Final Year Project: Gamma-Star-2

Eoghan Hogan
Data Science

October 18, 2020

1 Problem Statement

For my Final Year Project I would like to use StarCraftII (SC2) To explore interesting questions in Reinforcement Learning such as comparing the effectiveness of different learning algorithms, Memory Replay, the use of transfer learning, using a curriculum to guide learning, or using self-play to train agents.

This will happen using the python wrappers made around Blizzards SC2 protocols. The Challenges I will use to Train and Test the Agents will be Micromanagement Tasks in SC2 instead of trying to play a whole game due to complexity. The Tasks will be of my own devising and from the SC2 Multi-Agent Challenge.

2 Background

Reinforcement Learning is one of the most exciting areas of current Artificial Intelligence Research and Development. Many Real-World applications require Artificial Agents to compete and coordinate with other agents in complex environments.

The domain of StarCraft2 has emerged as an important challenge for Artificial Intelligence Research[6], owing to its iconic and enduring status among the most difficult professional ESports and its relevance to the real world in terms of its raw complexity and Virtually Infinite ways to play.

3 Related Work

Note: *On arxiv alone there have already been 40+ papers referencing SC2 since the start of 2020 alone.* [4]

The biggest paper that has influenced me is the Alpha Star paper [7]. Their Research was to produce a system that could play Professional SC2 players and win. While I am completely Drawn to the Idea of replicating their research It is beyond my scope.

One area of particular Interest is Learning from "Experience"[8]. That is using real Game play from Humans to Train the Agent. I believe this could be very useful as the SC2 community has Replays [1] of games at all levels of play since it came out giving me a treasure trove of data to use.

One key component of SC2 is balancing out Micro and Macro and while I have not outlined the Problem statement to involve building a full scale Agent. I have looked at work relating to ways to train full agent. While I mentioned "Experience" above another thing to consider is architecture [5] of the Agent.

A big part of macro is build order and learning a build order based on what is going on in the game. Learning build order alone is a challenge in itself and as such is in itself an area of research[9].

Another field of work is a hierarchical approach, where the hierarchy involves two levels of abstraction. One is the macro-actions extracted from expert's demonstration trajectories, which can reduce the action space in an order

of magnitude yet remains effective. The other is a two-layer hierarchical architecture, which is modular and easy to scale [10]

4 Data Sets

Replays

- https://liquipedia.net/starcraft2/OlimoLeague
- http://gggreplays.com
- https://lotv.spawningtool.com
- http://sc2.replays.net
- https://drop.sc

5 Bibliography

References Below

6 Personal Contribution

As a personal contribution I will be building a wrapper around PySC2 to emulate smac[2] so that I can have multiple challenges for the Agent.

I will be implementing a Reinforcement Learning[17] Agent and test a variety of methods [18].

The main contenders I wish to look at are Reinforcement Learning[18]. Deep Reinforcement Learning [11] [12].Q-Learning [12], Deep Q Learning, Dueling Networks [13], and Learning from Replays [15] [8].

I wish to Compare and Contrast these methods on SC2 to evaluate which would give rise to the most intelligent Agent and possibly which agent could play a full game.

References

[1] https://liquipedia.net/starcraft2/Replay_Websites

[2] https://github.com/oxwhirl/smac

[3] https://github.com/deepmind/pysc2

[4] https://arxiv.org/search/?query=starcraft2&searchtype=all&source=header

[5] ennis Lee, Haoran Tang, et. al.Modular Architecture for StarCraft II with Deep Reinforcement Learning. *Proceedings of the Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference* (AIIDE 2018)

[6] Oriol Vinyals, Timo Ewalds Sergey Bartunov, et. al. *StarCraft II: A New Challenge for Reinforcement Learning.* arXiv:1708.04782

[7] Vinyals, O., Babuschkin, I., Czarnecki, W.M. *et al.* Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature **575**, 350–354 (2019). https://doi.org/10.1038/s41586-019-1724-z

[8] Zhao, Dongbin and Wang, Haitao and Shao, Kun and Zhu, Yuanheng (2016). Deep reinforcement learning with experience replay based on SARSA. doi 10.1109/SSCI.2016.7849837

[9] Zhentao Tang, Dongbin Zhao, and Yuanheng Zhu. *Reinforcement Learning for Build-Order Production in StarCraft II* Chinese Academy of Sciences University of Chinese Academy of Sciences, Beijing, China 100190

[10] Zhen-Jia Pang, et. al. *On Reinforcement Learning for Full-length Game of StarCraft.* National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

[11] Vincent François-Lavet, et. al. An Introduction to Deep Reinforcement Learning https://arxiv.org/pdf/1811.12560.pdf

[12] Volodymyr Mnih, et. al. Playing Atari with Deep Reinforcement Learning https://arxiv.org/pdf/1811.12560.pdf

[13] Ziyu Wang, et. al. Dueling Network Architectures for Deep Reinforcement Learning https://arxiv.org/abs/1511.06581

[14] Andrei A. Rusu, et. al. Progressive Neural Networks https://arxiv.org/abs/1606.04671v3

[15] Marcin Andrychowicz, et. al. Hindsight Experience Replay https://arxiv.org/abs/1707.01495v3

[16] Greg Wayne, et. al. Unsupervised Predictive Memory in a Goal-Directed Agent https://arxiv.org/abs/1803.10760

[17] Richard S. Sutton, et. al. Policy Gradient Methods for Reinforcement Learning with Function Approximation https://papers.nips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-functio pdf

[18] Csaba Szepesv ari. Algorithms for Reinforcement Learning https://sites.ualberta.ca/~szepesva/papers/RLAlgsInMDPs.pdf