Diamonds are a Steve's Best Friend: Minecraft Imitation/Reinforcement Learning

Gabe Barney and Griffin Young
STANFORD UNIVERSITY SYMBOLIC SYSTEMS PROGRAM
barneyga@stanford.edu, gcyoung@stanford.edu

Motivation

- NeurIPS is hosting a reinforcement learning challenge for the game of Minecraft. The goal is to train an agent to obtain diamond.

Challenges:

- Sparse rewards
- Hierarchical task learning
- Subtask similarity
- Continuous action space

Figure 1: A visualization of expert data

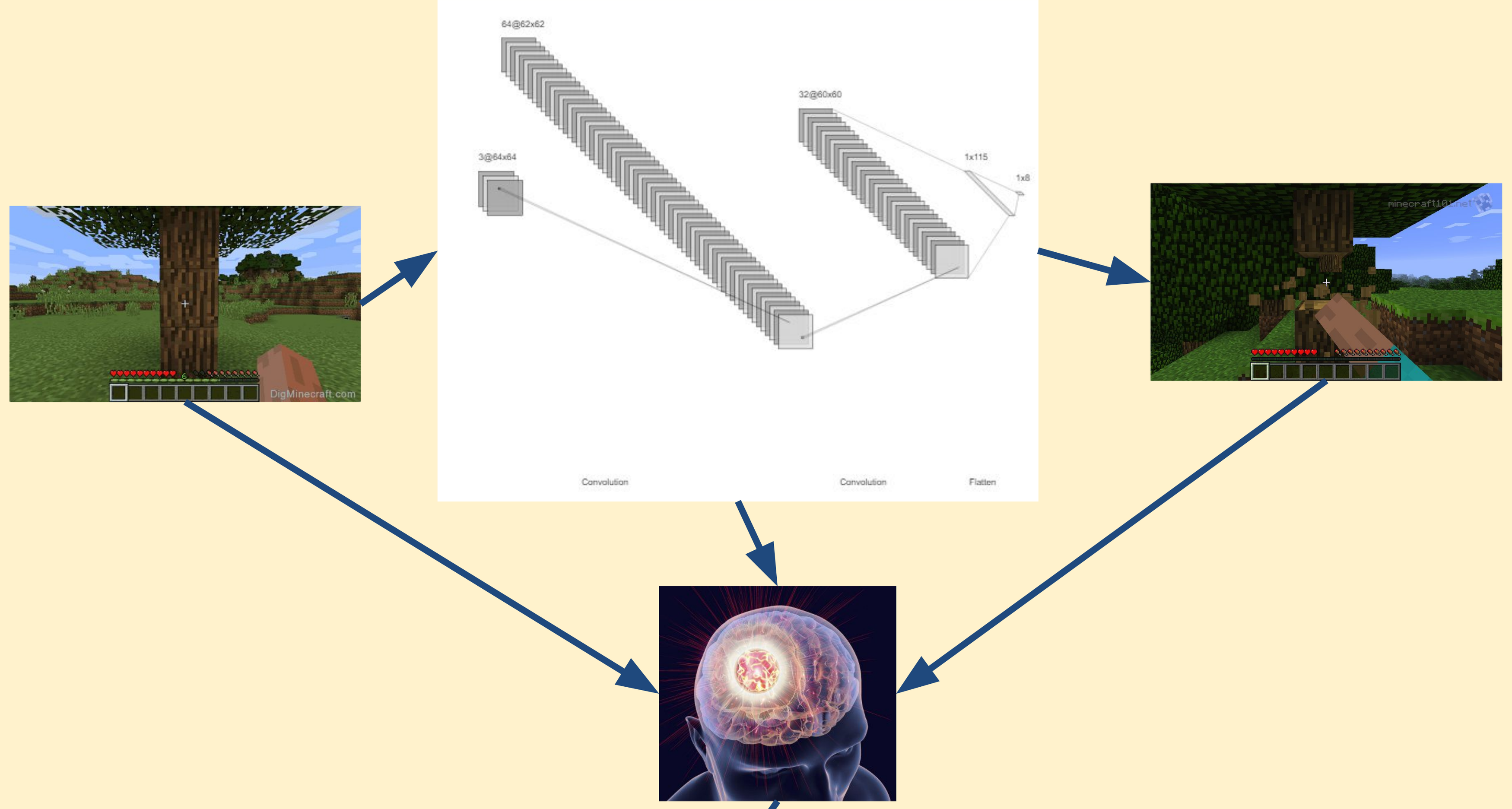
Imitation Dataset

- 1600 episodes of human gameplay (SARS')...
  - State (the RGB values of the screen at that frame)
  - User's action
  - User's current total reward
  - Next state
- ...divided across environments with different rewards...
  - Treechop: +1 for each unit of wood
  - Navigate: +100 upon reaching destination
  - NavigateDense: Rewards every tick for distance from destination
  - ObtainIronPickaxe: Reward once per item in hierarchy below iron pickaxe
  - **ObtainDiamond: Reward once per item in hierarchy below diamond**
- ...observation spaces...
  - Equipped Items (dictionary)
  - Inventory (dictionary)
  - Point of View (64x64x3 array of RGB values)
  - Compass Angle
- ...and action spaces.
  - Actions: attack, back, pitch/yaw of camera, forward, jump, left, right, place, sneak, sprint, equip, craft
  - Discretized the continuous pitch/yaw action space
  - Restricted action space based on environment (e.g. navigate only had yaw and jump)

Methods and Models

1. **Deep Q Learning**

- Epsilon greedy: Given state, choose random action with probability epsilon, else choose action with highest q value
- Store state, action, reward, new state tuples in sliding replay buffer
- Randomly sample from replay buffer and perform SGD on loss



$$L_i(\theta_i) = \mathbb{E}_{a\sim\mu}\left[\left(y_i - Q(s,a;\theta_i)\right)^2\right]$$

$$\text{where}\quad y_i := \mathbb{E}_{a'\sim\pi}\left[r + \gamma \max_{a'} Q(s',a';\theta_{i-1}) | S_t = s, A_t = a\right]$$

2. **Deep Q Learning from Demonstrations** -

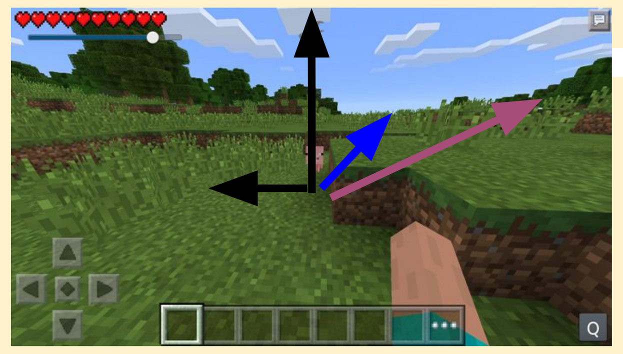
- Deep Q Learning plus imitation learning: prepopulates replay buffer with expert SARS' tuples and pretrains
- Additional imitation loss function:

0 if action same as expert action

$$J_E(Q) = \max_{a\in A}\left[Q(s,a) + \ell(a,a_E)\right] - Q(s,a_E)$$

Discussion

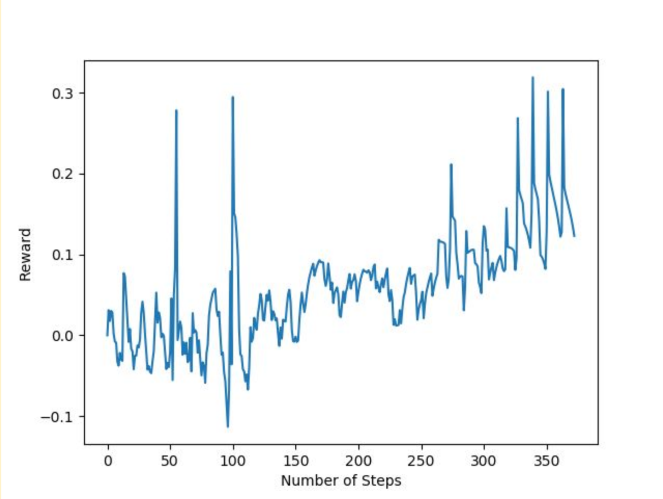
- Deep Q Learning fails in environments with sparse rewards
- In order to identify if the agent performed the same action as the expert within a huge action space, we used cosine similarity to find action in agent repertoire most similar to expert action



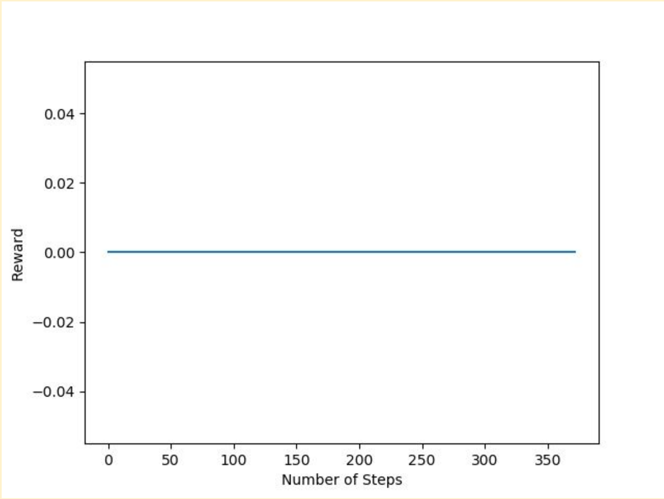
Results

Deep Q Learning

Navigate Dense

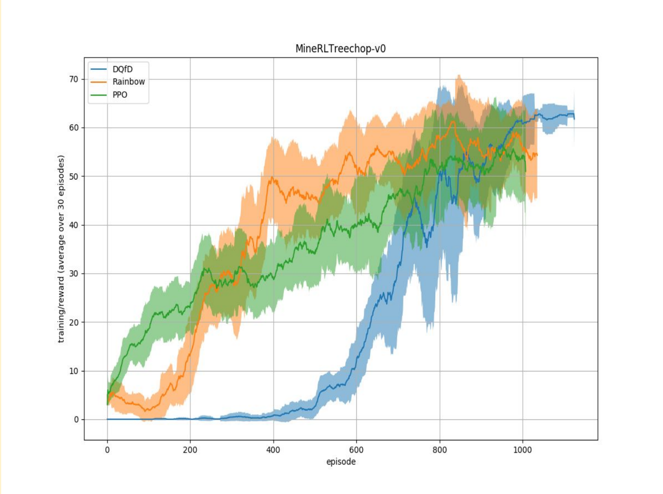


Treechop

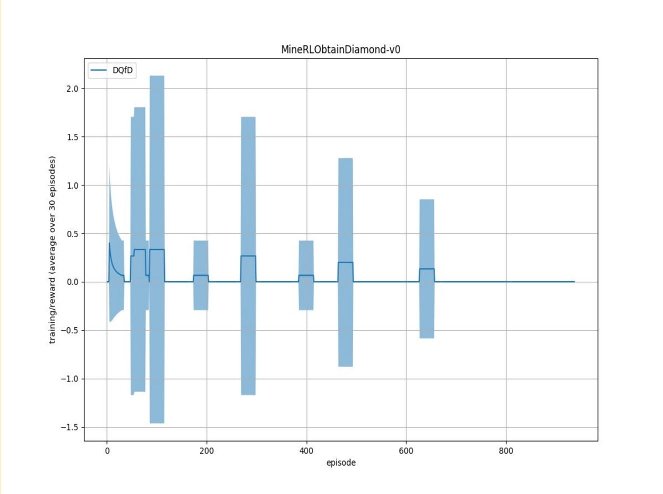


Deep Q Learning from Demonstrations

Projected Results: Treechop



Diamond



Future

- In order to tackle the ObtainDiamond task, we need to combine hierarchical reinforcement learning with the methods we've already considered
- Determine the differences between pure IL and DQfD, to make stronger inferences about the importance and brittleness of the different parts of DQfD.

References

[1] "MineRL: Towards AI in Minecraft". http://minerl.io/

[2] Hester, T.; Vecerik, M.; Pietquin, O.; Lanctot, M.; Schaul, T.; Piot, B.; Horgan, D.; Quan, J.; Sendonaris, A.; Osband, I.; et al. 2018. Deep q-learning from demonstrations. In Thirty-Second AAAI Conference on Artificial Intelligence. doi:10.1109/cvpr.2016.90

[3] H. M. Le, N. Jiang, A. Agarwal, M. Dudík, Y. Yue, and H. D. III, "Hierarchical imitation and reinforcement learning," in ICML, 2018.