Exploration method for reducing uncertainty using Q-entropy in deep reinforcement learning

*Seok-Jeong Song[1,2], Minji Kim[2], Wonseok Jung[2]
[1]Dept. of Information Display, Kyung Hee University
[2]Deep Learning College, Modulabs
e-mail : tenoclo@khu.ac.kr

Abstract

In this paper, we propose a novel exploration method for a Q-learning based deep reinforcement learning. According to how the agent has uncertainty on current state which can be estimated by entropy value of action-values at each state, the agent decides whether to do exploration or do exploitation.
Also we adopt the state visit-counter to handle the ambiguous states which means that several optimal actions exist.

I. Introduction

Reinforcement learning is a method in which an agent learns behaviors that maximize the accumulated rewards due to interaction with the unknown environment. It is important for agent to balance exploration and exploitation. Exploration is the agent explore the environment to find states and actions to get high rewards while exploitation is the agent make a decision using current knowledge. Especially, efficient exploration is central challenge in infinite state space. The agent which is trained with conventional exploration method such as epsilon-greedy [1] has two problem that it dose exploration when it should not and it doesn't exploration enough when it should do enough. For example, Even if the agent thoroughly experience some situation, it still conduct the exploration with very low probability. These exploration can cause critical result likes crash in autonomous driving or losing life in games. Also the agent conduct almost the only exploitation when it run into unfamiliar state later in the training. It can make the agent impossible to improve more. In this paper, we introduce a novel exploration method for deep reinforcement learning which is named q-entropy. In our method, the agent decide the ratio between exploration and exploitation according to the entropy of q-values. Thus it can eliminate the chance of exploration at the state that is experienced enough. Moreover, it makes the agent to do exploration when the agent encounter the new situation. We achieved considerable improvement on an Atari game, the Pong.

II. Methodology

With the q-entropy algorithm, the agent decide whether to exploit or to explore based on entropy of q-values and how many times the agent visit the state. The proposed network

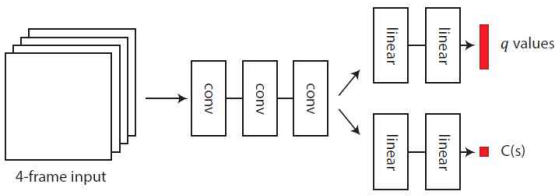

Figure 1 Proposed q-entropy network

architecture has two streams to separately estimate q values for each actions and visit counter C(s) as shown in fig. 1.

2.1 q-entropy

At each state, the agent estimates q-values for each actions and calculate the entropy of q-values with equation (1).

$$Q_{entropy} = -\sum_{i=0}^{A} \frac{\exp(q_i)}{Q} \log(\frac{\exp(q_i)}{Q}) \qquad (1)$$

Where A is number of total actions a and Q is sum of all q_is. If the value of q-entropy is larger than the predefined threshold value, $\alpha \cdot \max Q_{\text{entropy}}$, it means that the agent is uncertain about which action to do. Thus the agent conduct random action to explore state space. On the other hand, the q-entropy is less than threshold, the agent can be regarded as having confidence about which is the best action. In this case, the agent does only greedy action based on q-values.

2.2 Visit-counter

Depending on game, some states has similar true value for each q(s; a). If the agent experienced these states enough, it need not to conduct exploration anymore even if the q-entropy is high. Thus the agent can distinguish whether the state is familiar or not. We adopt visit-counter C(s) for state which can estimate how many times the agent visit each state in the past. We use similar method with [2] but we need only counter for a states, not an action a. We used sigmoid function for last activation function in C(s) path in network and parameters of last fully-connected layer for C(s) is initialized to 0 to set the C(s) value to 0.5 for any input state. C(s) is trained on dummy MDP that is same for original MDP except for true values. The true values for all state-action pairs in dummy MDP is 0, that is, r_t for every time step t is 0. Thus, whenever the agent visit the state s_t, $C(s_t)$ is decreased to 0.

2.3 Loss function

The loss function is composed of two part, L_q and L_C. L_q is squared error between target q-value and estimated q-value.

Where s_t is state at time step t and $Q(s_t; a)$ is

Algorithm 1 Q-Entropy

Input: state s_t, action space A, α, β
  estimate q-values
  calculate $Q_{entropy}$
  if q-entropy $> \alpha \max Q_{entropy}$:
    if $C(s) > \beta$:
      choose random action a_t
    else:
      choose greedy action a_t
  else :
    choose random action a_t
Output: a_t

$$L_q = (r + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a))^2 \qquad (2)$$

output of Q-network at state st and action a.

$$L_C = (\gamma C(s_{t+1}) - C(s_t))^2 \qquad (3)$$

$$L = L_q + L_C \qquad (4)$$

2.4 Policy

The final policy is shown in algorithm 1. The action is selected according to both of q-entropy and C(s).

Ⅲ. Experimental Result

As shown in the Fig.2, the q-entropy method exceed the others on the Pong. The NoisyNet [3] get highest score at the early stages, but q-entropy method rapidly outperform and converge to the max score soon.

Ⅳ. Conclusion

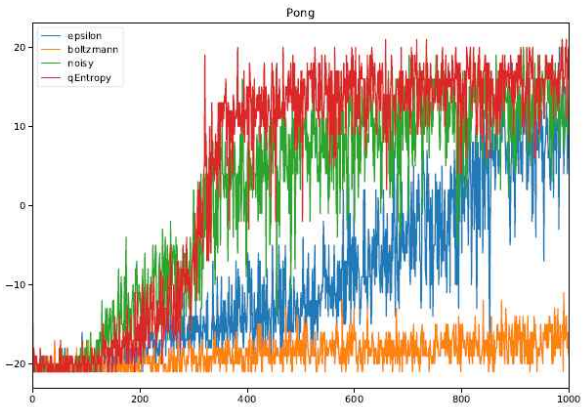

Figure 2 Rewards at each episode in Pong

We have presented a novel method for exploration in deep reinforcement learning that shows performance improvements on an Atari game Pong than conventional methods. The proposed method control the ratio of exploration to exploitation according to the uncertainty of current state. We will perform the experiments on more complicated environment.

References

[1] Mnih, et al., Human-level control through deep reinforcement learning. Nature, 2015.

[2] Choshen, et al., Dora the explorer: Directed outreaching reinforcement action-selection. 2018.

[3] Fotunato, et al., Noisy networks for exploration. 2017.