Udacity Machine Learning Capstone project  
Dennis Rossman

8/17/2019

# Overview

## Domain Background

|  |  |
| --- | --- |
|  | StarCraft Agent  It was a little hard to settle on a project since there are so many cool things to work on, but I tried to settle on a project that was challenging but constrained enough to be done in one month. My three top ideas were: manufacturing scheduler, StarCraft agent and self-driving car. I dropped the manufacturing scheduler because I couldn’t find a ready-made manufacturing environment simulator and I didn’t think I had time to create the simulator and do the machine learning model to improve things like bottle neck management. I was torn between the StarCraft agent and the self-driving car. Both have good environments, and I was excited about Amazon’s DeepRacer, but it doesn’t look like you can currently train the DeepRacer on your own machine and are tied to AWS. I thought that might make the project difficult to evaluate, so I unfortunately dropped it. That left the StarCraft agent. I have loved reading about DeepMinds progress in coming up with their StarCraft agent, I wanted to take a shot at it. |

## Problem Statement

|  |  |
| --- | --- |
|  | Create an agent using the StarCraft environment and the PySC2 library that can use reinforcement learning to play the game. |

## Datasets and Inputs

|  |  |
| --- | --- |
|  | The dataset will be coming from the StarCraft game using the pysc2 library. I am not currently planning on using any other external datasets currently. |

## Solution Statement

|  |  |
| --- | --- |
|  | I would love to say that my agent will be able to take on world class professionals, but I think I would be happy if the agent can beat the internal StarCraft game AI on the hard setting. What I want to try is to first create a basic agent that can complete the game. My stretch goal is to come up with a hierarchical reinforcement model where the lower level models handle basic tasks (i.e. collecting resources) and the higher-level model handles scheduling the lower agents depending on the game state (whether enemies are attacking). |

## Benchmark Model

|  |  |
| --- | --- |
|  | DeepMind is obviously an impressive benchmark that I think would be hard to beat by a student in a month, so right now the benchmark is just being able to beat the internal game’s AI. This will be easy to measure than having the agent play against a person since different people will have wildly different abilities. |

## Evaluation Metrics

|  |  |
| --- | --- |
|  | The two metrics that I am planning to use is the ability of the agent to beat the game on different levels of the games internal AI (easy up to hard). Also, I would like to track the amount of time that it takes the agent to win. |

## Project Design

|  |  |
| --- | --- |
|  | Step 1 – get the StarCraft environment installed and interfaced with the PySC2 library  Step 2 – Create a basic agent and train it through reinforcement learning with the starting task of being able to collect basic materials and expand the base  Step 3 – Continue training the agent until it can beat the game AI on easy  Step 4 – Improve the agent to beat the game on medium and then hard AI  Step 5 – Modify the agent to use hierarchical reinforcement learning so that the low level models will handle things like collecting resources and building new structures and the high level model will coordinate the lower level models. |

## Presentation

|  |  |
| --- | --- |
|  | I hope to be able to present:   1. A completed hierarchical reinforcement learning agent that is able to complete the game against the hard AI. 2. An agent that a person can play against that is as good or better than the game AI. |