HCIR Lab Seminar

A distributional Perspective on Reinforcement Learning

Marc G.Bellemare, Will Dabney, Remi Munos
Deep Mind

PMLR 2017

2019.05.04

Presented by Soo-Han Kang

Dept. of Computer Science and Engineering

SEOULTECH

Contents

Summary

Summary

- RL은 Value Function이 Expectation

$$Q_\pi(x,a) = E[\,G_t \mid X_t = x, A_t = a]$$

- Bellman Equation

$$Q(x,a) = ER(x,a) + \gamma EQ(X',A')$$

- Value Function을 Distributional Perspective로 보자!

$$Q_\pi(x,a) \coloneqq EZ_\pi(x,a) = E[\sum_{t=0}^{\infty} \gamma^t R(x_t,a_t)]$$

- Bellman Equation

$$Z(x,a) \stackrel{D}{=} R(x,a) + \gamma Z(X',A')$$

- 이렇게 Bellman Equation으로 표현가능 하면, Q-Learning이나 Sarsa를 사용할 수 있겠다!

Motivation

Motivation

- RL은 Value Function이 Expectation

$$Q_\pi(s,a) = E[\, G_t \mid S_t = s, A_t = a]$$

- Distributional Perspective RL이 존재 하였으나,
  - To model parametric uncertainty
  - To design risk-sensitive algorithms
  - Theoretical analysis

- 이 논문은 이렇게 한 측면으로의 Distributional Perspective RL이 아닌 Value Distribution을 사용

Motivation

- Contraction of the policy evaluation Bellman Operator
  - Rosler(1992)의 연구를 바탕으로 Fixed Policy에서, Value Distribution에 대한 Bellman Operator의 Contraction은 Wasserstein Metric을 최대화 하는 것

- Instability in the control setting
  - Distributional Bellman Optimality Equation에서의 Instability를 증명
  - Expectation value로 optimality operator를 축약 하였을 때, Distribution에 대한 metric의 축약이 아님

- Better Approximations
  - 알고리즘 관점으로 볼 때 Expectation 근사를 할 때보다 Distribution을 근사 할 때 많은 이점
  - Distributional 관점에서는 Value의 Multimodality를 보존(안정적인 학습을 한다고 믿음)
  - Nonstationary Policy에 대한 학습효과를 완화

Main Method

Main Method

- Bellman Optimality Equation은 unique한 Fixed Point가 Q*가 존재
- 이러한 Bellman Equation Contraction을 Bellman Operator로 표현 가능

$$\tau^{\pi} Q(x,a) := ER(x,a) + \gamma EQ(x',a')$$
$$\tau Q(x,a) := ER(x,a) + \gamma E_p \max_{a` \in A} Q^*(x',a')$$

- Our first aim is to gain an understanding of the theoretical behaviour of the distributional analogues of the Bellman operators, in particular in the less well-understood control setting. The reader strictly interested in the algorithmic contribution may choose to skip this section.

- 3장 내용은 우리가 기존에 사용했던 Q-Value에 대한 Bellman Equation에 Distribution Value인 Z로 Q-Value대신 사용할 수 있음을 증명하는 파트

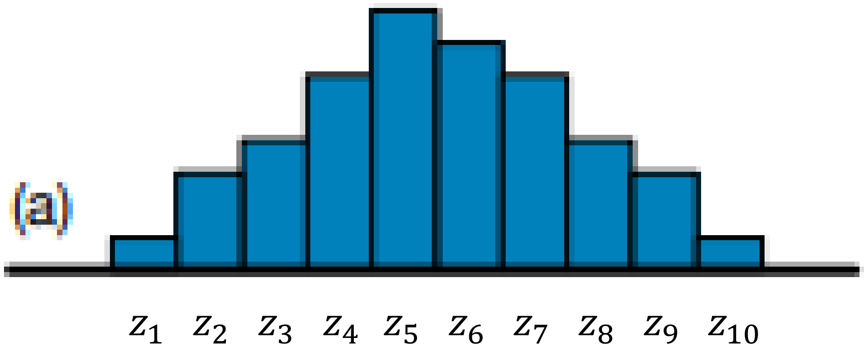
Parametric Distribution

- 이 논문에서는 Distributional을 표현하기 위해 support(N)라는 파라미터를 사용(z를 atom이라 표현)

$$\{z_i = V_{\min} + i\Delta z : 0 \le i < N\}, \Delta z := \frac{V_{max} - V_{min}}{N-1}$$

- 이러한 atom은 어떤 의미에서 분포에 대한"표준적인 결과 "

$$Z_\theta(x,a) = z_i \quad w.p.(with\ probability\ 1)\ \ p_i(x,a) := \frac{e^{\theta_i(x,a)}}{\Sigma e^{\theta_j(x,a)}}$$

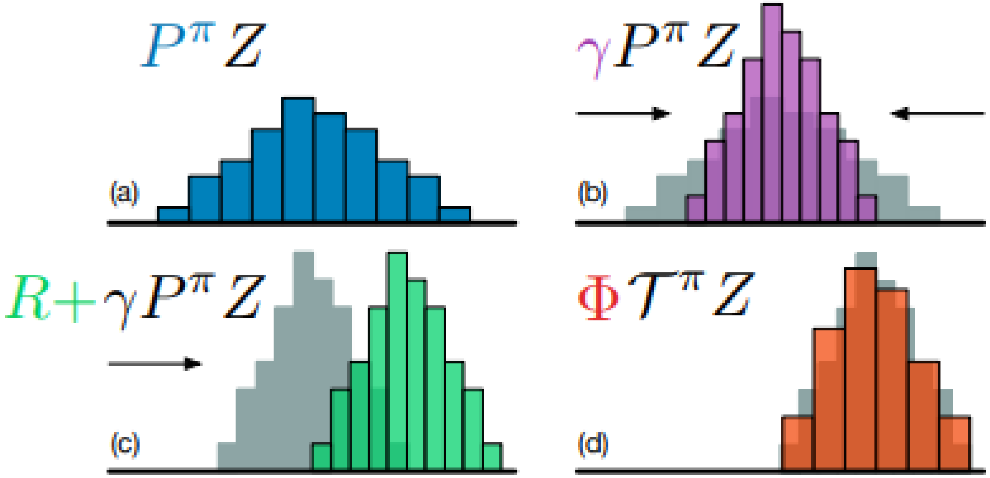


Projected Bellman Update

- 하지만 이렇게 Discrete 분포를 사용하면 문제가 발생
- Bellman Update τZ_θ와 Z_θ의 Support가 불일치
- 여기서 Section 3의 분석을 토대로 τZ_θ와 Z_θ의 Wasserstein Metric을 Minimize하는 것이 매우 Natural

- 하지만 두번째 문제가 발생
- Wasserstein Loss는 Sample transition에서 사용할 수가 없음

- 대신에 sample Bellman Update τZ_θ를 Support Z_θ에 Projection



Projected Bellman Update

- 이러한 Projection은 Bellman Update를 Multi Class Classification으로 감소

$$\pi : Greedy\ Policy(:= EZ_\theta)$$
$$Given\ (x, a, r, x')$$
$$\tilde{\tau} z_j := r + \gamma z_j$$
$$for\ each\ atom\ z_j\ distribution\ probability\ p_j\big(x', \pi(x')\big)$$
$$i^{th}\ component\ of\ the\ projected\ update\ \Phi\tilde{\tau}Z_\theta(x, a)$$
$$\big(\Phi\tilde{\tau}Z_\theta(x, a)\big)_i = \sum_{j=0}^{N-1} \left[1 - \frac{\left|\left[\tilde{\tau} z_j\right]_{V_{min}}^{V_{max}} - z_i\right|}{\Delta z}\right]_0^1 p_j\big(x', \pi(x')\big)$$
$$\text{Loss } \mathrm{L}_{\mathrm{x,a}}(\theta) = D_{KL}(\Phi\tilde{\tau}Z_\theta(x, a) || Z_\theta(x, a))$$

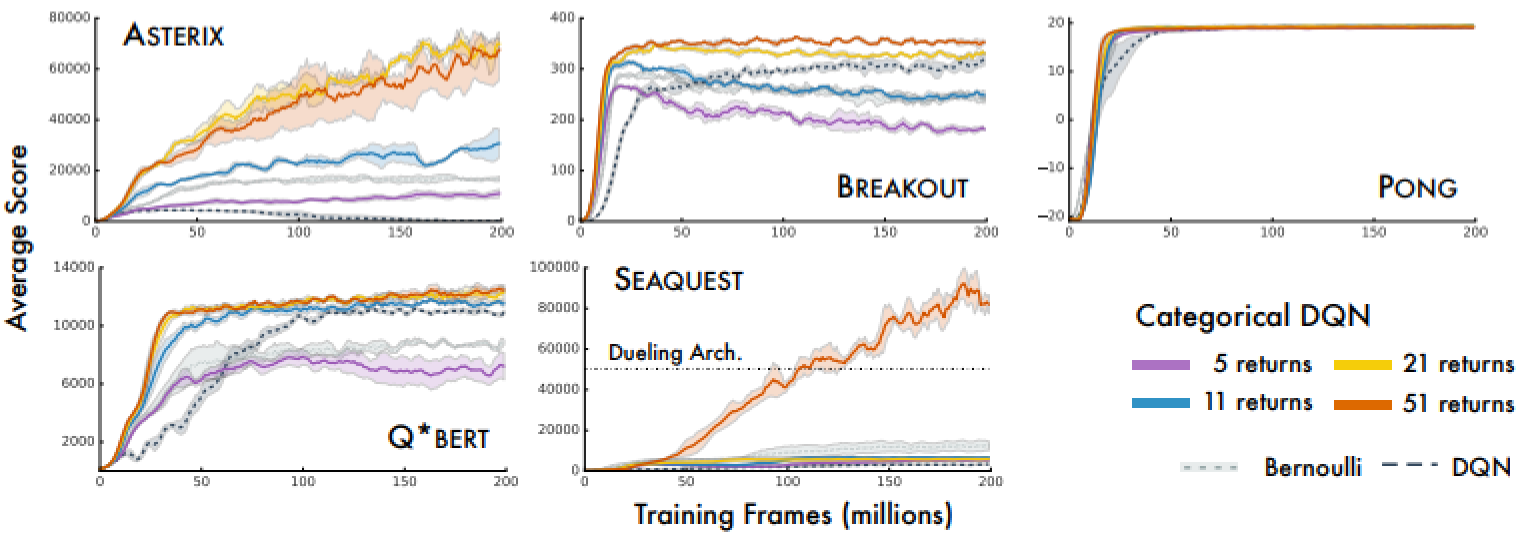
Projected Bellman Update

Algorithm 1 Categorical Algorithm

input A transition $x_t, a_t, r_t, x_{t+1}, \gamma_t \in [0, 1]$
 $Q(x_{t+1}, a) := \sum_i z_i p_i(x_{t+1}, a)$
 $a^* \leftarrow \arg\max_a Q(x_{t+1}, a)$
 $m_i = 0, \quad i \in 0, \dots, N-1$
 for $j \in 0, \dots, N-1$ **do**
  # Compute the projection of $\hat{\mathcal{T}} z_j$ onto the support $\{z_i\}$
  $\hat{\mathcal{T}} z_j \leftarrow [r_t + \gamma_t z_j]_{V_{\text{MIN}}}^{V_{\text{MAX}}}$
  $b_j \leftarrow (\hat{\mathcal{T}} z_j - V_{\text{MIN}})/\Delta z$ # $b_j \in [0, N-1]$
  $l \leftarrow \lfloor b_j \rfloor, u \leftarrow \lceil b_j \rceil$
  # Distribute probability of $\hat{\mathcal{T}} z_j$
  $m_l \leftarrow m_l + p_j(x_{t+1}, a^*)(u - b_j)$
  $m_u \leftarrow m_u + p_j(x_{t+1}, a^*)(b_j - l)$
 end for
output $-\sum_i m_i \log p_i(x_t, a_t)$ # Cross-entropy loss

Result

Result



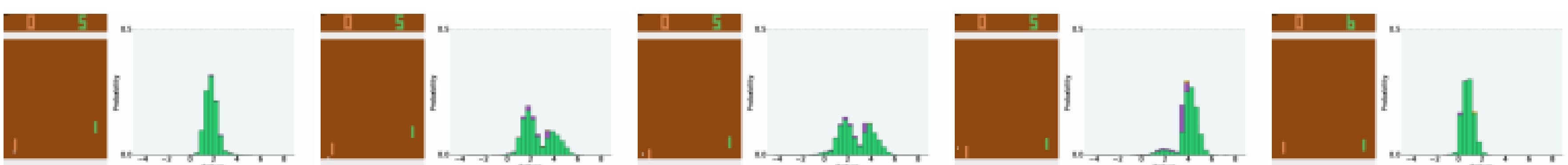
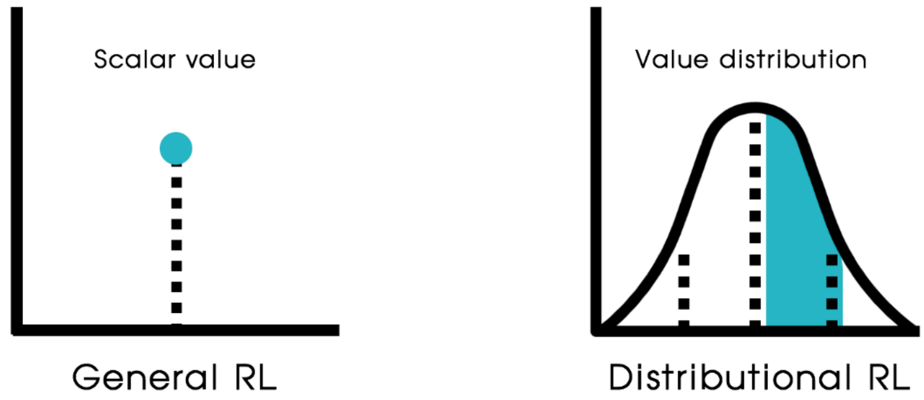
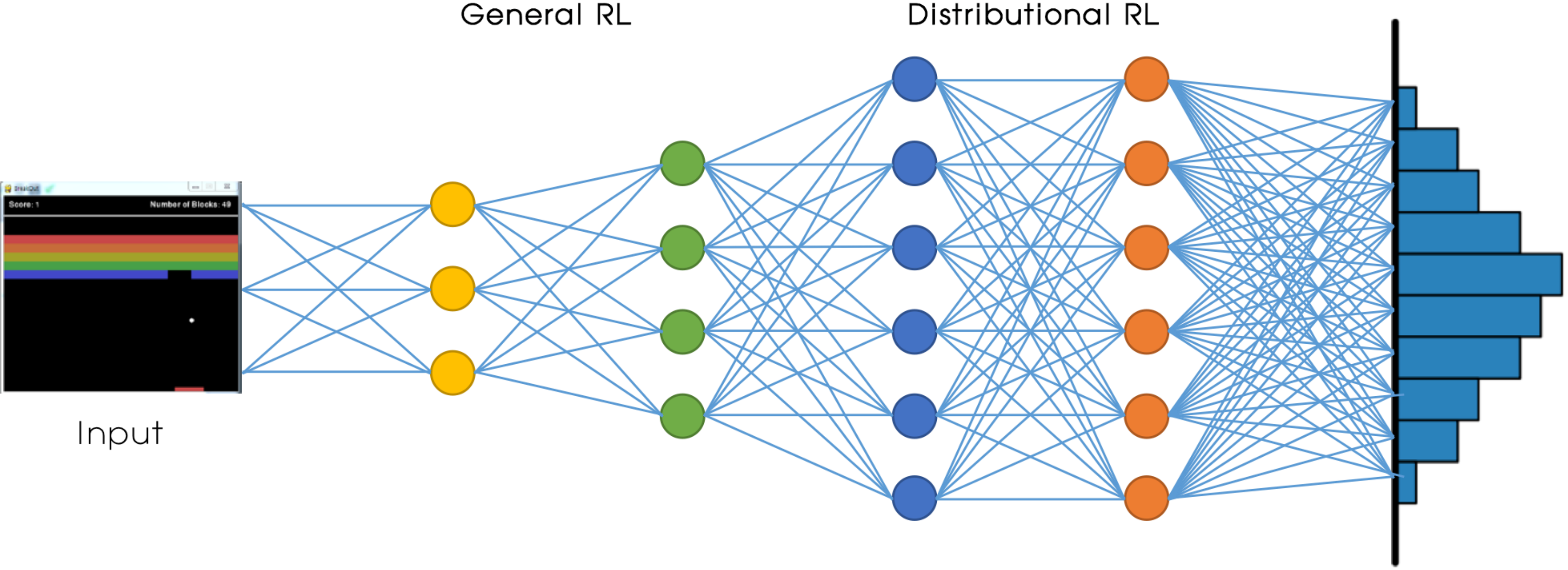


Figure 5. Intrinsic stochasticity in PONG.

Summary

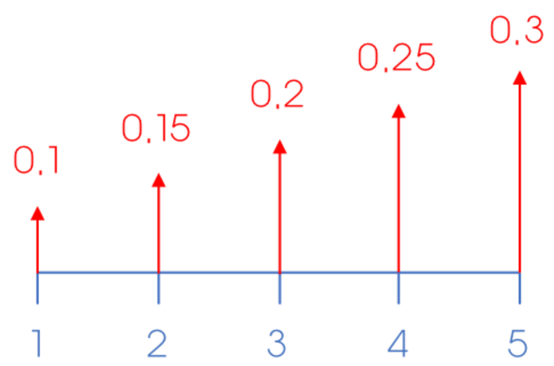
❑ 추후에 Wasserstein Loss를 사용함으로 QR-DQN -> IQN



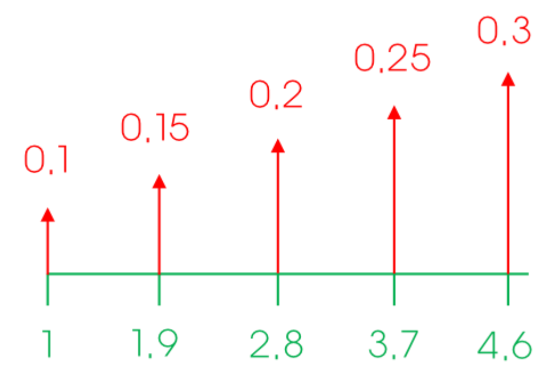


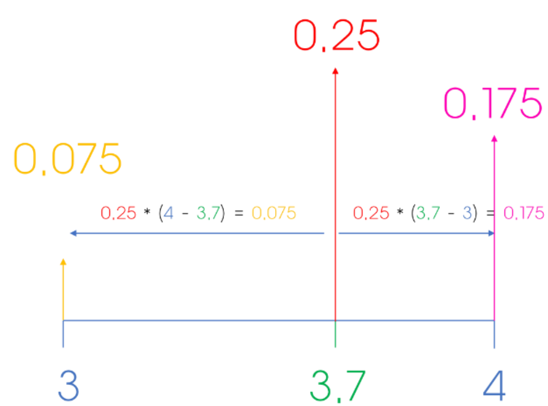
Summary

- Support: [1, 2, 3, 4, 5]
- Probability: [0.1, 0.15, 0.2, 0.25, 0.3]
- Reward: 0.1
- Discount factor: 0.9

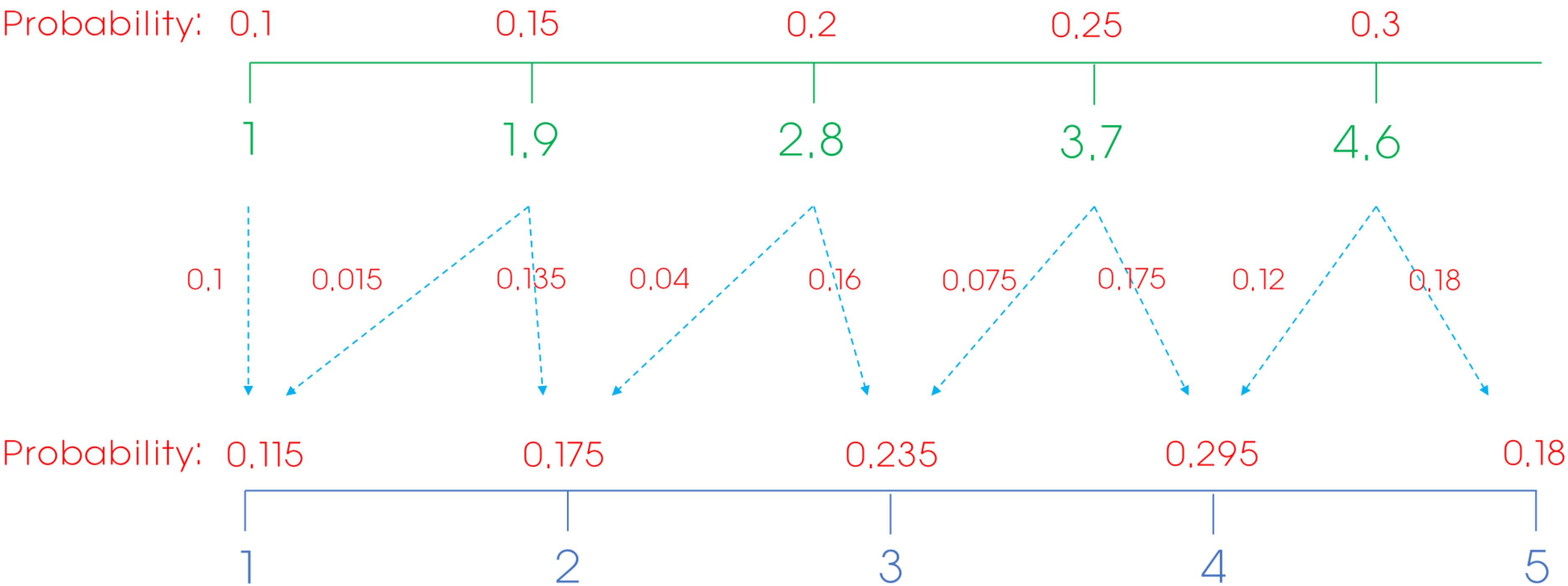


$$\hat{\mathcal{T}} z_i \leftarrow [r_t + \gamma z_i]_{V_{min}}^{V_{max}}$$





Summary


Probability: 0.1
0.15
0.2
0.25
0.3
1
1.9
2.8
3.7
4.6
0.1
0.015
0.135
0.04
0.16
0.075
0.175
0.12
0.18
Probability: 0.115
0.175
0.235
0.295
0.18
1
2
3
4
5

Summary

Algorithm 1 Categorical Algorithm

input A transition $x_t, a_t, r_t, x_{t+1}, \gamma_t \in [0, 1]$

$Q(x_{t+1}, a) := \sum_i z_i p_i(x_{t+1}, a)$

$a^* \leftarrow \arg\max_a Q(x_{t+1}, a)$

$m_i = 0, \quad i \in 0, \ldots, N-1$

for $j \in 0, \ldots, N-1$ **do**

Compute the projection of $\hat{\mathcal{T}} z_j$ onto the support $\{z_i\}$

$\hat{\mathcal{T}} z_j \leftarrow [r_t + \gamma_t z_j]_{V_{\text{MIN}}}^{V_{\text{MAX}}}$

$b_j \leftarrow (\hat{\mathcal{T}} z_j - V_{\text{MIN}})/\Delta z$ # $b_j \in [0, N-1]$

$l \leftarrow \lfloor b_j \rfloor, u \leftarrow \lceil b_j \rceil$ $\longrightarrow$ l: 버림, u: 올림

Distribute probability of $\hat{\mathcal{T}} z_j$

$m_l \leftarrow m_l + p_j(x_{t+1}, a^*)(u - b_j)$

$m_u \leftarrow m_u + p_j(x_{t+1}, a^*)(b_j - l)$

end for

output $-\sum_i m_i \log p_i(x_t, a_t)$ # Cross-entropy loss

- Distribution의 기대값을 통해 Q fuction 을 계산
- Q값을 최대로 하는 action 선택
- Target distribution 계산
- Target distribution 각각에 해당하는 support 결정
- Target distribution 을 기존 support 에 분배
- Target distribution 과 예측된 distribution 간 cross entropy loss