Deep RL for Long Term Strategy Games

Akhila Yerukola (akhilay), Ashwini Pokle (ashwini1), Megha Jhunjhunwala (meghaj)
Dept. of Computer Science, Stanford University

CS 229
Fall 2017

Motivation

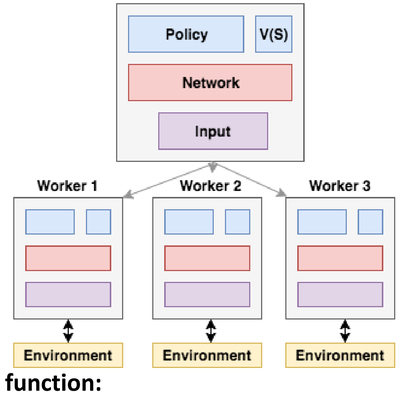
- Issues with Long-term strategy games
  - Large state space to explore
  - Delayed-sparse rewards
- Environment specific information needs to be provided
- We demonstrate our approach with 2 environments:
  1. Discrete stochastic decision process
  2. ATARI game "Montezuma's Revenge"

Approaches

- **Baseline**: Deep Q-Network (DQN), Asynchronous Advantage Actor Critic (A3C)
- **Our Approaches**: A3C-CTS (Context Tree Switching), Hierarchical Deep Q-Network

Asynchronous Advantage Actor Critic - Context Tree Switching (A3C - CTS)

- Multiple actor learners run in parallel to explore different parts of the game
- Agent's receives intrinsic rewards from the CTS density model as an exploration bonus



Advantage function:
$A(s,a) = Q(s,a) - V(s)$
Loss function:
$L(\theta) = \Delta_\theta \log \pi(a_t|s_t)\, A(s,a)$ where A(s,a) gives how much better this action is compared to average

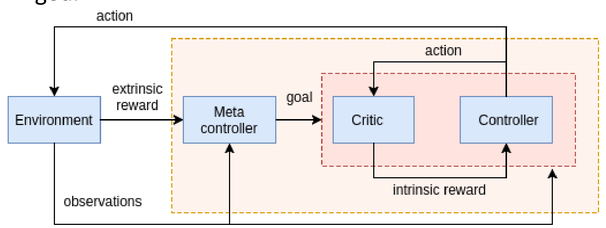
References

[1] Kulkarni, Tejas et al., "Hierarchical deep reinforcement learning: integrating temporal abstraction and intrinsic motivation", NIPS 2016
[2] Bellemare, Marc et al., "Unifying count-based exploration and intrinsic motivation.", NIPS 2016

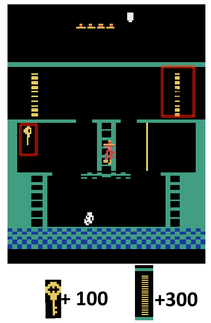
Hierarchical Deep-Q Network (hDQN)

The agent takes decisions over two levels of hierarchy

- **Meta-controller** : Learns policy over sub-goals
- **Actor-controller**: Learns policy over actions given the current state and sub-goal



Hierarchical Deep Q-Network Architecture

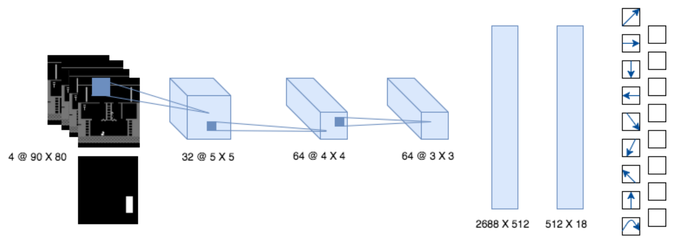


Montezuma's Revenge

ATARI Montezuma's Revenge

Architecture

- **Actor-Controller:** DQN network
- **Meta-Controller:** Finite State Machine to get the goal, given current state
- **Input to DQN:** Set of **4 consecutive image frames** from the game (gray scaled and down-sampled to 90 X 80) along with a **binary mask of the next goal**



Temporal diff. error: $\delta = (r + \gamma \max_{a'} Q(s', a'; \theta_{i-1,g})) - Q(s, a;\ \theta_i, g)$

Huber Loss: $L(\theta_i) = \frac{1}{|B|} \begin{cases} \Sigma\ (\delta^2/2) & for\ |\delta| \le 1 \\ \Sigma\ (|\delta| - 1/2) & otherwise \end{cases}$ $where\ |B| = batch\ size$

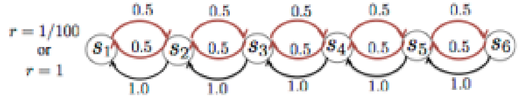
Tools Used: Open AI Gym, PyTorch

Hyperparameters: ϵ-greedy strategy, ϵ annealed from 1 to 0.1, $\alpha = 0.0025$

Optimization: RMSProp

Discrete Stochastic MDP

- **States:** {s1, s2,..., s6}
- **Actions:** {Left, Right}
- **Transition Probabilities:** T(s, Left) = 1; T(s, Right) = 0.5



Results and Analysis

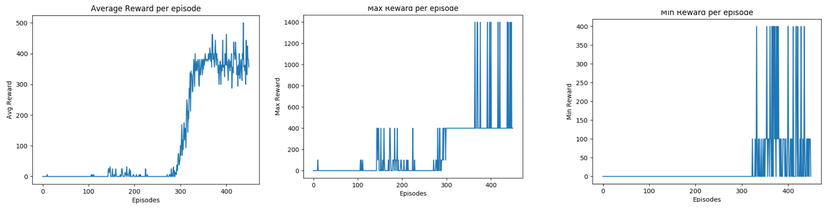
	Average Reward	Std. Dev	Max Reward	Min Reward	#Training Steps	Training time
DQN	0	0	0	0	6.2M	8 hours
A3C	0	0	0	0	0.15M	6 hours
A3C-CTS	362.5	22	1400	0	6.3M	8 hours
DQN-CTS	2	28	100	0	7.8 M	5 hours
H-DQN	Currently training		100	0	70K	3 hours elapsed

1. DQN and A3C

- Agent fails to learn the optimal strategy since there was insufficient exploration and the rewards were sparse

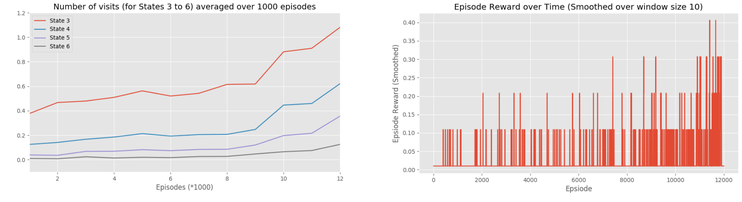
2. A3C-CTS

- 16 actor learners were trained in parallel and synchronized after every 120k steps
- All of them learnt to cross the first room thereby getting an avg. score of 400
- Intrinsic rewards from the CTS density model helps in better state-space exploration



3. hDQN: Stochastic MDP Environment

- Agent learns the long term-strategy of reaching state s1 after visiting state s6 thus achieving a reward of 1, instead of 0.01



4. Sub-goal Detection in Montezuma's Revenge

Future Work

- Analyze performance of hDQN against baseline on Montezuma's Revenge
- Analyze performance of intrinsic based heuristics for hDQN