StarCraft AI Research Summary

Qinyu Chen

henryqychen@gmail.com,
BeiJing, China

1 Why StarCraft?

StarCraft has proven to be a challenging environment for artificial intelligence research. While hand-crafted strategies are still the state-of-art, the deep networks approach is able to express a wide range of different strategies and thus improving the networks performance with deep reinforcement learning is an immediately promising avenue for future research.[1]

1.1 Advantages

1. Large-scale game replays data in the community
2. Classic RTS game with perfect designed balance, competitiveness and purity

1.2 Challenges

1. Enormous Search Space(10 times complexer than *Go* of game states and actions)
2. Partially observed environment
3. Memory preserving agent
4. Inference both in time and space
5. Multi-agent corporation
6. High time efficiency

2 Online Resources

2.1 Replays

GosuGamers [1], *ICCup* [2], *TeamLiquid* [3]

2.2 Competitions

AIIDE StarCraft AI Competition [4], *CIG StarCraft RTS AI Competition* [5], *Student StarCraft AI Competition* [6]

[1] http://www.gosugamers.net/
[2] https://iccup.com/
[3] http://www.teamliquid.net/
[4] https://sites.google.com/view/aiide2017/
[5] https://ls11-www.cs.tu-dortmund.de/rts-competition/start
[6] https://sscaitournament.com/

3 Game AI

3.1 Thread in game AI

Usually there is only one thread to handle AI logic, otherwise there could be multi-threaded for computation and one thread for rendering.

3.2 Difficulty level of game AI design

The most difficult AI design is for sports game, which is very similar to robotics

$$MMO < FPS < RTS < Sports \tag{1}$$

3.3 Traditional game AI methods

State Machine

Hierarchical State Machine

Behavior Tree

4 RTS Game AI

Real-time strategy(RTS) games have historaically been a domain of interest of the planning and decision making research communities.

4.1 Macromanagement AI

Or called General's AI which determine the time series of constructing buildings, conducting research, and producing units, among other things involving the intake and expending of resources. This is actually a form of micromanagement done to a relatively large number of units

Rule-based methods Implemented by if-else.

1. TODO

Advanced methods

1. planner
2. scoring system, power map
3. machine learning methods

4.2 Agent AI

Or called Soldier's AI, human player's micro-operation.

State Machine Behaviors will be triggered in certain condition(environment, resource, time)

Behavior Tree Theoretically, the behavior tree is equivalent to state machine but could reduce the complexity of control logic.
Practically, behavior tree need only 1/3 lines of more readable code comparing with the same logic implemented by FSM.

Hierarchical State Machine This is a traditional and efficient method to reduce the complexity of game AI design.

5 Modeling RTS Game

5.1 Reinforcement Learning Framework

Reinforcement Learning (RL) is an area of machine learning inspired by behaviourist psychology, concerned with how agents should take actions in an environment so as to maximize some notion of cumulative reward.
There are four main reseasons make RL quite different from other machine learning paradigms:[2]

1. There is no supervisor, only a reward signal
2. Feedback is delayed, not instantaneous
3. Time really matters (sequential, non i.i.d data)
4. Agents actions affect the subsequent data it receives

RL focus on online performance, which involves finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge). The $E\&E$ trade-off in RL has been most thoroughly studied through the multi-armed bandit (MAB) problem and in finite MDPs.

Table 1. Formal Definition of Reinforcement Learning[2]

Symbols	Explanation
a	agent
e	environment
t	timestamp, increments at each step
O_t^a	agent's observation(input) from environment at step t
A_t^a	agent's actions(ouput) to environment at step t
R_t^a	agent's reward from environment at step t, is a scalar feedback signal
S_t^a	agent's state at step t
S_t^e	environment's state at step t

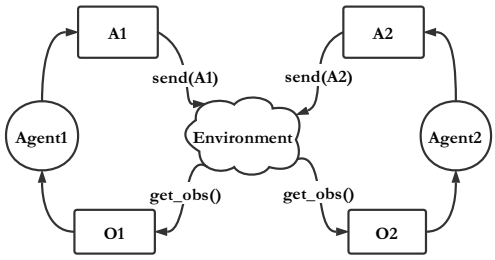
5.2 Inperfect Information Game

In full observability game, agent directly observes the environment stat, where:

$$O_t^a = S_t^a = S_t^e \tag{2}$$

As mentioned in 1.2.2, because of the game mechanisms: *fog-of-war* and confusing magic(invisibility, hallucination, etc.), agents receive inperfect information from partially observed environment, which leads more challenging AI design than *Chess* and *Go*.

Fig. 1. 1v1 Adversarial RL Framework



where:

$$S_{1,t}^a \neq S_{2,t}^a \tag{3}$$

$$S_{1,t}^a \cup S_{2,t}^a = S_t^e \tag{4}$$

6 Developing Environment

6.1 Tools

BWAPI The Brood War Application Programming Interface(BWAPI) is a free and open source C++ framework used to interact with Starcraft: BroodWar®. Researchers can create AI agents to play the game.[3]

Torch *Torch* is a scientific computing framework with wide support for machine learning algorithms that puts GPUs first. Torch is mainly implemented by *LuaJIT*.[4]

TorchCraft *TorchCraft* is a bridge between Torch and StarCraft.[5][6]

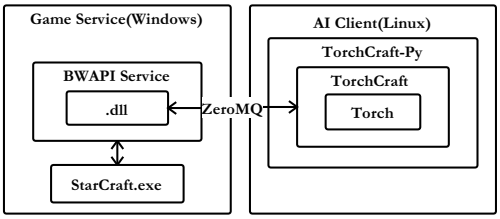
TorchCraft-Py A Python wrapper for TorchCraft developed by Alibaba.[7]

gym An open source reinforcement framework

gym-starcraft

6.2 Architecture

Fig. 2. TorchCraft-based Client-Server Architecture



6.3 BWAPI

Client

ServerState

Frame

Order

Unit

Resources

Bullet

Action

7 Macromanagement AI

7.1 Build Order Planning

Search Problem the goal is to find the build order that optimizes a specific heuristic.

Notes and Comments.

7.2 Dataset

Format Replay files are in binary format and require preprocessing before knowledge can be extracted.

7.3 Learning from replays

7.4

Algorithm 2: INSERTION-SORT(A)

```
1  for j = 2 to A.length
2      key = A[j]
3      // Insert A[j] into the sorted sequence A[1..j-1].
4      i = j - 1
5      while i > 0 and A[i] > key
6          A[i+1] = A[i]
7          i = i - 1
8      A[i+1] = key
```

8 Micromanagement AI

8.1

References

1. Niels Justesen and Sebastian Risi. Learning macromanagement in starcraft from replays using deep learning. *CoRR*, abs/1707.03743, 2017.
2. David Silver. Intruduction to reinforcement learning. `http://101.96.8.165/www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/intro_RL.pdf`, 2015.
3. `https://github.com/bwapi/bwapi`.
4. `http://torch.ch`.
5. `https://github.com/TorchCraft/TorchCraft`.
6. Gabriel Synnaeve, Nantas Nardelli, Alex Auvolat, Soumith Chintala, Timothée Lacroix, Zeming Lin, Florian Richoux, and Nicolas Usunier. Torchcraft: a library for machine learning research on real-time strategy games. *CoRR*, abs/1611.00625, 2016.
7. `https://github.com/alibaba/torchcraft-py`.