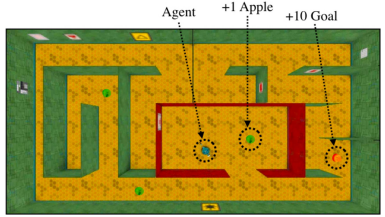
Surprise Pursuit on Deepmind Lab Maze

Luke Johnston (lukej@stanford.edu)

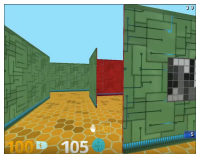
Introduction

The Deepmind Lab environment is a recently released platform for reinforcement learning in complex 3D environments [1]. In the recent paper "Reinforcement Learning with Unsupervised Auxiliary Tasks" [2], Mnih et al. modify the successful A3C algorithm [3], adding "unsupervised auxiliary tasks" such as policy control and reward prediction to speed up training and improve final results. The resulting "UNREAL (UNsupervised REinforcement Auxiliary Learning)" agent achieves state-of-the-art results in this environment and others. In this paper, I develop and implement an additional unsupervised auxiliary task, called "surprise pursuit", that I add to an existing UNREAL implementation in order to improve performance on the "maze_random_goal_01" environment of Deepmind Lab.

Maze Random Goal 01



- Agent begins at set location in the maze
- Apples give reward +1
- Goal gives reward +10
- When agent reaches goal, is randomly placed somewhere else in the maze
- Agent must explore new maze quickly, and remember goal location for next random placement
- Goal location randomized across episodes



Actions

- Left, right, forward, backward
- Strafe view left / right

State

- 84 x 84 x 3 image of environment from agent's perspective
- Current reward

UNREAL Auxiliary Tasks

Pixel Control

- Goal: agent needs to learn how actions affect the environment
- Learn a policy to maximize pixel change in a particular area of agent's view

Reward Prediction

- Goal: agent should recognize when rewards will occur so that it can pursue them
- Learn to predict reward of next state
- Can replay more frequently on rewarding states without introducing bias into policy or value function

Surprise Pursuit

Pixel Prediction

- Goal: agent should generally be able to predict how the environment will look after taking an action
- Architecture:
  - A3C architecture convolution + lstm
  - Lstm output is passed through 3 conv2d_transpose + batch_norm layers to reconstruct same shape as input
  - L2 loss over difference between prediction and actual

Maximizing Surprise

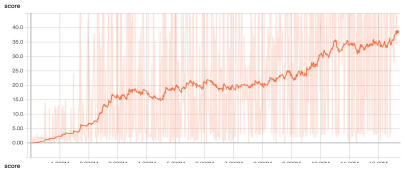
- Goal: in order to explore the maze, the agent needs to go where it hasn't been before. This policy seeks out actions that will "surprise" the model with a high pixel prediction loss.
  - Architecture: A3C architecture convolution + lstm
  - Lstm outputs passed through a single fully-connected layer to obtain q-values for each action
  - The reward for this policy network is the loss for the pixel prediction. The n-step loss for the Q network is then

$$L(\theta) = (Q(s_t, a_t) - r_t - \gamma * V_{est}(s_{t+1}))^2$$

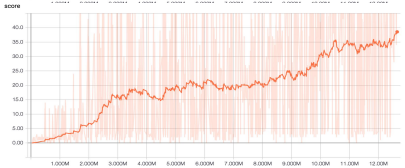
$$V_{est}(s_{t+1}) = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{N-1} r_{t+N-1} + \gamma^N \max_a Q(s_{t+N}, a)$$

Results

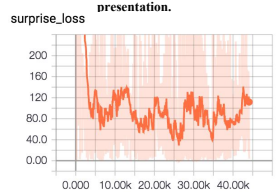
Average Scores Unmodified UNREAL

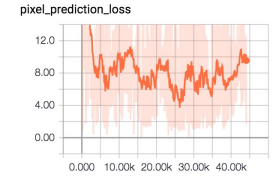


UNREAL + Surprise Pursuit. Will be updated for presentation, currently training.

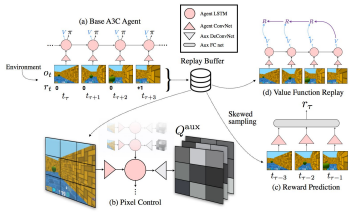


Surprise Pursuit Task Losses. Will update for presentation.





UNREAL Visualization [1]



Conclusions

The graphs in the "Results" section are being created right now. They should be ready in the early hours of tomorrow morning, so I will update this portion of the poster before presenting it tomorrow.

Future Work:

- Visualize frame predictions to see if auxiliary function is being learned successfully
- It may be better to predict feature activations rather than pixels directly. This would bypass need for deconvolutional layers that may be insufficient for fully reconstructing a scene of the environment.

References

[1] Beattie, Charles, et al. "DeepMind Lab." arXiv preprint arXiv:1612.03801 (2016).

[2] Jaderberg, Max, et al. "Reinforcement learning with unsupervised auxiliary tasks." arXiv preprint arXiv:1611.05397 (2016).

[3] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." International Conference on Machine Learning. 2016.

printed by MegaPrint Inc. www.postersession.com