A Deep Hierarchical Approach to Lifelong Learning in Minecraft

SHAHAR GIVONY, CHEN TESSLER
SUPERVISORS: DANIEL MANKOWITZ, TOM ZAHAVY, SHIE MANNOR
LAB ENGINEER: KOBY KOHAI
MACHINE LEARNING

Introduction

Google Translate

Object Detection

CAT, DOG, DUCK

Control Robotics and Machine Learning Laboratory
MACHINE LEARNING

Introduction
A Deep Hierarchical Approach to Lifelong Learning in Minecraft

* equally contributed
Technion Israel Institute of Technology, Haifa, Israel
chen.tessler, shahargiv, tomzahavy {@campus.technion.ac.il }, danielm@tx.technion.ac.il, shie@ee.technion.ac.il

Cited 21 times (so far)
LIFELONG LEARNING DEFINITION

Introduction

Systems Approach
1. Efficiently learn multiple tasks
2. Transfer knowledge to new tasks

Selective Transfer
Temporal/Spatial

Agent

Knowledge Retention

Shared Knowledge base
Introduction

MINECRAFT
REINFORCEMENT LEARNING

Introduction
DQN – DEEP Q NETWORK
DQN – DEEP Q NETWORK

\[ Q(s_t, a_t) = r_t(s_t, a_t) + \sum_{i=t+1}^{T} \gamma^{i-t} \cdot r_i(s_i, a_i) \]

Immediate Future

GOAL

Given reward

Learned Q function
DQN – DEEP Q NETWORK

Introduction

argmax $\pi \left( Q(s_t, a) \right)$

$Q(s_{foward})$
$\vdots$
$Q(s_{left})$
MINECRAFT - CHALLENGES

Introduction
PROBLEM DEFINITION

“Perform Lifelong Learning using Hierarchical Deep RL in a simulated ‘real-world’ environment”
HIERARCHICAL LIFELONG LEARNING

Complex lifelong problem
Overview

PART 1
Learn a skill

PART 2
Solve a lifelong learning problem
LEARNING IN MINECRAFT

Part 1

Minecraft

Socket Interface

JAVA
- Domain design
- Reward modeling
- Create API

DQN

LUA
- Configuration optimization
- Simulate
- Algorithmic changes
Part 1

DEEP SKILL NETWORK (DSN)

Pickup

Break / Place

Navigate
Part 1

DSN - RESULTS

1. Pickup
   - Success rate (%)
   - Max ≈ 1

2. Break / Place
   - Success rate (%)
   - Max ≈ 1

3. Navigation
   - Success rate (%)
   - Max ≈ 0.9
Overview

PART 1
Learn a skill

PART 2
Use skills to solve lifelong learning
HIERARCHICAL DEEP RL NETWORK (HDRLN)
HDRLN FRAMEWORK

Primitive Actions

Primitive Actions
HIERARCHICAL DEEP RL NETWORK (HDRLN)

\[
\begin{align*}
\text{arg\thinspace max}_{a \in A} Q(s, a) &= \text{Pickup} \\
\text{arg\thinspace max}_{a \in A} Q(s, a) &= \text{Forward}
\end{align*}
\]

Part 2
THE DOMAIN

Navigation 2 Domain

Pickup Domain

Placement Domain

Navigation 2 Room

Complex Domain

Placement Room

Start

Exit

Exit

Goal

Goal

Object to place

Pickup object

Part 2
Set the time to 1000
Changing to clear weather
HDRLN RESULTS

![HDRLN Comparison Chart](chart.png)

- **Mission success [%]**
- **Epoch**
- **HDRLN Comparison**
  - DQN-DoubleQ
  - HDRLN
  - HDRLN-DoubleQ
ANALYSIS

Results
SUMMARY

Part 1
• Proof of concept
• Deep Skill Network
  • Knowledge retention

Part 2
• HDRLN
  • Selective transfer by learning to select skills
  • Effective knowledge retention using the Distilled multi-skill network
CONTRIBUTIONS

- First API with Minecraft for Deep RL
- First work to utilize Hierarchical Deep RL
- First implementation SMDP (skills) loss over Deep RL

- Created interest in Academia & Industry
- Collaborations
WHAT’S NEXT

• Automatic skill learning
• Automatic skill refinement
QUESTIONS?
MORE INFORMATION - REFERENCE

Our descriptive website
http://chentessler.wixsite.com/hdrlminecraft

Google Deepmind’s DQN Algorithm

Burlapcraft – Machine learning environment for Minecraft Forge
http://h2r.cs.brown.edu/announcing-burlapcraft/

Nature publication of the DQN
http://www.nature.com/nature/journal/v518/n7540/full/nature14236.html

Microsoft project Malmo
http://blogs.microsoft.com/next/2016/03/13/project-malmo-using-minecraft-build-intelligent-technology/
MORE INFORMATION - REFERENCE

Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning, Sutton at al


Policy distillation, Google DeepMind


Deep Reinforcement Learning with Double Q-learning, Google DeepMind

REINFORCEMENT LEARNING

State of the art results
- Autonomous vehicles
- Robotics
REINFORCEMENT LEARNING

Introduction

Transition tuples: \((s_t, a_t) \rightarrow (s_{t+1}, r_t)\)

\[
Q_t(s_t, a_t) = r_t(s_t, a_t) + \sum_{i=t+1}^{T} \gamma^{i-t} \cdot r_i(s_i, a_i)
\]

Immediate \quad Future

\[
\pi(s_t) = \arg\max_{a \in A} Q(s_t, a)
\]
FUNCTION APPROXIMATION

DQN – Deep Q Network

\[ \bar{Q}(\bar{s}, a_1) \]

\[ \bar{Q}(\bar{s}, a_2) \]
SUCCESS MEMORY

• Prioritize success trajectories
• Smaller memory (1% = 1K states)
Press ‘E’ to open your inventory.
IJCAI workshop review
HIERARCHICAL LIFELONG LEARNING

Part 2

Plan
Navigate
Gather wood

Navigate
Build
Gather wood
SKILLS FRAMEWORK

Main agent:
• Primitive actions (forward, turn left, turn right, ...)
• Skills

Skills:
• Temporally extended actions / options / macro actions
Observation = \((s_t, a_t, \{r_t, \ldots, r_{t+n}\}, S_{t+n}, T_{t+n-1})\)

\[
Q(s_t, a_t) = \begin{cases} 
\sum_{i \in [t, t+n-1]} \gamma^{i-t}r_i, & T_{t+n-1} = \text{terminal} \\
\sum_{i \in [t, t+n-1]} \gamma^{i-t}r_i + \gamma^n \cdot \underset{a \in A}{\operatorname{argmax}} Q(s_{t+n}, a; \theta), & \text{else}
\end{cases}
\]
HIERARCHICAL DEEP RL NETWORK (HDRLN)
SKILL DISTILLATION
DISTILLED MULTI-SKILL NETWORK

Teacher Networks

Pickup  Navigate 2  Break  Place

Student Network

Pickup  Navigate 2  Break  Place
STRUCTURE

The Problem → Our Solution

Alternative Approaches

Minecraft → DQN → DDQN

Skills

HDRLN

Distilled Network
OUR WORK

Adapt BURLAP mod for Minecraft

- Domain design
- Reward modeling
- Allow external control

Connect Minecraft to DQN

DQN hyper parameter optimization

Success memory

Simulate
## DQN Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Original value</th>
<th>Modified value</th>
</tr>
</thead>
<tbody>
<tr>
<td>n-replay</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Replay memory</td>
<td>1,000,000</td>
<td>100,000</td>
</tr>
<tr>
<td>Eps endt</td>
<td>Replay memory (1,000,000)</td>
<td>400,000</td>
</tr>
<tr>
<td>Lr</td>
<td>0.00025</td>
<td>0.0025</td>
</tr>
<tr>
<td>Eval freq</td>
<td>250,000</td>
<td>5,000</td>
</tr>
<tr>
<td>Eval steps</td>
<td>125,000</td>
<td>500</td>
</tr>
</tbody>
</table>
Double Q Learning (DDQN)

- Unbiased

\[ Q(S_t, a_t) = \begin{cases} 
    r_t, & T_t = \text{terminal} \\
    r_t + \gamma \cdot Q(S_{t+1}, \operatorname{arg\,max}_{a \in A} Q(S_{t+1}, a; \theta), \theta^-), & \text{else} 
\end{cases} \]
OUR DDQN RESULTS

Double Q Learning Comparison

Mission success [%]

Epoch

0 20 40 60 80 100 120 140 160 180 200 220 240

DQN Double DQN
POLICY DISTILLIATION

- Multiple teachers, single student = shared representation

Table B2: Performance of multi-task distilled agents on 3 Atari games. Best relative scores are outlined in bold.

<table>
<thead>
<tr>
<th></th>
<th>DQN</th>
<th>Multi-DQN</th>
<th>Multi-Dist-NLL</th>
<th>Multi-Dist-KL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freeway</td>
<td>25.8</td>
<td>23.3</td>
<td>90.3</td>
<td>26.5</td>
</tr>
<tr>
<td>Pong</td>
<td>16.2</td>
<td>12.0</td>
<td>74.1</td>
<td>14.8</td>
</tr>
<tr>
<td>Q*bert</td>
<td>4589.8</td>
<td>3987.3</td>
<td>86.9</td>
<td>5678.0</td>
</tr>
<tr>
<td>Geometric Mean</td>
<td></td>
<td>83.5</td>
<td></td>
<td>105.1</td>
</tr>
</tbody>
</table>
HDRLN RESULTS

HDRLN Comparison

Mission success [%]

Epoch

- HDRLN-DoubleQ
- HDRLN-Distilled-DoubleQ