MBC: 오프라인 강화학습 기반 거대 목표-제약 행동 모방 알고리즘

장현성, 김정현

세종대학교

23012724@sju.ac.kr, j.kim@sejong.ac.kr

MBC: Massive Goal-Conditioned Behavior Cloning Algorithm Based on Offline Reinforcement Learning

Hyeonseong Chang, Junghyun Kim

Sejong Univ.

요 약

본 논문은 오프라인 강화학습 알고리즘인 Goal-Conditioned Supervised Learning (GCSL)의 성능 한계를 극복하기 위해 심화된 정책 신경망 구조와 초구체 정규화를 적용한 모델을 제안한다. 기존 벤치마크와 달리 고수준 작업을 평가하는 OGBench의 Antmaze-medium 환경에서 5개 작업을 수행한 결과, 제안 모델은 GCSL 대비 모든 작업에서 작업 성공률이 크게 향상되었으며 평균 22.12%p의 개선을 달성하였다. 이는 정규화를 통한 데이터 분포의 보존과 신경망의 일반화 성능 향상에 기인한다.

Ⅰ. 서론

로봇의 움직임을 제어하기 위해선 로봇 관절의 회전 각도와 토크, 상대적 위치 등을 계산해야 하는데, 이를 일일이 제어하기란 쉽지 않다 [1]. 그래서 상대적으로 간단한 신경망을 사용하여 로봇의 관절을 제어하는 강화학습 알고리즘이 각광 받고 있다 [1]. 강화학습은 상태나 관찰값이 신경망으로 모델링 된 정책 함수에 입력되면 적절한 행동을 생성한다.

강화학습은 크게 온라인 강화학습과 오프라인 강화학습 두 가지로 구분할 수 있다. 두 방법 모두 로봇을 에이전트로 만들어서 환경과 상호작용한 경험을 통해 학습한다는 공통점이 있다. 온라인 강화학습은 에이전트가 실시간으로 환경과 상호작용하여 경험을 스스로 생성하지만, 오프라인 강화학습은 이미 생성된 경험을 사용한다는 차이점이 있다.

전통적으로 온라인 강화학습은 유명 알고리즘인 Proximal Policy Optimization [2]나 Soft Actor Critic [3]과 같이 에이전트가 좋은 경험을 생성하여 정책 함수를 효율적으로 업데이트하는 방법을 주로 연구하였다. 하지만 에이전트가 직접 경험을 생성하는 방법은 알고리즘 구조에 굉장히 민감하고 정교한 구현이 요구되어 모델링 하는 것이 매우 어렵다 [4]. 따라서 이와 같은 어려움을 해결하기 위해 대안으로 오프라인 강화학습이 제안되었다 [5].

오프라인 강화학습은 사람이 직접 상호작용한 데이터나 정교한 조작을 통해 생성된 데이터를 에이전트의 경험으로 사용한다. 결과적으로, 오프라인 강화학습은 경험 생성에 대한 부담이 줄어들고, 모델링이 온라인 강화학습보다 수월해진다는 장점이 있다.

본 논문에서는 Goal-Conditioned Supervised Learning (GCSL) [5] 알고리즘을 비교 대상 모델로 사용한다. GCSL은 여러 가지 오프라인 강화학습 중에서도 특정 목표를 달성하기 위해 수집된 경험 데이터를 사용하여 정책을 업데이트하는 알고리즘이다. 한편, GCSL은 정책 함수를 업데이트하기 때문에 정책 신경망 구조에 따라 성능에 한계가 존재한다. 실제로 GCSL의 신경망 구조는 3층에 불과하다. 따라서 본 논문에서는 향상된 성능을 얻기 위해 GCSL의 정책 신경망 구조를 정교하게 심층화한 Massive Goal-Conditioned Behavior Cloning (MBC)을 제안한다. MBC는 신경망의 심층화와 더불어 초구체 정규화를 적용한 구조 [6]을 사용하였다.

Ⅱ. 본론

본 논문에서는 에이전트와 환경을 정의하고 성능을 비교하기 위해 OGBench [7]를 사용하였다. OGBench는 기존 벤치마크보다 난이도가 높은 고수준 작업을 포함하므로, 다양한 환경에 적응 가능한 에이전트를 모델링 하기에 적합하다. OGBench에는 수많은 작업이 존재하는데, 본 논문에서는 4족 보행 로봇 Ant가 중간 크기의 미로 속에서 목표지점까지 이동하는 Antmaze-medium 환경을 사용하였다. Antmaze-medium 환경의 예시는 그림 1에 제시되어 있다. Ant 로봇이 가로, 세로 15m의 정사각형 미로를 돌아다니며 시작 지점에서 목표지점까지 이동하는 서로 다른 5가지 작업을 수행한다.

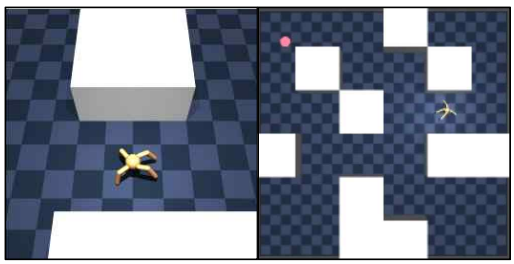

그림 1. OGBench의 Antmaze-medium 환경 예시.

GCSL 알고리즘은 오프라인 데이터에 기반하여 단일 정책 신경망을 학습하고 이를 통해 로봇을 제어한다. OGBench에서 제공하는 오프라인 데이터셋은 품질이 높기 때문에 알고리즘의 성능은 정책 신경망이 데이터셋의 행동 확률 분포를 얼마나 잘 반영하는지에 따라 크게 좌우된다. 따라서 본 논문에서는 GCSL 정책 신경망 구조의 은닉층을 깊게 설계해 더 정교한 확률 분포를 학습하도록 설계하였다. 그러나 은닉층을 단순히 깊게 쌓는 방식은 과적합을 유발할 수 있으므로, [6]의 초구체 정규화를 적용하여 이를 방지하였다.

표 1. GCSL과 제안 모델 (MBC)의 작업별 성공률 비교.

알고리즘	작업 1 성공률	작업 2 성공률	작업 3 성공률	작업 4 성공률	작업 5 성공률	작업 성공률 평균
GCSL [5]	0.360±0.124	0.225±0.093	0.218±0.075	0.256±0.091	0.465±0.165	0.305±0.102
MBC	**0.578±0.204**	**0.526±0.192**	**0.407±0.149**	**0.536±0.212**	**0.683±0.231**	**0.546±0.193**

초구체 정규화는 데이터 $\mathbf{x}\in\mathbb{R}^n$가 주어졌을 때, 이를 $\mathbb{R}^{n+1}$의 고차원 공간에 매핑 (mapping)한 후, L_2-정규화를 적용하는 것이다. 초구체 정규화를 적용하면 스케일 차이로 인해 무시될 수 있는 데이터 구분이 보존되어, 신경망의 표현력이 향상된다 [6]. 초구체 정규화를 하기 위해 먼저 가공되지 않은 형태의 관찰값에 이동 통계 정규화 (Running Statistics Normalization, RSNorm)를 적용한다. 이동 평균과 표준편차가 수식 (1)과 같을 때, RSNorm은 수식 (2)와 같다.

$$\mu_t=\mu_{t-1}+(1/t)\delta_t,\ \sigma_t^2=((t-1)/t)(\sigma_{t-1}^2+(1/t)\delta_t^2), \quad (1)$$

$$\overline{\mathbf{o}}_t=\mathrm{RSNorm}(\mathbf{o}_t)=\frac{\mathbf{o}_t-\mu_t}{\sqrt{\sigma_t^2+\epsilon}}, \quad (2)$$

t는 타임 스텝이고 δ_t는 $\delta_t=\mathbf{o}_t-\mu_{t-1}$로 현재와 이전의 관찰값 평균의 차이다. 그다음 이동 평균 정규화가 적용된 데이터를 고차원 공간으로 매핑하고 L_2-정규화를 진행한다:

$$\tilde{\mathbf{o}}_t=\mathrm{L_2-Norm}([\overline{\mathbf{o}}_t;\mathbf{c}]), \quad (3)$$

$\mathbf{c}$는 하나의 원소만 0이 아니고 나머지는 모두 0인 벡터이다. 고차원 매핑은 두 벡터를 연접 (concatenate)하여 수행한다. 일반적인 신경망은 편향 b가 추가되지만, 본 논문에서는 학습 가능한 파라미터로 구성된 스케일링 벡터 $\mathbf{s}_{\mathrm{h}}^{1}\in\mathbb{R}^{d_{\mathrm{h}}}$를 사용한다. 가중치가 $\mathbf{w}_{\mathrm{h}}^{1}\in\mathbb{R}^{(|O|+1)\times d_{\mathrm{h}}}$일 때, 첫 번째 은닉층의 출력은 다음과 같다:

$$\mathbf{h}_t^0=\mathrm{L_2-Norm}(\boldsymbol{s}_h^0\odot(\tilde{\mathbf{o}}_t^{\top}\mathbf{w}_{\mathrm{h}}^0)), \quad (4)$$

$\odot$은 벡터 또는 행렬의 원소별 곱셈을 의미한다.

수식 (1)~(3)은 입력 임베딩에 해당한다. 임베딩된 값은 수식 (4)와 (5)로 표현되는 블록으로 들어가 N번 반복된다. 각 블록은 차원이 4배로 늘어났다가 원래대로 돌아오는 inverted bottleneck 구조의 비선형 변환을 사용했다. 가중치가 각각 $\mathbf{w}_{\mathrm{h},1}^{l}\in\mathbb{R}^{d_{\mathrm{h}}\times 4d_{\mathrm{h}}}$, $\mathbf{w}_{\mathrm{h},2}^{l}\in\mathbb{R}^{4d_{\mathrm{h}}\times d_{\mathrm{h}}}$이고 스케일링 벡터 $\mathbf{s}_{\mathrm{h}}^{l}\in\mathbb{R}^{4d_{\mathrm{h}}}$가 주어졌을 때, 중간 출력은 다음과 같다:

$$\tilde{\mathbf{h}}_t^l=\mathrm{L_2-Norm}(\mathrm{ReLU}(\mathbf{s}_{\mathrm{h}}^l\odot(\mathbf{h}_t^{l^{\top}}\mathbf{w}_{\mathrm{h},1}^{l}))\mathbf{w}_{\mathrm{h},2}^{l}). \quad (5)$$

마지막으로 블록의 최종 출력은 두 데이터를 보간한 뒤에 초구체 상으로 투영하는 방법으로 계산된다:

$$\mathbf{h}_t^{l+1}=\mathrm{L_2-Norm}((\mathbf{1}-\boldsymbol{\alpha}^l)\odot\mathbf{h}_t^l+\boldsymbol{\alpha}^l\odot\tilde{\mathbf{h}}_t^l), \quad (6)$$

$\mathbf{1}\in\mathbb{R}^{d_h}$과 $\boldsymbol{\alpha}^l\in\mathbb{R}^{d_h}$는 각각 단위 벡터와 보간 벡터를 의미한다.

데이터 간 관계를 명확히 반영하는 값을 이용하여 행동 확률 분포를 예측하는 단계에서는 관찰값과 행동을 동시에 고려하여 Q값을 추정해야 한다. 본 논문에서는 Q값 추정을 위해 [6]에서 사용한 것과 같이 Distributional Reinforcement Learning 알고리즘 [8]을 사용하였다.

실험 결과는 표 1에 제시하였다. 서로 다른 랜덤 시드로 총 8회 실험되었고, 실험 결과는 평균과 표준편차의 형태로 나타냈다. 최고 성능은 볼드체로 표시하였다. 표 1에서 한눈에 알 수 있듯이 제안 모델인 MBC는 모든 작업에서 큰 폭의 성공률 향상을 달성했다. 작업별 평균 성공률을 기준으로 작업 1부터 5까지 각각 성공률이 21.8%p, 30.1%p, 18.9%p, 18.0%p, 21.8%p만큼 향상되었다. 더불어 전체 작업에 대한 평균 성공률은 22.12%p 향상되었다. 모든 작업에서 성공률이 개선되었다는 것은 신경망 구조의 일반화 성능이 증가했음을 의미한다. 즉, 은닉층이 깊어지고 초구체 정규화가 잘 적용되었기 때문에 정책 신경망이 데이터의 특정 행동 확률 분포에 치우쳐 과적합되지 않았다고 해석할 수 있다.

Ⅲ. 결론

본 논문에서는 오프라인 강화학습 알고리즘인 GCSL의 정책 신경망 구조를 기존보다 심층화하고, 초구체 정규화를 적용하여 성능을 개선한 MBC를 제안하였다. OGBench의 Antmaze-medium 환경에서 5가지 작업을 대상으로 실험을 수행한 결과, 제안 모델은 GCSL 대비 모든 작업에서 성능이 크게 향상되었으며, 평균 22.12%p의 성공률 개선을 보였다. 이는 단순히 은닉층을 깊게 설계하는 것뿐만 아니라 초구체 정규화를 통해 데이터의 확률 분포를 효과적으로 보존하고, 정책 신경망이 안정적으로 행동 확률을 근사할 수 있었기 때문이다. 향후 연구로는 학습 시간을 효율적으로 단축하는 방안에 대해 탐구할 계획이다.

ACKNOWLEDGMENT

이 논문은 2023년도 정부 (교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (RS-2023-00271991).

참 고 문 헌

[1] Zhang, T. and Mo, H., "Reinforcement learning for robot research: A comprehensive review and open issues," *Int. J. Adv. Robotic Syst.*, vol. 18, no. 3, pp. 1-22, Jun. 2021.

[2] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O., "Proximal policy optimization algorithms," *arXiv preprint* arXiv:1707.06347, 2017.

[3] Haarnoja, T., Zhou, A., Abbeel, P. and Levine, S., "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Machine Learning (ICML)*, 2018.

[4] Lu, M., Zhong, H., Zhang, T. and Blanchet, J., "Distributionally robust reinforcement learning with interactive data collection: Fundamental hardness and near-optimal algorithms," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024.

[5] Ghosh, D., Gupta, A., Reddy, A., Fu, J., Devin, C., Eysenbach, B. and Levine, S., "Learning to reach goals via iterated supervised learning," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2021.

[6] Lee, H., Lee, Y., Seno, T., Kim, D., Stone, P. and Choo, J., "Hyperspherical normalization for scalable deep reinforcement learning," in *Proc. Int. Conf. Machine Learning (ICML)*, 2025.

[7] Park, S., Frans, K., Eysenbach, B. and Levine, S., "Ogbench: Benchmarking offline goal-conditioned RL," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2025.

[8] Bellemare, M. G., Dabney, W. and Munos, R., "A distributional perspective on reinforcement learning," in *Proc. Int. Conf. Machine Learning (ICML)*, 2017.