

이차원 대전 액션 게임을 위한 신경망 기반의 지능 캐릭터 구현

나종민*, 오하령**, 성영락**, 정성훈***, 조병헌*

*국민대학교 전자공학과

**국민대학교 전자전파통신공학부

***한성대학교 정보공학부

e-mail: rorn77@empal.com

An Implementation of Neural Networks Based Intelligent Characters for Two Dimensional Fighting Action Games

Jong-Min Na*, Ha-Ryoung Oh**, Yeong-Rak Seong**,

Sung-Hoon Jung***, Byeong-Heon, Cho*

*Dept of Electronics Engineering, Kook-Min University

**Dept of Electronic and Telecommunication Engineering, Kook-Min University

***Dept of Information Engineering, Han-Sung University

요 약

현재 인공지능 기법은 우리 주위의 여러 분야에서 사용되고 있으며, 그 중요도가 점점 높아지고 있다. 이러한 분야 중에 하나가 바로 컴퓨터 게임 분야로 전통적인 인공지능 기법부터 근래에 게임과의 결합이 시도되고 있는 비전통적 기법에 이르기까지 다양한 종류의 방법이 시도되었고, 시도될 예정이다. 본 논문에서는 게임 내에서 그 비중이 커져가고 있는 인공지능을 통한 캐릭터의 지능화 방안으로 신경망 기반의 지능 캐릭터를 제안하였다.

1. 서론

인공지능을 구현하는 방법에는 오래전부터 사용되어왔던 FSM(Finite State Machine)과 같은 전통적인 기법과 최근에 들어 사용되고 있는 A-Life와 같은 기법 등 여러 기법들이 존재한다. 그러나 아직 까지도 적은 투자로 높은 효율을 보인다는 이유로 가장 널리 쓰이는 전통적인 인공지능 기법으로는 게임이 쉽게 단조로워 지고, 변화된 환경에 대한 적응이 불가능하다는 단점이 존재한다.

위의 문제점을 해결하기 위한 방안으로 신경망 기반의 지능형 캐릭터에 대한 연구가 이미 발표되었으나 본 논문에서는 이를 더 확장하여 이차원 대전 액션 게임을 위한 신경망 기반의 지능 캐릭터를 제안하고 간단한 게임을 모델링하여 제안된 알고리즘을 확인해 본다. [1-3]

2. 신경망 지능 캐릭터의 구현

신경망을 통하여 지능형 캐릭터를 구현하기 위해서 우선 게임의 규칙에 대하여 다음과 같은 가정을

하였다. 첫째, 캐릭터의 각 행동은 거의 대부분 단일 스텝에 이루어 지지 않고 여러 스텝에 걸쳐 이루어 진다. 둘째, 캐릭터의 다음 행동의 시작은 현재의 행동이 완료되거나, 완료되지 않았을 때 상대방의 공격으로 인해 취소가 되는 경우에만 가능하며, 현재의 행동을 자의로 취소할 수 없다. 셋째, 일반적인 전진 또는 후진이 아닌 횡이동을 할 때에는 캐릭터 간의 거리에는 변화가 없으며, 상대 캐릭터가 바라보는 방향이 변경되게 된다. 단 변화되는 방향의 각도는 캐릭터간의 거리에 따라 차이가 있다. 넷째, 상대 캐릭터가 횡이동을 하여 상대 캐릭터가 있는 곳과 현재 바라보는 방향의 차이는 현재의 동작이 끝난 후 몇몇 행동을 제외한 다음 행동이 시작될 때 보정된다.

2.1 신경망의 구성

신경망은 처음에 입력을 받아들이는 최하위의 뉴런들인 입력계층과, 신경망에 의해 처리된 데이터를 출력하는 뉴런들의 층인 출력계층, 그리고 그 사이

에서 값을 전달하는 은닉계층으로 구성된다. 여기에서는 하나의 은닉계층을 가지고 있는 다층 퍼셉트론 구조로 신경망을 구성하였다.

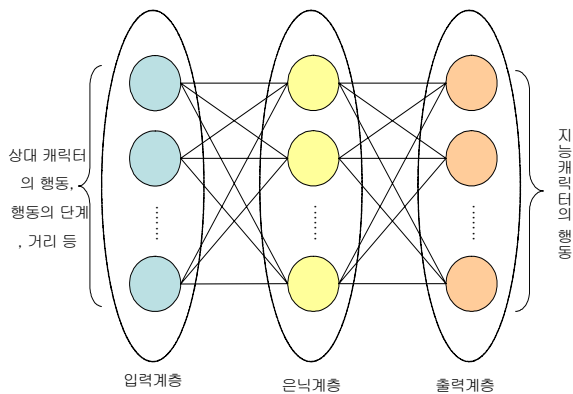


그림 1 - 지능형 캐릭터를 위한 신경망의 구성

지능형 캐릭터가 자신의 행동을 결정하기 위해서는 당연히 상대 캐릭터의 행동이 무엇인지를 확인해야 할 것이다. 이와 같은 값들이 신경망의 입력 계층에 입력되게 되면 은닉계층을 거쳐서 출력계층의 각 노드들에 0부터 1사이의 실수 값들이 나오게 되고 이 출력 값들 중에 가장 큰 수를 가지는 노드를 자신의 행동으로 선택한다. 입력 계층에 입력되는 값들에 따라 은닉 계층의 노드와 출력 계층의 노드의 값들을 계산하는 식은 일반적으로 전방향 신경망에서 사용되는 식과 같다.[4]

2.2 오류역전과 알고리즘을 이용한 강화학습

역전과 알고리즘(Backpropagation algorithm)은 다층 퍼셉트론 구조이고, 전방향 신경망에서 사용되는 학습 알고리즘이다. 역전과 알고리즘을 적용하기 위해서는 입력 데이터와 원하는 출력(o) 데이터가 있어야 한다. 간단히 개념을 살펴보면, 먼저, 입력이 신경망의 가중치와 곱하고 더하는 과정을 몇 번 반복하면 입력의 결과값인 출력(y)이 나온다. 이 때 출력(y)은 데이터에서 주어진 원하는 출력(o)과 다르다. 결국, 신경망에서는 (y - o)만큼의 오차(e = y - o)가 발생하며, 오차에 비례하여 출력계층과 은닉계층 사이의 가중치를 갱신하고, 그 다음 은닉계층과 입력계층 사이의 가중치를 갱신한다. 가중치를 갱신하는 방향이 신경망의 처리 방향과는 반대 방향이다. 이런 이유로 ‘역’전과 알고리즘이라고 한다.

교사 학습의 경우에는 위에서 원하는 출력(o)으로 표현 했던 것처럼 최적의 해를 알고 있는 경우에

사용되나 이 논문에서처럼 어떠한 행동이 최적의 해인지 알 수 없는 경우에는 사용이 불가능하나 현재 행동에 대한 적절성을 판단할 수 있는 경우에는 그것을 이용한 강화 학습 방법을 사용할 수 있다.

강화학습(Reinforcement Learning)이란 이름 그대로 자신이 수행한 행동에 대하여 보상값을 받아 조금씩 좋은 방향으로 행동을 강화시키는 학습방법이다. 현재 상태에서 최적의 행동을 계산을 통해 결정하지 않고, 여러 번의 시행착오에 기반을 두어 경험에 의해 각 상태에서의 최적의 행동을 조금씩 학습해 나간다.

본 논문에서는 두 캐릭터가 대전을 하며 발생하는 점수의 차를 이용하여 신경망을 학습시킨다. 지능 캐릭터의 구현은 두 단계를 통해서 이루어진다. 첫 번째 단계는 게임의 규칙에 대한 기본 정보가 전혀 없는 상태에서 규칙을 학습하기 위한 단계이며, 두 번째 단계는 첫 학습 단계에서 학습하게 된 내용을 바탕으로 실제 대전을 하는 단계이다.

2.2.1 규칙 학습 단계

초반의 신경망은 지능 캐릭터가 게임의 규칙에 대한 아무런 지식이 없는 상태에서 대전을 통해서 게임의 규칙을 습득하는 단계이다. 초기 신경망의 계층 간의 가중치는 일정한 범위 안에서 무작위로 부여된다. 규칙 학습 단계에서는 여러 가지 경우에 대한 학습을 하는 것이 지능형 캐릭터의 대전 능력 향상에 도움이 되므로 지능형 캐릭터와 상대 캐릭터의 행동 역시 무작위로 부여된다.

이렇게 입력된 상황(상대방 캐릭터의 행동, 행동의 단계 등의 입력 노드에 들어가는 데이터들)과 그때 지능 캐릭터가 선택한 행동 그리고 그 행동으로 인해서 점수를 잃거나 얻는 결과가 나오게 되면 이를 이용해서 역전과 알고리즘으로 학습을 수행한다. 먼저 출력 계층과 은닉 계층의 가중치는 다음의 식에 의해 수정된다.

$$\delta_j = z_j \times (1 - z_j) \times (d_j - z_j) \quad <1>$$

$$v_{ij} = v_{ij} + (a(t) \times \delta_j \times h_i) \quad <2>$$

d_j 는 j번째 출력노드에서의 원하는 출력값을 의미하며, z_j 는 j번째 출력 노드의 값, v_{ij} 는 i번째 은닉 노드와 j번째 출력 노드간의 링크 가중치, $a(t)$ 는 시간에 대한 학습률 함수를 나타낸다. 학습률 함수는 학습률 $\times e^{-t/\Delta t}$ 를 사용한다. Δt 는 $t_{end} - t_{start}$ 이고 t_{end} 는 학습 종료 시간, t_{start} 는 학습 시작 시간을 나타낸다. 이런 과정으로 출력 계층과 은닉 계층의 가중치 수

정이 끝나면 다음의 식을 통하여 은닉 계층과 입력 계층간의 가중치를 수정한다.

$$\delta_j = h_j \times (1 - h_j) \times \left(\sum_{k=0}^{N_o} (w_{jk} \times \delta_k) \right) \quad <3>$$

$$w_{ij} = w_{ij} + (a(t) \times \delta_j \times x_i) \quad <4>$$

여기서 x_i 는 i 번째 입력 노드의 값, h_j 는 j 번째 은닉 노드의 값, N_o 는 출력 노드의 개수, w_{jk} 는 j 번째 은닉노드와 k 번째 출력노드간의 링크 가중치, w_{ij} 는 i 번째 입력 노드와 j 번째 은닉 노드간의 링크 가중치, $a(t)$ 는 학습률 함수를 나타낸다.

2.2.2 실전 단계

실전 단계에서는 입력 노드에 입력되는 데이터들을 바탕으로 은닉 계층을 통해 출력 계층의 각 노드들에 대한 값들이 정해지면 가장 높은 값이 발생한 노드의 행동을 선택하여 대전을 진행하게 된다.

3. 실험 및 결과

본 논문에서 제안하는 신경망 지능 캐릭터의 가능성을 확인하기 위해 이차원 대전 액션 게임을 간단하게 모델링 해 보았다. 캐릭터가 선택할 수 있는 행동의 종류는 모두 23개며 이 행동의 종류와 세부 사항에 대해서 <표 1>과 같이 정의하였다.

행 동	틱의 수	거리 변화가 발생하는 틱	공격 판정이 있는 틱	방향 보정
정지	2	×	×	×
전진	2	2	×	0
후진	2	2	×	0
앞기	2	×	×	×
앞으로 점프	3	2	×	0
제자리 점프	3	×	×	×
뒤로 점프	3	2	×	0
왼쪽 횡이동	2	×	×	0
오른쪽 횡이동	2	×	×	0
앞으로 점프중 손공격	5	3	4	0
제자리 점프중 손공격	5	3	4	0
뒤로 점프중 손 공격	5	3	4	0
위 손 공격	4	×	4	0
중간 손 공격	4	×	4	0
앞아서 손 공격	4	×	4	0
앞으로 점프중 발 공격	6	3	5	0
제자리 점프중 발 공격	6	×	5	0
뒤로 점프중 발 공격	6	3	5	0
위 발 공격	5	×	5	0
중간 발 공격	5	×	5	0
앞아서 발 공격	5	×	5	0
서서 막기	2	×	×	0
앞아서 막기	2	×	×	0

표 1 - 실험을 위해 간략화한 게임의 세부 규칙

3.1 이동

캐릭터의 이동은 크게 두 가지로, 첫 번째는 일차원적인 이동을 하는 행동으로 전진, 후진, 앞으로 점프, 뒤로 점프의 4가지가 존재한다.

두 번째는 이차원적인 이동을 하는 행동으로 왼쪽 횡이동, 오른쪽 횡이동의 두 가지가 존재한다. 이 때 양 캐릭터간의 거리에는 변화가 없으며 이동한 캐릭터를 A, 이동하지 않은 캐릭터를 B라고 할 때, B가 바라보고 있는 방향과 실제 A가 위치하는 곳이 차이가 나게 된다. 거리가 1인 상태에서 횡이동은 90° 방향의 차이를 발생시키고 그 이상의 거리에서는 거리에 비례하여 더 작은 각도의 차이를 발생시킨다. 이처럼 이차원 이동으로 발생하는 실제 캐릭터의 위치와 캐릭터가 바라보는 방향의 차이는 특정 행동을 제외한 행동들의 첫 번째 틱에서 자동으로 보정되도록 하였다. 정면이 아닌 곳을 공격받게 되면 대비하지 못한 방향에서의 공격을 받게 되는 것이므로 정면을 공격하여 성공하였을 때보다 상대 캐릭터가 더 많은 점수를 얻을 수 있도록 하였다. 방향을 크게 3가지로 분류하여 상대방 캐릭터가 바라보고 있는 각도의 차이가 315°~45°일 때는 정면으로 1배의 점수를, 45°~135°이거나 225°~315°일 때는 옆면으로 1.2배의 점수를, 135°~225°일 때는 후면으로 1.5배의 점수를 획득할 수 있도록 하였다. 또한 공격하는 캐릭터가 바라보고 있는 방향과 상대 캐릭터가 실제로 존재하는 방향을 비교하여 정면 이외의 방향이라면 공격을 실패하도록 하여 방어의 수단으로도 활용될 수 있게 하였다.

3.2 공격과 방어

공격행동은 크게 유효거리 1~2의 손을 통한 공격행동과 유효거리 2~3의 발을 통한 공격행동으로 나눌 수 있다. 캐릭터의 상태는 서있는 자세, 앉아있는 자세, 점프인 자세의 3가지로 구분하여 이에 따라 공격의 성공, 실패를 판별하게 된다.

공격 행동의 종류	성공 조건 (상대방의 상태)	실패 조건 (상대방의 상태)
위 공격	점프	서기, 앉기 서서 막기
중간 공격	서기, 앉기	점프 서서 막기
앞아서 공격	서기, 앉기	점프 앉아 막기
점프중 공격	서기, 점프	앉기 서서 막기

표 2 - 공격 행동의 종류에 따른 성공, 실패 조건

공격이 성공한 경우에 얻게 되는 점수를 <표 3>에 나타내었다. 정면이 아닌 경우에서의 공격은 앞

에서 설명한 것처럼 표에 표시된 점수의 1.2배에서 1.5배까지 획득이 가능하다.

공격의 종류	점수
점프중 손 공격	2
일반 손 공격	1
점프중 발 공격	3
일반 발 공격	2

표 3 - 공격의 성공시 획득하는 점수

3.3 시뮬레이션

본 실험에서는 규칙 학습의 횟수를 1,000틱부터 1,048,576,000틱까지 2배씩 증가시켜 실험을 진행하였고, 공정한 데이터를 얻기 위해서 각기 다른 랜덤 시드를 부여한 20개의 실험을 진행하여 평균을 구하였다. 학습률은 0.001, 0.01, 0.1, 1의 4가지 경우에 대하여 실험을 진행하였다

결과에 대한 분석은 지능 캐릭터의 획득점수를 양 캐릭터가 획득한 점수로 나눈 수치를 그 척도로 삼았으며 20개를 나눈 평균값과 평균에 대한 표준편차를 확인하였다.

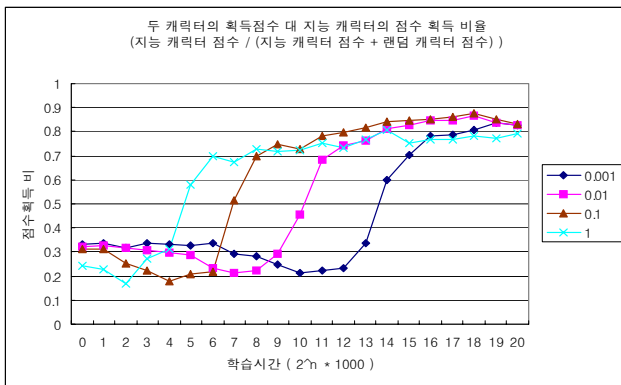


그림 2 - 두 캐릭터의 획득점수 대 지능 캐릭터의 점수 획득 비율

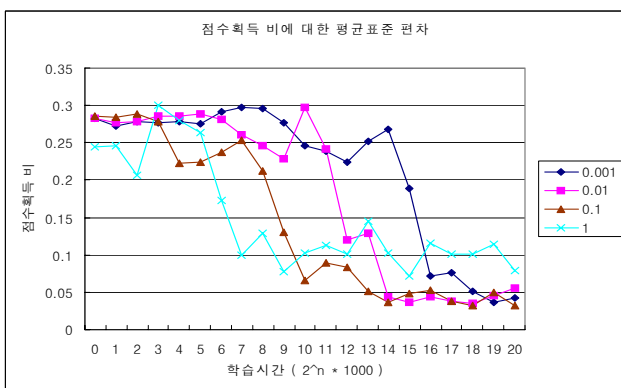


그림 3 - 점수 획득 비에 대한 평균 표준 편차

학습률 1의 경우 다른 학습률에 비해 더 적은 학습 시간에서도 좋은 점수 획득 비율을 보이지만 전 구간에서의 최대 점수 획득비는 다른 학습률의 최대

점수 획득비보다 떨어지고 다른 학습률보다 평균 표준 편차가 더 적은 학습 시간에서 낮게 나타나나 그래프 전체적으로 볼 때에는 다른 학습률에 비해서 일정한 수치로 수렴되지 않고 진동폭이 크다는 것을 확인할 수 있다. 이는 학습률 1의 경우 다른 학습률의 경우보다 한 번의 학습을 통하여 수정되는 링크 간의 가중치 값이 상대적으로 크기 때문이다. 나머지 학습률의 그래프에서는 학습률 1의 대전 능력을 보이기까지 더 많은 학습 시간이 필요하지만 전체 영역에서 확인했을 때 최고 점수 획득 비율도 높고 표준 편차면에서도 안정적인 형태를 나타내는 것을 확인할 수 있다.

4. 결론

본 논문에서는 이차원 대전 액션 게임을 위한 학습이 가능한 신경망 기반의 지능 캐릭터 구현에 대해서 제안하고 이를 검증해 보았다. 실험을 통해서 신경망을 통한 지능형 캐릭터가 학습 시간이 많아지면 많아질수록 무작위로 행동하는 캐릭터에 비해서 더 좋은 대전 능력을 가지며 큰 학습률은 빠르게 좋은 대전 능력을 가지게 되는 지능형 캐릭터를 만들 수 있으나 안정적인 대전 능력에서는 약간 떨어지는 것을 확인할 수 있었다.

현재는 단순히 무작위로 발생하는 행동에 대하여 학습을 하고 대응을 하고 있으나, 몇 가지의 행동의 패턴을 지정하여 특정 패턴에 대한 지능 캐릭터의 반응이나 규칙 학습 이후에 변경되는 규칙에 대한 적응성도 추후에 계속 연구할 예정이다.

참고문헌

- [1] 조병현, 정성훈, 성영락, 오하령, "대전 액션 게임을 위한 신경망 지능 캐릭터의 구현", 한국퍼지및지능시스템학회 논문지, v.14, n.4, pp.383-389, 2004. 7. 25.
- [2] 조병현, 정성훈, 성영락, 오하령, "대전 게임에서 상대방 캐릭터의 행동 패턴을 학습하여 대응하는 신경망 지능 캐릭터", 전자공학회 논문지 제 41권 CI 편 제6호, pp.579-590.
- [3] 조병현, 정성훈, 성영락, 오하령, "신경망을 이용한 지능형 게임 캐릭터의 구현", 한국정보처리학회 논문지, 제11-B권 제6호, pp831-840
- [4] Richard. P. Lippman, "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, pp.4-22, April, 1987.