Playing Atari with Deep Reinforcement Learning - Mnih et al., 2013

Brahma S. Pavse

November 12 2018

Cons

RL Exploration

ensures adequate exploration of the state space. In practice, the behaviour distribution is often selected by an ϵ-greedy strategy that follows the greedy strategy with probability $1 - \epsilon$ and selects a random action with probability ϵ.

- What is “random” to choose the random action?
  - Softmax Exploration?
  - Uniform?

Reward Clipping

of the games during training only. Since the scale of scores varies greatly from game to game, we fixed all positive rewards to be 1 and all negative rewards to be -1, leaving 0 rewards unchanged. Clipping the rewards in this manner limits the scale of the error derivatives and makes it easier to use the same learning rate across multiple games. At the same time, it could affect the performance of our agent since it cannot differentiate between rewards of different magnitude.

- Despite acknowledging potential impact
  - Why not scale them between a range?

These values work, don't change anything!

- No hyperparameter insights
  - No (in)formal intuition
- Examples
  - Action every "k" frames. Why 4?
  - Experience replay of "N" trajectories? Why 1 million?
  - Exploration probability of ε. Why does 0.1 (after annealing) work for games with 4 AND 18 actions?

Clarity of Graphs

how much discounted reward the agent can obtain by following its policy from any given state. We collect a fixed set of states by running a random policy before training starts and track the average of the maximum[2] predicted Q for these states. The two rightmost plots in figure 2 show that average

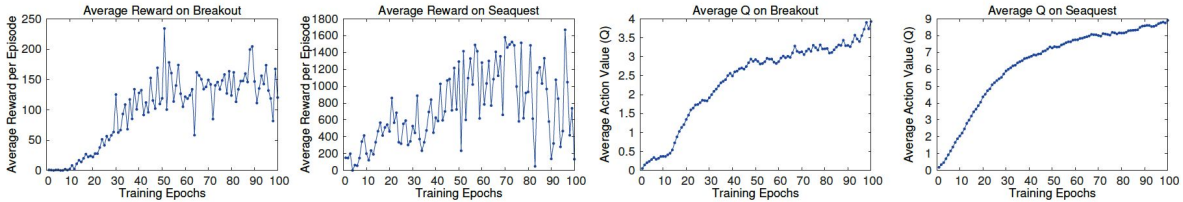

Figure 2: The two plots on the left show average reward per episode on Breakout and Seaquest respectively during training. The statistics were computed by running an ϵ-greedy policy with $\epsilon = 0.05$ for 10000 steps. The two plots on the right show the average maximum predicted action-value of a held out set of states on Breakout and Seaquest respectively. One epoch corresponds to 50000 minibatch weight updates or roughly 30 minutes of training time.

How many are in the “fixed set of states”?

Tells us reliability

Transfer Learning

- Admittedly, not focus of paper
- However, since focus was on 7 games
  - Why not encourage it in conclusions?
  - Provide possible insights?

Thanks!