University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

MAgent RL

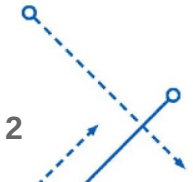
Tyler Perison

CSE 410: Reinforcement Learning

December 8, 2020

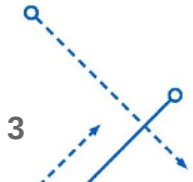
Project Description & Background

- PettingZoo MAgent environments:
  - Battle
  - Battlefield
- Key features of MAgent environments used:
  - Completely individual reward system
  - Mixed cooperative-competitive
  - Premature death of agents
  - Large observation sizes (13x13x41 = 6929)
  - Large gridworld (45x45)
  - Many agents (24 or 162)

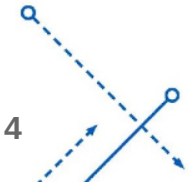


Implementation

- Simple Gridworld:
  - Q-Learning
  - Cooperative
  - Large observation space (7x7x7x7 = 2401)
  - Combined rewards
  - Alternate which Q-Table gets updated
  - Colab Widgets for visuals
- MAgent Gridworld:
  - Parallel environment configuration
  - DQN
  - MSE loss function
  - Individual rewards
  - Configurable number of Q-Networks
  - Configurable number of agents per network
  - Matplotlib animations for visuals

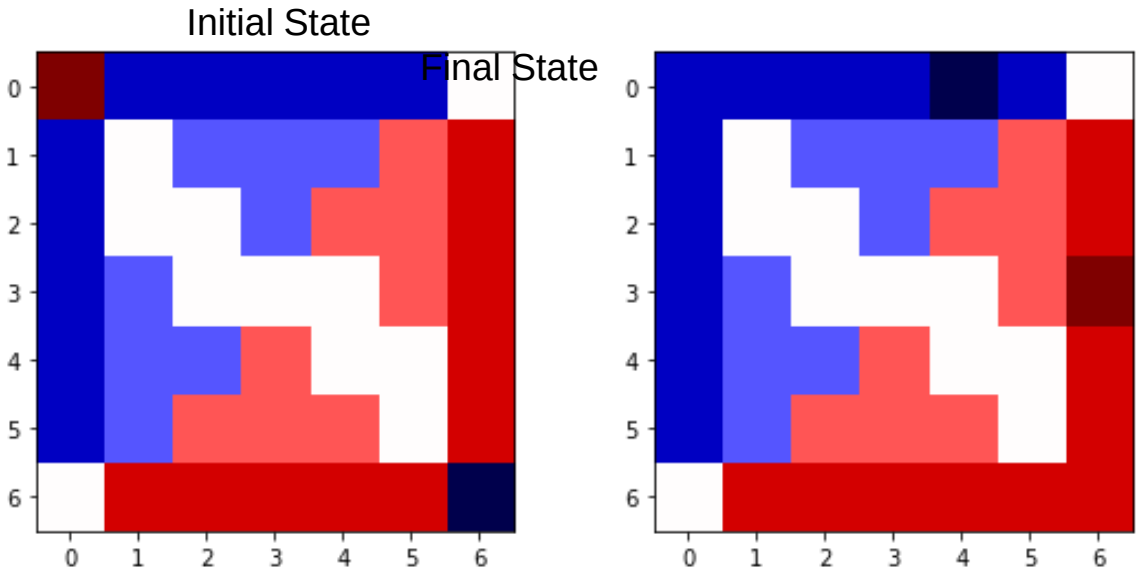


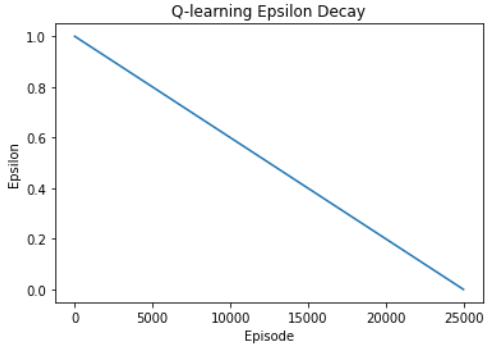
Demo

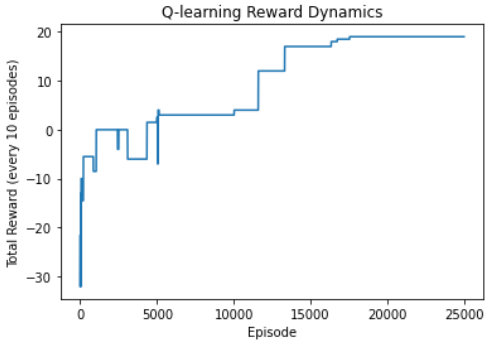


Simple Gridworld Results

- Runtime: ~ 1 minute 30 seconds for 25000 episodes

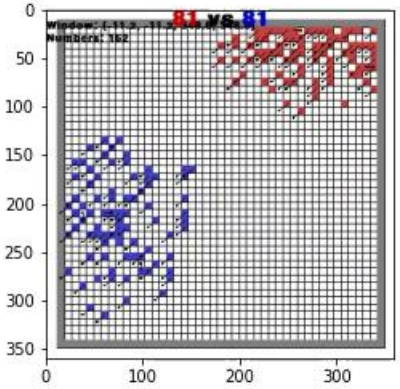


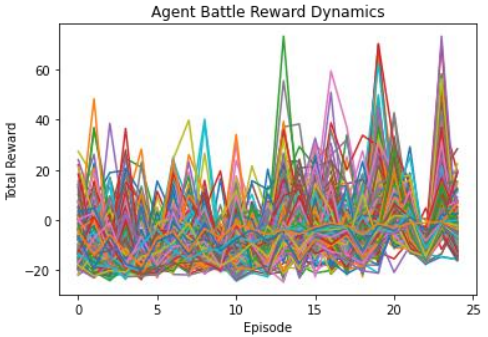


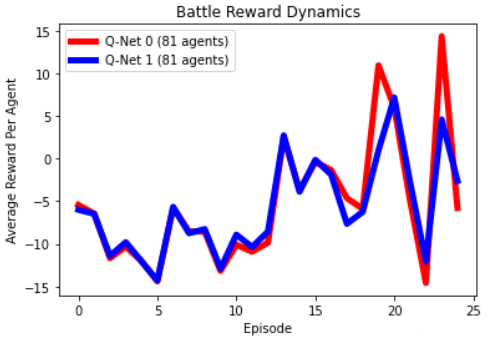


MAgent Battle Results

- Runtime (steps maxed at 500):
  - Synchronous: ~1 minute 45 seconds per episode
  - Batch: ~3 minutes per episode

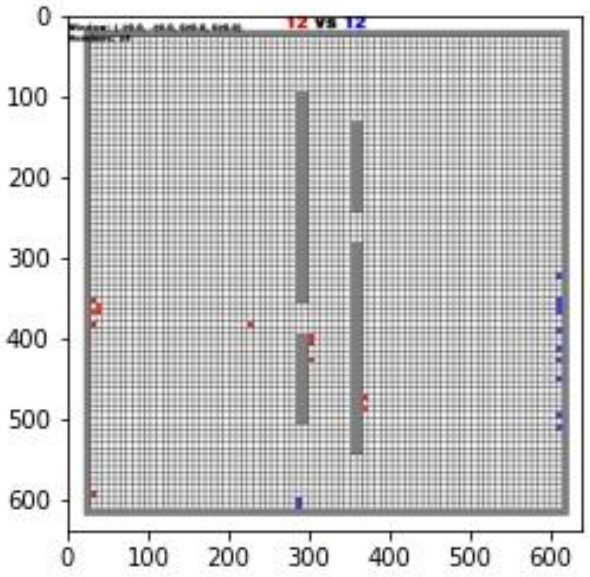


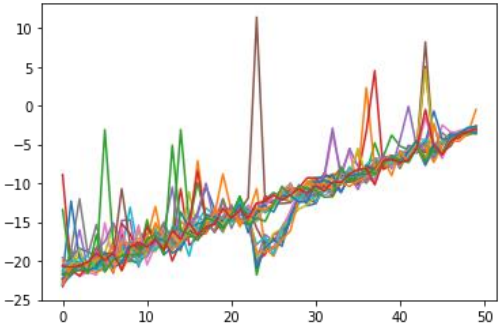


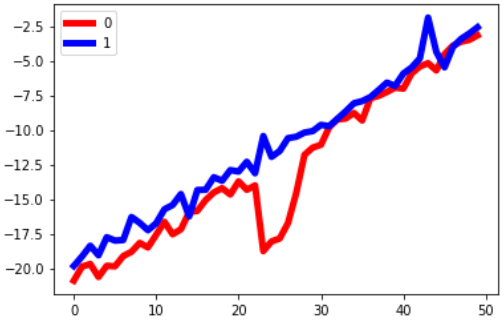


MAgent Battlefield Results

- Parameters copied from Battle Q-Networks

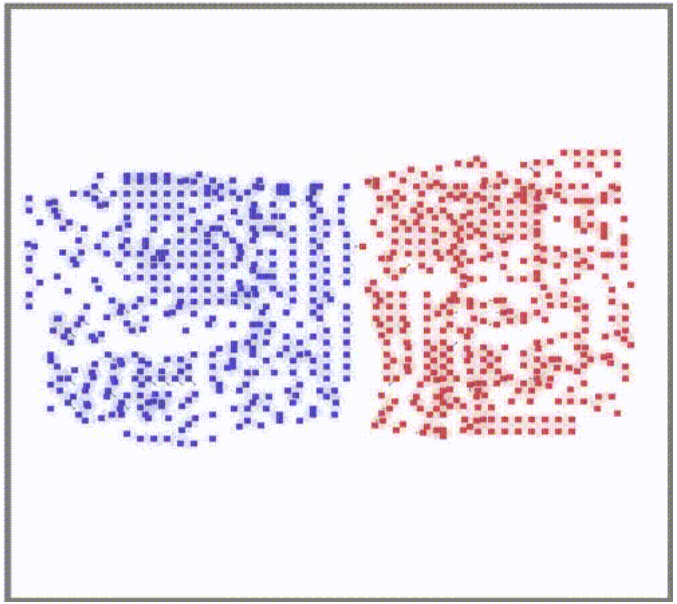


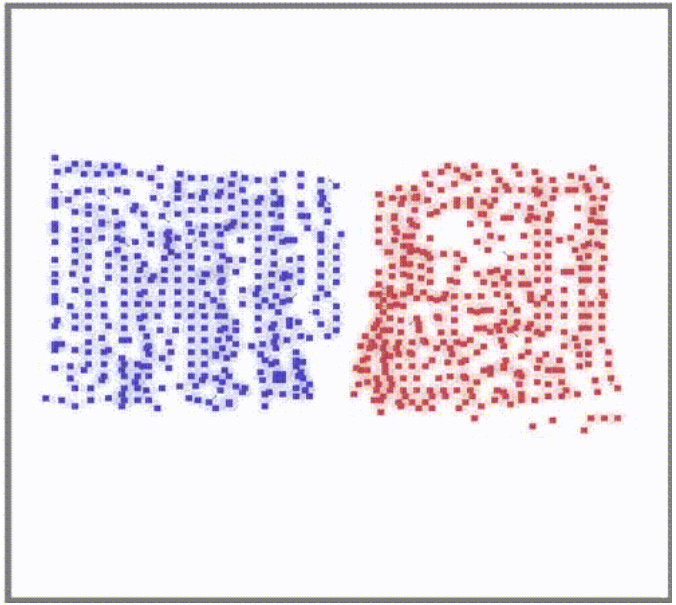




MAgent Battle Baseline

- Examples from MAgent's baseline (~ 1 day to train according to MAgent's GitHub):





Gifs from MAgent's GitHub

Summary

- What I was able to accomplish:
  - Accounting for deaths of agents during runtime
  - Mixed cooperative-competitive
  - Became more comfortable with PyTorch
- What I can improve upon in the future:
  - Implement faster batch processing
  - Tuning hyperparameters with longer runtimes
  - Transfer learned parameters to similar environments

Thank you