RLS 기반 Actor-Critic 학습을 이용한 로봇이동

Robot Locomotion via RLS-based Actor-Critic Learning

김종호, 강대성, 박주영
고려대학교 제어계측공학과
Jongho Kim, Daesung Kang, Jooyoung Park
Dept. of Control & Instrumentation Engineering, Korea University
E-mail:{oyeasw,mpkds,parkj}@korea.ac.kr

요약

강화학습을 위한 많은 방법 중 정책 반복을 이용한 actor-critic 학습 방법이 많은 적용 사례를 통해서 그 가능성을 인정받고 있다. Actor-critic 학습 방법은 제어입력 선택 전략을 위한 actor 학습과 가치 함수 근사를 위한 critic 학습이 필요하다. 본 논문은 critic의 학습을 위해 빠른 수렴성을 보장하는 RLS(recursive least square)를 사용하고, actor의 학습을 위해 정책의 기울기(policy gradient)를 이용하는 새로운 알고리즘을 제안하였다. 그리고 이를 실험적으로 확인하여 제안한 논문의 성능을 확인해 보았다.

키워드 : Actor-Critic, RLS, policy gradient, 강화학습

1. 서론

강화학습은 지도학습과 비지도 학습의 중간적인 특성을 가지고 있어 시도와 오류를 통해서 정책이 결정되기 때문에 다루고자 하는 대상의 구체적인 모델이 필요없는 장점을 가지고 있다. 강화학습의 주요 방법 중 하나인 actor-critic방법은 정책 반복을 통하여 actor와 critic의 파라미터 개선을 하며, 개선된 파라미터들은 다음상태 제어 입력을 선택하는데 이용된다. Critic학습은 정책의 실행에 관련된 부분으로 일반적으로 현재 상태와 다음상태의 가치의 차에 의해서 계산되며 계산된 값들은 actor의 학습에 이용된다. Actor학습은 정책의 조정과 관련된 부분으로 최적의 제어 입력을 선택하는 과정이다. 본 논문에서는 정책 기울기 방법과 RLS 기법을 이용하여 actor와 critic의 학습을 대체하는 새로운 알고리즘을 제안하였다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 정책 기울기 방법을 소개하고 critic학습을 위해 RLS 기법을 적용한 수정된 actor-critic 학습 방법을 제안한다. 3장에서는 제안된 학습방법을 로봇에 적용했을 경우에 대한 결과를 설명하고 마지막 4장에서는 토의 및 향후 연구 방향을 제시한다.

2. Policy gradient- RLS기법을 이용한 학습

2.1 RLS기법을 이용한 critic 학습

참고 논문[4][5]에서 언급된 것처럼, 최소 자승법(least square)은 데이터의 비효율적인 활용과 학습에 사용되는 파라미터 선택의 문제를 해결하기 위한 학습 방법이다. 일반적으로 최소 자승법은 다음과 같은 선형 시스템의 해를 찾는 문제로 접근할 수 있다.

$$A W + b = 0 \ , \quad W\text{는 연결강도벡터} \qquad (2.1)$$
$$(A \in R^{K\times K},\ b \in R^{K} \ , \ K\text{는 feature vector의 수})$$

식 2.1의 해는 $W_t = -A_t^{-1} b_t$와 같고, [4][5]의 LSTD(λ)를 이용한 A, b의 적격성을 고려한 해의 형태는 다음과 같다.

$$\begin{aligned} b_t &:= b_t + z_t r_t \\ A_t &:= A_t + z_t(\phi(x_t) - \phi(x_{t+1}))' \\ z_{t+1} &:= \lambda z_t + \phi(x_t) \end{aligned} \qquad (2.2)$$

z_t는 적격성 트레이스 벡터를 ϕ_t는 기저 벡터를 나타낸다. 각 step t 에서 A_t를 연산하기 위해 초기 매트릭스 A의 값을 다음과 같이 나타낼 수 있다.

$$A_0 = \delta I + \Phi(x_t)(\gamma\Phi(x_{t+1}) - \Phi(x_t)) \qquad (2.3)$$

한편, critic이 다루고자 하는 목적 함수는 아래와 같은 형태를 갖는다.

$$J=\| \sum_{t=1}^{T} A(X_t) W - \sum_{t=1}^{T} b(X_t) \|^2 \quad (2.4)$$

즉, critic은 식(2.4)의 에러값을 최소화 하기 위해 파라미터 벡터 W를 개선하게 된다. Critic의 목적 함수가 다루는 값은 Bellman 방정식의 형태로 나타난다.

$$Q^{\pi}(x,\mu) = r(x,u) + \gamma \int_X p(x'|x,\mu) V^{\pi}(x') dx' \quad (2.5)$$

참고 논문[2]에서 가치함수와 입력 가치함수의 차를 advantage value function으로 정의하고 다음과 같이 표현하였다.

$$A^{\pi}(x,\mu) = Q^{\pi}(x,\mu) - V^{\pi}(x) \quad (2.6)$$

Critic은 식(2.5)와 식(2.6)사이의 차이를 최소화 하기위한 방향으로 파라미터벡터를 개선한다. 한편 식(2.6)은 compatible function으로 근사되며 각 step 마다 가치함수 $V^{\pi}(s)$는 기저벡터와 현재 상태의 선형결합으로 표현된다. 근사된 에러의 값은 아래와 같이 표현가능하다.

$$\begin{aligned}
&\tilde{\Psi}_t(v,w) \\
&\triangleq \| \sum_{k=0}^{t} z_k [(\widetilde{V}_v(x_k) + \widetilde{A}_w(x_k,a_k)) - (r_k + \gamma \widetilde{V}_v(x_{k+1}))] \|^2 \\
&= \| \sum_{k=0}^{t} z_k [\phi'(x_k) - \gamma\phi'(x_{k+1}), \nabla_\theta \log\pi(a_k|x_k)' \begin{bmatrix} v \\ w \end{bmatrix} - \\
&\quad \sum_{k=0}^{t} z_k r_k \|^2
\end{aligned} \quad (2.7)$$

매트릭스 A의 역행렬을 구할 수 있으면,[3][8]의 보조 정리(lemma)를 이용하여 매트릭스 역 공식(matrix inversion formula)을 사용할 수 있다.

$$(A+BC)^{-1} = A^{-1} - A^{-1}B(I+CA^{-1}B)^{-1}CA^{-1} \quad (2.8)$$

식 (2.8)을 이용하면 식(2.7)에서 언급된 최소 자승문제를 critic은 RLS 기법으로 해를 구할 수 있다.

$$\begin{aligned}
&K_{t+1} = P_t z_t / (\mu + (\phi'(x_t) - \gamma\phi'(x_{t+1})) P_t z_t) \\
&W_{v_{t+1}} = W_{V_t} + K_{t+1}(r_t - (\phi^T(x_t) - \gamma\phi^t(x_{t+1})) W_{V_t} \\
&P_{t+1} = \frac{1}{\mu}(P_t - P_t z_t (1 + (\phi'(x_t) - \gamma\phi'(x_{t+1}) P_t z_t))^{-1} \\
&\qquad \times (\phi'(x_t) - \gamma\phi'(x_{t+1}) Pt),
\end{aligned} \quad (2.9)$$

where $P_t = A_t^{-1}$, $K_t = P_t z_t$

2.2 Actor의 학습

Actor를 위한 학습은 할인된 보상값의 함으로 표현되는 목적함수를 최대화 하기 위한 방향으로 이루어 진다.

$$J(\Theta) = \int_X d^{\pi}(x) \int_u \pi(\mu|x) r(x,u) du dx \quad (2.10)$$

Θ는 목적함수의 특징 파라미터를 나타낸다. 정책 기울기를 이용한 일반적인 학습 방법은 파라미터 공간의 기울기(gradient)를 따르는 방향으로 학습이 진행된다.

$$\Theta_{i+1} = \Theta_i + \alpha \nabla_\Theta J(\Theta_i)$$

한편 목적 함수의 기울기 방향은 아래와 같다.

$$\nabla_\Theta J(\Theta) = \int_X d^{\pi}(x) \int_U \nabla_\Theta \pi(\mu|x) \delta^{\pi}(x,\mu) d\mu dx \quad (2.11)$$

식 (2.11)을 확률적으로 샘플링하고 compatible 함수 근사를 이용한 actor의 파라미터 update 방법은 아래와 같다. 이는 참고논문 [2]에서 언급된 것처럼 actor의 학습은 평균 파라미터 벡터의 Fisher information을 이용한 것으로 볼 수 있다.

$$\Theta_{t+1} \leftarrow \Theta_t + \alpha F_{t+1}(\Theta) w_{t+1}, \quad (2.12)$$

where

$$F_{t+1} = \nabla_\Theta \log\pi(\mu_{t+1}|x_{t+1}) \nabla_\Theta \log\pi(\mu_{t+1}|x_{t+1})'$$

한편 가치함수(value function)와 compatible 함수 근사를 위해서 아래와 같은 기저함수를 사용하였다.

$$\Phi_t = [\phi(x_t)', \nabla_\Theta \log\pi(\mu_t|x_t)']', \quad \Phi_t = [\phi(x_{t+1})', 0']' \quad (2.13)$$

식 (2.13)의 기저 벡터는 제어입력 μ_t에 독립적으로 움직이고, 낮은 분산을 가지고 있어서 함수 근사형태에 적합하다.

2.3 Policy gradient- RLS

2.1과 2.2에서 언급한 critic을 위한 RLS와 actor를 위한 policy gradient를 결합한 알고리즘의 형태는 다음과 같다.

(1) 파라미터를 초기화함
w_Θ=0(actor의 연결강도), z_t=0(eligibility)
(2) 시간 스텝 t의 관측 변수 x_t관찰
(3) 확률분포 $\phi(.;\Theta(x_t;w_\Theta))$에 따른 제어 입력 μ_t를 샘플링하여 실행
(4) 제어입력에 따른 다음 상태변수(x_{t+1})와 보상값(r_t) 관찰
(5) 파라미터 개선

$\zeta_t \triangleq [\phi'(x_t) - \gamma\phi(x_{t+1}), \nabla_{w_\theta} \log\pi(a_t|x_t)']$,
γ=할인율

a. **Recursive update**

$A_0 = \delta I + z_0[\phi' - \gamma\phi'(x_t), \nabla_\theta \log\pi(a_0|x_0)']'$, $\delta > 0$

$z_0 = [\phi'(x_0), \nabla_\theta \log\pi(a_0|x_0)']'$

$A_t = \beta A_{t-1} + z_t\zeta_t$ for $t \geq 1$

where β=forgetting factor

$P_t \triangleq A_t^{-1}, K_t \triangleq P_t z_t$

b. Critic 파라미터 개선

$z_t = \gamma\lambda z_{t-1} + \zeta_t'$

$$P_t = \frac{1}{\beta}\left(P_{t-1} - \frac{P_{t-1}z_t\zeta_t P_{t-1}}{\beta + \zeta_t P_{t-1} z_t}\right)$$

$$K_t = \frac{P_{t-1}z_t}{\beta + \zeta_t P_{t-1} z_t}$$

$$\begin{bmatrix} v_{\theta_t} \\ w_{\theta_t} \end{bmatrix} = \begin{bmatrix} v_{\theta_{t-1}} \\ w_{\theta_{t-1}} \end{bmatrix} + K_t\left(r_t - \zeta_t \begin{bmatrix} v_{\theta_{t-1}} \\ w_{\theta_{t-1}} \end{bmatrix}\right)$$

$\theta_{t+1} \leftarrow \theta_t + \alpha F(\theta) w_{\theta_t}$

(6) go to step (2)

3. 모의 실험

3.1 Kimura 로봇

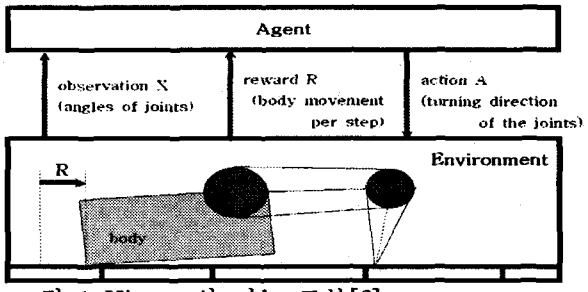

그림 1 Kimura의 기는 로봇[6]

참고문헌 [6]에서 Kimura 등은 강화학습의 효용성을 보이기 위해 간단한 기는 로봇을 응용 문제로 고려하였다. 이 로봇은, 중력이 가해지는 환경 아래에서 두 개의 링크를 가지고 기는 동작을 수행하는 평면형 머니퓰레이터(planar manipulator)로써 그림 1의 구조를 갖는다.

이 로봇에 부과된 임무는 최대한 빨리 전진하는 것인데, 에이전트(agent)는 로봇 및 환경에 대한 구체적인 모델 또는 정보가 주어지지 않은 상태에서 직접적인 경험을 통해 관찰된 보상값(rewards) r 만을 가지고 효과적인 제어 규칙을 발견해내야 한다. 각 시간 스텝 때마다 에이전트는 조인트의 각도를 읽어 들이고 확률적 제어입력 선택 전략에 따라 조인트에 연결된 모터의 회전 방향 및 회전각도를 결정한다.

그리고, 학습 과정에서 이용되는 보상값 r 을 위해서는 해당 시간 스텝 동안 전진한 거리가 사용된다. 만일 로봇이 후진하는 경우에는 후진한 거리만큼의 음의 보상값(negative reward)이 생성됨은 물론이다. 직관적으로 생각할 때에, 위의 로봇이 최대한 빨리 전진하기 위해서는 기면서 앞으로 나아가는 패턴을 신속하게 습득해야 함을 알 수 있다. 본 논문에서 고려하는 로봇 관련 데이터는 [6]의 경우와 같다.

몸체와 위쪽 팔을 잇는 조인트의 움직임은 몸체와 수평인 방향에서 [-4, 35] 도 범위에서만 가능하고, 위쪽 팔과 아래쪽 팔을 잇는 두 번째 조인트의 움직임은 위쪽 팔과 수평인 방향에서 [-120, 10] 도 범위에서만 가능하다. 그리고 아래쪽 팔의 뾰족한 끝부분이 지면에 닿아 있을 때에는, 뾰족한 끝부분은 미끄러지지 않고 몸체만 미끄러짐을 가정한다.

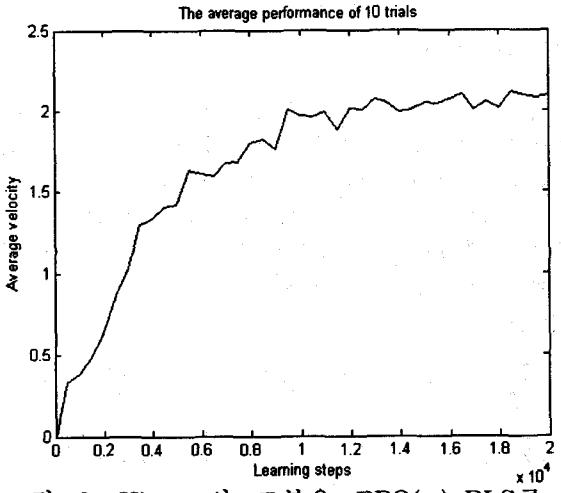

그림 2. Kimura의 로봇을 RPO(λ)-RLS를 적용하여 학습시킨 결과[9]

3.2 Kimura 로봇과 policy gradient -RLS를 적용

본 논문에서는 [6]에서의 이론 전개를 참고하여, σ에는 1의 값을 actor에는 각 조인트의 제어입력 선택 전략을 위한 확률분포 ϕ로 다음과 같은 정규분포를 고려하였다:

$$\phi(\mu; c) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\mu - c)^2}{2\sigma^2}\right)$$

그리고 각 조인트에서는 로봇의 과도한 움직임을 막기 위해서 각 시간 스텝 당 [-12도 , 12도]범위까지의 움직임만 허용하는 한계성을 부여하였다. 한편 기는 로봇이 받아들이는 입력값은 변위가 [-1,1] 범위가 되도록, 관련 측 변수인 조인트를 적절하게 스케일링한 값을 사용하였다.

논문에서 사용된 기저 벡터는 다음과 같다. 각 조인트의 스케일링 값인 θ_1과 θ_2를 입력으로 하고 고정된 분산과 평균을 갖는 RBF(radial basis function)을 사용하여 각 조인트가 갖는 RBF함수

의 값으로 나눈 NRBF(normalized radial basis function)를 기저 벡터로 사용하였다.

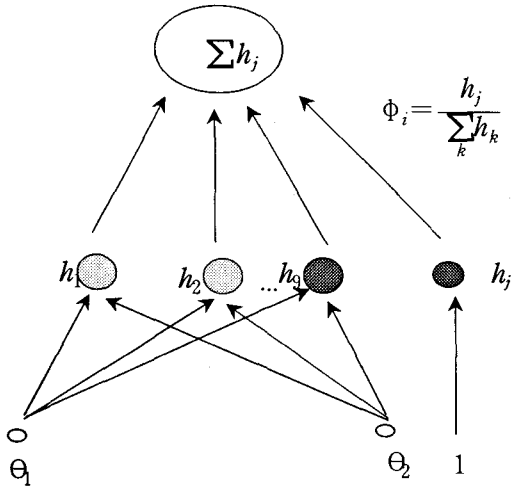

그림 3. NRBF-Network를 이용한 기저 함수근사

실험에서는 10번의 episode를 실행했으며, 각 episode는 모두 20000번의 step으로 구성되어 있다. 평균속도는 각 500step의 배수에 그동안 학습된 actor의 파라미터를 이용하여, 한정된 거리를 이동하게 한 후 그에 대한 평균속도를 구했다.

학습에서 사용된 그 밖의 관련 파라미터는 다음과 같다. 할인율 $\gamma=0.95$, 감쇠율 $\lambda=0.75$, 학습율 $\alpha=0.003$, 초기 분산상수 $\delta=0.5$

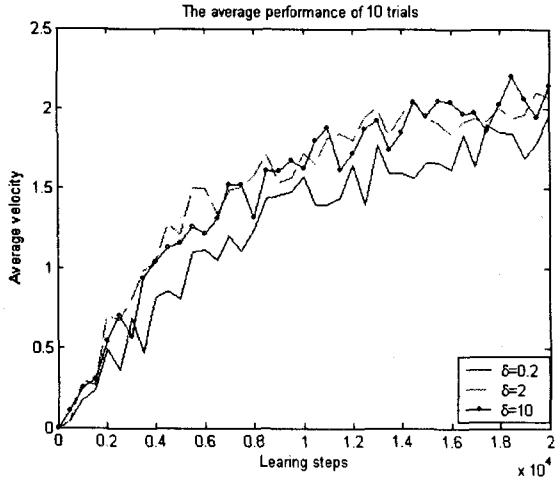

그림 4. Kimura의 로봇을 policy gradient-RLS를 이용하여 학습시킨 결과

4. 토의 및 향후과제

그림4와 2를 비교했을때, policy gradient-RLS를 이용한 방법이 RPO(λ)-RLS의 방법과 비슷한 평균 속도를 갖고 있음을 보여준다. RLS는 함수 근사를 이용해 지역적 최적해(local minimum)에 빨리 도달 하는 장점을 가지고 있다. 이런 장점을 이용하여 함수 근사에 적합한 형태임을 실험을 통해서 확인해 볼 수 있었다. 향후 과제로는 제안한 학습 방법을 다른 예에 적용해 보는 문제나, 최근 기계학습 분야에 큰 영향을 미치고 있는 커널 기법을 강화 학습 분야에 접목 시킨 학습 알고리즘 개발 후 이를 시뮬레이터를 통해 확인해 보는 문제 등을 들 수 있다.

참고문헌

[1] A. Nedic and D. P. Bertsekas "Least Square Policy Evaluation Algorithms With Linear Function Approximation", *Journal of Discrete Event Dynamic Systems*

[2] J. Peters, S. Vijayakumar and S. Schaal "Reinforcement Learning for Humanoid Robotics," *Proceedings of 3rd IEEE-RAS International Conference on Humanoid Robots, Karlsruhe, Germany.*

[3] X. Xu, H. He and D. Hu, "Efficient Reinforcement Learning Using Recursive Least-Square Methods," *Journal of Artificial Intelligence Research*, vol 16, pp. 259-292, 2002

[4] Boyan, J. "Least-squares temporal difference learning." *Machine Learning : Proceedings of the sixteenth International Conference(ICML)*

[5] Boyan, J.. Technical update : least-squares temporal difference learning. *Machine Learning, Special Issue on Reinforcement Learning,* to appear.

[6] H. Kimura, K. Miyazaki, and S. Kobayashi, "Reinforcement learing in POMDPs with function approximation," In *Proceedings of the 14th International Conference on Machine Learning (ICML '97)*, pp. 152-160, 1997

[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998

[8] Ljung L. "Analysis of recursive stochastic algorithm." *IEEE Transactions on Automatic Control*, vol, 22, pp. 551

[9] 김종호, 강대성, 박주영, "RPO기반 강화학습 알고리즘을 이용한 로봇제어" 한국 퍼지 및 지능시템 학회 2005년도 춘계학술 대회 논문집, 15권 1호, pp, 505 , 2005년 4월