discounted episodic **Markov decision process (MDP)** (S, A, γ, P, r), the agent chooses an action at according to the **policy** π(at|st) at **state** st. The environment receives the **action**, produces a **reward** rt+1 and transfers to the next state st+1 according to the transition probability P(st+1|st, at). This transition probability is unknown in RL domain. The process continues until the agent reaches a terminal state or a maximum time step. The objective is to maximize the expected discounted cumulative rewards Eπ[Rt] = Eπ[ Sumi=0 -> ∞ γi rt+i ], (1) where γ ∈ (0, 1] is the ***discount factor***.

RL can be devided into off-policy and on-policy methods, and also 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. Q-learning is a typical off-policy value-based method.

DRL makes a combination of DL and RL, achieving rapid developments since proposed.

**A. Value-based DRL methods**

DQN = Deep Q-network is the most famous DRL model which learns policies directly from high-dimensional inputs. It receives raw pixels, and outputs a value function to estimate future rewards.

DRQN = Deep Recurrent Q-Network replaces the first fully-connected layer with a recurrent neural network in DQN, to address the limited memory and imperfect game information at each decision point.

LS-DQN = The Least Squares DQN combines DQN’s rich feature representations with the stability of a linear least squares method.

DQfD = Deep Q-learning from Demonstrations combines DQN with human demonstrations, which improves the sample efficiency greatly.

Ape-X DQfD uses a new transformed Bellman operator to process rewards of varying densities and scales, and applies human demonstrations to ease the exploration problem to guide agents towards rewarding states

DQV uses TD learning to train a Value neural network, and uses this 4 network to train a second Quality-value network to estimate state-action values. DQV learns significantly faster and better than double-DQN.

Soft DQN is an entropy-regularized versions of Q-learning, with better robustness and generalization.

QR-DRL = Distributional reinforcement learning with Quantile regression in which the distribution over returns is modeled explicitly instead of only estimating the mean.

IQN = Implicit Quantile Networks is a state-of-the-art distributional DQN, it approximates the full Quantile function for the return distribution with Quantile regression.

RUDDER is a novel reinforcement learning approach for finite MDPs with delayed rewards, which is also a return decomposition method, RUDDER is exponentially faster on tasks with different lengths of reward delays.

Distributional DRL learns the value distribution, in contrast to common RL that models the expectation of return, or value

C51 focuses on the distribution of value, and designs distributional DQN algorithm to learn approximate value distributions.

**B. Policy gradient DRL methods**

-> optimizes the parameterized policy directly. Actor-critic architecture computes the policy gradient using a value-based critic function to estimate expected future reward.

Asynchronous DRL is an efficient framework for DRL that uses asynchronous gradient descent to optimize the policy. Asynchronous advantage actor-critic (A3C) trains several agents on multiple environments, showing a stabilizing effect on training.

UNsupervised REinforcement and Auxiliary Learning (UNREAL) learns separate policies for maximizing many other pseudo-reward functions simultaneously.

PAAC is a novel framework for efficient parallelization of DRL, where multiple actors learn the policy on a single machine.

PGQ (=the new method) which combines policy gradient with Q-learning, establishing an equivalency between regularized policy gradient techniques and advantage function learning algorithms.

Retrace(λ) takes the best of the importance sampling, off-policy Q(λ), and tree-backup(λ), resulting in low variance, safety, and efficiency.

Reactor is a sample-efficient and numerical efficient reinforcement learning agent based on a multi-step return off-policy actor-critic architecture.

IMPALA = Importance Weighted Actor Learner Architecture is a new distributed DRL, which can scale to thousands of machine. IMPALA uses a single reinforcement learning agent with a single set of parameters to solve a mass of tasks.

1. Trust region method:

TRPO: Trust Region Policy Optimization is proposed for optimizing control policies, with guaranteed monotonic improvement. This algorithm is effective for optimizing large nonlinear policies.

PPO: Proximal policy optimization samples data by interaction with the environment, and optimizes the objective function with stochastic gradient ascent. PPO has some benefits over TRPO, and is much simpler to implement, with better sample complexity.

ACER: Actor-critic with experience replay introduces several innovations, including stochastic dueling network, truncated importance sampling, and a new trust region method, which is stable and sample efficient.

ACKTR: Actor-critic using Kronecker-Factored Trust Region bases on natural policy gradient, and uses Kronecker-factored approximate curvature (K-FAC) with 5 trust region to optimize the actor and the critic. ACKTR is sample efficient compared with other actor-critic methods.

1. Deterministic policy:

DDPG: Apart from stochastic policy, deep deterministic policy gradient is a kind of deterministic policy gradient method which adapts the success of DQN to continuous control. DDPG is an actor-critic, off-policy algorithm, and is able to learn reasonable policies on various tasks.

D4PG: Distributed Distributional DDPG is a distributional update to DDPG, combined with the use of multiple distributed workers all writing into the same replay table. This method has a much better performance on a number of difficult continuous control problems.

1. Entropy-regularized policy gradient:

SAC: Soft Actor Critic is an off-policy policy gradient method, which establishes a bridge between DDPG and stochastic policy optimization.

**C. Model-based DRL methods**

Combining model-free reinforcement learning with on-line planning is a promising approach to solve the sample efficiency problem.

TreeQN is proposed to address these challenges: it’s a tree-structured model that serves as a drop-in replacement for any value function network in DRL with discrete actions.

ATreeC is an actor-critic variant that augments TreeQN with a softmax layer to form a stochastic policy network

STRategic Attentive Writer (STRAW) neural network architecture to build implicit plans. It purely interacts with an environment, and is an end-to-end method.

Value propagation (VProp) bases on value iteration, and is an efficient differentiable planning module. It can successfully be trained to learn to plan using RL.

Section IV: brief introduction of research platforms and competitions, and present performances of DRL methods in classical single-agent Arcade games, first-person perspective games, and multi-agent real-time strategy games.

**DRL IN VIDEO GAMES**

Playing video games like human experts is challenging for computers. With the development of DRL, agents are able to play various games end-to-end. Real-time strategy (RTS) games are very challenging for reinforcement learning method.

**A. Game research platforms**

Platforms and competitions make great contributions to the development of game AI, and help to evaluate agents’ intelligence.

* General Platforms:
  + Arcade Learning Environment (ALE) is the pioneer evaluation platform for DRL algorithms, which provides an interface to plenty of Atari 2600 games (?). ALE presents both game images and signals, such as player scores, which makes it a suitable testbed. To promote the progress of DRL research, OpenAI integrates a collection of reinforcement learning tasks into a platform called Gym, which mainly contains Algorithmic, Atari, Classical Control, Board games, 2D and 3D robots.
  + OpenAI Universe is a platform for measuring and training agents’ general intelligence across a large supply of games. Gym Retro is a wrapper for video game emulator with a unified interface as Gym, and makes Gym easy to be extended with a large collection of video games, not only Atari but also NEC, Nintendo, and Sega (?), for RL research.
  + General Video Game Playing is intended to design an agent to play multiple video games without human intervention. The General Video Game AI (GVGAI) competition is proposed to provide a easy-to use and open-source platform for evaluating AI methods, including DRL.
  + DeepMind Lab is a first-person perspective learning environment, and provides multiple complicated tasks in partially observed, large-scale, and visually diverse worlds.
  + Unity ML-Agents Toolkit is a new toolkit for creating and interacting with simulation environments. This platform has sensory, physical, cognitive, and social complexity, and enables fast and distributed simulation, and flexible control.
* Specific Platforms:
  + 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.
  + TORCS is a racing car simulator which has both low-level and visual features for the self-driving car with DRL.
  + ViZDoom is a first-person shooter game platform, and encourages DRL agent to utilize the visual information to perform navigation and shooting tasks in a semi-realistic 3D world.
  + Facebook proposes TorchCraft for StarCraft I, and DeepMind releases StarCraft II learning environment.They expect researchers to propose powerful DRL agents to achieve high-level performance in RTS games and annual StarCraft AI competitions.
  + CoinRun provides a metric for an agent’s ability to transfer its experience to novel situations.
  + Google Research Football is a new environment based on open-source game Gameplay Football for DRL research.

**B. Atari games**

In this section, we will introduce the main achievements in the ALE domain, including the extremely difficult Montezuma’s Revenge (is one of the most difficult Atari video games. It is a goal-directed behavior learning environment with long horizons and sparse reward feedback signals. Players must navigate through a number of different rooms, avoid obstacles and traps, climb ladders up and down, and then pick up the key to open new rooms. It requires a long sequence of actions before reaching the goal and receiving a reward, and is difficult to explore an optimal policy to tackle tasks. Learning from human data is also a proper method to reach better performance in this problem).

* DQN achieves human-level performance across 49 games.
* Averaged-DQN examines the source of value function estimation errors, and demonstrates significantly improved stability and performance on the ALE benchmark.
* UNREAL outperforms the previous best performance on Atari, averaging 880% expert human performance.
* PAAC achieves sufficiently good performance on ALE after a few hours of training.
* DQfD has better initial performance than DQN on most Atari games, and receives more average rewards than DQN on 27 of 42. In addition, DQfD learns faster than DQN even when given poor demonstration data.
* Noisy DQN replaces the conventional exploration heuristics with NoisyNet, and yields substantially higher scores
* C51 obtains impressive results, and demonstrates the importance of the value distribution in approximated RL.
* Rainbow provides improvements in terms of sample efficiency and final performance.
* Ape-X DQN achieves better final score in less wall-clock training time
* Ape-X DQfD algorithm exceeds the performance of an average human on 40 games using a common set of hyperparameters.

**C. First-person perspective games**

Different from Atari games, agents in first-person perspective video games can only receive observations from their own perspectives, resulting from imperfect information inputs. In RL domain, this is a POMDP problem which requires efficient exploration and memory.

* ViZDoom: First-person shooter (FPS) games play an important role in game AI research. Doom is a classical FPS game, and ViZDoom is presented as a novel testbed for DRL. Agents learn from visual inputs, and interact with the ViZDoom environment in a first-person perspective.
* TORCS: TORCS is a racing game where actions are acceleration, braking and steering. This game has more realistic graphics than Atari games, but also requires agents to learn the dynamic of the car.
* Minecraft: Minecraft is a sandbox construction game, where players can build creative creations, structures, and artwork across various game modes. Recently, it becomes a popular platform for game AI research, with 3D infinitely varied data. Project Malmo is an experimentation platform that builts on the Minecraft for AI research. It supports a large number of scenarios, including navigation, problem solving tasks, and survival to collaboration.
* DeepMind lab: DeepMind lab is a 3D first-person game platform extended from OpenArena, which is based on Quake3. Comparable to other first-person game platforms, DeepMind lab has considerably richer visuals and more realistic physics, making it a significantly complex platform.

**D. Real-time strategy (RTS) games**

* StarCraft: players need to perform actions according to real-time game states, and defeat the enemies. Generally speaking, designing an AI bot have many challenges, including multi-agent collaboration, spatial and temporal reasoning, adversarial planning, and opponent modeling. Currently, most bots are based on human experiences and replays, with limited flexibility and intelligence. DRL is proved to be a promising direction for StarCraft AI, especially in micromanagement, build order, mini-games and full-games.
* MOBA and Dota2: MOBA (Multiplayer Online Battle Arena) is originated from RTS games, which has two teams, and each team consists of five players. To beat the opponent, five players in a team must cooperate together, kill enemies, upgrade heros, and eventually destroy the opponent base. King of Glory (a simplified mobile version of Dota) is the most popular mobile-end MOBA game in China.

Section V: CHALLENGES IN GAMES WITH DRL

Since DRL has achieved large progress in some video games, it is considered as one of most promising ways to realize the artificial general intelligence. However, there are still some challenges should be conquered towards goal:

-tradeoff between exploration and exploitation

-low sample efficiency

-dilemma in generalization and overfiting

-multi-agent learning

-incomplete information

delayed sparse rewards

1. Exploration-exploitation

Exploration can help to obtain more diversity samples, while exploitation is the way to learn the high reward policy with valuable samples => trade-off between exploration and exploitation = major challenge for RL :) Common methods for exploration require a large amount of data, and can not tackle temporally-extended exploration. Combining exploration with deep neural networks can help to learn much faster, which greatly improves the learning speed and final performance in most games

1. Sample efficiency

Humans can quickly master highly rewarding actions of an environment, while DRL algorithms usually take millions of samples to achieve human-level performance, but some solutions can be used for improving data efficiency, 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. + FeUdal Networks (FuNs) include a Manager module and a Worker module: the Manager sets abstract goals at high-level, the Worker receives these goals, and generates actions in the environment. FuN dramatically outperforms baseline agents on tasks that involve long-term credit assignment or memorization.

1. Generalization and Transfer

Multi-task learning with shared neural network parameters can solve the generalization problem, and efficiency can be improved through transfer across related tasks. Hybrid reward architecture takes a decomposed reward function as input (much smoother, which can be easily approximated with a low-dimensional representation) and learns a separate value function for each component (more effectively) => IMPALA shows the effectiveness for multi-task reinforcement learning, using less data and exhibiting positive transfer between tasks. To successfully learn complex tasks with DRL, we usually need large task-specific networks and extensive training to achieve good performance. Distral shares a distilled policy which can learn common knowledge across multiple tasks. Each worker is trained to solve individual task and to be 11 close to the shared policy, while the shared policy is trained by distillation. Mix & Match is a training framework that is designed to encourage effective and rapid learning in DRL agents. It allows to automatically form a curriculum over agent, and progressively trains more complex agents from simpler agents

1. Multi-agent learning

Multi-agent learning is very important in video games, such as StarCraft. In a cooperative multi-agent setting, curse-of-dimensionality, communication, and credit assignment are major challenges. Team learning uses a single learner to learn joint solutions in multi-agent system, while concurrent learning uses multiple learners for each agent. Recently, the centralised training of decentralised policies is becoming a standard paradigm for multi-agent training. Multi-agent DDPG considers other agents’ action policy and can successfully learn complex multi-agent coordination behavior.

1. Imperfect information

In partially observable and first-perspective games, DRL agents need to tackle imperfect information to learn a suitable policy. Making decisions in these environments is challenging for DRL agents. A critical component of enabling effective learning in these environment is the use of memory. DRL agents have used some simple memory architectures, such as several past frames or an LSTM layer. But these architectures are limited to only remember transitory information. Model-free episode control learns difficult sequential decision-making tasks much faster, and achieves a higher overall reward.

1. Delayed spare (=disponibil/de rezerva) rewards

The sparse (risipit/imprastiat) and delayed reward is very common in many games, and is also one of the reasons that reduce sample efficiency in reinforcement learning. In many scenarios, researchers use curiosity as an intrinsic reward to encourage agents to explore environment and learn useful skills. Curiosity can be formulated as the error that the agent predicts its own actions’ consequence in a visual space. This can scale to high-dimensional continuous state spaces. Moreover, it leaves out the aspects of environment that cannot affect agents. Curiosity search for DRL encourages intra-life exploration by rewarding agents for visiting as many different states as possible within each episode.

Section VI:

Game AI with deep reinforcement learning is a challenging and promising direction. Recent progress in this domain has promote the development of artificial intelligence research. In this paper, we review 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 game. A high-level game AI requires to explore more efficient and robust DRL techniques, and needs novel frameworks to be implemented in complex environment.