

신경망을 이용한 멀티 에이전트 기반 대공방어 단위 학습모형

Anti-air Unit Learning Model Based on Multi-agent System Using Neural Network

최 명 진* 이 상 현*
Choi, Myung-Jin Lee, Sang-Heon

ABSTRACT

In this paper, we suggested a methodology that can be used by an agent to learn models of other agents in a multi-agent system. To construct these model, we used influence diagram as a modeling tool. We present a method for learning models of the other agents at the decision nodes, value nodes, and chance nodes in influence diagram. We concentrated on learning of the other agents at the value node by using neural network learning technique. Furthermore, we treated anti-air units in anti-air defense domain as agents in multi-agent system.

주요기술용어(주제어) : Neural Network(신경망), Anti-air Unit(대공방어 유닛), Multi-agent System(멀티 에이전트 시스템), Influence Diagram(영향도)

1. 머리말

최근 첨단 과학기술의 발전으로 미사일 등 유도무기의 사정거리와 정밀도가 향상되어 이에 대비한 미사일 방어의 중요성이 부각되고 있다. 본 연구는 대공방어 지역 내에 다수의 방어 유닛(Unit) 간에 각자 어느 공격 미사일을 담당할 것인지에 대한 의사결정 모형 구축에 대한 방법론을 제시한다. 이는 유닛 간의 실시간 의사결정을 통해 동일 표적에 대한 중복 담당을 방지하는 지능적 표적 담당 모형구축에 대한 연구에 해당된다.

멀티 에이전트 시스템(MAS : Multi-Agent System)에서 에이전트는 다른 에이전트와 상호작용을 하게 된

다. 이러한 다수 에이전트가 서로 상호작용을 통해 자신들의 행위가 전체 시스템 효율을 저해하지 않도록 조화시키는 기술을 찾아내는 핵심으로서 현재 활발히 연구가 진행되고 있는 분야이다.

일부 연구에서 에이전트 모형을 구축하는 방법이 제시되었는데 류경현 등^[1]은 파레토 최적(Pareto Optimal) 관점에서 에이전트들이 제안한 해를 가지고 퍼지 의사결정에 기반한 멀티 에이전트의 효율적인 조정방안을 제시하였고, Kaminka *et al.*^[4]은 공중 전투 시뮬레이션 환경에서 인식작업을 계획하는 규칙(Rule) 기반 모형을 연구하였으며, Carmel *et al.*^[1]은 상대 에이전트의 전략을 모형화하는 유한한 오토마타(자동장치)의 활용에 관하여 연구하였다. 이러한 에이전트 모형을 구성할 수 있는 상이한 대안의 의사결정 도구로 영향도(Influence Diagram)가 사용될 수 있다.

본 연구는 하나의 에이전트인 대공방어 유닛이 다른 유닛의 행동패턴을 학습하여 서로 간 중복 없이

† 2008년 9월 1일 접수~2008년 10월 10일 게재승인

* 국방대학교(National Defense University)

주저자 이메일 : ch01mj@naver.com

공격해 오는 미사일의 요격을 담당하는 의사결정 모형 구축에 대한 방법론 개발을 목적으로 하여 모형 구축의 도구로 영향도를 사용하였다. 영향도를 통해 유닛 간의 의사결정 모형의 프레임웍을 구축하고, 영향도의 각 노드에서의 다른 유닛에 대한 학습문제를 취급한다. 특히 신경망(Neural Networks)을 통해 가치 노드(Value Node)에서의 학습문제를 비증 있게 제시한다.

2. 이론적 배경

가. 영향도(Influence Diagram)

영향도는 복잡한 의사결정 문제를 묘사하는 가시적인(Graphical) 도구로서 불확실한 데이터와 정보 간의 조건적인 의존성을 표현하는 노드(Node)들과 조건적이고 정보적인 방향성을 가지는 아크(Directed Arc)들로 이루어진 네트워크(Networks)이다.

영향도는 네 개의 노드로 이루어지는 방향성이 있는 비주기적인(Non-cyclic) 그래프로서 구성되어 있으며 기회 노드(Chance Node), 결정 노드(Decision Node), 가치 노드(Value Node), 그리고 결정적 노드(Deterministic Node)가 있다. 일반적으로 원형으로 표현되는 기회 노드는 불확실한 양(Random Quantity), 사각형으로 표현되는 결정 노드는 결심(Decisions), 다이아몬드형으로 표현되는 가치 노드는 효용(Utility), 그리고 이중 원형으로 표현되는 결정적 노드는 결정적인 양(Deterministic Quantity)을 표현한다. 방향성이 있는 아크(Arc)는 연결된 노드 간의 관계를 나타내고 기회 노드로 향하는 아크는 확률적인 관련성을 나타낸다. 결정 노드로 향하는 아크는 결심점에서의 정보의 가용성을 나타내고 두 결정 노드 간의 아크는 결정의 순차적인 순서를 나타낸다. 또한 가치 노드로 향하는 아크는 가능한 결과에 대한 선호도를 나타낸다^[10].

나. 신경망(Neural Network)

신경망 모형은 경험을 통해 학습하는 인간 두뇌의 신경망 활동을 모방하여 입력 데이터를 통한 반복적 학습을 거쳐 패턴을 찾아내고, 이를 일반화하여 향후

미래 입력 데이터에 대한 출력값을 예측하는데 유용하게 사용되는 기법이다. 1943년 매컬러와 피츠^[6]에 의해 뉴런의 모형화가 최초로 시도된 이래, 1949년 캐나다의 심리학자 헵^[3]에 의해 두 뉴런 사이의 연결강도를 조절하는 학습규칙이 개발되었다. 1958년 로젠블릿^[7]에 의해 새로운 학습방법인 델타규칙에 입각한 최초의 실제적인 신경망 모형이 제시되었고, 1986년 러멜하트^[8]가 일반화된 델타규칙을 다층 신경망에 적용하여 신경망 연구에 새로운 전기를 마련하였다.

다층 퍼셉트론(Multi Layer Perceptron)에 일반화된 델타규칙을 학습규칙으로 사용한 BP 알고리즘(Backpropagation Algorithm)은 특정 뉴런의 활성이 다른 뉴런의 오류 출력에 공헌한다면 두 뉴런 간의 연결 가중치를 오류치에 비례하여 조절해야 하며, 이러한 과정은 그 아래 층(Layer)에 있는 신경세포들까지 계속된다는 특징을 갖는다. 이처럼 출력층에서 발생한 에러를 아래층으로 역전파시키는 오류 역전파(Error Backpropagation) 알고리즘을 적용한다.

입력층(Input Layer), 은닉층(Hidden Layer), 그리고 출력층(Output Layer)으로 이루어지는 신경망은 뉴런(Neuron)을 모형화한 처리 요소들을 기본 구성 단위로 하는데, 처리 요소는 다른 처리 요소들로부터 여러 입력값을 받아들여 단 하나의 출력값만을 생성하여 이를 연결된 또 다른 층의 처리 요소들에게 전달한다. 처리 요소 i, j 간의 연결강도는 연결 가중치를 사용하여 나타내고 이를 W_{ij} 로 표기한다.

각 처리요소들은 전달받은 입력값들과 연결가중치를 사용하여 순 입력값을 계산한 후, 이를 전이함수(Transfer Function) 혹은 활성함수(Activation Function)에 의해 변환하여 출력값을 결정한다. 신경망에서 사용되는 전이함수에는 계단함수(Hard Limiter), 임계논리(Threshold Logic), 그리고 시그모이드(Sigmoid) 함수 등이 있는데 시그모이드 함수가 가장 많이 사용된다.

입력층, 은닉층, 그리고 출력층의 3층 구조로 이루어진 신경망 모형의 표기 및 계산절차는 다음과 같다.

<Notation>

m : 입력 및 출력패턴 개수
 p : p 번째 입력패턴

- i : 입력층
- j : 은닉층
- k : 출력층
- O_{pi} : p 번째 입력패턴에서의 입력층 출력값
- net_{pj} : p 번째 입력패턴에서의 은닉층 입력값
- W_{ji} : 은닉층과 입력층 간 연결가중치
- θ_j : 은닉층 바이어스(Bias)
- O_{pj} : p 번째 입력패턴에서의 은닉층 출력값
- net_{pk} : p 번째 입력패턴에서의 출력층 입력값
- W_{kj} : 출력층과 은닉층 간 연결가중치
- θ_k : 출력층 바이어스
- O_{pk} : p 번째 입력패턴에서의 출력층 출력값
- t_{pk} : p 번째 입력패턴에서의 출력층 목표 출력값
- δ_{pk} : p 번째 입력패턴에서의 출력층 델타(Delta)
- δ_{pj} : p 번째 입력패턴에서의 은닉층의 델타
- η_w : 연결가중치 학습률
- η_θ : 바이어스 학습률
- α_w : 연결가중치 모멘텀 상수(Momentum Constant)
- α_θ : 바이어스 모멘텀 상수

<Error Backpropagation Algorithm>

[단계 1] 입력층과 은닉층, 은닉층과 출력층 간 연결가중치를 초기화 한다.

[단계 2] 입력패턴에 대한 전방향 전파를 계산한다.

- 입력층 : 입력층에 입력한 값을 그대로 출력한다.
- 은닉층 : 식 (1)과 같이 입력층 출력값과 입력층과 은닉층 간 연결가중치 곱에 바이어스를 더해 은닉층 입력값 net_{pj} 를 계산한다.

$$net_{pj} = \sum_i W_{ji} O_{pi} + \theta_j \quad (1)$$

은닉층 입력값을 활성화함수 $f(net_{pj})$ 에 대입하여 출력값을 구한다. 시그모이드 함수를 활성화함수로 적용했을 경우 은닉층 출력값 O_{pj} 는 식 (2)와 같다.

$$O_{pj} = f(net_{pj}) = \frac{1}{1 + \exp(-net_{pj})} \quad (2)$$

- 출력층 : 은닉층 출력값과 은닉층과 출력층 간 연

결가중치 곱에 바이어스를 더해 출력층 입력값 net_{pk} 를 계산한다.

- 출력층 입력값을 활성화함수에 대입하여 출력층 출력값 O_{pk} 를 계산한다.
- 목표 출력값 t_{pk} 와 출력값 O_{pk} 를 비교하여 식 (3)과 같이 오차 E_p 를 계산하고, 식 (4)와 같이 출력층 전체 오차 E 를 계산한다.

$$E_p = \frac{1}{2} \sum_k (t_{pk} - O_{pk})^2 \quad (3)$$

$$E = \sum_{p=1}^m E_p \quad (4)$$

[단계 3] 오차를 역전파 한다.

- 출력층 : [단계 2]에서 구한 오차를 활성화함수의 미분식에 취해 식 (5)와 같이 δ_{pk} 를 구하고, 이 값을 바탕으로 식 (6), 식 (7)과 같이 연결가중치 및 바이어스 변화량을 계산한다.

$$\delta_{pk} = (t_{pk} - O_{pk}) f'(net_{pk}) = (t_{pk} - O_{pk}) O_{pk} (1 - O_{pk}) \quad (5)$$

$$\Delta W_{kj} = \eta_w \delta_{pk} O_{pj} + \alpha_w \{ W_{kj}(now) - W_{kj}(old) \} \quad (6)$$

$$\Delta \theta_k = \eta_\theta \delta_{pk} + \alpha_\theta \{ \theta_k(now) - \theta_k(old) \} \quad (7)$$

식 (8), 식 (9)와 같이 연결가중치 및 바이어스 변화량을 현재값에 더하여 출력층 연결가중치 및 바이어스를 최신화한다.

$$W_{kj}(new) = W_{kj}(now) + \Delta W_{kj} \quad (8)$$

$$\theta_k(new) = \theta_k(now) + \Delta \theta_k \quad (9)$$

- 은닉층 : 출력층에서 구한 δ_{pk} 를 이용해 식 (10)과 같이 은닉층의 δ_{pj} 를 구하고, 식 (11), 식 (12)와 같이 연결가중치 및 바이어스 변화량을 계산한다.

$$\delta_{pj} = f'(net_{pj}) \sum_k \delta_{pk} W_{kj} = O_{pj} (1 - O_{pj}) \sum_k \delta_{pk} W_{kj} \quad (10)$$

$$\Delta W_{ji} = \eta \delta_{pj} O_{pi} + \alpha_w \{ W_{ji}^r(now) - W_{ji}^r(old) \} \quad (11)$$

$$\Delta \theta_j = \eta \delta_{pj} + \alpha_\theta \{ \theta_j(now) - \theta_j(old) \} \quad (12)$$

식 (13), 식 (14)와 같이 연결가중치 및 바이어스 변화량을 현재값에 더하여 은닉층 연결가중치 및 바이어스를 최신화한다.

$$W_{ji}(new) = W_{ji}(now) + \Delta W_{ji} \quad (13)$$

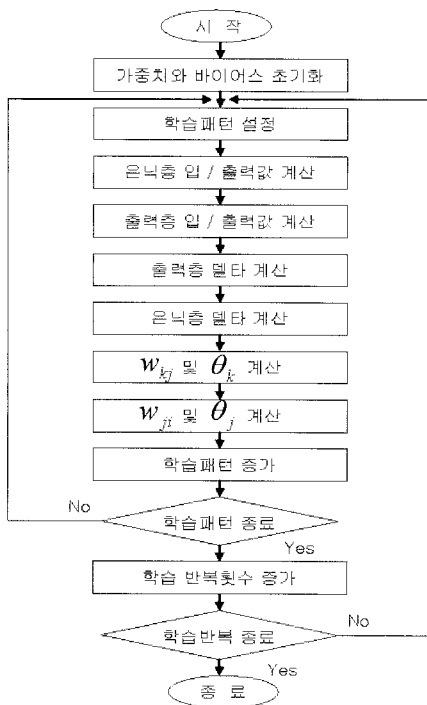
$$\theta_j(new) = \theta_j(now) + \Delta \theta_j \quad (14)$$

[단계 4] 다음 패턴을 학습시키고, 모든 입력패턴에 대해 학습할 때까지 [단계 2]로 돌아간다.

[단계 5] 학습 반복 횟수를 세고, 반복 횟수가 제한 횟수보다 작으면 [단계 2]로 돌아간다.

[단계 6] 종료

위 절차를 도식화하면 그림 1과 같다.

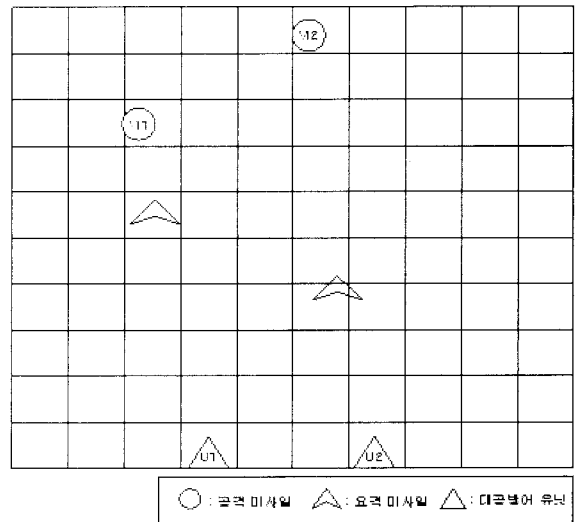


[그림 1] 오류 역전파 알고리즘 절차도

3. 문제정의 및 모형구축

가. 문제정의

10×10 2차원 평면으로 표현한 대공방어 영역에서 두 개의 접근중인 공격 미사일에 대한 방어 유닛은 오직 하나의 요격 미사일(Interceptor)만을 가지고 있으며 두 미사일중 어느 것을 요격해야 할지를 결정해야 한다. 미사일이 지상에 도달했을 때 장착한 탄두 크기에 비례하여 피해가 발생하고 전체 피해를 최소화할 목적으로 한다. 이 때, 유닛1은 동일 미사일에 대한 중복 담당을 피하기 위해 유닛2의 행위 패턴에 대한 과거 관측치 학습을 통해 유닛2가 어느 공격 미사일을 요격할 것인지 예측하고자 한다. 그림 2는 이러한 대공방어 영역을 나타낸다.



[그림 2] 대공방어 영역

나. 모형구축

본 연구의 모형에서는 대공방어 영역 하 유닛1의 유닛2의 행위에 대한 모형을 표현하는 영향도를 작성하고, 종속적인 인과관계를 가지는 결정 노드, 기회 노드, 그리고 가치 노드에서의 학습 문제를 다음으로써 이를 구체화한다.

1) 영향도

그림 3은 대공방어 영역에서 유닛1이 모형화해야할

유닛2의 의사결정모형을 나타내는 영향도이다. 영향도에서 유닛2의 행위는 결정 노드로 표현되며 이는 미사일1과 미사일2 중에서 하나를 선택할 수 있는 유닛2의 능력을 나타낸다. 유닛2가 미사일1을 요격하기로 결정했다면 이는 유닛2의 미사일1 요격이라는 기회 노드에 영향을 준다. 이 기회 노드는 미사일1 요격시의 비용에 대한 기회 노드 및 미사일1의 유닛2에 의한 요격성이라는 기회 노드에 영향을 준다. 동일한 방식으로 각 기회 노드 간 종속관계가 설정되어, 미사일1과 미사일2에 의한 각각의 피해 노드가 전체 피해 노드에 영향을 주고, 이는 최종적으로 효용노드에 영향을 준다.

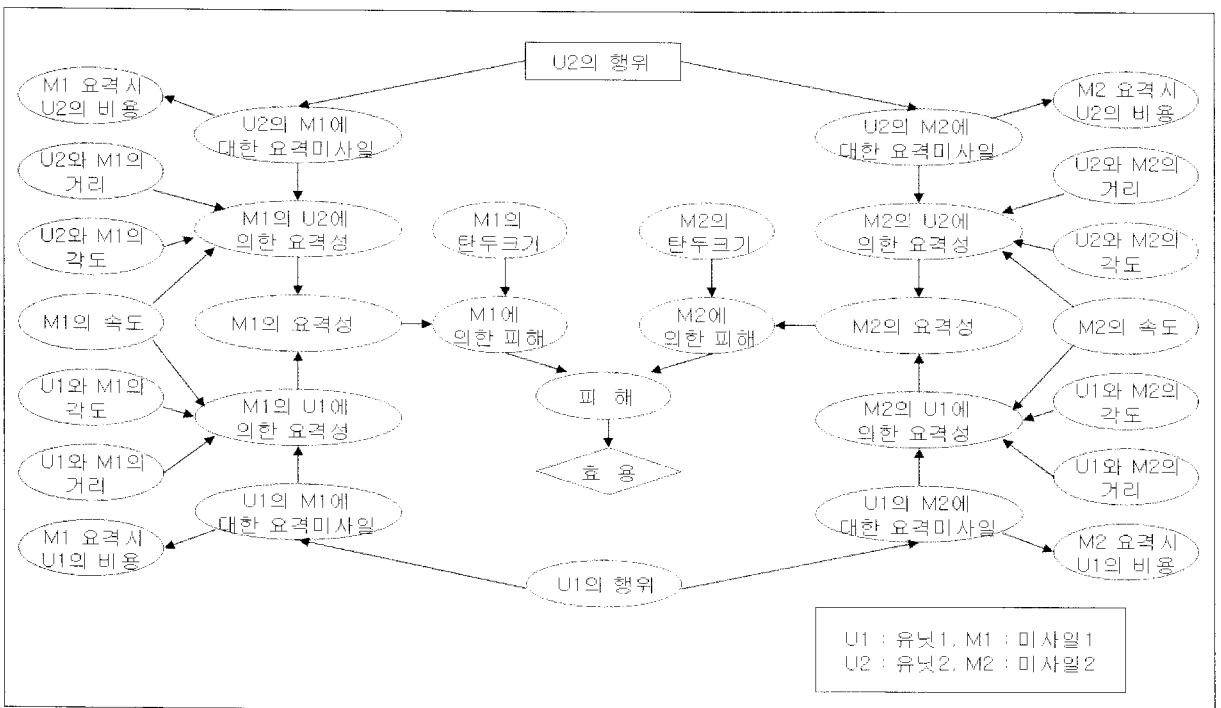
유닛2가 어떤 미사일을 담당할 것인지에 영향을 주는 요소가 해당 미사일 요격에 따른 비용과 해당 미사일에 의한 피해라고 가정하면 유닛2는 비용 최소화 와 피해 최소화에 대한 선호도에 의해 자신의 행위를 결정하게 된다. 즉 유닛의 선호도를 나타내는 가치 노드에서의 출력값이 유닛2의 비용과 피해에 대한 선호가중치에 따른 미사일1과 미사일2 중에서의 선택을

나타내는 값이라면 유닛1은 이러한 행위로 예측된 유닛2의 선택에 기반하여 자신의 행위를 결정하게 된다.

2) 결정 노드

에이전트의 능력은 시스템과 상호작용을 수행할 수 있는 여러 가지 대안적인 행위의 과정이다. 영향도에서 에이전트의 능력은 결정 노드에 할당되는 값으로 제시된다. 이러한 값의 할당은 가능한 모든 경우에 대해 검토할 필요가 있는데 이는 멀티 에이전트 시스템에서의 다른 에이전트에 대한 모형이 그 에이전트가 할 수 있는 모든 가능한 행위들을 고려해야함을 의미한다.

예를 들어, 에이전트를 대공방어 영역에서의 방어 유닛1로 간주하고, 유닛 1은 최초로 유닛 2가 장거리 요격미사일을 미보유한 것으로 가정한다. 이 때 유닛2의 능력은 유닛2에 대한 유닛1의 모형에서 결정 노드 값의 집합 {S_M1, S_M2, Other}으로 모형화 된다. 여기서 S_M1, S_M2는 각각 공격 미사일 M1, M2에 대한 단거리 요격 미사일의 발사 행위를 나타



[그림 3] 대공방어 유닛2의 의사결정 모형

낸다. 어떤 경우에 유닛1은 유닛2가 장거리 요격 미사일을 발사하는 것을 관측할 수 있는데, 이 때 유닛1은 자신의 유닛2에 대한 모형을 최초로 판단했던 것과 달리 변경된 결정 노드 값의 집합 $\{S_M1, S_M2, L_M1, L_M2, Other\}$ 으로 수정해야 한다. 여기서 L_M1, L_M2 는 각각 공격 미사일 $M1, M2$ 에 대한 장거리 요격 미사일의 발사 행위를 나타낸다.

이와 반대로 어떤 경우에 다른 에이전트의 능력은 제거되거나 다른 값의 일부로 흡수될 수 있는데, 이런 경우는 유닛2가 무력화되어 더 이상 어떤 행위를 수행할 수 없을 때 발생한다. 유닛1의 관점에서 유닛2가 수행할 수 있고 또 반드시 취해야할 행동이 누락되었을 때 유닛1은 유닛2의 과거 행위패턴으로부터 특정 행위의 누락을 인지할 수 있다. 이러한 상황은 유닛2의 능력을 모형화하는 과정에서의 부정확성에 의하여 야기될 수 있다. 이러한 경우에는 모형 내에서 다음 순위의 바람직한 행위를 식별하여 그 것이 과거 행위패턴과 일관성을 보인다면 모형이 여전히 유효하다고 추론할 수 있다. 결론적으로 유닛2에 대한 유닛1의 모형은 결정 노드에서 누락된 행위의 값을 제거함으로써 수정된다.

3) 가치 노드 및 실험

가치 노드는 노드의 효용(Utility)을 나타내며 이는 곧 에이전트의 선호도를 의미한다. 멀티 에이전트 시스템에서 다른 에이전트에 대한 모형이 그 것의 행위패턴을 제대로 예측하지 못하는 가능한 이유 중의 하나는 에이전트의 선호도가 모형에 의해 정확히 표현되지 않았기 때문이다. 이 때 효용함수를 최신화하여 모형을 수정하며 이를 통해 에이전트의 과거 행위패턴의 모든 결과가 항상 해당 상태에서의 효용을 최대화시킬 수 있다.

상태 S 에서의 효용함수를 $U(S)$ 라 하고 식 (15)와 같이 정의한다.

$$U(S) = g(F_1, \dots, F_M) \quad (15)$$

여기서 $F = \{F_1, \dots, F_M\}$ 는 에이전트의 선호도에 직접적으로 영향을 미치는 특성들의 집합으로 영향도에서 가치 노드에 대한 부모 노드의 집합이다. 효용함

수 g 는 다중 속성 효용 이론^[5]에서 특성 $F_m (m = 1, \dots, M)$ 값 요소들의 가중화된 합이다. 단순성을 위해 가중화된 요소가 특성 F_m 에 선형으로 종속된다고 가정하면 에이전트의 효용 함수 g 를 식 (16)과 같이 표현할 수 있다.

$$U(S) = w_1 f_1 + \dots + w_M f_M \quad (16)$$

여기서, 각 가중치 w_m 은 특성 F_m 에 대응하고, 에이전트의 효용에 대한 특성의 영향력의 크기를 나타낸다. 또한 f_m 은 상태 S 에서 특성 F_m 의 값을 의미한다. 에이전트의 선호도를 학습하는 방법으로 가중치 w_m 을 수정한다.

B^* 를 기대 효용을 최대화하는 행위라고 하면 식 (17)과 같이 표현할 수 있다.

$$B^* = \operatorname{argmax}_{b_n} \sum_{l=1}^L P(S_l | b_n, C) \times U(S_l) \quad (17)$$

여기서, $B = \{b_1, \dots, b_N\}$ 는 에이전트의 행위의 집합이고, S_l 은 비확정적인 행위 b_n 이 주어졌을 때 가능한 결과 상태이며 l 은 L 개의 결과 범위를 가진다. 조건부 확률에서 C 는 과거 행위패턴에서 제공되는 시스템 내에서의 알려진 데이터를 의미한다.

식 (16), 식 (17)에 의해 B^* 는 식 (18)과 같이 표현된다.

$$B^* = \operatorname{argmax}_{b_n} \sum_{l=1}^L P(S_l | b_n, C) \sum_{m=1}^M w_m f_m^l \quad (18)$$

여기서, f_m^l 은 상태 S_l 에서의 특성 F_m 의 값이고, 이 식은 식 (19)와 같이 변형될 수 있다.

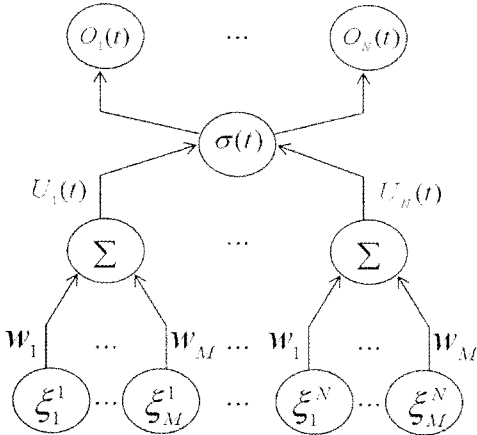
$$B^* = \operatorname{argmax}_{b_n} \sum_{m=1}^M w_m \zeta_m^n \quad (19)$$

여기서, ζ_m^n 은 b_n 과 C 가 주어졌을 때 특성 F_m 의 기댓값으로 $F_m: (f_{m,1}, \dots, f_{m,r_m})$ 에 그들의 조건부 확률을 곱한 값의 합으로 식 (20)과 같이 주어진다.

$$\zeta_m^n = \sum_{q=1}^{F_m} f_{m,q} \times P(F_m = f_{m,q} | b_n, C) \quad (20)$$

다른 에이전트의 과거 행위패턴으로부터 주어진 시간 구간 동안에 에이전트의 관측된 행위의 집합을 얻을 수 있는데 그러한 집합을 $A = \{A(1), \dots, A(T)\}$ 로 정의한다. 여기서 T 는 행위의 가능한 총 수이고, $A(t) \in B, t = 1, \dots, T$ 이다. 효용함수는 초기 모형에서 초기 효용함수를 유도하고, 집합 A 에 기반하여 가중치 w_m 을 계속적으로 조정함으로써 수정된다. 이러한 과정을 거쳐 $A(t)$ 는 모든 상황 t 에서 B^* 와 같아진다.

가중치 조정은 2장에서 설명한 오류 역전파 알고리즘의 일반적 절차에 따른다. 식 (19)를 기반으로한 가중치 학습을 위한 신경망을 그림 4에 제시하였고 이는 의사결정상황 t 에서의 의사결정과정을 나타낸다.



[그림 4] 가중치 학습을 위한 신경망

그림 4에서 $\sigma(t)$ 는 의사결정 결과를 구성하는 활성화함수이고^[2], 여기서 출력값 $O_n(t)$ 는 $n = 1, \dots, N$ 일때 식 (21)과 같다.

$$O_n(t) = \begin{cases} 1 & U_n(t) > U_l(t), \forall l \neq n \\ 0 & otherwise \end{cases} \quad (21)$$

$t_n(t)$ 는 $A(t)$ 로부터의 목표 출력값으로 식 (22)와 같다.

$$t_n(t) = \begin{cases} 1 & A(t) = b_n \\ 0 & otherwise \end{cases} \quad (22)$$

가중치 조정에는 잘 알려진 점진적 하강 기법인 델타 규칙^[3]이 적용되는데 에러 신호 $e_n(t)$ 에 기반한 비용 함수를 최소화하여 각 가중치 조정을 통해 출력값을 목표 출력값에 근접하게 한다. 일반적으로 사용되는 비용함수는 최소자승오차(mean-square-error) 기준에 의한 것으로 $e_n(t) = t_n(t) - O_n(t)$ 로 정의될 때 2장의 식 (3)과 같다.

또한 가중치 정규화를 위해 ζ 의 정규화가 네트워크 적용 전에 필요한데 이는 식 (23)과 같다.

$$\xi_m^n = \frac{\zeta_m^n}{\sqrt{\sum_{n=1}^N \sum_{m=1}^M (\zeta_m^n)^2}} \quad (23)$$

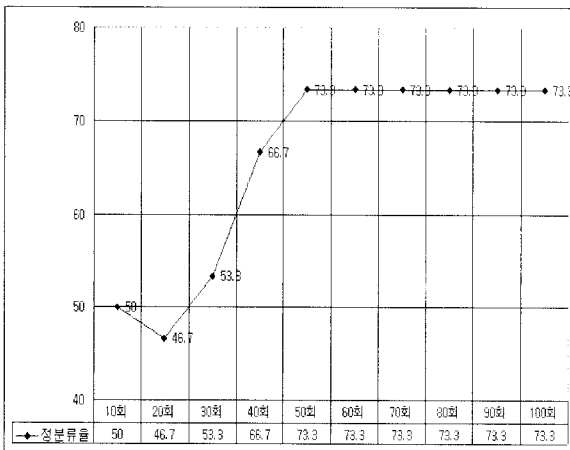
상황 t 에서의 최종 가중치 변화량은 $m = 1, \dots, M$ 이고 η 가 학습률을 나타낼때 식 (24)와 같다.

$$\Delta w_m(t) = \eta \sum_{n=1}^N e_n(t) \xi_m^n(t) \quad (24)$$

VC++를 활용한 실험에서는 공격 미사일의 위치와 탄두 크기를 임의 생성하여 총 100개의 의사결정 상황을 생성하였다. 식 (20)의 $f_{m,q}$ 값은 피해 (0, 50, 100), 비용 (0, 500, 1000)으로 각각 설정하였고, $P(F_m = f_{m,q} | b_n, C)$ 값은 피해 (0.1, 0.4, 0.5), 비용 (0.0, 0.25, 0.75)로 각각 적용하였다. 산출된 ζ_m^n 값은 식 (23)과 같이 정규화하여 입력패턴으로 활용하였고, ζ^1 이 ζ^2 보다 크면 유닛2가 미사일1을 요격하는 것으로 하여 값을 1, 반대의 경우 값을 0으로 하여 목표 출력값으로 활용하였다. 유닛2의 행위는 피해와 비용 최소화를 같은 비율로 선호하도록 설계하였다. 또한 유닛2와 미사일 간의 거리가 멀수록 피해는 적게 유도하고, 이 때 사정거리가 긴 요격 미사일이 요구되므로 비용은 크게 설정하였다. 미사일의 탄두 크기가 클수록 피해는 크게 적용하고, 이 때 더 큰 피해를 야기하는 큰 탄두를 가진 미사일에 대한 요격 필요성이 커짐에 따라 보다 우수한 성능의 요격 미사일을

사용해야 하므로 비용도 크게 설정하였다. 연결가중치 학습률 η_w 는 0.9, 바이어스 학습률 η_b 는 0.85로 각각 설정하였으며 연결가중치 조절에 관성을 주어 학습 성능 및 속도 향상을 위해 모멘텀 상수 α_w, α_b 를 각각 0.045, 0.04로 적용하였다.

입력 및 출력 패턴은 모형 구축에 가장 일반적인 훈련용 70%, 검증용 30%를 적용하였다. 총 100회의 반복 실험결과 출력값의 목표 출력값에 대한 정분류율이 그림 5와 같이 나타났다. 50회 반복시 정분류율이 73.3%로 높아졌고, 이후 수렴하였다.



[그림 5] 정분류율

4) 기회 노드

영향도에서 기회 노드는 노드 간에 존재하는 확률적 관계에 의해 표현된다. 결정 노드와 가치 노드가 이미 명백해진 경우에, 다른 에이전트에 대한 모형이 여전히 그 행위 결과를 예측할 수 없다면 이는 모형의 기회 노드가 정확한 종속관계에 따른 확률을 할당 받지 못한 결과이다.

모형의 부정확성을 유발하는 특성노드의 집합을 ψ 라 정의하면 이러한 부정확성의 야기 원인을 조사하여 모형을 수정하는 방법에는 다음과 같은 것들이 내포될 수 있다.

첫 번째, 부정확성의 원인이 ψ 에 속하는 특성들의 부정확한 국소 확률 분포에 기인한다고 가정하는 것이다. 이러한 가정에 기반하여 ψ 에 속하는 특성 노드들의 조건부 확률을 최신화하여 모형을 수정한다.

두 번째, 원인이 ψ 에 속하는 특성들을 포함하는 노드 간의 부정확한 종속관계에 기인한다고 가정하는 것이다. 특정 부모 노드로부터의 연결을 제거하거나 다른 노드로부터의 연결을 추가하여 모형을 수정한다.

세 번째, 문제가 ψ 에 속하는 특성 때문이 아닌 그들의 부모노드에 의해 야기된다고 가정하는 것이다. 부정확한 확률 분포 혹은 종속 관계를 가지는 특정 부모노드를 수정하여 모형을 최신화한다.

4. 맺음말

본 연구에서는 멀티 에이전트 시스템에서 하나의 에이전트가 다른 에이전트의 행위패턴을 학습하는 모형구축에 대한 방법론을 다루었다. 모형 구축에는 영향도를 활용하고, 영향도의 결정 노드, 가치 노드, 그리고 기회 노드에서의 학습 방법을 제시하였으며 특히 신경망 이론을 적용하여 가치 노드에서의 학습을 구체화하였다. 이의 활용 분야로 대공방어 유닛을 예로 들었으며, 유닛 간 공격 미사일에 대한 중복 담당을 회피할 수 있는 방법론을 제시하였다.

본 연구에서는 신경망 모형의 제어요소를 임의로 고정하여 실험하였다. 향후에 신경망의 학습률, 모멘텀 상수 등의 제어요소를 변경하면서 최적의 조건을 찾아내는 연구가 수행되어야 한다. 또한 3개 이상 다수 방어 유닛 간 의사결정에 대한 연구도 필요하다.

참 고 문 헌

- [1] Carmel, D. and Markovitch, S., "Learning Models of Intelligent Agents", AAAI/IAAI 1, pp. 62~67, 1996.
- [2] Haykin, S., "Neural Network : A Comprehensive Foundation", New York : Macmillan College Publishing Company, 1994.
- [3] Hebb, D. O., "The Organization of Behavior : A Neuropsychological Theory", New York : Wiley, 1949.

- [4] Kaminka, G., Tambe, M. and Hopper, C., "The role of Agent-modeling in Agent Robustness", In Proceedings of the Conference on AI Meets the Real World, 1998.
- [5] Keeney, R. L. and Raiffa, H., "Decisions with Multiple Objectives : Preferences and Value Tradeoffs", New York : Wiley, 1976.
- [6] McCulloch, W. S. and Pitts, W., "A Logical Calculus of the Ideas Immanent in Nervous Activity", Bulletin of Mathematical Biophysics, Vol. 5, pp. 115~133, 1943.
- [7] Rosenblatt, F., "The Perceptron : A Probabilistic Model for Information Storage and Organization in the Brain", Psychological Review, Vol. 65, pp. 386~408, 1958.
- [8] Rumelhart, D. E., Hinton, G. E. and Williams, R. J., "Learning Internal Representation by Error Propagation : in Parallel Distributed Processing", The MIT Press, Vol. 1, 1986.
- [9] Widrow, B. and Hoff Jr., M. E., "Adaptive Switching Circuits", IRE WESCON Convention Record, pp. 96~104, 1960.
- [10] X. Wu and K. I. Poh, "Decision-model Construction with Multilevel Influence Diagrams", The Knowledge Engineering Review, Vol. 15, No. 3, pp. 233~256, 2000.
- [11] 류경현, 정환목, "퍼지 의사결정에 기반한 멀티 에이전트의 효율적인 조정방안", 퍼지 및 지능시스템학회 논문지 Vol. 17, No. 1, pp. 66~71, 2007.