Deep Reinforcement Learning
Introduction and State-of-the-art

Arjun Chandra
Research Scientist
Telenor Research / Telenor-NTNU AI Lab
arjun.chandra@telenor.com
@boelger

24 October 2017

https://join.slack.com/t/deep-rl-tutorial/signup
The Plan

• Some history
• RL and Deep RL in a nutshell
• Deep RL Toolbox
• Challenges and State-of-the-art
  • Data Efficiency
  • Exploration
  • Temporal Abstractions
  • Generalisation
Brief History

late 1980s

RL for robots using NNs, L-J Lin. PhD 1993, CMU

Gerald Tesauro

1995

2004

Stanford

2013 — Vlad Mnih et. al.

Google DeepMind

David Silver et. al.

2015 —

http://heli.stanford.edu/
Problem Characteristics

dynamic

uncertainty/volatility

uncharted/unimagined/
exception laden

delayed consequences

requires strategy
Solution

machine with **agency** which **learn**, **plan**, and **act** to find a strategy for solving the problem autonomously to some extent probe and **learn from feedback** focus on the **long-term objective** explore and **exploit**
Reinforcement Learning

- **Observation** and **feedback** on actions

**Problem/Environment**

**Agent**
- **Goal**
- **Model**
- **\(\pi/Q\)**

**Agent**
- **Goal**
- **Model**
- **\(\pi/Q\)**

**Goal**
- Maximize return \(E\{R\}\)

**Model**
- Dynamics model

**\(\pi/Q\)**
- Policy/value function

**Diagram**

- **Action**
The MDP game!

Inspired by Prof. Rich Sutton's tutorial: https://www.youtube.com/watch?v=gqgnxyjaKe4

Goal

maximise return $E\{R\}$
The MDP (S,A,P,R,Γ)

R: immediate reward function R(s, a)
P: state transition probability P(s'|s, a)

R=10±3
P=1.00

R=-10±3
P=0.01

R=20±3
P=0.01

R=40±3
P=0.99

https://github.com/traai/basic-rl
Terminology

- state or action
- value function
- policy
- dynamics model
- reward
- goal

home
Terminology

state or action
value function
$Q(s,a)$ $V(s)$
policy
dynamics model
reward
goal

home

Q
V
Q
Q
**Terminology**

| state or action value function | policy \(\pi(s|a)\) | dynamics model | reward | goal |
|--------------------------------|---------------------|----------------|--------|------|

**Home**
<table>
<thead>
<tr>
<th>Terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td>state or action value function</td>
</tr>
<tr>
<td>policy</td>
</tr>
<tr>
<td>dynamics model</td>
</tr>
<tr>
<td>reward</td>
</tr>
<tr>
<td>goal</td>
</tr>
</tbody>
</table>

If I go South, I will meet home.
Terminology

- state or action
- value function
- policy
- dynamics model
- reward
- goal

home
Terminology

- state or action
- value function
- policy
- dynamics model
- reward
- goal
Deep Reinforcement Learning

- Observation and feedback on actions
- Maximise return $E\{R\}$
- Model dynamics model $\pi/Q$
- Agent
- Problem/Environment
Deep Reinforcement Learning

Sensors → Perception → World Model → Planning → Control → Action

pixels → vision/detection → prediction/physics sim/kinematics → motion planner → low level controller set torques → motor

abstractions ~ info loss (manual craft)

Sensors → Deep Neural Networks (abstractions/representation adapted to task) → Action
Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car, Bojarski et. al.,
2017

data mismatch
Toolbox

Standard algorithms to give you a
flavour of the norm!
Human-level control through deep reinforcement learning,
Mnih et. al., Nature 518, Feb 2015
experience replay buffer

**save** transition in memory

randomly **sample** from memory for training = i.i.d
\[ (r + \gamma \max_{a'} Q(s', a', w^-) - Q(s, a, w))^2 \]
prioritised experience replay

sample from memory based on surprise

\[ r + \gamma \max_{a'} Q(s', a', w^-) - Q(s, a, w) \]

Prioritised Experience Replay, Schaul et. al., ICLR 2016
dueling architecture

\[ Q(s, a) = V(s) + A(s, a) \]

*Dueling Network Architectures for Deep RL* Wang et. al., ICML 2016
however
training is
SLOOOOo....W
Parallel Asynchronous Training

value and policy based methods

Asynchronous Methods for Deep Reinforcement Learning, Mnih et. al., ICML 2016
Parallel learners

HOGWILD!

Updates

Shared params

Agent

Copy

Copy

Copy

Copy

Copy

Copy

Copy

Copy

https://github.com/traai/async-deep-rl
So 2016…
Can we train even faster?
PAAC
(Parallel Advantage Actor-Critic)

Efficient Parallel Methods for Deep Reinforcement Learning,
A. V. Clemente, H. N. Castejón, and A. Chandra, RLDM 2017

https://github.com/alfredvc/paac

1 GPU/CPU
Reduced training time
SOTA performance

Alfredo Clemente
Challenges and SOTA

Data Efficiency
Exploration
Temporal Abstractions
Generalisation
Data Efficiency
Demonstrations

Observation and feedback on action

Past observations, action, feedback

Action

Learning from Demonstrations for Real World Reinforcement Learning, Hester et. al., arXiv e-print, Jul 2017
Deep RL with Unsupervised Auxiliary Tasks

Use replay buffer wisely

observation and feedback on actions

Buffer

Agent

Goal

Problem/Environment

action

Reinforcement Learning with Unsupervised Auxiliary Tasks,
Jaderberg et. al. ICML 2017
Reinforcement Learning with Unsupervised Auxiliary Tasks, Jaderberg et al. ICML 2017
learn to act to affect pixels

e.g. if grabbing fruit makes it disappear, agent would do it
predict
short term reward
e.g. replay pick key
series of frames
predict long term reward
10x less data!

Reinforcement Learning with Unsupervised Auxiliary Tasks, Jaderberg et. al. ICML 2017
https://deepmind.com/blog/reinforcement-learning-unsupervised-auxiliary-tasks/
A Distributional Perspective on Reinforcement Learning,
Bellemare et. al., ICML 2017
Normal DQN target:
[sample reward after step + discounted previous return estimate from then on]

BUT this:
[fuse R with discounted previous return distribution]

\[
P^\pi Z
\]

\[
R + \gamma P^\pi Z
\]

\[
\Phi T^\pi Z
\]

A Distributional Perspective on Reinforcement Learning, Bellemare et. al., ICML 2017
Figure 4. Learned value distribution during an episode of SPACE INVADERS. Different actions are shaded different colours. Returns below 0 (which do not occur in SPACE INVADERS) are not shown here as the agent assigns virtually no probability to them.

“If I shoot now, it is game over for me”
A Distributional Perspective on Reinforcement Learning, Bellemare et. al., ICML 2017
A Distributional Perspective on Reinforcement Learning, Bellemare et. al., ICML 2017
Exploration
Curiosity Driven Exploration

observation and feedback on action

Agent

Model

Goal

NN

action
Curiosity Driven Exploration

---

Curiosity-driven Exploration by Self-supervised Prediction,
Pathak, Agrawal et al., ICML 2017.
Curiosity Driven Exploration by Self-Supervised Prediction

ICML 2017

Deepak Pathak, Pulkit Agrawal, Alexei Efros, Trevor Darrell
UC Berkeley

https://github.com/pathak22/noreward-rl
https://pathak22.github.io/noreward-rl/
Temporal Abstractions
HRL with pre-set Goals


pre-defined goal
selected by meta-controller
FeUdal Networks for Hierarchical Reinforcement Learning

Manager tries to find good directions.

Worker tries to achieve them.

FeUdal Networks for Hierarchical Reinforcement Learning, Vezhnevets et al. ICML 2017
FeUdal Networks for Hierarchical Reinforcement Learning, Vezhnevets et. al. ICML 2017
Generalisation
Meta-learning (Learn to Learn)

Versatile agents!

Transfer learning works with images

Good features for decision making?

learn to go East

learn to reduce learning time to go to X

MAML

0 gradient steps

0 grad/opt step: policy ready to learn

MAML

0 gradient steps

1 grad/opt step: learnt to achieve goal

http://bair.berkeley.edu/blog/2017/07/18/learning-to-learn/
Code: https://github.com/cbfinn/maml_rl
Videos: https://sites.google.com/view/maml
Domain Randomisation

Generalising from Simulation
Sim-to-Real Transfer of Robotic Control with Dynamics Randomization, Peng et al. arXiv preprint, 18 Oct 2017

https://blog.openai.com/generalizing-from-simulation/
Generalisation via Self-play
Deep RL in **AlphaGo Zero**

Improve thinking and intuition with **feedback from self-play** [zero human game data]

Very High Level Mechanics

\[ \begin{align*}
&\text{[Xt, Yt, Xt-1, Yt-1, ..., Xt-7, Yt-7, C]} \\
&\text{residual block of conv layers} \\
&\text{[39 to 79 layers]} \\
&\text{+ p and v heads} \\
&\text{[2 layers, 3 layers]} \\
&\text{play to the end} \\
&\text{z}
\end{align*} \]
Self-play to end of game

NN training: learn to evaluate

Self-play step: select move by simulation + evaluation

AlphaGo Zero
Discovering new knowledge

https://deepmind.com/blog/alphago-zero-learning-scratch/
https://www.youtube.com/watch?v=WXHFqTvfFSw
Inspired to study RL much?

Next lecture:
Building Blocks of (Deep) RL
November 8, 2017

https://join.slack.com/t/deep-rl-tutorial/signup