2002년 한국정보처리학회 춘계학술발표논문집 제9권 제1호

# 다중 에이전트 강화학습을 위한 SOM 기반의 일반화

임문택*, 김인철**
경기대학교 전자계산학과
e-mail : {asublim*, kic**}@kyonggi.ac.kr

# SOM_Based Generalization for Multiagent Reinforcement Learning

Mun-Tack Lim*, In-Cheol Kim**
Department of Computer Science, Kyonggi University

요 약

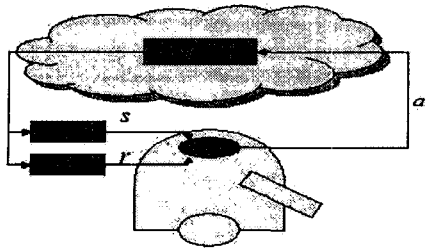
본 논문에서는 에이전트간의 통신이 불가능한 다중 에이전트 환경에서 각 에이전트들이 독립적이면서 대표적인 강화학습법인 Q-학습을 전개함으로써 서로 효과적으로 협조할 수 있는 행동전략을 학습하려고 한다. 하지만 단일 에이전트 경우에 비해 보다 큰 상태-행동공간을 갖는 다중 에이전트환경에서는 강화학습을 통해 효과적으로 최적의 행동 전략에 도달하기 어렵다는 문제점이 있다. 이 문제에 대한 기존의 접근방법은 크게 모듈화 방법과 일반화 방법이 제안되었으나 모두 나름의 제한을 가지고 있다. 본 논문에서는 대표적인 다중 에이전트 학습 문제의 예로서 the Prey and Hunters Problem 를 소개하고 이 문제영역을 통해 이와 같은 강화학습의 문제점을 살펴보고, 해결책으로 신경망 SOM 을 이용한 일반화 방법을 제안한다. 이 방법은 다층 퍼셉트론 신경망과 역전파 알고리즘을 이용한 기존의 일반화 방법과는 달리 군집화 기능을 제공하는 신경망 SOM 을 이용함으로써 명확한 다수의 훈련 예가 없어도 효과적으로 채 경험하지 못한 상태-행동들에 대한 Q 값을 예측하고 이용할 수 있다는 장점이 있다.

## 1. 서론

다중 에이전트 학습이란 다중 에이전트 환경에서 에이전트간의 조정을 위한 행동전략을 학습하는 것을 말한다. 본 논문에서는 에이전트간의 통신이 불가능한 다중 에이전트 환경에서 각 에이전트들이 독립적이면서 대표적인 강화학습법인 Q-학습을 전개함으로써 서로 효과적으로 협조할 수 있는 행동전략을 학습하려고 한다. 하지만 단일 에이전트 경우에 비해 보다 큰 상태-행동공간을 갖는 다중 에이전트환경에서는 강화학습을 통해 효과적으로 최적의 행동 전략에 도달하기 어렵다는 문제점이 있다. 이 문제에 대한 기존의 접근방법은 크게 모듈화 방법과 일반화 방법이 제안되었으나 모두 나름의 제한을 가지고 있다. 본 논문에서는 대표적인 다중 에이전트 학습 문제의 예로서 the Prey and Hunters Problem 를 소개하고 이 문제영역을 통해 이와 같은 강화학습의 문제점을 살펴보고, 해결책으로 신경망 SOM 을 이용한 일반화 방법 QSOM 을 제안한다. 이 방법은 기존의 일반화 방법과는 달리 군집화 기능을 제공하는 신경망 SOM 을 이용함으로써 명확한 다수의 훈련 예가 없어도 효과적으로 이전에 경험하지 못한 상태-행동들에 대한 Q 값을 예측하고 이용할 수 있다는 장점이 있다. 후속 연구를 통해 본 논문에서 제시한 QSOM 학습법의 일반화 효과와 성능을 평가할 계획이다.

## 2. 강화학습

강화학습(reinforcement learning)이란 에이전트 행동에 따라 주어지는 누적 보상 값(reward)을 최대화 할 수 있는 최적의 행동 전략을 학습하는 것이다.




[그림 1] 강화학습에서 에이전트와 환경의 상호작용

강화학습에서는 [그림 1]과 같은 에이전트와 환경과의 상호작용을 가정한다. 즉, 에이전트는 환경의 현재 상태에 적합한 행동을 선택하여 수행하고 환경은 그 행동에 따라서 보상값과 다음 상태를 에이전트에게 제공한다, 그러면 에이전트는 다시 새로운 상태에 적합한 행동을 수행하는 과정을 반복해 나간다. 그리고 이러한 과정이 진행되는 동안 에이전트의 목표는 환경으로부터 얻을 수 있는 보상값의 합이 최대가 될 수 있는 행동전략을 학습하는 것이다. 따라서 강화학습에서 최적의 행동 전략 $\pi^*$ 는 [식 1]과 같이 누적 보상값을 최대화하는 행동전략을 말한다.

$$V^{\pi}(s_t) = \gamma^0 r_t + \gamma^1 r_{t+1} + \gamma^2 r_{t+2} + \ldots$$
$$\pi^* = \arg\max_{\pi} V^{\pi}(s), (\forall s) \qquad \text{[식 1]}$$

가장 널리 이용되는 Q-학습은 환경 모델을 필요로 하지 않는 대표적인 강화학습 방법이다. Q-학습에서는 상태 s 와 행동 a 쌍에 대한 평가함수 Q (s, a) 값을 예측하고 이 Q 값을 기초로 행동을 선택한다. 즉, 통상적으로 Q 값이 가장 큰 행동을 선택한다. 그러나 사전에 정확한 평가함수 Q(s,a)를 알 수는 없고 단지 에이전트가 실제로 상태 s 와 행동 a 를 직접 반복 경험함으로써 추정할 수 있을 뿐이다. 따라서 보다 정확한 Q 값을 추정해야만, 이것을 기초로 보다 올바른 행동을 결정할 수 있으므로, 최적의 Q 함수값에 수렴한 것은 곧 최적의 행동전략을 학습한 것을 의미한다. 따라서 Q-학습에서는 모든 상태-행동(s, a) 쌍에 따른 평가함수 값 Q (s, a)을 저장 운영하는 Q-표를 필요로 한다. [식 2] 는 Q 학습의 학습규칙으로서, Q 함수값은 기존의 Q 함수값과 실제 경험으로 얻어지는 보상값 r 에 의해 새롭게 갱신된다.

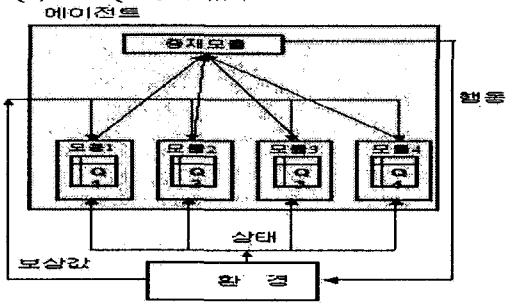
$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)) \text{ [식 2]}$$

Q-학습에서는 한번의 행동 경험을 통해 특정 상태-행동쌍(s, a)에 대한 평가함수값 Q (s, a)가 한번만 갱신된다. 따라서 Q-학습이 수렴하기 위해서는 가능한 모든 상태-행동 쌍에 대한 충분한 반복 경험을 필요로 하기 때문에 큰 상태-행동 공간을 갖는 복잡한 학습문제의 경우에는 이전에 경험해보지 못했거나 경험이 충분치 않은 상태-행동들이 많아 효과적인 학습을 기대하기 힘들다.

## 3. 모듈화와 일반화

### 3.1 모듈화

Q-학습을 비롯한 대부분의 강화학습에서 문제가 되는 큰 상태-행동 공간문제를 해결하고자 하는 기존의 연구들은 크게 모듈화 방법과 일반화 방법으로 나눌 수 있다. 모듈화 방법에서는 큰 상태-행동 공간을 몇 개의 작은 모듈로 나누어, 모듈별로 별도의 Q 학습을 전개한 뒤 각 모듈의 Q-학습 결과를 취합하여 행동을 결정하는 방법이다. 모듈화 방법에는 Modular-Q, AMQL, AMMQL 등이 있다.




[그림 2] Modular-Q 학습법

[그림 2]는 대표적인 모듈화 방법인 Modular-Q 학습법을 표현한 것이다. Modular-Q 학습을 전개하는 에이전트는 Q-학습을 전개하는 n 개의 모듈과 그 결과를 결합함으로써 행동을 결정하는 1 개의 중재 모듈을 가지고 있다. Modular-Q 학습과정은 다음과 같다. 먼저 에이전트의 행동선택 단계에서는 현재 상태에서 가능한 각 행동들에 대한 Q 값을 각 모듈들이 제공하면 중재모듈에서 이 값들을 취합하여 [식 3]과 같이 Q 값의 합이 최대가 되는 행동을 선택하고 이것을 실행하게 된다. 그 결과 환경으로부터 보상값이 주어지면 이 보상값은 각 모듈에 입력되어 Q 함수값을 새롭게 갱신하는데 사용되어진다.

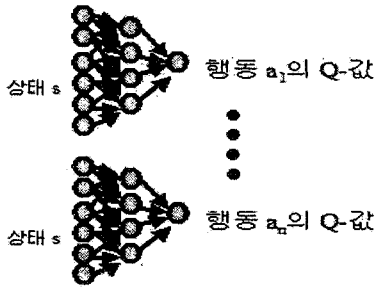
$$a^* \leftarrow \arg\max_{a \in A} \sum_{i=1}^{n} Q_i(s,a) \qquad \text{[식 3]}$$

이러한 모듈화 방법은 다음과 같이 몇 가지 문제점을 가지고 있다. 첫째, 이 방법은 사전에 잘 정의된 모듈들을 필요로 하며, 효과적인 모듈화를 위해서는 많은 영역지식이 필요하다. 둘째, 각 모듈별 학습 결과를 결합하는 중재모듈의 결합 방식이 매우 단순하고 고정적이다. 셋째는 결과적으로 모듈화를 통해 각 모듈에서 고려하는 환경요소는 한 두개 정도로 줄어들어 실제 문제의 복잡도에 비해 지나치게 단순화된다는 것이다.

### 3.2 일반화

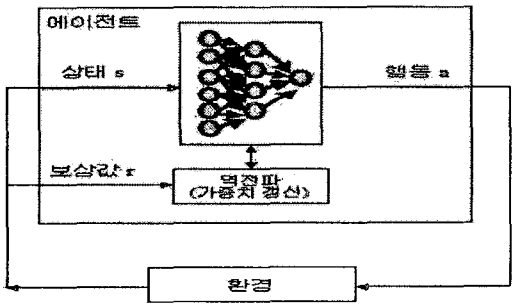
일반화(generalization) 방법은 문제공간을 작은 조각으로 나누는 모듈화와는 달리 귀납적 학습(inductive learning) 기법을 적용함으로써 경험해본 일부 상태-행동 쌍의 Q 함수값을 기초로 나머지 경험하지 못한 상태-행동 쌍에 대한 Q 함수값을 예측하는 방법들을 말한다. 기존의 일반화 방법에는 적용하는 귀납적 학습 기법에 따라 신경망(neural net)을 이용한 QCON 과

SGA 알고리즘 그리고 결정트리(decision tree)를 이용한 G-학습 알고리즘 등이 있다..




[그림 3] 다층 퍼셉트론(입력:상태, 출력:Q 값)

대표적인 일반화 방법인 QCON 알고리즘은 [그림 3]과 같이 각 행동마다 상태를 입력으로, Q 함수값을 출력으로 하는 단일 출력의 다층 퍼셉트론(multi-layer perceptron)을 두고 이것을 이용함으로써 경험하지 못한 상태들에 대한 Q 함수값도 예측할 수 있게 하였다.




[그림 4] QCON 학습법

[그림 4]는 전체적인 QCON 학습과정을 나타낸다. 에이전트는 현재 상태에서 취할 행동을 선택하기 위해 현재 상태를 [그림 3]과 같은 각 신경망의 입력으로 제공함으로써 가능한 모든 행동의 Q 함수값을 예측할 수 있고 그러면 [식 4]와 같이 이들 중 가장 큰 Q 함수값을 가지는 행동을 선택한다.

$$a^* = \arg\max_{a \in A} QNET_a(s) \qquad \text{[식 4]}$$

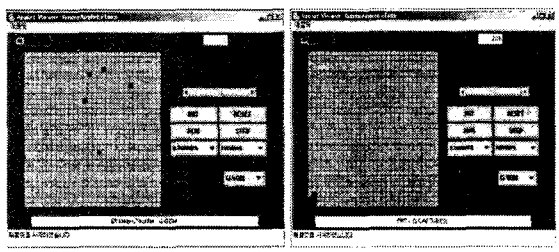
선택된 행동이 실행되고 나서 환경으로부터 새로운 상태와 보상값 r 이 주어지면, 이 보상값 r 은 [식 5]에 의해 새로운 Q 함수값 $Q_{a^*}(s)$ 을 구하는데 이용된다. 그리고 이 새로운 Q 함수값 $Q_{a^*}(s)$ 와 신경망의 출력으로 제시된 기존의 Q 함수값 $QNET_{a^*}(s)$ 의 차(difference)를 오차(error)로 삼아 역전파 알고리즘(backpropagation algorithm)을 적용하여 신경망의 가중치를 갱신한다.

$$Q_{a^*} \leftarrow r + \gamma(\max_{a \in A} QNET_a(s')) \qquad \text{[식 5]}$$
$$\Delta Q_{a^*} \leftarrow Q_{a^*} - QNET_{a^*}(s)$$

그러나 다층 퍼셉트론과 역전파 알고리즘에 기초한 교사학습(supervised learning)을 적용한 QCON 은 효과적인 학습을 위해서는 다수의 정확한 훈련 예가 필요하다. 하지만 훈련 예로 사용할 정확한 Q 값을 미리 알 수 없으므로 역시 오차를 포함한 추정치인 Q 값을 실제 훈련에 사용함으로써 오차가 커져 높은 성능을 기대하기 어렵다.

## 4. The Prey and Hunters Problem

The Prey and Hunters Problem 은 [그림 5]와 같이 격자세계(grid world)에서 도망가는 하나의 prey 와 이를 포위해 잡으려는 다수의 hunter 들로 이루어진 문제로서, 서로 통신이 불가능한 hunter 들이 어떤 행동전략을 써야 효과적으로 협조하여 prey 를 잡을 수 있느냐 하는 다중 에이전트 학습 문제이다. Prey 및 hunter 에이전트는 매 단계마다 동시에 현재 위치에서 5 개의 이동 동작(동, 서, 남, 북, 현재위치) 중 하나를 실행한다. [그림 5]의 왼쪽은 문제의 초기 상태에서 각 에이전트의 위치를 나타내고, 오른쪽은 prey 에이전트가 잡힌 최종 상태를 나타낸다. 만약 hunter 에이전트가 각자 독립적인 Q 학습을 전개하여 행동을 결정한다고 가정해보자. 격자세계의 크기가 30x30 인 경우, 5 명의 에이전트의 위치정보로 표현되는 상태공간은 $900x900x900x900x900=900^5$ 의 크기를 갖게 된다.

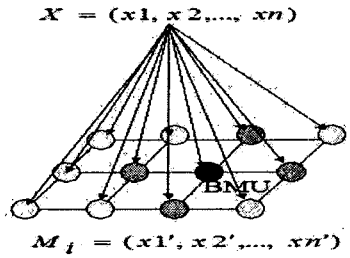


[그림 5] The Prey and Hunters Problem

따라서 이와 같이 상태공간이 큰 문제에 단순히 Q 학습과 같은 강화학습을 적용할 경우, 제한된 시간 내에 효과적인 행동전략을 학습할 수 없게 된다.

## 5. QSOM 학습법

신경망 SOM(Self Organizing Map)은 격자 모양으로 연결된 단층의 뉴런들로 구성된 신경망으로서, [그림 6]의 X 와 같은 입력벡터들이 주어지면 이들을 가장 잘 매치되는 뉴런(Best Matching Unit, BMU)에 배정함으로써 유사한 입력들에 대한 군집화(clustering)를 해준다. SOM 에서 각 뉴런은 하나의 클러스터를 나타내며, 뉴런별로 이 클러스터에 속한 구성원들의 속성값에 기초한 가중치벡터(weight vector)를 유지한다. 각 뉴런의 가중치벡터와의 비교를 통해 새로운 입력 X 가 BMU 에 배정이 되면 이 BMU 와 이웃한 뉴런들의 가중치는 입력 X 쪽에 가깝도록 갱신된다. 이와 같은 SOM 의 학습과정은 결국 유사한 입력 데이터들을 몇 개의 클러스터로 군집화함과 동시에 유사한 클러스터들을 인접한 위치에 배치시킴으로써 군집화의 결과를 시각화하는데 유리한 특징을 가지고 있다.



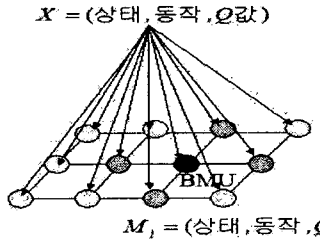

[그림 6] 신경망 SOM

[식 6]은 BMU(Best Matching Unit) 계산식으로서, 입력벡터 X 와 뉴런의 가중치 벡터 $M_i$와의 거리가 가장 가까운 뉴런이 BMU 로 선택된다.

$$\|X - M_{bmu}\| = \min\{\|X - M_i\|\} \quad [식\ 6]$$

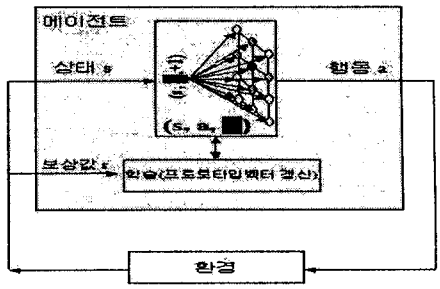
유사한 입력 데이터들에 대한 군집화 기능을 제공하는 신경망 SOM 을 일반화에 이용하는 방법은 간단하다. 서로 다른 상태나 행동을 하나씩 독립적으로 구분하지 않고 유사한 것들끼리 묶어 군집화를 이루고 이 클러스터별로 동일한 출력을 공유하도록 함으로써 일반화를 이룰 수 있다. 이와 같이 신경망 SOM 을 일반화에 이용하면 기대할 수 있는 가장 큰 장점은 비교사 학습(unsupervised learning)으로서 정확한 훈련 예가 없어도 높은 성능의 일반화가 가능하다는 점이다. 따라서 본 논문에서는 강화학습을 위한 일반화에 신경망 SOM 을 적용한 SOM 알고리즘을 제안한다.

QSOM 에서 각 뉴런의 가중치벡터와 입력벡터는 [그림 7]과 같이 (상태, 동작, Q 값)으로 구성된 3 차원 벡터로 표현된다.




[그림 7] QSOM 의 입력벡터 및 가중치벡터

QSOM 의 전체적인 학습과정은 [그림 8]과 같다. 먼저 현재 상태에서 에이전트가 취할 행동을 결정하기 위해서는 입력벡터 X=(현재상태, ■ ,+1) 에 대한 BMU 을 선택한 뒤 그 BMU 의 가중치벡터 $M_i$ =(상태,동작,Q 값)에 기술된 동작에 따라 행동한다. 그 결과 환경으로부터 보상값 r 이 주어지면 이번에는 입력벡터 X'=(현재상태, 실행한 동작, ■ ) 에 대한 BMU 을 선택하고, 벡터 (현재상태, 실행한 동작, 보상값)를 이용하여 BMU 와 이웃 뉴런들의 가중치벡터 $M_i$ =(상태,동작,Q 값)를 갱신한다.




[그림 8] QSOM 학습법

## 6. 실험평가 계획

The Prey and Hunters Problem 영역의 실험을 통해 본 논문에서 제시한 QSOM 학습방법의 일반화 효과를 평가 할 것이다. 평가항목으로는 평가함수 값의 수렴속도, 행동의 최적성 그리고 Prey 에 포획될 때 까지 소요된 총 동작 수(시간)를 산출하고 각 Hunter 에이전트의 학습 진행 상황인 SOM 에서 출력 뉴런 $M_i$ =(상태,동작,Q 값)의 변화과정을 추적하게 될 것이다. 그리고 실험평가 전에 결정해야 할 학습인자로는 SOM 에서 Map 을 구성하는 뉴런 수와 배열방식과 학습률, 경감률 등을 결정해야 한다.

## 7. 결론 및 향후연구

본 논문에서는 Q 학습의 큰 상태공간 문제를 해결하기 위한 방법으로 신경망을 이용한 일반화 방법을 제안 하였다. 특히 일반화 방법으로 SOM 을 이용한 QSOM 을 제안하였으며 QSOM 학습방법은 Q-표 대신 신경망 SOM 을 사용하였고 QSOM 의 특징으로는 비교사 학습으로 정확한 훈련 예 없이도 효과적인 학습이 가능하다는 것이다. 향후 연구로는 the Prey and Hunters Problem 에 대한 실험을 통해 본 논문에서 제안한 QSOM 의 성능과 효과를 입증하는 것이다.

## 참고문헌

[1] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore, "Reinforcement Learning: A Survey", Journal of AI Research Vol.4, pp. 147-166, 1996.

[2] Long-Ji Lin, "Self-Improving Reactive Agent Based On Reinforcement Learning, Planning and Teaching", Machine Learning, vol. 8, pp.293-321, 1992.

[3] Norihiko Ono, Kenji Fukumoto, "Multi-agent Reinforcement Learning : A Modular Approach", Proc. of ICMAS-96, pp. 252-258, 1996.

[4] Rishard S. Sutton and A.G. Barto, "Reinforcement Learning : An Introduction", MIT Press, 1998.

[5] Takayuki Kohri, Kei Matsubayashi, Mario Tokoro, "An Adaptive Architecure for Modular Q-Learning", Journal of AI Research, 1998