Review of recent advances in Deep Reinforcement Learning

Maxime Toquebiau

25th February 2021

Markov Decision Process

$$(X, A, p, r)$$

- X: State space
- A: Action space
- p: Transition probability
- r: Reward function
- Policy: $\pi(x) = a$
- Value: $V_\pi(x) = E_\pi[R_t | S_t = x]$

- Temporal Difference Learning (TD Learning)
  - TD Learning (Sutton, 1988)
  - Q-Learning (Watkins and Dayan, 1992)
  - SARSA (Rummery and Niranjan, 1994)
- Policy Optimization
  - REINFORCE (Williams, 1992)
  - Actor-Critic
  - Cross-Entropy Methods (Szita and Lorincz, 2006)
- Model-based learning
  - Dyna-Q (Sutton, 1990)
- Evolution Strategies

⇒ **Successful use of RL:** TD-Gammon (Tesauro, 1995), TD search for Go (Silver et al., 2012), Policy Gradient for quadrupedal locomotion (Kohl and Stone, 2004)

- Use of handcrafted features
- Limited to low dimensional problems
  - State space
  - Action space
- Hard to generalize to new situations/environments/tasks

$\Rightarrow$ Lack of scalability

- Powerful (non-linear) function approximation
- Powerful representation learning

$\Rightarrow$ Scale to high-dimensional state and action spaces

$\Rightarrow$ Generalize better

$\Rightarrow$ Learn from high-dimensional sensory input (e.g. vision, speech)

$\Rightarrow$ Perform better in partially observable environments

Different branches of Deep Reinforcement Learning (DRL)

- Value-based
- Policy-based
- Model-based
- Other approaches:
  - Meta-Learning
  - Intrinsic Objectives
  - Imitation Learning
  - Inverse RL
  - Hierarchical RL
  - ...

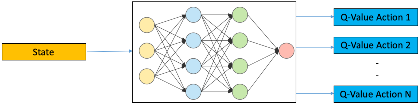
Deep Q-Network (DQN):

"Playing Atari with Deep Reinforcement Learning", Mnih et al., 2013
"Human-Level control through deep reinforcement learning", Mnih et al., 2015

PROBLEM: Learning to control agents directly from pixel images

⇒ DQN: CNN to analyse raw pixels and output Q-values

↳ Single architecture achieves professional human level across 49 Atari games



2013	Deep Reinforcement Learning
2015	Deep Q-Network
2015	Prioritized Experience Replay

Prioritized Experience Replay (PER):

"Prioritized Experience Replay", Schaul et al., 2015

PROBLEM: Sample efficiency

⇒ PER: Replay more frequently transitions with high TD-Error

↳ Sample efficiency x2
↳ Beat all previous methods on the Atari benchmark
↳ Also favor stochastic transitions

2013	Deep Reinforcement Learning
2015	Deep Q-Network
2015	Prioritized Experience Replay
2016	Double DQN

Double DQN (DDQN):

"Deep Reinforcement Learning with Double Q-Learning", van Hasselt et al., 2016

PROBLEM: Overestimation of Q-values in (Deep) Q-Learning

$\Rightarrow$ Decoupling action selection and evaluation with one model for each

$$Y_t^{DoubleQ} \equiv R_{t+1} + \gamma Q(S_{t+1}, \arg\max_a Q(S_{t+1}, a; \boldsymbol{\theta_t}); \boldsymbol{\theta_t}')$$

↳ Reduces overestimation of Q-values
↳ Increases performance

Dueling DQN:

"Dueling Network Architectures for Deep Reinforcement Learning", Wang et al., 2016

PROBLEM: Find better suited neural network architectures for model-free RL

$\Rightarrow$ Separately estimate state value and advantage for each action

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left(A(s, a; \theta, \alpha) - \frac{1}{|A|} \sum_{a'} A(s, a'; \theta, \alpha) \right)$$

↳ Less variance in training
↳ Increased performance

Distributional DQN:
"A Distributional Perspective on Reinforcement Learning", Bellamare et al., 2017

PROBLEM: When learning the expected (**average**) return of an action, we lose some information about possible outcomes

⇒ Learn a distribution of random return for each action (*value distribution*)
Distributional Bellman equation:

$$Z(x, a) \equiv R(x, a) + \gamma Z(X', A')$$

↳ Greatly increased performance on Atari benchmark
⇒ Learning a value distribution matters, even when using a policy which aims to maximize expected return

2013	Deep Reinforcement Learning
2015	Deep Q-Network
2015	Prioritized Experience Replay
2016	Double DQN
2016	Dueling DQN
2017	Distributional DQN
2017	Noisy DQN

Noisy DQN:

"Noisy Networks for Exploration", Fortunato et al., 2017

PROBLEM: Most exploration strategies are not really efficient

⇒ Parametric noise added to model's weights to induce stochasticity in agent's policy

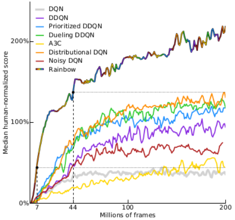
↳ Greatly increased performance on Atari benchmark

2013 Deep Reinforcement Learning
2015 Deep Q-Network
2015 Prioritized Experience Replay
2016 Double DQN
2016 Dueling DQN
2017 Distributional DQN
2017 Noisy DQN
2017 Rainbow

Rainbow:
"Rainbow: Combining Improvements in Deep Reinforcement Learning", Hessel et al., 2017

PROBLEM: Many improvements have been made on the DQN algorithm in previous years

$\Rightarrow$ Study these improvements and combine them in a single architecture



↳ Greatly increased performance on Atari benchmark

2013	Deep Reinforcement Learning
2015	Deep Q-Network
2015	Prioritized Experience Replay
2016	Double DQN
2016	Dueling DQN
2017	Distributional DQN
2017	Noisy DQN
2017	Rainbow
2019	R2D2

Recurrent Replay Distributed DQN (R2D2):
"Recurrent Experience Replay in Distributed Reinforcement Learning", Kapturowski et al., 2019

PROBLEM: In increasingly difficult partially observable domains, there is a need for more advanced memory-based approaches

⇒ Distributed Dueling Double DQN + Prioritized Experience Replay + LSTM to capture temporal information

↳ Outperforms (x4) previous methods on Atari benchmark, current state-of-the-art value-based approach

2015 DDPG

Deep Deterministic Policy Gradient (DDPG):

"Continuous control with deep reinforcement learning", Lillicrap et al., 2015

PROBLEM: DQN is limited to discrete action domains

⇒ Combine Actor-Critic with the recent successes of DQN: Policy network learned to maximize Q:

$$\max_{\theta} \mathbb{E}_{\tau}[Q_{\Phi}(s, \mu_{\theta}(s))]$$

DQN to find Q-function (MSBE minimization):

$$L(\Phi) = \mathbb{E}_{\tau}[(Q_{\Phi}(s,a) - (r + \gamma Q_{\Phi}(s', \mu_{\theta}(s'))))^2]$$

↳ Learns good policies for continuous actions
↳ Struggles to learn from raw pixels on some tasks
↳ Can overestimate Q-values, possibly leading to bad policy

2015 DDPG
2017 PPO

Proximal Policy Optimization (PPO):

"Proximal Policy Optimization Algorithms", Schulman et al., 2017

PROBLEM: Taking too large policy updates can lead to unstable training

$\Rightarrow$ Change the Policy Gradient objective and clip the loss values: Clipped surrogate objective:

$$L^{CLIP}(\theta) = \mathbb{E}[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$$

with $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}$

↳ Outperforms other policy gradients methods on robotic locomotion and Atari games
↳ Very simple algorithm, easy to implement

2015 DDPG
2017 PPO
2018 TD3

Twin Delayed DDPG (TD3):
"Addressing Function Approximation Error in Actor-Critic Methods", Fujimoto et al., 2018

PROBLEM: DDPG also overestimates Q-values

⇒ Use a combination of tricks to have a more stable algorithm:

- Double DQN and Clipped Double-Q trick
- Target networks (for policy and Q-functions)
- Delayed policy updates
- Target policy smoothing

↳ Reduces overestimation of Q-values
↳ Outperforms all previous policy-based algorithms

2015 DDPG
2017 PPO
2018 TD3
2018 Soft Actor-Critic

Soft Actor-Critic:

"Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor", Haarnoja et al., 2018

PROBLEM: Sample efficiency and meticulous hyperparameter tuning

$\Rightarrow$ Entropy-regularized RL:
Maximize expected reward AND entropy:

- Bonus reward proportional to entropy of policy
- + Double DQN with Clipped Double-Q trick
- + Target networks (for policy and Q-functions)

↳ Outperforms DDPG and PPO (similar to TD3)
↳ More stable w.r.t. random seeds

2016 AlphaGo

2017 AlphaGo Zero

PROBLEM: Leverage recent progress in deep RL to tackle the full-game of Go

AlphaGo:
"Mastering the game of Go with deep neural networks and tree search", Silver et al., 2016

⇒ Using value and policy networks combined with Monte Carlo tree search, learned with both supervised training and self-play

↳ First computer program to beat professional players in the full-sized game of Go

AlphaGo Zero:
"Mastering the game of Go without human knowledge", Silver et al., 2017

↳ Outperforms AlphaGo with only self-play

2016	AlphaGo
2017	AlphaGo Zero
2017	AlphaZero

AlphaZero:
"Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm", Silver et al., 2017

PROBLEM: Find a general solution for board games

$\Rightarrow$ Generalise AlphaGo Zero to learn Chess and Shogi

↳ Achieve superhuman level on both games in 24 hours of self-play

2016 AlphaGo
2017 AlphaGo Zero
2017 AlphaZero
2018 World Models

World Models:
"World Models", Ha and Schmidhuber, 2018

PROBLEMS:

- Human base their decisions and actions on a mental model of the world, itself based on their senses and predictions of the future
- Deep RL algorithms would benefit from using larger Neural Networks to learn rich representations of the world

$\Rightarrow$ Large Recurrent world model + small controller to act in world model

↳ Achieve good performance by training an agent entirely inside the simulated dream world

2016	AlphaGo
2017	AlphaGo Zero
2017	AlphaZero
2018	World Models
2018	PlaNet

PlaNet:
"Learning Latent Dynamics for Planning from Pixels", Hafner et al., 2018

PROBLEM: Learning the dynamics of high-dimensional environments is hard

⇒ Learn compact latent representations of high-dimensional environment fitted for multi-step prediction

↳ Outperforms model-free approaches on DeepMind control suite
↳ 200x more data efficient

2016 AlphaGo
2017 AlphaGo Zero
2017 AlphaZero
2018 World Models
2018 PlaNet
2019 MuZero

MuZero:
"Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model", Schrittwieser et al., 2019

PROBLEM: In real-world situations, a perfect simulator of the environment is not available

$\Rightarrow$ Learn a recurrent model to predict only quantities relevant to tree-based planning: reward, policy, value

↳ Achieves new state-of-the-art performance on Atari benchmark
↳ With no knowledge of game rules, match performance of AlphaZero in Go, Chess and Shogi

2016	AlphaGo
2017	AlphaGo Zero
2017	AlphaZero
2018	World Models
2018	PlaNet
2019	MuZero
2019	Dreamer

Dreamer:
"Dream to Control: Learning Behaviors by Latent Imagination", Hafner et al., 2019

PROBLEM: Model-based approaches can be shortsighted when using a finite imagination horizon

$\Rightarrow$ PlaNet to learn latent dynamics + Actor-Critic to predict actions and state values in learned latent space

↳ Outperforms previous model-based and model-free methods on DeepMind control suite (data efficiency, computation time and performance)

Plan2Explore:
"Planning to Explore via Self-Supervised World Models", Sekar et al., 2020

PROBLEM: RL algorithms are tend to be task-specific and have poor sample efficiency

$\Rightarrow$ PlaNet + Dreamer + Exploration induced as an intrinsic reward
$\Rightarrow$ Zero-shot or few-shot adaptation to downstream task

↳ Zero-shot performance competitive to Dreamer
↳ Few-shot matching or outperforming Dreamer
↳ Highly scalable and data-efficient approach

Value-based

Key Concept: Prioritized Experience Replay
Limitation: Discrete action space
Current SOTA: R2D2 (Kapturowski et al., 2019)

Policy-based

Key Concepts: Trust regions, Entropy regularized RL
Current SOTA: PPO (Schulman et al., 2017) / Soft Actor-Critic (Haarnoja et al., 2018)

Model-based

Key Concepts: Planning in latent imagination, Zero(Few)-shot(s) adaptation
Benefits: Data efficiency, Transfer learning, Increased computational budget $\Rightarrow$ Increased planning performance (Silver et al., 2017)
Current SOTA: MuZero (Schrittwieser et al., 2019) / Dreamer (Hafner et al., 2019)

Thank you!