Training Reinforcement Learning Agents to React to an Ambush for Military Simulations

Tim Aris*, Volkan Ustun, Rajay Kumar****

*U.S. Army Combat Capabilities Development Command – Soldier Center (DEVCOM SC) Simulation and Training Technology Center (STTC), 12423 Research Parkway, Orlando, FL 32826,

**University of Southern California Institute for Creative Technologies, Playa Vista, CA, USA

timjaris@gmail.com, ustun@ict.usc.edu, kumar@ict.usc.edu

Abstract

There is a need for realistic Opposing Forces (OPFOR) behavior in military training simulations. Current training simulations generally only have simple, non-adaptive behaviors, requiring human instructors to play the role of OPFOR in any complicated scenario. This poster addresses this need by focusing on a specific scenario: training reinforcement learning agents to react to an ambush. It proposes a novel way to check for occlusion algorithmically. It shows vector fields showing the agent's actions through the course of a training run. It shows that a single agent switching between multiple goals is possible, at least in a simplified environment. Such an approach could reduce the need to develop different agents for different scenarios. Finally, it shows a competent agent trained on a simplified React to Ambush scenario, demonstrating the plausibility of a scaled-up version.

Introduction

Reinforcement learning (RL) aims to produce optimal policies in given environments. While there has been significant progress, learning to navigate a 3D terrain remains challenging for RL systems.

The specific task chosen was for the agent to react to an ambush, and this paper presents preliminary work towards creating robust agents capable of such responses. It expands on (Aris et al. 2023) by reusing the waypoint-based navigation, and transfers the problem over from taking cover.

Reacting to an ambush is a complex behavior with many moving parts, so we iteratively built up to it by first training agents to walk to a goal, then training agents to walk to a goal while avoiding a static enemy, and finally reacting to an enemy spawning mid episode. We also show an experiment showing that one agent that can dynamically switch goals is possible.

Scenarios

This paper presents three main scenarios. The first has three agents moving to a goal while trying to stay hidden from a static enemy. The second is the Charge or Flee scenario, which takes place on a flat strip with the goal randomly chosen between moving through enemy fire towards a goal or fleeing to survive as long as possible. The last is the Simple React to Ambush scenario, where agents move on a flat plane towards a goal, and they have to react when an enemy spawns next to them and attacks mid episode.

The first set of scenarios was conducted in the Rapid Integration and Development Environment (RIDE), a military training simulation environment that can interface with the Unity game engine (Hartholt et al. 2021). The latter two scenarios were done in the ICT MLAgents API (Kumar 2023). All agents were trained leveraging the ML-Agents framework within Unity (Juliani et al. 2018) and using the Proximal Policy Optimization (PPO) algorithm (Schulman et al. 2017).

Move While Hidden Scenario

This scenario went through a number of iterations. First, agents simply walked to the goal to make sure the environment was working properly. Then, the agents were rewarded for getting closer to the goal, and punished for how visible they were. However, any punishment that is correlated with exploration usually results in catatonic agents, so the visibility punishment was replaced with a reward for being hidden.

Both versions of visibility rewards or punishments led to an issue of it being a delicate balance between those and the proximity to goal rewards. Giving the enemy the ability to fire at the agent led to more stable learning, and was what the visibility metrics were supposed to be an abstract representation of anyway.

Charge or Flee Scenario

The motivation for this scenario was to investigate whether one RL agent can choose between multiple goals based on information in its observation space. DeepMind has demonstrated that something similar is possible (O.E.L. Team et al. 2021), but their model was much bigger than ours, and they processed goals as natural language fed through the architecture of gpt-2, while our goal is represented as a single boolean given to the agent.

The Charge or Flee scenario is one where the RL agents can either run through enemy fire to get to a goal, usually losing one or two agents in the process, or move the other direction to live as long as possible.

The intention for this is to eventually enable hierarchical reinforcement learning, with a commander agent being able to choose the goals of the subordinate agents as actions.

Simple React to Ambush Scenario

The final scenario, a simplified version of the intended React to Ambush scenario. Agents move on a flat plane from the start to the goal, and when they reach the middle, an enemy spawns and the agents must stop moving in order to fire back accurately enough to kill the enemy before the enemy kills the agent.

Results

As mentioned above, the agents in the Move While Hidden scenario require a careful balancing of rewards when visibility is a parameter in the reward function, while the agents trained with enemy fire learned more robustly, as the lack of a goal reward on death was incentive enough to avoid enemy fire.

We observed an interesting behavior when fine-tuning the reward function for the Move While Hidden Scenario: agents developed a cooperative strategy where one agent out of three would move in the opposite direction of the goal. This occurred because agents were rewarded for staying on the goal (to try and prevent them wandering off before their teammates could arrive, as the episode ended when all three were close to the goal). So, two agents standing on the goal for the entire episode gave a higher reward than three agents arriving at the goal as early as possible. The end-of-episode reward was adjusted to always be higher than this strategy.

In the Charge or Flee scenario, the agents were able to converge on the strategy of charging when the goal variable was 1, and fleeing when 0, demonstrating a simple RL agent can move between multiple goals, though it is unknown how this scales for more complex tasks or a greater quantity of goals.

In the Simple React to Ambush scenario, the trained agents successfully move towards the goal, stop to defeat the enemy, and then continue towards the goal. However, there is occasional behavior where the agents loop back towards the start before reaching the goal, so the agents can't be said to be optimal.

Conclusion

This paper presents preliminary work on developing RL agents capable of reacting to ambushes and dynamically choosing between multiple goals. Further work will expand on the React to Ambush scenario by introducing realistic terrains, altering the time and place the ambush takes place, and incorporating multiple goals, so the agent can prioritize getting to the goal as fast as possible, eliminating the enemy, or surviving.

Acknowledgments

The project/effort/work depicted here was or is sponsored by the U.S. Army Research Laboratory (ARL) under contract number W911NF-14-D-0005. Statements and opinions expressed and content included do not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

Aris, T., Ustun, V., & Kumar, R. (2023, May). Learning to Take Cover with Navigation-Based Waypoints via Reinforcement Learning. In The International FLAIRS Conference Proceedings (Vol. 36).

Hartholt, A., K. McCullough, E. Fast, A. Reilly, A. Leeds, S. Mozgai, V. Ustun, and A. S. Gordon. 2021. "Introducing RIDE: Lowering the Barrier of Entry to Simulation and Training through the Rapid Integration & Development Environment". 2021 Virtual Simulation Innovation Workshop.

Juliani, A., Berges, V. P., Teng, E., Cohen, A., Harper, J., Elion, C., ... & Lange, D. (2018). Unity: A general platform for intelligent agents. arXiv preprint arXiv:1809.02627.

Kumar, R. (2023). ICTMLAgentsAPI. GitHub. https://github.com/HATS-ICT/ICTMLAgentsAPI

Schulman J., Wolski F.,Dhariwal F., Radford A., and Klimov O. 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

Team, O.E.L., Stooke, A., Mahajan, A., Barros, C., Deck, C., Bauer, J., Sygnowski, J., Trebacz, M., Jaderberg, M., Mathieu, M. and McAleese, N., 2021. Open-ended learning leads to generally capable agents. arXiv preprint arXiv:2107.12808