Training agent for first-person shooter (FPS) game with Actor-critic Curriculum learning.

Yuxin Wu,Yuandong Tian

Published as a conference paper at ICLR 2017

Abstract

- This paper propose a new framework for training vision-based agent for First-Person Shooter (FPS) Game, in particular Doom.
- Their framework combines the state-of-the-art reinforcement learning approach (Asynchronous Advantage Actor-Critic (A3C) model) with curriculum learning.
- On a known map, our agent won 10 out of the 11 attended games and the champion of Track1 in ViZDoom AI Competition 2016.

Introduction

- Recently, Asynchronous Advantage Actor-Critic (A3C) model shows good performance for 3D environment exploration. However, in general, to train an agent in a partially observable 3D environment from raw frames remains an open challenge.
- In this paper, we train an AI agent in Doom with a framework that based on A3C with convolutional neural networks (CNN).

DOOM AS A REINFORCEMENT LEARNING PLATFORM

- In Doom, the player controls the agent to fight against enemies in a 3D environment. The agent can only see the environment from his viewpoint and thus receives partial information upon which it makes decisions.
- On modern computers, the original Doom runs in thousands of frames per second, making it suitable as a platform for training AI agent.
- ViZDoom [Kempka et al. (2016)] is an open-source platform that offers programming interface to communicate with Doom engine,

Method - NETWORK ARCHITECTURE

- Author use convolutional neural networks to extract features from the game frames and then combine its output representation with game variables.
- For input, they use the most recent 4 frames plus the center part of them, scaled to the same size (120 × 120). Therefore, these centered “attention frames” have higher resolution than regular game frames, and greatly increase the aiming accuracy. The policy network will give 6 actions.

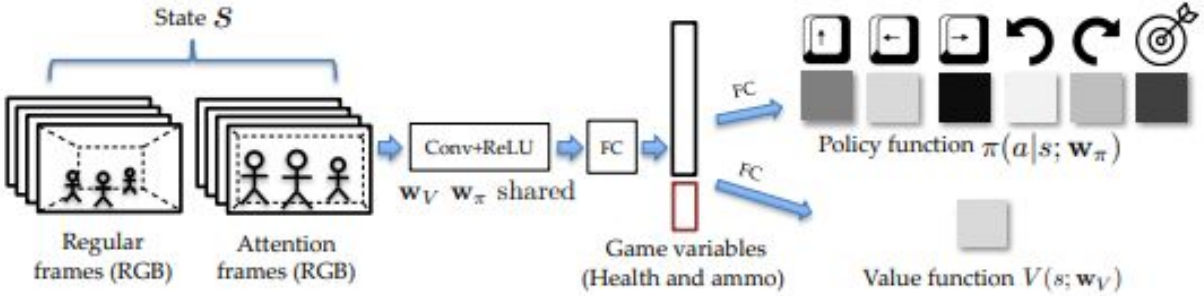


Figure 3: The network structure of the proposed model. It takes 4 recent game frames plus 4 recent attention frames as the input state s, and outputs a probability distribution $\pi(a|s)$ of the 6 discrete actions. The policy and value network share parameters.

Method - TRAINING PIPELINE

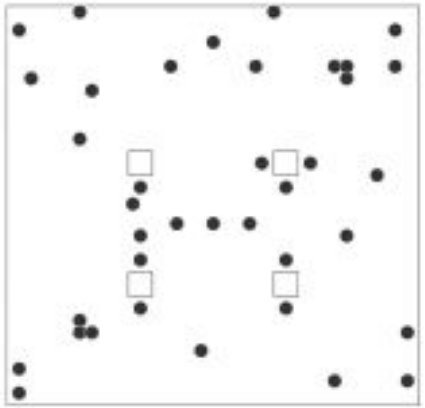
- The training procedure is implemented with TensorFlow and tensorpack5 .
- The main process collects frames from different game instances to create batches, and optimizes on these batches asynchronously on one or more GPUs.
- They use Adam [Kingma & Ba (2014)] with $\varepsilon = 10^{-3}$ for training.

Method - TRAINING PIPELINE

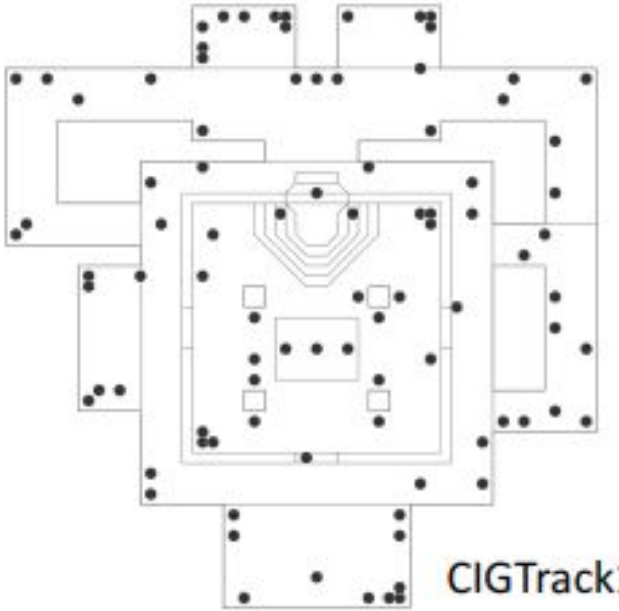
Parameters	Description	FlatMap	CIGTrack1
living	Penalize agent who just lives	-0.008 / action	
health_loss	Penalize health decrement	-0.05 / unit	
ammo_loss	Penalize ammunition decrement	-0.04 / unit	
health_pickup	Reward for medkit pickup	0.04 / unit	
ammo_pickup	Reward for ammunition pickup	0.15 / unit	
dist_penalty	Penalize the agent when it stays	-0.03 / action	
dist_reward	Reward the agent when it moves	9e-5 / unit distance	
dist_penalty_thres	Threshold of displacement	8	15
num_bots	Number of built-in bots	8	16

Table 2: Parameters for different maps.

Method - TRAINING PIPELINE



FlatMap



CIGTrack1

Figure 2: Two maps we used in the paper. `FlatMap` is a simple square containing four pillars . `CIGTrack1` is the map used in Track1 in ViZDoom AI Competition (We did not attend Track2). Black dots are items (weapons, ammo, medkits, armors, etc).

Method - CURRICULUM LEARNING

- When the environment only gives very sparse rewards, or adversarial, A3C takes a long time to converge to a satisfying solution.
- To address this, we use curriculum learning that trains an agent with a sequence of progressively more difficult environments. By varying parameters in Doom, they could control its difficulty level.
- Author consider the agent to perform well if its frag count is greater than 10 points.

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Speed	0.2	0.2	0.4	0.4	0.6	0.8	0.8	1.0
Health	40	40	40	60	60	60	80	100

Table 3: Curriculum design for `FlatMap`. Note that enemy uses `RocketLauncher` except for Class 0 (`Pistol`).

Result

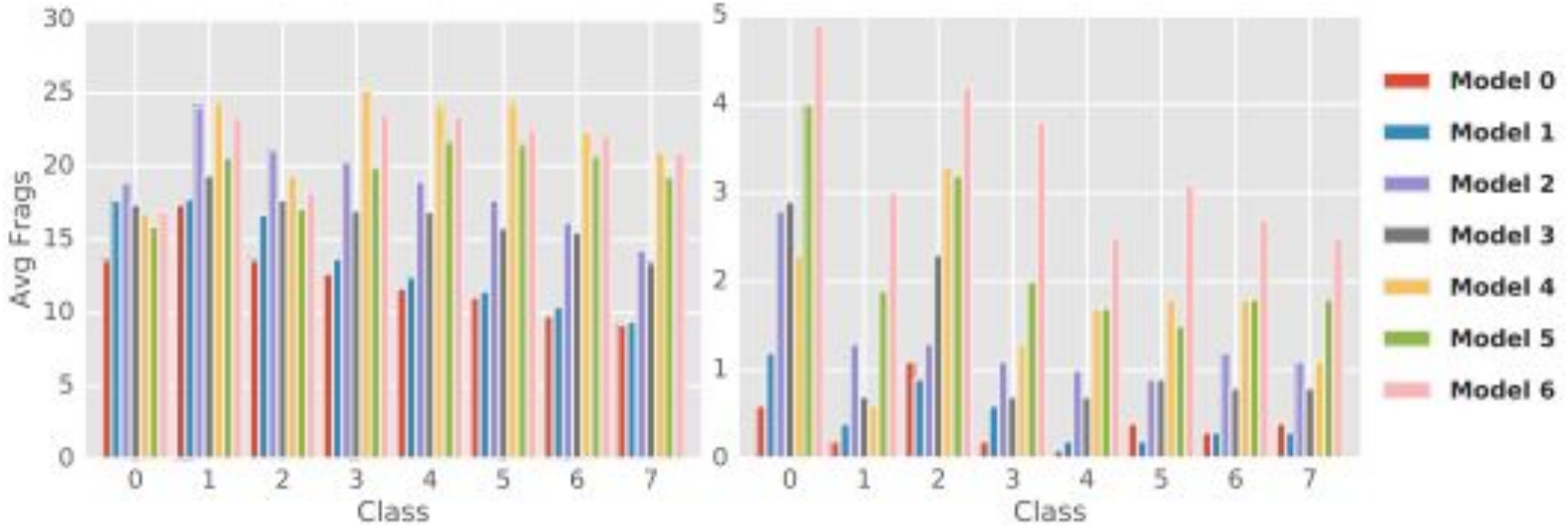
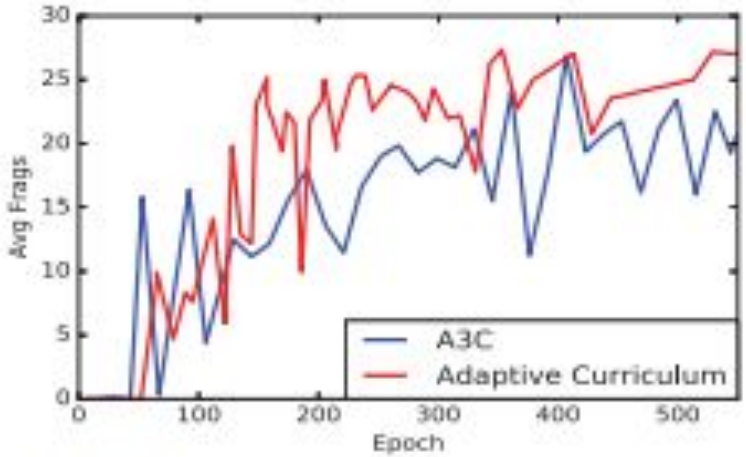


Figure 4: Average Frags over 300 episodes evaluation, on `FlatMap`(left) and `CIGTrack1`(right) with different levels of enemies (See Tbl. 3 for curriculum design). Models from later stages performs better especially on the difficult map, yet still keeps a good performance on the easier map.

Result



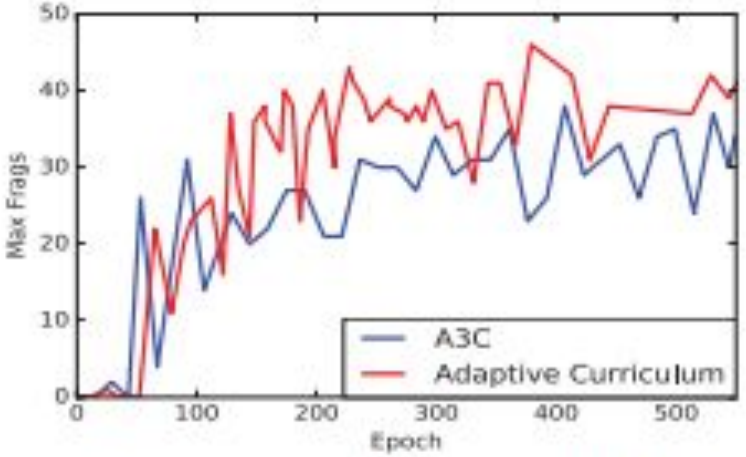


Figure 5: Performance comparison on Class 7 (hardest) of `FlatMap` between A3C [Mnih et al. (2016)] and adaptive curriculum learning, at different stage of training. Average frags and max frags are computed from 100 episodes. Adaptive curriculum shows higher performance and is relatively more stable.

Result

- Fig. 4 shows that the curriculum learning increases the performance of the agents over all levels. When an agent becomes stronger in the higher level of class, it is also stronger in the lower level of class without overfitting.
- Fig. 5 shows comparison between adaptive curriculum learning with pure A3C. They can see that pure A3C can learn on FlatMap but is slower. Moreover, in CIGTrack1, a direct application of A3C does not yield sensible performance.

Conclusion

- In this paper, author propose a new framework to train a strong AI agent in a First-Person Shooter (FPS) game, Doom, using a combination of state-of-the-art Deep Reinforcement Learning and Curriculum Training.
- It learns to use motion features and build its own tactics during the game, which is never taught explicitly.
- Currently, their bot is still an reactive agent that only remembers the last 4 frames to act. Ideally, a bot should be able to build a map from an unknown environment and localize itself, is able to have a global plan to act, and visualize its reasoning process.