2005년도 대한전기학회 하계학술대회 논문집 2005. 7. 18 - 20

네트워크 기반의 강화학습 알고리즘과 시스템의 정보공유화를 통한 최단경로 검색과 갱신

민성준·장종수·김홍윤·허훈
고려대학교 제어계측공학과

Search of Optimal Path and Renewal via network based Reinforcement Learning Algorithm and sharing of System Imformation

Seong-Joon Min·Jong-Soo Chang·Hong-Yoon Kim·Hoon Heo
Dept. of Control & Instrument Engineering, Korea univ.

Abstract - 본 논문에서는 환경과 시스템의 상호작용을 통한 경험에 의해 습득된 정보를 개체간 네트워크를 통하여 갱신하는 과정을 구성하는 연구를 하였다.

기존의 연구에서는 강화학습 알고리즘을 이용하여 임의의 구역에 대한 지도 정보를 습득하고 이를 바탕으로 개체들 각각의 최적의 행동 정책을 구성하는 바 이 때 각각의 체개체가 가지고 있는 최단경로에 대한 정보의 우위를 결정하는 과정을 추가하였다. 이를 바탕으로 최종적으로 선택된 경로에 대한 정보를 업데이트하여 구성된 네트워크를 통한 개체간 데이터를 동시에 공유하는 과정을 거쳐서 각각의 시스템이 스스로 정보를 갱신하는 방법을 제안하였다. 또한 이 제안한 개념의 적합성을 입증하기 위하여 개체간의 정보를 통합하고 비교하는 실험을 수행하여 성공적인 결과를 얻었다.

1. 서 론

강화학습(Reinforce Learning)의 역사는 두 개의 주요 맥락을 가지고 있다. 하나의 맥락은 trial and error에 의한 학습과 관련이 있다. 이는 심리학에서의 동물을 통한 학습에서 출발한 것이다. 이런 기조는 인공지능 분야에서 초기 작업을 통해서 발전되어 왔으며 1980년대 초에 강화학습의 부활을 이끌어 왔다. 다른 하나의 맥락은 최적 제어 문제와 가치 학습을 사용한 해결 및 동적 프로그래밍과 관련되어 있다. 50년대 중반 Richard Bellman과 19세기의 Hamilton과 Jacobi에 의해 발전되어 왔으며 최적 제어는 1950년대 후반 동적 시스템의 동작의 측정을 최소화 하기위한 제어기의 설계 문제에서 묘사되었다. 이렇게 발전해온 강화학습의 개념은 현재 다양한 범위에서 그 가능성을 보이고 있다. 이는 현대 과학이 발전함에 따라 인간의 지능을 대체할 무인 시스템에서의 가능성을 엿볼 수 있는 것이다. 본 논문에서는 무인 개체를 이용하여 지형에 대한 정보를 인식하고 이를 통해 최적의 정보를 구성하여 이러한 정보를 공유화하는 데 목적을 두고 있다. 그리하여 강화학습을 이용한 최적경로 탐색에 대한 시뮬레이션을 구성하여 경로 탐색에 대한 실험을 수행하였다. 먼저 trial-error 방식에 의하여 맵에 대한 정보를 습득한다. 그리고 습득된 정보를 토대로 하여 Reinforcement Learning을 통하여 얻어진 가능한 경로를 구하게 된다. 그리고 이 경로 중에서 최적의 경로를 구하여 개체에 인식하여 준다. 또한 다양한 맵을 구성하여 각각의 맵에 대한 정보를 각 개체가 소유하게 되어 이를 네트워크를 통하여 공유하는 과정으로 구성하였다.

2. 본 론

2.1.1 The Agent-Environment Interface

강화학습 문제는 목표를 얻고자 하는 상호작용으로부터 시작된다고 볼 수 있다. 그러므로 강화학습에서는 학습자와 결정권자를 agent라 칭하고 모든 외부 환경을 environment라 한다. 계속되는 상호작용 속에서 agent는 actions을 결정하게 되고 environment는 이런 action에 대응하게 되고 agent에 대하여 새로운 상황을 연출하게 된다. 또한 environment는 agent가 극대화하려는 특별한 수학적 값인 reward를 받게 된다. 즉, 각각의 연속된 스텝에서 agent와 environment는 상호작용하여 이에 대한 보상값인 reward가 주어지게 되는 것이다.

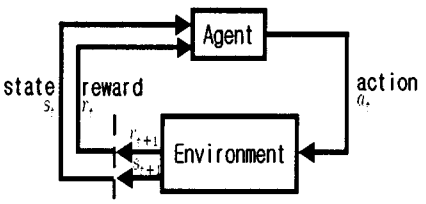

그림1. The agent-environment interaction

2.1.2 Returns

궁극적으로 agent의 목표는 reward를 최대화하는데 있다 할 수 있다. 간단하게 이를 보자면 아래의 식 (2.1)과 같은 형태로 볼 수 있다.

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \cdots + r_T \quad (2.1)$$

하지만 많은 경우에서 이런 상호작용이 각각의 에피소드에서 자연스럽게 종료되는 것이 아니기에 이에 대한 추가적인 개념이 discounting이다. 이러한 접근에 따르면 agent는 action을 결정하는데 향후에 걸쳐서 받는 discounted rewards의 합을 고려하게 된다.

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (2.2)$$

여기에서 gamma는 0과 1사이의 값을 가지게 된다.

2.1.3 Unified Notation for Episodic

간단하게 각각의 에피소드에 따른 state를 살펴보면 아래와 같은 diagram으로 나타낼 수 있다.

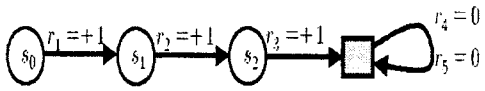

그림.2 The state transition diagram

여기에서 박스 형태는 에피소드의 종료에 대응하는 것으로 나타내었다. s_0에서 시작하여 연속적으로 각각의 reward를 얻을 수 있다.

2.1.4 Value Functions

모든 강화학습 알고리즘의 대부분은 value function을 계산하는 데 기반을 두고 있다. 이는 주어진 state에서 agent가 얼마나 좋은 action을 택하는 function을 말하는 것이다. 이러한 value function은 policy에 의해 정의된다 할 수 있다.

policy π는 각각의 state와 action으로부터 위치하는 것으로 probability π(s,a)는 state s에서 action a를 취하는 가능성을 나타내는 것이다. 따라서 $V^{\pi}(s)$를 정의하자면 state-value function for policy π 라 한다.

$$V^{\pi}(s)=E_{\pi}[R_t|s_t=s]=E_{\pi}[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t=s] \quad (2.3)$$

위와 유사하게 policy π아래 state s에서 actiona a를 취하는 값을 $Q^{\pi}(s)$라 하고 이를 정의하면 action-value function for policy π 라 한다.

$$Q^{\pi}(s,a)=E_{\pi}[R_t|s_t=s,\ a_t=a] \quad (2.4)$$
$$=E_{\pi}[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t=s,\ a_t=a]$$

2.1.5 Optimal Value Functions

optimal state-value function V^*와 optimal action-valu e function Q^*를 정의하면 아래와 같다.

$$V^*(s)=\max_{\pi} V^{\pi}(s) \text{ for all } s\in S \quad (2.5)$$

$$Q^*(s)=\max_{\pi} Q^{\pi}(s,a) \text{ for all } s\in S \text{ and } a\in A(s) \quad (2.5)$$

$$Q^*(s,a)=E_{\pi}[r_{t+1}+\gamma V^*(s_{t+1})|s_t=s,\ a_t=a] \quad (2.6)$$

이러한 Q-learning을 도식화하면 아래와 같다.

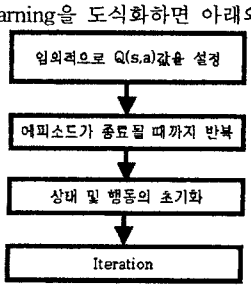

그림.3 Q-learning algorithm

2.2.1 경로 검색 환경 구성 및 시뮬레이션

본 논문에서는 matlab을 기반으로 임의적으로 다양한 map을 구성하여 종단점에 다다르면 reward를 주는 것으로 환경을 설정하였다. 이동 개체는 네 개 방향에 대한 센서를 장착한 것으로 가정하여 한 step마다 그 위치에 따른 센서값을 저장하여 이동 가능 방향을 알 수 있도록 구성하였다. 그리고 map의 크기는 5×5 사이즈로 구성하였으나 이 map의 크기는 그 이상으로 확장하여도 올바른 결과를 얻을 수 있음을 알 수 있다.

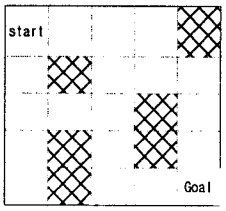

그림.4 임의로 구성한 map

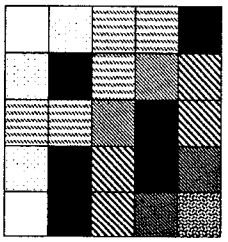

그림.5 구성된 map에 대한 이동가능한 path

map상에서 빗금친 부분은 block으로 설정하였다. 먼저 설정된 map의 시작 지점에서 trial-error방식으로 goal을 찾게 하였다. 이렇게 goal을 찾게 되는 과정에서 개체는 각각의 state에서의 네 개 방향의 값들을 저장하게 된다. 즉, 이동가능 여부를 판별하여 이를 정해진 값들로 환산하여 행렬 형태로 저장한다.

goal을 찾게 되면 에피소드는 종료되게 되고 각 state의 방향값들을 저장한 데이터를 바탕으로 각 위치에 따른 이동 가능 방향을 표시하게 된다.

위의 그림에서 보면 진하게 채워진 부분은 block이고 골인 지점에 가까운 경로일수록 사선 무늬가 길어지면서 촘촘하게 표시하여 경로를 표시하였다.

	1	2	3	4	5
1	4.3044	4.7827	5.3141	5.9046	-9.9997
2	4.7827	-9.9997	5.9046	6.5607	7.2897
3	5.3141	5.9046	6.5607	-9.9997	8.0997
4	4.7827	-9.9997	7.2897	-9.9997	8.9997
5	4.3045	-9.9997	8.0997	8.9997	9.9997

그림.6 각 state에 따른 value

앞에서의 trial-error방식에서는 각각의 cell에서 개체는 네 개 방향 모두 움직일 수 있다. 하지만 goal의 위치를 알게 되면 value iteration 과정을 수행하여 최종적으로 위의 표와 같은 값으로 수렴하게 된다. 이 때 개체는 현재의 위치에서 가장 큰 값의 방향으로 policy를 결정하게 된다. 이는 goal로 가는 가장 좋은 정책으로 인식하게 된다.

	1	2	3	4	5
1	ES	E	ES	S	BLOCK
2	S	BLOCK	ES	E	S
3	E	E	S	BLOCK	S
4	N	BLOCK	S	BLOCK	S
5	N	BLOCK	E	E	GOAL

그림.7 각각의 state에서의 최적 이동 방향

계산된 value 값들로부터 최적 방향을 제시받게 된다. 이 때 제시된 이동 방향의 value값이 같을 경우는 모두 가능하게 표시하였다. 이렇게 가능한 path를 표시하고 최적의 path를 구하기 위해 시스템을 구성하였다. 이 시스템에서는 value값의 비교를 통하여 goal을 찾는 가장 최적의 path를 표시하였다. 단, value값이 같을 경우는 임의적으로 선택하게 구성하였다.

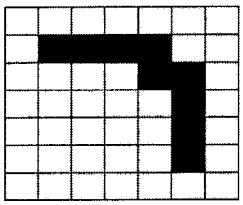

그림.8 Optimal path

2.2.2 네트워크를 이용한 다개체간 데이터 공유

위에서 구한 optimal path를 바탕으로 두 개체에 각각 다른 map에 대한 데이터를 주었다. 이 때 각각의 데이터는 optimal path를 줄 경우와 단순 이동 가능 path를 임의로 주었다. 그리고 아래의 순서도와 같은 알고리즘으로 구성하였다.

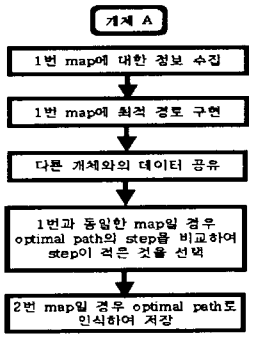

그림.9 네트워크를 통한 데이터 공유 흐름도

위의 흐름도를 바탕으로 같은 map에서 다른 optimal path를 가진 경우와 서로 다른 map에서 각각 하나만의 optimal path를 가진 경우를 설정하였다. 앞에서의 실험에서 나온 데이터들을 각각의 개체들이 가지고 있다고 가정하여 먼저 각각의 센서값들을 개체로부터 받아 연산한 후 optimal path를 개체로 이식하는 과정을 simulation 하였다.

개체는 이식된 최적 경로를 통하여 map에 대한 검색 과정을 수행하지 않고 바로 goal을 찾아내었다. 이렇게 이식된 데이터의 유효성을 판단한 후 네트워크를 통하여 개체간 데이터 통신을 하였다.

같은 map에서 최적 경로가 다른 경우 개체는 이식된 최적 경로의 step의 수를 비교하여 그 중 최소의 것을 선택하여 자신의 최적 경로로 설정하게 된다. 물론 이것이 궁극적인 최적 경로라고 확신할 수 없다. 그래서 향후 같은 map에 대한 다른 최적 경로를 가진 개체와의 네트워크 통신을 이용하여 최적 경로를 비교하여 갱신하는 과정을 수행함으로서 궁극적인 최적 경로를 구하는 과정을 수행하였다.

다른 map의 경우에는 개체가 그 map에 대한 정보가 없을 경우 그 정보를 최적 정보로 설정하고 이 후 위의 경우와 같은 과정을 반복하여, 최적 경로를 구하였다.

3. 결 론

본 논문에서는 네트워크를 기반으로 한 강화학습을 이용하여 임의의 map에 대한 최적 경로를 산출하여 이를 공유 및 갱신하는 과정을 제안하였다. 이 과정에서 강화학습은 정확한 데이터를 바탕으로 최적의 정책을 결정하는 데 매우 우수한 결과를 보여주었다. 또한 trial-error 과정의 반복을 방지하기 위해 각 개체간의 네트워크를 구성함으로서 최적 경로만을 공유하게 되어 새로운 경로 산출에 대한 계산 과정을 줄일 수 있을 뿐만 아니라 계속되는 갱신 과정을 통하여 데이터의 정확성 및 다양화를 추구할 수 있다는 데 그 의의가 있다.

향후에는 simulation 환경이 아닌 실제적인 하드웨어를 바탕으로 이를 구현하여 데이터의 다양화를 통한 강화학습의 가능성을 극대화하여 실제 적용 범위의 다양화를 꾀함과 동시에 네트워크 기반 환경 아래서 이를 구현함으로써 본 연구 결과를 실제에 적용하고자 한다.

[참 고 문 헌]

[1] Richard S.Sutton, "Reinforcement Learning", MIT Press, 1999

[2] Abhijit Gosavi, "Simulation-based Optimization", Kluwer Academic Publishers, 2003

[3] 김병천, "미로 환경에서 최단 경로 탐색을 위한 실시간강화학습", 정보처리학회, 2002

[4] 고광원, "강화학습과 SVM을 이용한 인지제어에 관한연구", 소음진동학회, 2003