복수의 부분 작업을 위한 확장된 Q-Learning

오도훈○ 윤소정 오경환
동양 Systems 서강대학교 컴퓨터학과 인공지능 연구실
dhoh@tysystems.com {sjyoun, kwoh}@ailab.sogang.ac.kr

Extended Q-learning under Multiple Tasks

Do-Hoon Oh○ So-Jung Youn Kyung-Hwan Oh
TongYang Systems Dept. of Computer Science, Sogang University

요 약

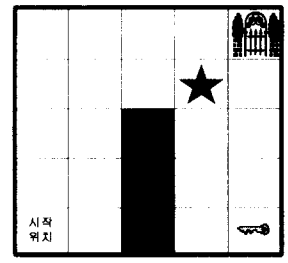
많은 학습 방법 중에서 비교적 최근에 제시된 강화학습은 동적인 환경에서 뛰어난 학습 능력을 보여주었다. 이런 장점을 바탕으로 강화학습은 학습을 기초로 하는 에이전트 연구에 많이 사용되고 있다. 하지만, 현재까지 연구 결과는 강화학습으로 구축된 에이전트로 해결할 수 있는 작업의 난이도에 한계가 있음을 보이고 있다. 특히, 복수의 부분 작업으로 구성되어 있는 복합 작업을 처리할 경우에 기존의 강화학습 방법은 문제 해결에 한계를 보여주고 있다. 본 논문에서는 복수의 부분 작업으로 구성된 복합 작업이 왜 처리하기 힘든가를 분석하고, 이런 문제를 처리할 수 있는 방안을 제안한다. 본 논문에서 제안하고 있는 EQ-Learning은 강화학습 방법의 대표적인 Q-Learning을 개량하여 기존의 문제를 해결한다. 이 방법은 각각의 부분 작업 해결 방안을 학습시키고 그 학습 결과들의 적절한 적용 순서를 찾아내 복합 작업을 해결한다. EQ-Learning의 타당성을 검증하기 위해 격자 공간에서 복수의 부분작업으로 구성된 미로 문제를 통하여 실험하였다.

1. 서론

학습 방법들 중에서 최근에 각광받고 있는 강화학습[1]은 비교적 많은 가능성을 보여주었지만 실생활에 많은 문제에 적용되지는 않았다. 이는 강화학습이 복수의 부분 작업들로 이루어져 있는 복합 작업을 처리하는 데에 한계를 보여주고 있는 것에서 기인한다. 강화학습은 작업을 완료했을 때에 주어지는 보상을 바탕으로 학습을 한다. 하지만 이런 특성으로 인해 커다란 작업에 포함되는 부분 작업들은 설사 완료되더라도 특정한 보상을 받기 힘들기 때문에, 이를 해결하는 방법은 학습되기 힘들고, 결과적으로 여러 개의 부분 작업으로 이루어진 작업은 기존의 방법으로 학습할 수 없게 된다. 이를 해결하기 위해 본 논문에서는 강화학습의 방법으로 많이 사용되는 Q-Learning[2]을 개량해서 복수의 부분 작업들로 구성된 복합 작업을 처리할 수 있는 EQ-Learning을 제안한다.

2. 복합작업

<그림 1>의 예를 통해 복합작업에 대해 알아보자. <그림 1>의 미로를 탈출하기 위해서는 오른쪽 아래에 있는 '열쇠'를 습득한 후에 왼쪽 위에 있는 '출구'를 열고 나가야 한다. 즉, [열쇠 탐색] - [출구 탐색] - [탈출]



<그림 1> 5 x 5 미로

이라는 일련의 과정을 거쳐야 한다. 이런 일련의 과정을 처리하는 방법을 학습시키려면 <그림 1>에서 '별'이 위치한 곳에서 열쇠의 획득 여부에 따라 다르게 행동해야 하는 것이 중요하다. 즉, 열쇠를 획득하지 못한 경우라면 열쇠를 획득하기 위해 아래로 가는 행동을 학습해야 하고, 이미 열쇠를 획득한 경우라면 출구를 향하는 행동을 학습해야 한다. 그러나 강화학습은 단순하게 행동에 따라 주어지는 보상을 크게 하려는 방향으로만 학습하게 된다. 이러한 강화학습의 특징으로 인해 결국 <그림 1>에 나타낸 것과 같이 간단한 문제조차 강화학습으로는 해결할 수 없게 된다.

복합작업은 전체 작업이 몇 개의 부분 작업으로 이루어진 것을 칭한다. <그림 1>에서의 예는 '열쇠를 탐색하는 작업'과 '출구를 탐색하는 작업', 2 개의 부분 작업으로 이루어진 복합작업이다. 강화학습이 이러한 복합작업들을 처리하기 힘든 이유는 강화학습이 단순히 전체 작업

에 대한 보상만을 고려하고 현재 처리되어야 할 부분 작업에 대한 고려를 못한다는 점에서 기인한다[3].

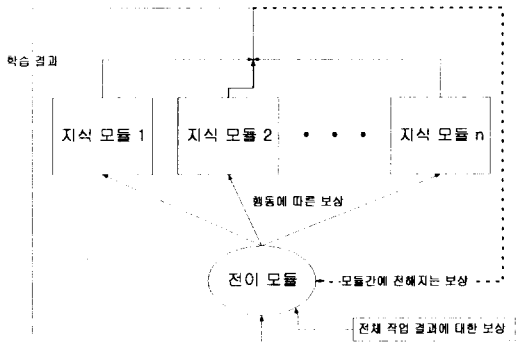
3. EQ-Learning

3.1 지식모듈

일반적으로 여러 개의 부분 작업으로 이루어진 복합 작업을 처리하기 위해서는 각각의 부분 작업을 처리하는 전략과 처리해야할 부분 작업의 순서를 정해야 한다. 이 논문에서는 부분 작업을 처리하기 위한 전략을 지식 모듈이라 정의하며, 이런 지식 모듈이 Q-Learning으로 학습되기 때문에 각각의 지식 모듈은 Q-Table형태로 되어 있다.

3.2 EQ-Learning의 구조

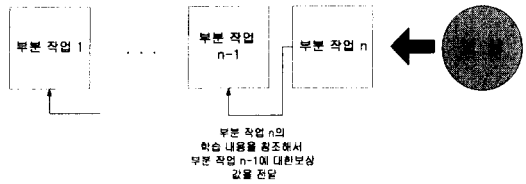
n 개의 부분 작업으로 구성된 복합작업을 수행하기 위한 EQ-Learner의 전체적인 구조는 <그림 2>와 같다.



<그림 2> EQ-Learning의 구조

<그림 2>에서 지식 모듈들은 각각 하나의 부분 작업을 수행할 수 있는 방법을 학습하게 된다. 그리고 이러한 지식 모듈들이 어떠한 부분 작업을 맡아서 학습하게 되는가를 전이 모듈에서 제어하게 된다. 전이 모듈에서 하는 일은 크게 두 가지이다. 첫째 현재 수행해야 하는 부분 작업을 인지하고 해결하는 방법을 학습시킬 지식 모듈을 선택하고, 둘째 전체 작업이나 부분 작업을 완료했을 경우에 해당 지식 모듈에 적절한 보상을 하는 임무이다. 전이 모듈은 초기에 임의의 지식 모듈을 하나 선택해 현재 작업을 수행하게 한다. 해당 지식 모듈은 Q-Learning의 방법을 이용하여 부분 작업을 처리하는 방법을 학습하게 되고, 학습의 매 단계마다 도출되는 결과가 다시 전이 모듈의 입력으로 들어가게 된다. 전이 모듈은 현재 활동하는 지식 모듈의 수행 결과가 현재 작업의 목표를 달성하면 다른 지식 모듈을 활동하게 하거나 작업을 종료시키고, 작업의 목표를 달성하지 못하면 계속해서 하나의 지식 모듈에게 작업을 수행하게 한다.

부분 작업이 종료되면 전이 모듈은 다른 지식 모듈에서 다른 부분 작업을 수행하기 시작한다. 이 때, 이전까지 수행하던 지식 모듈은 부분 작업이 완료되어도 특정한 보상이 주어지지 않기 때문에, 학습을 제대로 이루지 못하게 된다. EQ-Learning에서는 이 문제를 해결하기 위해 이어지는 지식 모듈에 저장되어 있는 축적된 보상을 참조해 이전까지 처리한 부분 작업에 대해 보상을 내린다. 결과적으로 지식 모듈은 최종 결과에 대한 보상을 이전까지 수행했던 부분 작업들에 대해서 역전파시키는 역할을 한다. 이런 일련의 과정을 그림으로 나타낸 것이 <그림 3>이다.



<그림 3> 전체 작업에 대한 보상의 역전파

3.3 EQ-Learning의 학습 방법

위에서 제시되었던 EQ-Learning은 아래와 같은 순서에 의해 구성된다. 기존 Q-Learning의 방법을 바탕으로 해서 본 논문에서 제시하고 있는 EQ-Learning은 (1)을 축으로 이루어진다.

$$EQ_t(ta,x,a) = (1-a)Q_{t-1}(ta,x,a) + a[r(T) + \gamma U(ta,y)] \quad (1)$$

단, $U(ta,y) = \max_{a \in A} EQ(ta,y,a)$

r(T)는 보상 함수, ta는 현재 수행하는 부분 작업, a 는 학습률을 나타내고 있다.

일반적인 Q-Learning이 단순히 상태와 해당 상태에서의 행동만을 고려해서 작업의 해답을 구축하는 데 반해, EQ-Learning은 기존의 상태 x와 행동 a를 취할 때 수행해야 하는 부분작업 ta를 고려해서 학습을 이루고 있다. 하지만, 전체 작업이 끝날 때에만 학습을 위한 보상이 내려지게 되므로, 위에서 보상 함수를 나타낸 $r(T)$는 이 점을 고려해서 설계해야 한다. 전체 작업을 T, T를 이루는 부분 작업들을 t_i(단, $0 \le i \le n$, n은 부분 작업의 갯수)라 하자. 부분 작업 t_{k-1}이 완료된 후에 부분 작업 t_k를 시작한다(단, $1 \le k \le n$). 그러면 전체 작업 T는 $T = \{t_1, t_2, t_3 \ldots t_n\}$라고 표기할 수 있다. 작업에 대한 보상은 부분 작업 t_{n-1}까지의 모든 부분 작업을 합당한 순서에 따라 모두 완료한 후에 부분 작업 t_n의 수행을 끝내면 주어지게 된다. 보상이 전체 작업이 끝났을 경우에만 주어지므로 각각의 부분 작업을 수행하는 방법을 학습하기 위해서는 위에서 언급했던 대로 전체 작업에 대한 보상을 각각의 부분 작업에 대해 역전파 시켜줘야 한다. 이를 위해 보상함수를 아래의 식들과 같이 정의한다. 우선 전체 작업이 끝났을 때의 보상 함수는 (2)와 같이 일정한 상수를 부여하는 방식으로 이루어진다.

$$r(T) = c \qquad \text{(단, c는 상수)} \qquad (2)$$

그리고, k 번째 부분 작업을 수행하는 중에 특정 상태에서 취한 행동에 의해 상태 y로 전이되면서 수행하던 부분작업이 완료하게 되면 (3)과 같이 정의된 보상이 주어지게 된다.

$$r_k(T) = \max_{a \in A} EQ(ta_{k+1}, y, a) \qquad (3)$$

r_k 는 k번째 부분작업이 완료되었을 때의 보상,
s_k는 k번째 부분작업을 수행하는 상황(단, $0 \le k \le n-1$)

위의 식은 하나의 부분 작업이 끝났을 때의 상태는 다음에 이어지는 부분 작업이 시작하는 상태와 일치한다는 점에서 착안한 것이며, 위에서 언급했던 전체 작업에 대한 보상을 부분 작업으로 역전파시킨다. 이 과정을 다음과 같이 자세히 표현할 수 있다.

t_n에 대한 보상 : $R_n = c$ (전체 작업이 끝났을 경우)

t_{n-1}에 대한 보상 : $R_{n-1} = \gamma^{m(n-1)} R_n = \gamma^{m(n-1)} c$

.
.
.

t_1에 대한 보상 : $R_1 = \gamma^{m(1)} R_2 = \gamma^{m(1)} \gamma^{m(2)} \cdots \gamma^{m(n)} c$

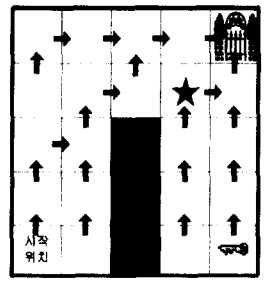
R_k는 부분 작업 k가 끝났을 때의 보상이고 c는 전체 작업 완료에 따른 보상 정도이며 γ 는 감소 인자, $m(k)$는 부분 작업 k를 수행할 때 거치게 되는 상태의 수를 나타낸다.

이와 같이 EQ-Learning은 전체 작업에 대한 보상을 각각의 부분 작업으로 역전파시킴으로써 학습의 과정에서 필요한 보상을 전달해주고 있고, 이를 통해서 부분 작업들의 해결 방법이 각각의 지식 모듈에 작성될 수 있다.

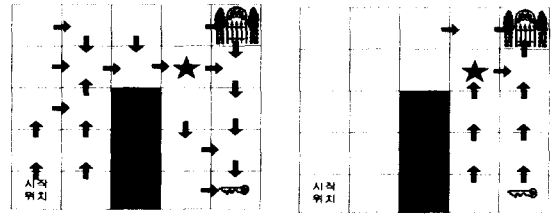
4. 실험 및 결과

실험 공간은 <그림 1>과 같다. 이 실험공간에서 약 30회에 걸친 학습 결과의 평균으로 구해진 값을 통해 특정 상태에서 선호하는 행동 양식을 알아보았다. 한 회의 학습은 500번의 학습 단계에 걸쳐 반복적으로 이루어지며, 1번의 학습 단계는 '시작위치'에서 출발한 에이전트가 열쇠를 획득한 후에 오른쪽 위의 출구에 도달하거나 1000회의 행동으로도 이 문제를 해결하지 못하면 문제를 해결하지 못하고 종료하게 된다. <그림 4>는 기존의 Q-Learning을 통해 위에서 언급한 실험 과정의 결과로써 얻어진 행동 양식을 나타내고 있으며 각각의 행동은 해당 상태에서 가장 높은 확률로 취해진 행동들이다.

<그림 4>에서 나타내고 있는 학습 결과는 시작 위치에서 열쇠를 획득하는 과정이 바르게 학습되지 않은 것을 보여주고 있다. 이렇듯 일반적인 강화학습은 하나의 상태에 대해 하나의 행동만을 학습하게 되기 때문에, 위와 같은 복합작업 문제를 해결할 수 없다. EQ-Learning을 사용한 학습 결과는 <그림 5>에서 나타난다. EQ-Learning은 복합작업을 구성하고 있는 부분 작업에 따라 나누어 각각 학습을 하며, 만족할만한 학습 결과를 보여준다. <그림 5>에서 보는 것과 같이 EQ-Learning은 부분 작업에 따라 지식 모듈을 나누어서 각각의 부분작업에 적절한 행동을 학습한다.



<그림 4> Q-Learning의 학습 결과



(a) 열쇠를 획득하는 작업의 학습 결과 (b) 출구를 탐색하는 작업의 학습 결과

<그림 5> EQ-Learning의 실제 실험 결과

이와 같이 본 논문에서 제시한 EQ-Learning은 해결해야 할 부분 작업을 처리하는데 지식모듈을 사용함으로써 첫 번째 언급되었던 문제를 극복하였고, 결과에 대한 보상이 일정한 감소 인자에 의해 다른 지식 모듈들로 감소되어 가면서 전이되어 가기 때문에 이 차이에 의해 지식 모듈간의 순서를 정함으로써 두 번째에 제시된 문제를 해결할 수 있었다.

5. 결론

본 논문에서는 기존의 강화학습 방법으로 쉽게 해결할 수 없었던 복합 작업을 처리할 수 있는 EQ-Learning의 방법을 제안하였다. 제안된 방법은 기존에 제시되었던 방법들이 가지고 있었던 본질적인 문제를 해결함과 동시에 그 전까지 존재했던 제약사항들을 극복하고 나아진 성능을 보여주었다. 무엇보다 이전에 제시되었던 방법들에 비해 구조적으로 간단해졌기 때문에 어느 문제에서나 적용하기 쉽다는 장점을 가지고 있다.

6. 참고문헌

[1] Kaelbling, Lesile P and Michael L. Littman. "Reinforcement Learning : A Survey", 1996

[2] Christoper J. C. H. Watkins and Peter Dayan, "Q-Learning", 1992

[3] S. P. Singh "On The Efficient Learning of Multiple Task Sequences", 1992