Double A3C on Playing Atari Game

Lingjie Kong, Ruixuan Ren

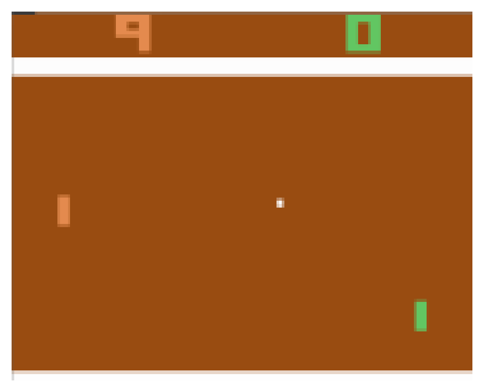
Video Link: https://drive.google.com/file/d/1nIwGI2CbrmnOKAOgaDvY4Rz6kFkhr9B9/view?usp=sharing

Motivation

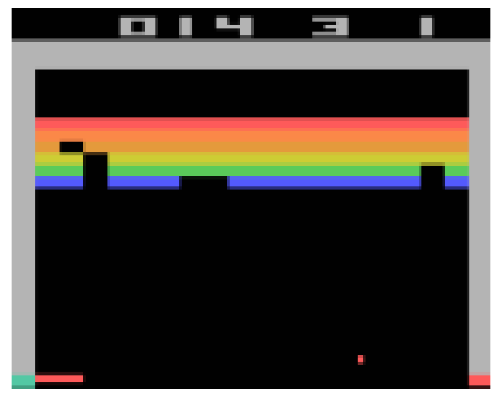
Deep reinforcement learning has been successfully applied to training Atari game agents. A lot of interests have been cast on designing new algorithms to improve performance. Here, we propose a new algorithm – double A3C and compare its performance with popular benchmarks.

Problem Definition

Reinforcement learning is an area regarding how agents act to an unknown environment for maximizing its rewards. Unlike Markov Decision Process in which agent has full knowledge of its state, rewards, and transitional probability, RL agent utilizes exploration and exploitation to cover model uncertainty. Because the model usually has a large input feature space, a neural network (NN) is often used to summarize the correlation between input feature and output state action value. Our goal is to improve state-of-the-art A3C (Asynchronous Advantage Actor-Critic) algorithm by implementing our own "double A3C". We compare its performance with Deep Q-network (DQN) and vanilla A3C on three Atari games: Pong, Ice Hockey and Breakout.



Pong



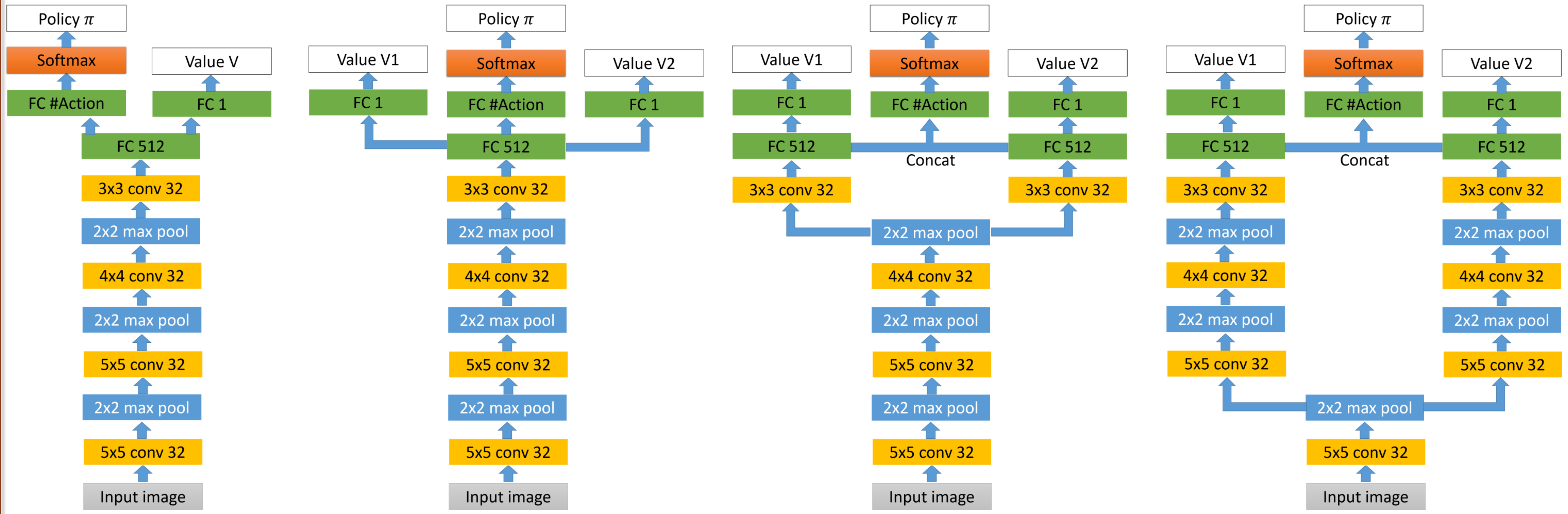
Breakout

Ice hockey

Challenge

- Requires large amount of hand-labelled data
- Sparse, noisy and delayed rewards
- Highly correlated sequence of states

A3C Network Structure



Policy π
Softmax
FC #Action
Value V
FC 1
FC 512
3x3 conv 32
2x2 max pool
4x4 conv 32
2x2 max pool
5x5 conv 32
2x2 max pool
5x5 conv 32
Input image

Value V1
FC 1
Policy π
Softmax
FC #Action
Value V2
FC 1
FC 512
3x3 conv 32
2x2 max pool
4x4 conv 32
2x2 max pool
5x5 conv 32
2x2 max pool
5x5 conv 32
Input image

Value V1
FC 1
FC 512
3x3 conv 32
Policy π
Softmax
FC #Action
Concat
Value V2
FC 1
FC 512
3x3 conv 32
2x2 max pool
4x4 conv 32
2x2 max pool
5x5 conv 32
2x2 max pool
5x5 conv 32
Input image

Value V1
FC 1
FC 512
3x3 conv 32
2x2 max pool
4x4 conv 32
2x2 max pool
5x5 conv 32
Policy π
Softmax
FC #Action
Concat
Value V2
FC 1
FC 512
3x3 conv 32
2x2 max pool
4x4 conv 32
2x2 max pool
5x5 conv 32
2x2 max pool
5x5 conv 32
Input image

A3C

Double A3C

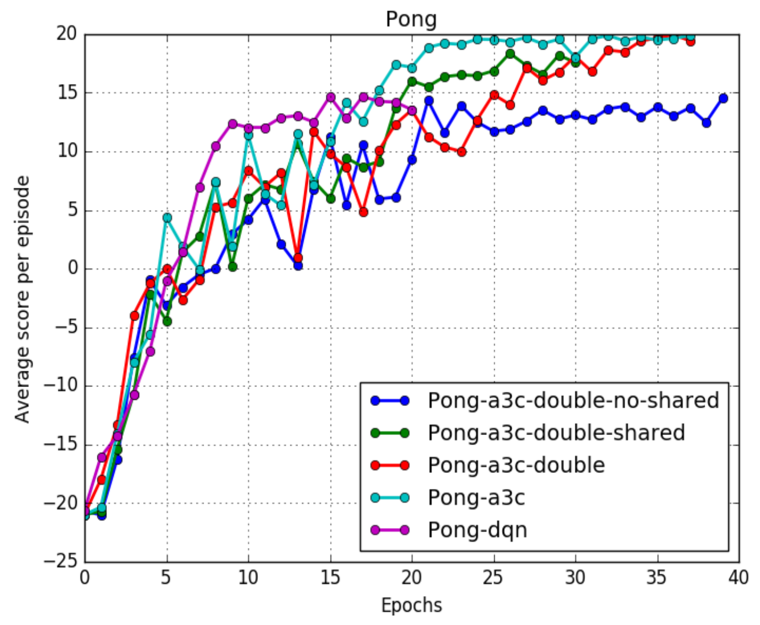
Less Shared Double A3C

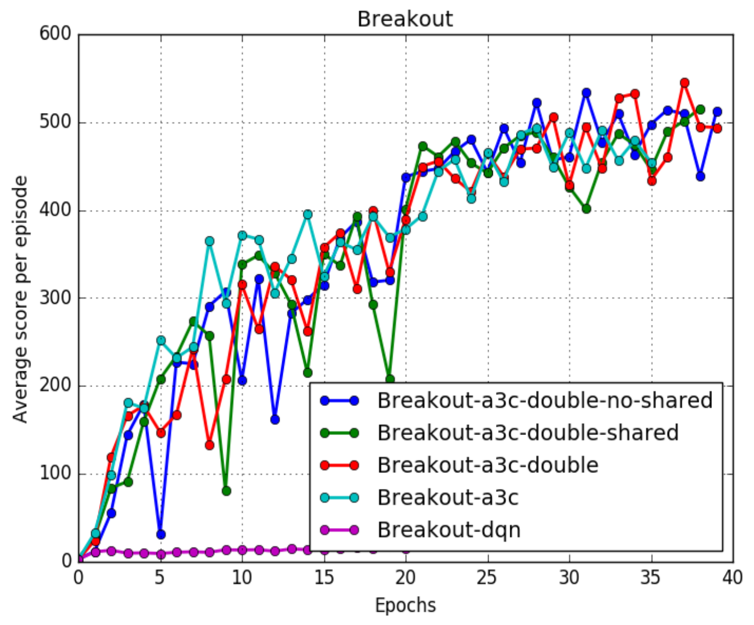
No Shared Double A3C

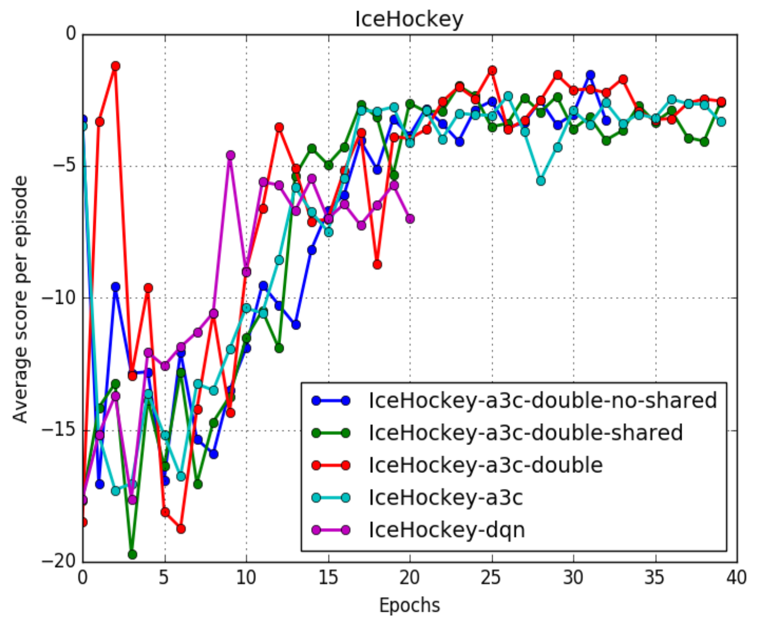
- Improved classical A3C network structure by introducing double A3C
- Double A3C maintains two independent value functions
- Double A3C concatenates the fully connected layers before value function output layers to generate one policy
- Randomly pick one value function to update during training
- Varied shared number of layers between two value functions

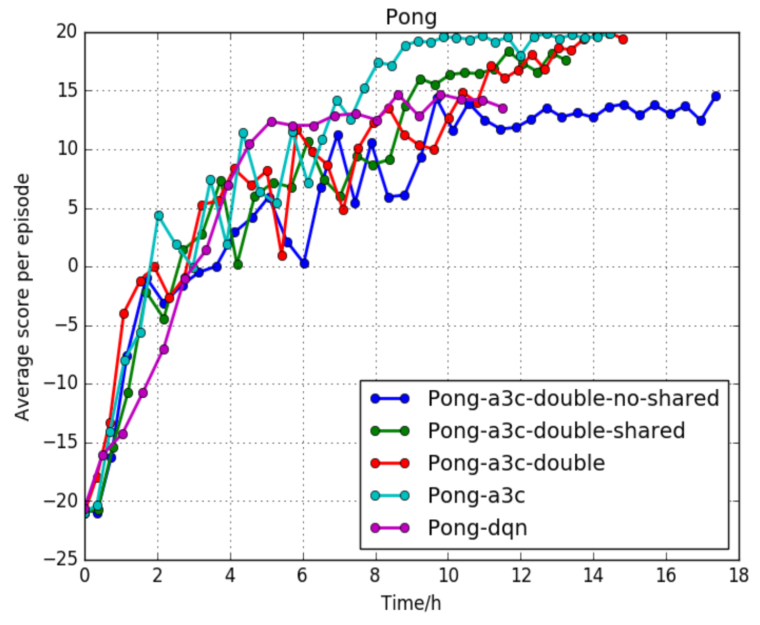
Result

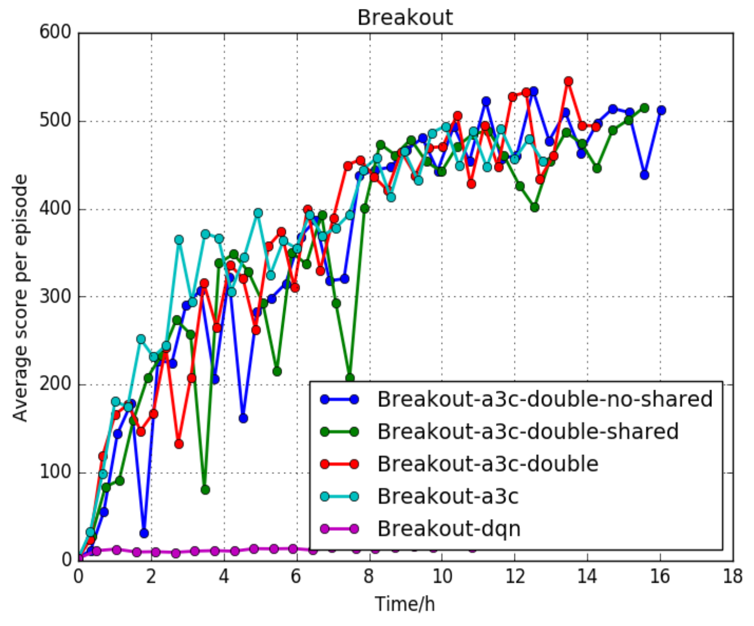
- Evaluated the performance by comparing average score over each epoch and over training time

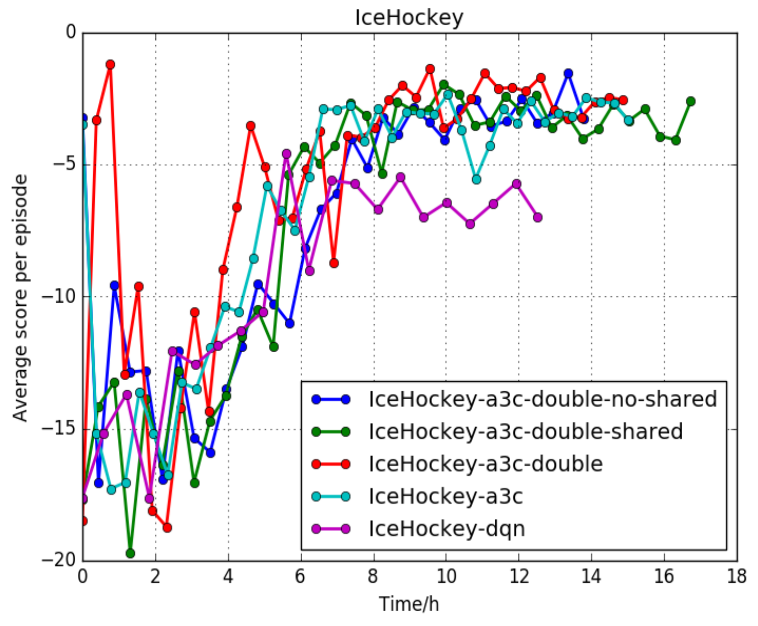












Analysis

- Double A3C outperforms DQN in most environment
- Double A3C is more resistant to environment variation compared to DQN
- Performances of classical A3C and double A3C are at the same level
- We hope to use double A3C to break the correlations between states for achieving better performance. However, A3C already used multiple parallel agents during update. Such parallel agents have already removed correlation in certain degree. Therefore, adding a second value function does not significantly improve the performance.
- Double A3C is inspired by double DQN to remove bias. However, classical A3C does not introduce maximization bias.
- Implement V-trace to reduce policy lag