John Balis (balis)
Yisen Wangv(ywang2557)
Milica Cvetkovic (mcvetkovic)

Project Proposal
CS 766

2021-03-25

1 Background

When learning to navigate through 3d environments in the context of Deep Q learning with pixel value inputs, the Deep Q Network(DQN) must implicitly perform localization and mapping, if this information is necessary to successfully navigate in a way which maximizes reward. Instead of requiring the DQN to implicitly perform localization and mapping from pixel vales, we are interested in determining how augmenting the features of the input to the DQN using SLAM affects the performance of standard reinforcement learning approaches in simulated 3d environments.

2 State-Of-The-Art

2.1 SLAM in RL

There is existing work exploring the use of SLAM for feature augmentation in the context of reinforcement learning. Chen et. al. [**1**] presents a technique for reinforcement learning in both real and simulated 3d environments which uses SLAM to provide an estimate of velocity as a feature to the DQN algorithm. It is worth noting that in this case, SLAM also extracts the absolute position, but this data is only given to a separate planning algorithm which then provides the DQN algorithm with ”local goals.” Bhatti et. al. presents a scheme for augmenting pixel values with features extracted using SLAM when training a DQN to play a DOOM reinforcement learning environment[**2**]. The method outlined in Bhatti et. al. is similar to the method presented in this proposal, although it may differ in terms of specifics such as how the augmented features are calculated from SLAM outputs.

2.2 Simulation Environment

Procedurally-generated environments are widely used in reinforcement learning research, as they not only simulate real world environments and provide realistic constraints such as first-person observable views, but also enable fast and efficient training before applying the algorithm to the real world. Miniworld[**3**] is one of the environments that we are going to use.
MiniWorld is selected as training environment for many RL projects. Paper[**4**] investigates exploration under sparse reward, where MiniWorld is used to generate 3D navigation tasks with large room number and room size. It used the default CNN architecture provided in OpenAI baselines.Paper[**5**] introduced a new algorithm that extends Hindsight Experience Replay(HER) to complex visual environments. By retroactively transforming failed trajectories to successful ones, HER allows the agent exploit the sample more efficiently. Experiments generated by MiniWorld contain two sets of navigation tasks with continuous control policy. The network architecture for DDQN agent and DDPG actor take 64x64 image as input, followed by four convolutional layer and two dense layer.

3 Methods

Reinforcement learning will be implemented using stable-baselines, a library containing reference implementations of state-of-the art reinforcement learning algorithms. At a minimum, the provided implementation of DQN will be used for analysis; other algorithms may also be considered. The reinforcement learning environment will be OpenAI gym-miniworld, which provides a simple 3d reinforcement learning environment with navigation based reward. In the control case, the features for DQN will be the pixel values from gym-miniworld. In the experimental case, the features used to train the DQN will include both pixel values and spatial data extracted using SLAM on the preceding sequence of images extracted from the environment.

4 Hypothesis

We hypothesize that using SLAM to augment DQN features will improve the speed at which DQN increases average reward during training. This seems to be the case in [**2**].

5 Milestones

Task	time
Create DQN Miniworld benchmark	February 25 - March 10
Set up SLAM with Miniworld	March 11- March 20
Design input encoding for SLAM features into DQN	March 21- April 6
Contrast performance of DQN on Miniworld	April 7 - April 21
Complete project writeup and presentation	April 22- May 5

6 Further Research

Assuming a satisfactory answer to the research question, other goals of interest include comparing the relative performance of reinforcement learning with feature augmentation performed by leading implementations of direct v.s. indirect SLAM. Additionally, it is worth exploring how the injection of artificial noise into the images generated by the reinforcement learning environment affects this performance.

7 Website

https://cmilica.github.io/cs766project/

References

[1] Chen, G., Pan, L., Chen, Y., Xu, P., Wang, Z., Wu, P., Ji, J., Chen, X. (2020). Robot Navigation with Map-Based Deep Reinforcement Learning. ArXiv:2002.04349 [Cs]. http://arxiv.org/abs/2002.04349

[2] Bhatti, S., Desmaison, A., Miksik, O., Nardelli, N., Siddharth, N., Torr, P. H. S. (2016). Playing Doom with SLAM-Augmented Deep Reinforcement Learning. ArXiv:1612.00380 [Cs, Stat]. http://arxiv.org/abs/1612.00380

[3] Maxime Chevalier-Boisvert, MiniWorld, GitHub repository, https://github.com/maximecb/gym-miniworld, 2018

[4] Daochen Zha, Wenye Ma, Lei Yuan, Xia Hu, Ji Liu, ”Rank the Episodes: A Simple Approach for Exploration in Procedurally-Generated Environments,” ICLR, 2021

[5] Sahni, H., Buckley, T., Abbeel, P., and Kuzovkin, I., “Addressing Sample Complexity in Visual Tasks Using HER and Hallucinatory GANs”, arXiv e-prints, 2019.