제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

다중에이전트 행동기반의 강화학습에 관한 연구

도 현 호*, 정 태 충
경희대학교 전자계산공학과
e-mail:mubious@iislab.kyunghee.ac.kr

A Study on Reinforcement Learning of Behavior-based Multi-Agent

HyunHo Do *, TaeChoong Chung
Dept of Computer Engineering, KyungHee University

요 약

다양한 특성들을 가지고 있는 멀티에이전트 시스템의 행동학습은 에이전트 설계에 많은 부담을 덜어준다. 특성들로부터 나오는 다양한 행동의 효과적인 학습은 에이전트들이 환경에 대한 자율성과 반응성을 높여줄 수 있다. 행동학습은 model-based learning과 같은 교사학습보다는 각 상태를 바로 지각하여 학습하는 강화학습과 같은 비교사 학습이 효과적이다. 본 논문은 로봇축구환경에 에이전트들의 행동을 개선된 강화학습법인 Modular Q-learning을 적용하여 복잡한 상태공간을 효과적으로 나누어 에이전트들의 자율성과 반응성을 높일수 있는 강화학습구조를 제안한다.

1. 서론

멀티에이전트 시스템은 자발성, 자율성, 사회성, 반응성을 가지고 있다. 위의 특성을 가지고 있는 독립적인 프로그램인 멀티에이전트 시스템에서, 적절한 성질들을 판단하여 행동하는 학습은 아주 중요한 역할을 가지고 있다. 실세계와 같이 복잡한 환경에서는 설계자가 멀티에이전트 시스템들의 특성들을 판단하여 설계하기는 아주 어려운 일이다. 여기서 에이전트 설계자의 부담을 덜어줄 수 있는 것이 학습이며, 현재 멀티에이전트의 학습에 대하여 많은 연구가 진행 중이다.

임의의 환경에서, 에이전트들은 선택할 수 있는 행동들의 집합을 가지고 있고, 환경에 대한 많은 변수들이 존재한다. 다수의 변수들을 포함한 환경에서 다양한 행동에 대한 학습은 기존의 교사 학습법(supervised learning)으로는 해결하기가 힘들다. 이 때, 정해진 전략을 사용하는 교사 학습보다는 환경을 감지하여 스스로 최적의 전략을 세울 수 있는 강화학습 같은 비교사 학습(un-supervised learning)이 더욱 효과적이다. 강화학습은 에이전트들의 환경에 대한 적응과 반응성을 높일 수 있다. 그러나 단순한 강화학습의 가장 큰 문제점은 큰 상태공간을 갖는 복잡한 환경들에 그대로 적용하기가 힘들다는 것이다. 본 논문에서는 복잡한 환경인 로봇 축구에 기존의 강화 학습을 개선한 Modular Q-learning 적용하여, 에이전트들의 자율성과 반응성을 향상시키기 위해 큰 상태 공간에서 에이전트들의 다양한 행동을 이용한 효과적으로 학습할 수 있는 구조를 제안한다.

2. 강화 학습

2.1 Q-learning

Q-learning은 대표적인 강화학습법으로서 기본적으로 에이전트가 동적인 환경에서 시행착오를 통해, 목적을 이루기 위한 수치적인 확률 보상(Reward)이 최대로 되는 행동(정책)을 학습하는 것이다. 이 학습 방법은 일정한 사전지식으로만 학습하기 힘든 환경에서 보다 더 좋은 학습 효율을 나타낼 수 있다.

Q-learning의 일반적인 정의는 다음과 같다. 임의의 시간 t에서, 에이전트는 상태 s_t에서 행동 a_t을 선택하고 확률적인 보상 r_t를 받는다. 최적의 정책을 찾기 위해서는 상태 전이 함수(state transition probability function)와 보상 함수(reinforcement function)가 필요하며, $T(s_t, a_t, s_{t+1})$ $R(s_t, a_t)$로 나타낸다. $T(s_t, a_t, s_{t+1})$는 상태 s_t에서 행동 a_t을 실행할 때 다음 상태 s_{t+1}로 전이 될 확률을 나타내는 함수이고, $R(s_t, a_t)$는 상태 s_t에서 행동 a_t을 선택할 때 받는 보상의 함수이다. 절감된 예상 보상 $R(s_t, a_t)$는 에이전트가 있는 상태와 선택한 행동에 의존하고 에이전트의 목표는 절감된 예상 보상을 최대로 하는 정책을 찾는 것이다. 절감된 예상 보상은 식(1)과 같다.

$$R(s_t, a_t) = E\left\{ \sum_{i=0}^{\infty} \gamma^i r_{t+i} \right\} \qquad (1)$$

여기서, γ는 절감 계수(discounted rate)이며 $0 \le \gamma < 1$ 의 범위를 갖는다. 절감 계수는 t시간 간격에서 받은 보상이 현재 받은 보상에 비해 γ^t 만큼 적다는 것을 의미한다. 상태 s에서 시작하여 정책(policy) π를 따를 때의 예상 보상을 정책에 대한 상태 값 함수

(state value function)로 나타내면 식(2)와 같이 정의한다. 이 때 최적의 정책 π^*가 존재하면 최적의 상태 값 함수는 식(3)과 같다.

$$V^{\pi}(s) = R(\pi(s), a) + \gamma \sum_{s \in S} T(\pi(s), a) V^{\pi}(s') \quad (2)$$

$$V^*(s) = \max_a \left\{ R(s, a) + \gamma \sum_{s \in S} T(s, a, s') V^*(s') \right\} \quad (3)$$

Q-learning에서는 정책 π에 대해 Q-value(action value)을 식(4)와 같이 정의한다. Q-learning은 Q-value가 최적의 정책에 대해 근사 하는 것이 최종 목표이며, Q-value와 V^*의 관계는 $V^* = \max_a Q^*(s, a)$로 표현된다. 현재의 상태와 행동을 각각 s, a이고 다음 상태와 그 상태의 행동을 s', a'이라 할 때 일반적인 Q-Value Update식은 식(5)와 같다.

$$Q^{\pi}(s, a) = R(\pi(s), a) + \gamma \sum_{s \in S} T(\pi(s), a) V^{\pi}(s') \quad (4)$$

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left(r + \gamma \max_a Q(s', a') - Q(s, a) \right) \quad (5)$$

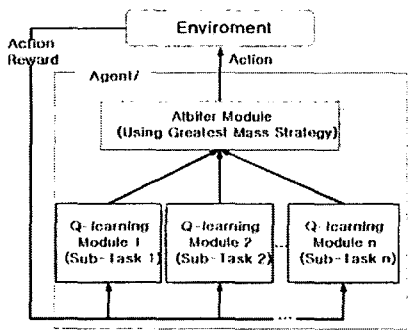
(α : 학습율 , γ : 보상값 r에 대한 절감 계수)

그러나 실세계의 문제에 적용시, Q-learning의 가장 큰 문제점은 큰 상태공간에서 학습에 필요한 Q-Value의 저장소 역할을 하는 Q-Table의 크기가 환경의 변수의 갯수에 따라 지수적으로 증가하는 것이다. 최적의 시간동안 에이전트가 가능한 모든 상태들을 학습하기가 힘들게 된다.[1,2]

2.2 Modular Q-learning

Whitehead는 이런 상태공간의 문제를 해결하기 위하여 효과적으로 상태공간을 줄일 수 있는 Modular Q-learning을 제안하였다.[3] Modular Q-learning은 설계자가 사전처리로서 큰 상태공간을 적절하게 sub_task들로 나누고, 각 sub_task별로 학습이 이루어진다. 각 learning module들의 학습한 Q-value 결과 값들을 식(6)과 같은 최대 질량 전략(GMS: Greatest Mass Strategy) 방식으로 결합하여 최적의 행동을 결정하게 된다.

$$a^* \leftarrow \arg\max_{a \in A} \sum_{i=0}^{n} Q_i(s, a) \quad (6)$$



[그림 1] Modular Q-learning 구조

[그림1]은 Modular Q-learning의 구조를 보여주고 있다. 설계자에 의해서 sub_task로 나누어진 learning module들은 순수한 Q-learning의 학습법을 사용하여 sub-task들의 Q-value를 학습한다. 학습된 Q-value들은 arbiter module에서 식(6)의 GMS방식으로 최적의 행동을 선택한다. 최적의 행동을 학습할 때까지 적은 Q-value의 저장공간과 최적의 행동의 수렴시간을 많이 줄일 수 있다.

3. 멀티에이전트의 행동 정의

복잡한 환경에서 에이전트들의 행동을 학습하기 위하여 로봇축구의 환경을 사용한 다양한 연구가 이루어지고 있다. 로봇축구의 환경에서 에이전트들의 행동을 Balch는 Motor Schema를 이용하여 정의하였다.[5]

Motor Schema는 에이전트의 다양한 물리적인 행동들을 간단하게 정의한다. 각각의 행동들은 환경에 대한 상태공간으로 정의 될 수 있고, 이러한 다양한 행동들은 일정한 행동들의 집합**(behavioral assemblage)**으로 그룹화 하여 복잡한 환경 속에 행동들을 상태에 대한 행동들의 집합으로 표현한다.

로봇 축구 환경에서 에이전트들은 같은 목표 즉, 득점을 많이 하고, 실점을 최대한으로 줄이기 위한 행동들을 취하게 된다. 각 에이전트의 최상위 행동들은 **move_to_ball (mtb), get_behind_ball (gbb), move_to_backfield (mtbf)**의 행동들로 나누어질 수 있다. 각 에이전트의 상태는 볼이 인접함과 인접하지 않음으로 구분한다. 즉,**behind_ball, not_behind_ball**로 나누어지게 된다. 로봇 축구 환경에서 가장 중심적인 역할을 하는 Forward, Goalie의 행동은 [표 1], [표 2]처럼 나누어진 행동들의 집합을 지각된 상태에 대해서 정의할 수 있다.[6]

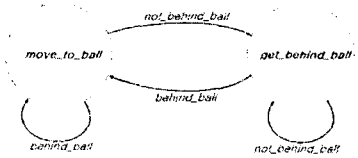
perceptual feature	assemblage		
	mtb	gbb	mtbf
not_behind_ball	0	1	0
behind_ball	1	0	0

[표 1] Team Forward strategy

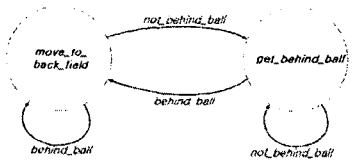
perceptual feature	assemblage		
	mtb	gbb	mtbf
not_behind_ball	0	1	0
behind_ball	0	0	1

[표 2] Team Goalie strategy

에이전트의 Motor Schema를 상태와 행동들을 도표로 표현할 수 있는 **FSA**(Finite State Automaton)로 [그림2], [그림3]과 같이 나타내며, 에이전트의 상태에서 각각 행동들은 에이전트의 센서(sensor)로부터 받은 위치정보들을 바탕으로 [그림4]와 같이 각각의 계층을 이루게 된다.[5]

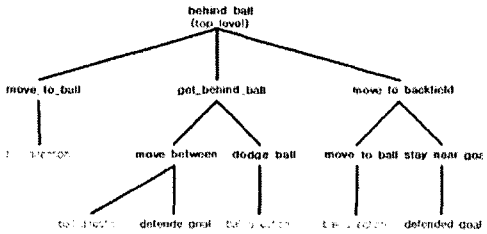


[그림 2] Team Forward FSA



[그림 3] Team Goalie FSA

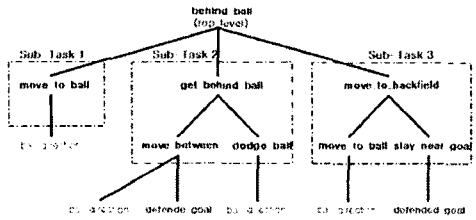
[그림4]와 같이 tree의 root(top_level)의 상태에서 실행 가능한 행동들을 tree에 표현처럼 최상위 행동들인 sub_tree(move_to_ball,get_behind_ball,move_to_backfield)를 가지게 된다. 각 sub_tree의 상태들은 에이전트의 지각 센서로부터 받은 환경정보를 하위레벨에서 전달받아 Motor Schema의 행동들로 표현한다.



[그림 4] 현재상태의 가능한 행동 Tree

4. Modular Q-learning을 사용한 행동의 학습

순수한 Q-learning으로 학습하는 경우, 에이전트들이 각자의 현재 상태에서 모든 행동들을 경험하여 최종적인 새로운 상태로의 전이하는 것은 많은 시간과 저장공간을 필요로 하게 된다.



[그림 5] 행동Tree에서의 sub-task분리

이런 문제점들을 본 논문에서는 Modular Q-learning을 사용함으로서 해결한다. learning module들의 학습 공간들을 나누는 것은 Modular Q-learning에서 아주 중요하다. 지각 센서로부터 받아 Motor Schema로 표현된 각각의 행동들을 기준으로 학습의 공간을 선택하였다. 각 최상위 행동들에 대하여 [그림5]와 같이 sub_task로 나누어 학습을 하기 때문에 보다 적은 시간으로 빠른 행동전략에 수렴할 수 있으며, 각 상태에 대한 학습 범위를 줄여줌으로서 Q-Value의 저장공간도 줄여준다. 학습의 빠른 행동전략수렴은 에이전트의 지각된 상태에 대하여 자율성과 반응성을 높여준다.

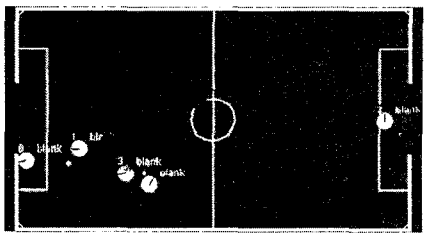
Modular Q-learning의 각 learning module들은 최상위 행동들로 분리된 sub_task들을 가지고 순수한 Q-learning의 식(3)의 Q-Value Update식을 사용한 학습을 하며, arbiter module에서 각 module의 Q-Value를 최대 질량 전략(GMS: Greatest Mass Strategy)을 사용하여 최적의 행동전략을 찾아 실행하게 된다. 각 학습시 learning module에서 사용되는 보상값(reward)은 식(7)과 같이 정의하여 사용을 하였다.

$$r(s_t) = \begin{pmatrix} +1 & (\text{우리팀이득점을하였을경우}) \\ -1 & (\text{상대팀이득점을하였을경우}) \\ 0 & (\text{나머지의모든경우}) \end{pmatrix} \quad (7)$$

5. 실험 환경 및 결과

5.1 실험 환경

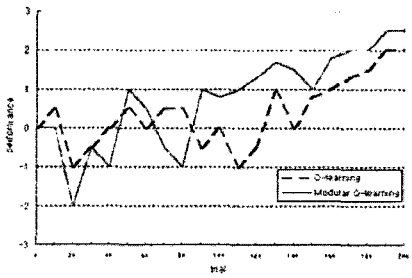
본 연구에서는 Modular Q-learning의 적용을 구현하기 위하여 Georgia공대의 Tucker Balch가 개발한 멀티 에이전트 및 멀티 로봇 시뮬레이션 Teambots을 사용하여 구현하였다. Teambots은 에이전트들의 행동을 Motor schema로 표현할 수 있도록 Clay라는 Java package로 구현되어있다.[6] Clay는 기본적인 에이전트들의 행동들을 학습하기 위해 순수한 Q-learning을 사용할 수 있도록 구현되어있다.
비교를 위해 학습기능이 없는 팀과 순수한 Q-learning으로 학습을 하는 팀, Modular Q-learning으로 학습하는 팀을 구현하여 평가를 하였다.



[그림 6] Robot Soccer Simulation (Using Teambots)

5.2 실험 결과

본 연구에서는 Modular Q-learning학습을 한 에이전트 팀과 Q-learning 학습을 한 에이전트 팀의 행동들을 통하여 비교 실험하였다. 각각의 에이전트들의 학습력과 반응성을 평가하기란 쉬운 일이 아니므로 팀 전체의 득점비율로 성능 평가를 하였다. 각 팀의 학습력이 높아질수록 환경에 대한 반응성이 좋아지고 경기의 득점이 높아지기 때문이다.



[그림 7] 학습 팀의 성능평가

[그림7]은 학습기능이 없는 팀과 각각 200번의 경기(trial)를 한 결과를 나타내고 있다. 각 팀의 학습시 사용되는 학습율(α: learning rate)과 학습 절감 계수

(γ: discount rate)는 똑같이 α=0.2 , γ=0.8로 고정하여 학습을 하였다. 득점 비율을 통한 성능 평가는 식(8)과 같은 방법으로 평가하였다.

$$P(performance) = \frac{\text{학습팀의득점} - \text{제어팀의득점}}{2} \quad (8)$$

제안한 Modular Q-learning을 이용한 팀이 순수한 Q-learning을 이용한 팀보다 초기에는 좋지 못한 성능을 나타내었으나, 일정 시간이 경과한 뒤에는 Q-learning학습 에이전트 팀보다 Modular Q-learning 학습 에이전트 팀의 성능이 향상되었다.

6. 결론 및 향후 연구 방향

본 논문은 복잡한 환경의 멀티 에이전트 시스템에서 다양한 행동들로 인하여 지수적으로 늘어나게 되는 상태 공간을 줄이기 위하여 강화학습인 Modular Q-learning을 사용한 구조를 제안하고, Q-learning을 사용한 구조와의 비교, 평가를 로봇축구 환경의 실험을 통하여 효과적임을 보였다. 또한 이 실험을 통하여 복잡한 환경 속의 에이전트 학습시 효과적인 상태 공간이 전체 학습 성능에 어떠한 영향을 주는지 보여주고 있다. 그러나 축구 전술문제와 같은 상위 레벨의 학습문제까지 고려를 할 경우에는 좀 더 복잡한 행동들이 나와 학습시간의 지연이 올 수도 있다. 이를 위한 효과적인 학습 모듈 설계와 다른 강화학습 Algorithm의 비교, 평가하여 다양한 환경에 적응할 수 있는 학습구조를 만들어 가는 것이 향후 연구 과제로 이루어 져야 할 것이다.

참고문헌

[1] R.S.Sutton, A.G.Barto, "Reinforcement Learning: An Introduction," MIT Press, 1998

[2] C. Wathins, P. Dyan, "Q-Learning,", Machine Learning, pp279-292, 1992

[3] S. Whitehead,"Learning Multiple Goal Behavior via Task Decomposition and Dynamic Policy Merging," Robot Learning, Kluwer Academic Press, 1993

[4] M. Tan, "Multiagent Reinforcement Learning: Independent Vs. Cooperative Agent," Proc. of the 10th International Conference On Machine Learning, 1993

[5] R.Arkin, T.Balch, "Aura: principles and practice in review," Journal of Experimental and Theoretical Artificial Intelligence, in press, 1997

[6] T. Balch, "Behavioral Diversity in Learning Robot Teams,"AAAI-97 workshop on Multiagent Learning, 1997

[7] P.V.C. Caironi, M. Dorigo, "Training and delay reinforcements in Q-Learning agent," Journal of Intelligent System 12, pp615-724, 1997

[8] Kostas Kostiadis, Huosheng Hu. "Reinforcement Learning and Co-operation in a Simulated Multi-Agent System," RoboCup-98: Robot Soccer World Cup II, pp.366-377, Springer Verlag, Berlin, 1999

[9] Kui-Hong Pack, Yong-Jae Kim, Jong-Hwan Kim, "Modular Q-Learning based Multi-Agent Cooperation for Robot Soccer," Robotics and Autonomous System Vol.35, pp.109-122, 2001