

Slide 0

지능형 게임 캐릭터 연구

연구 내용

- 박사학위 논문 I (국민대, 2004년, 조병헌박사)
 - 제목: 지능형 게임 캐릭터를 위한 학습 및 적응 방법에 관한 연구
 - 개요
 - * 신경망으로 지능형 캐릭터를 구현
 - * 상대 캐릭터와의 게임을 통해 획득한 점수를 이용하여 강화학습
 - * 강화학습을 통하여 게임의 규칙을 스스로 학습
 - * 행동하나만 아니라 행동패턴을 학습하는 방법 제안
 - * 개체수준 및 개체군 수준에서 게임 상대방의 전략이나 환경변화에 스스로 적응하는 방법 제안
(유전자 알고리즘을 사용하여 개체군 수준에서의 적응 방법 제안)

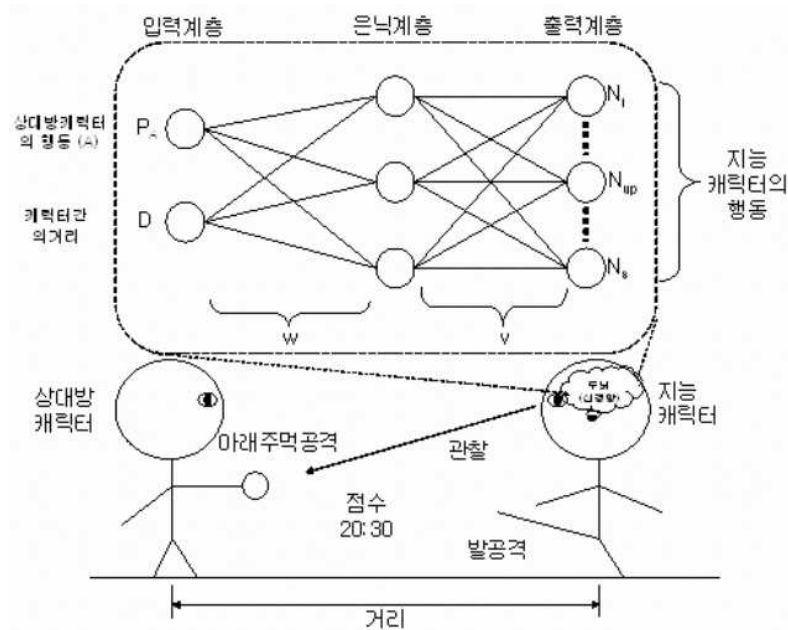
Slide 1

신경망 구조

- 가정
 - 단일 클릭 행동 게임
 - * 모든 행동이 하나의 클릭에서 종료되는 게임
 - cf. 다중 클릭 행동 게임: 하나의 행동에 여러 개의 클릭이 필요한 게임 (일반 게임)
- 신경망 구성
 - 입력
 - * 상대방 캐릭터의 행동: 주먹, 발차기등을 정수로 매핑하여 입력
 - * 캐릭터간 거리: 상대 캐릭터와의 거리를 숫자로 입력
 - 출력
 - * 지능 캐릭터의 행동 갯수만큼 출력노드 있음
 - * 값은 [0, 1] 실수 값 출력
 - * 출력 값 중 가장 큰 값을 갖는 행동을 함
 - 신경망: Feed-forward Neural Network 사용 (LGU model 사용)
 - * 입력-은닉층: $h_j = f(\sum_{i=0}^{N_i} (x_i \cdot w_{ij}))$
 - * 은닉-출력층: $z_j = f(\sum_{i=0}^{N_h} (h_i \cdot v_{ij}))$

Slide 2

신경망 구조 (Cont'd)



Slide 3

신경망 학습

- 학습과정
 - 초기 지능 캐릭터의 신경망의 가중치는 임의 값으로 임의의 행동을 수행
 - 상대 캐릭터와의 대결을 통해서 스스로 학습하여 게임의 규칙을 익힘
 - 상대 캐릭터는 사람이 조작하는 캐릭터나 혹은 기존의 방법으로 구현된 캐릭터 등이 될 수 있음
 - 상대 캐릭터는 자신만의 전략과 공격 및 방어법을 갖고 있음

Slide 4

- 학습
 - 오류 역전파를 이용한 강화학습
(환경으로부터 받는 누적 보상 값을 최대화 할 수 있는 최적의 행동 전략을 학습)
 - 점수의 차이를 강화 값으로 사용
 - (지능 캐릭터 점수 - 상대방 캐릭터 점수) 가
 - * 양이면 같은 상황에서 해당 행동을 하도록 장려,
 - * 아니면 해당 행동을 하지 않도록 학습
 - 절대값이 클수록 크게 강화

신경망 학습 (Cont'd)

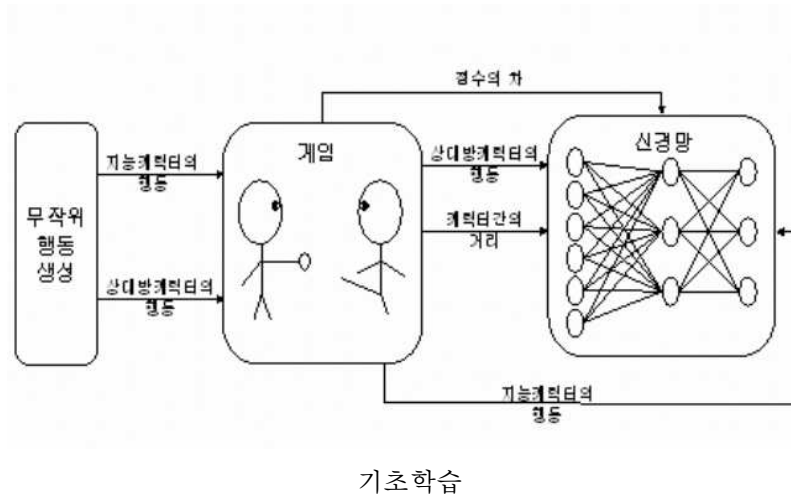
- 두 종류의 학습 단계 (기초학습, 실전학습)
 - 기초학습: 게임의 규칙을 학습하는 단계
 - 실전학습: 게임 중 환경이나 상대방의 전략이 변경되었을 경우에 대처하기 위한 학습

Slide 5

- 기초학습 단계
 - 게임의 규칙을 학습하기 위하여 상대방 캐릭터와 지능 캐릭터의 행동을 무작위적으로 생성
 - 획득한 점수로 게임의 규칙을 학습
 - 무작위적으로 생성하기 때문에 모든 경우 (상대 캐릭터 행동 × 거리 × 지능 캐릭터 행동)가 학습되지는 않음

신경망 학습 (Cont'd)

Slide 6



신경망 학습 (Cont'd)

- 강화학습
 - 출력층과 은닉 계층 사이의 링크 가중치 수정

Slide 7

$$\delta_j = z_j \cdot (1 - z_j) \cdot (d_j - z_j)$$

$$v_{ij} += \alpha(t) \cdot \delta_j \cdot h_i$$

- * d_j 는 목표값
- * z_j 는 j 번째 출력노드
- * v_{ij} 는 i 번째 은닉 노드와 j 번째 출력 노드간 링크 가중치
- * $\alpha(t)$ 는 시간에 따른 학습률의 함수로서 학습률 $\times e^{-t/\Delta t}$ 를 사용 (Δt 는 $t_{end} - t_{start}$ 로 t_{start} 는 학습시작 시간, t_{end} 는 학습종료 시간)

신경망 학습 (Cont'd)

- d_j 설정 방법

* 지능 캐릭터의 행동에 해당하는 출력 노드

Slide 8

$$d_j = \begin{cases} (s_i - s_o)/sd_{max} & \text{if } (s_o - s_i) > 0 \\ 0 & \text{otherwise} \end{cases}$$

- s_i 는 지능 캐릭터의 획득 점수
- s_o 는 상대 캐릭터의 획득 점수
- sd_{max} 는 최대 점수 차

* 그외 출력 노드 (학습할 필요가 없음): $d_j = z_j \rightarrow \delta_j = 0 \rightarrow v_{ij} = 0$

신경망 학습 (Cont'd)

- 입력층과 은닉 계층 사이의 링크 가중치 수정

$$\delta_j = h_j \cdot (1 - h_j) \cdot \sum_{k=0}^{N_h} v_{jk} d_k$$

$$w_{ij} += \alpha(t) \cdot \delta_j \cdot x_i$$

Slide 9

- * x_i 는 i 번째 입력노드 값
- * h_j 는 j 번째 은닉노드 값
- * N_h 는 은닉노드의 갯수
- * w_{ij} 는 i 번째 입력노드와 j 번째 은닉노드 간의 링크 가중치

- 학습이 완료되면

- * 특정 상황(상대방의 행동 및 거리)에서 출력 노드의 값이 가장 큰 값의 행동을 선택
- 특정 상황에서 가장 유리한 행동 (점수차가 큰 행동)을 선택

신경망 학습 (Cont'd)

- Slide 10**
- 실전학습 단계
 - 기초학습 단계 후에 실제 게임을 진행하면서 학습하는 단계
 - 실전학습 이유
 - * 기초학습 단계에서 모든 상황에 대한 학습이 이루어지기 어려움
 - * 게임 규칙의 일부 변경 및 게임 점수 체계의 변경 등에 적응력을 갖음
 - * 게임의 흥미를 더하기 위해 기초학습 단계에 학습률을 작게하고 실전에 학습하게 함
 - 기초학습 단계와의 차이점
 - * 실전 게임 중이기 때문에 캐릭터의 행동이 무작위적이 아니라 신경망 출력에 의해 결정됨

실험

- Slide 11**
- 캐릭터 행동
 - 11가지 행동
 - * 정지, 전진, 후진, 막기, 점프, 앉기, 아래/위 주먹공격, 아래/위 발공격, 필살기
 - 단일클릭: 모든 행동은 동시에 시작해서 동시에 끝남
 - 행동 소요 시간: 1클릭
 - 공격별 유효거리

공격행동	공격점수	유효거리
아래주먹	1	0~2
위주먹	2	
아래발	3	2~3
위발	4	
필살기	5	3~5

실험 (Cont'd)

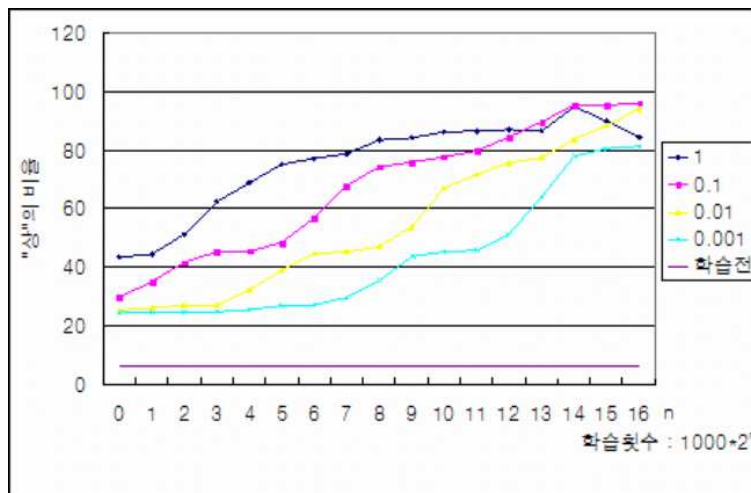
Slide 12

- 방어
 - 아래주먹, 아래발 공격은 상대방이 막거나 점프하면 점수 획득 못함
 - 위주먹, 위발 공격은 상대방이 막거나 앉으면 점수 획득 못함
 - 필살기는 상대방이 막으면 점수 반만 획득
- 학습 평가 기준
 - 지능 캐릭터의 규칙 학습 여부를 판단하기 위한 기준
 - 1) 모든 행동에 대하여 획득 가능한 점수와 지능 캐릭터가 실제로 획득한 점수를 비교
 - 2) 일정 시간 게임을 수행한 후 상대방 캐릭터와 지능 캐릭터의 획득 점수를 비교

실험 (Cont'd)

Slide 13

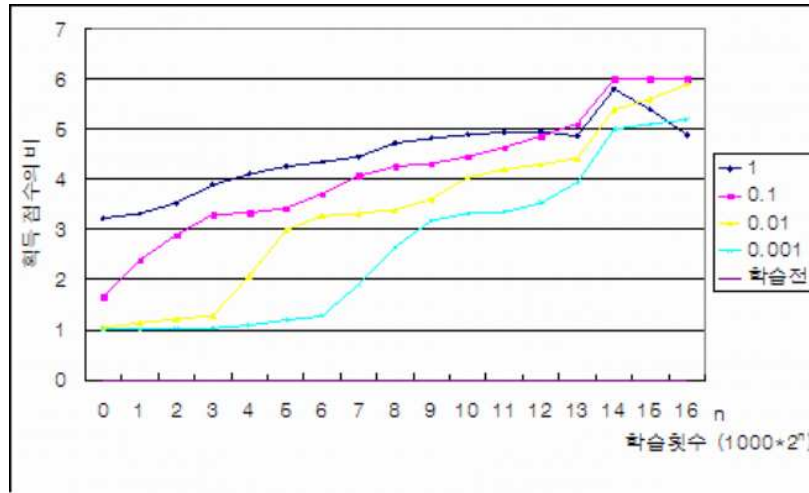
- 실험결과
 - 1) 번 평가 기준에 따라 획득한 점수 차를 상,중,하로 나누고 상을 획득한 비율을 측정



실험 (Cont'd)

- 2)번 평가 기준에 따라 지능캐릭터와 상대 캐릭터가 획득한 점수의 비를 측정

Slide 14



- 지능캐릭터 행동과 최적행동을 비교 (학습률 1.0에서 250만번 학습 후 비교)

실험 (Cont'd)

Slide 15

표 4 최적 행동

		상대방의 행동											
		0	1	2	3	4	5	6	7	8	9		10
거리	0	7	7	7	Any	7	6	7	7	7	7	7	정지 0
	1	7	7	7	Any	7	6	7	7	7	7	7	전진 1
	2	9	9	9	Any	9	8	9	9	9	9	9	후진 2
	3	10	10	10	10	10	10	10	10	10	10	10	막기 3
	4	10	10	10	10	10	10	10	10	10	10	10	점프 4
	5	10	10	10	10	10	10	10	10	10	10	10	앉기 5
	6	Any										{1} ^c	아래주먹 6

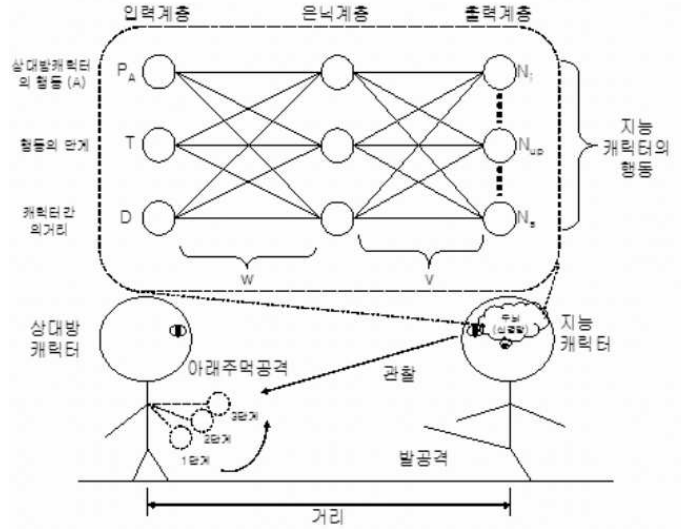
표 5 학습 후의 행동

		상대방의 행동											
		0	1	2	3	4	5	6	7	8	9		10
거리	0	7	7	7	7	7	7	7	7	7	7	7	정지 0
	1	7	7	7	9	9	9	7	7	7	7	7	전진 1
	2	9	9	9	10	10	8	9	9	9	9	9	후진 2
	3	10	10	10	10	10	10	10	10	10	10	9	막기 3
	4	10	10	10	10	10	10	10	10	10	10	10	점프 4
	5	10	10	10	10	10	10	10	10	10	10	10	앉기 5
	6	10	10	10	10	10	10	10	10	10	10	10	아래주먹 6

다중 클릭 행동 게임

- 행동이 하나의 클릭에서 종료되는 것이 아니라 여러 클릭에 걸쳐 일어남
- 신경망 구조

Slide 16



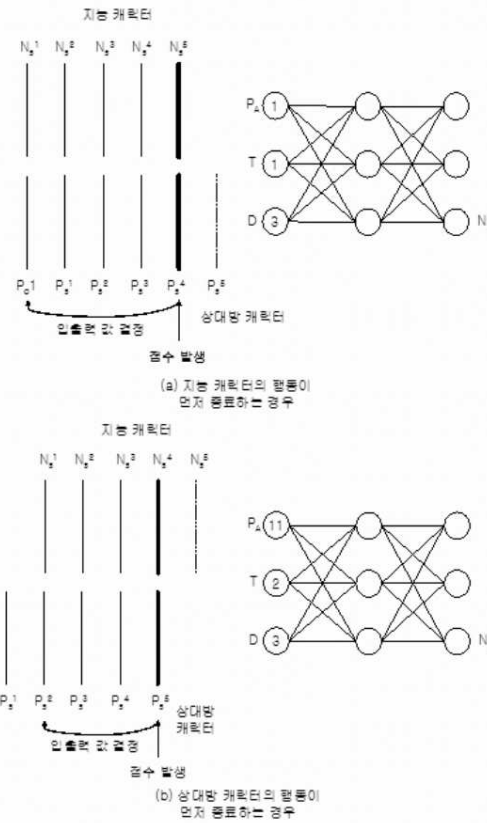
다중 클릭 행동 게임 (Cont'd)

Slide 17

- 가정
 - 행동별로 여러 개의 단계가 필요
 - 캐릭터가 행동을 수행하고 있을 때 상대의 공격으로 인하여 타격을 받으면 해당 행동은 무효화
 - 수행 중이던 행동을 임의로 취소하거나 다른 행동으로 바꿀 수 없음
- 점수 발생의 시점 차 발생
 - 두 캐릭터의 행동 중 하나가 완료되어 타격을 주는 시점을 중심으로 함
 - 다만, 지능 캐릭터의 행동 시작을 중심으로 상대 캐릭터의 행동, 단계, 거리를 학습

다중 클럭 행동 게임 (Cont'd)

Slide 18



다중 클럭 행동 게임 (Cont'd)

Slide 19

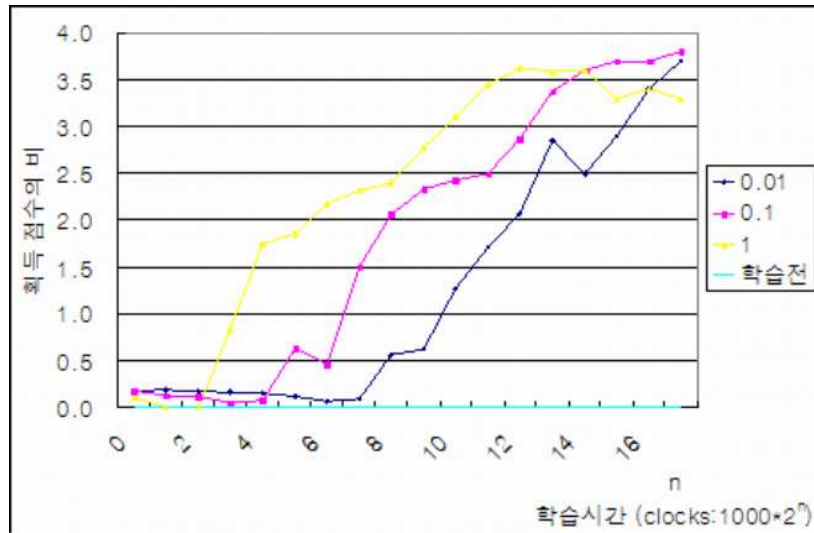
- 공격 별 필요시간/공격점수/유효거리

공격행동	필요시간(클럭)	공격점수	유효거리
아래주먹	2	1	0~2
위주먹	4	2	
아래발	6	3	2~3
위발	8	4	
펼살기	10	5	3~5

다중 클럭 행동 게임 (Cont'd)

- 실험 결과

Slide 20



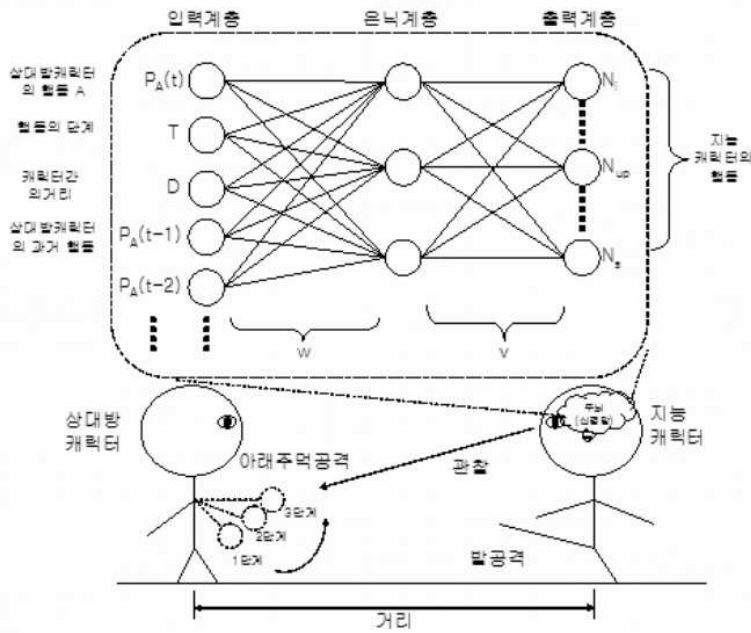
행동 패턴 학습

- 배경
 - 위의 연구는 상대 캐릭터의 현재 행동만을 관찰해서 대응하는 연구
 - 사람의 경우 대부분 특정 행동 패턴이 있음
 - 상대 캐릭터의 특정 행동 패턴을 학습하여 대응하는 지능 캐릭터 구현
- 행동 패턴 예
 - 손 공격 후 뒤돌려 차기
 - 특정 거리에서는 특정 공격 반복
- 신경망 구조
 - 행동 패턴에 대응하기 위하여 상대 캐릭터의 과거 행동도 입력
 - 과거 행동의 갯수는 미리 정해 놓음

Slide 21

행동 패턴 학습 (Cont'd)

Slide 22



행동 패턴 학습 (Cont'd)

Slide 23

- 예) 상대 캐릭터의 행동: 전진, 위주먹, 앉기, 아래주먹 공격 단계 3
- 상대 캐릭터의 행동은 미리 정해진 패턴에 따라 생성됨
- 지능 캐릭터의 행동은 무작위로 생성됨
- 점수 체계의 변경
 - 점수가 오로지 현재 행동에 의해서만 결정된다면 행동패턴을 학습해도 장점이 없음 (행동 패턴을 학습할 이유가 없음)
 - 연속적인 공격 형태에 따라 다른 점수가 부여되는 경우 행동패턴을 학습하면 좋음 (실제 대부분의 경기에서 이와 같음)

행동 패턴 학습 (Cont'd)

Slide 24

- 이동 행동에 대한 학습
 - 상황에 따라서 이동이 더 좋은 경우가 많음
 - * 상대 캐릭터의 어떤 공격에 대하여
 - * 남은 단계 동안의 지능 캐릭터의 어떤 공격도 점수를 획득하지 못함 (무효이거나 공격이 약해서)
 - * 방어를 해도 점수를 잃을 경우
 - * 만약, 이동으로 유효거리를 벗어날 수 있다면
 - * 이동이 더 좋은 행동
 - 문제는 이동은 공격이 아니기 때문에 점수가 없음 → 학습할 방법이 없음
 - 이동 후 공격에서 점수를 획득하면 이동에도 1/2의 점수를 부여함
 - 연속적인 이동에 대하여 이전 이동의 1/2의 점수를 부여함

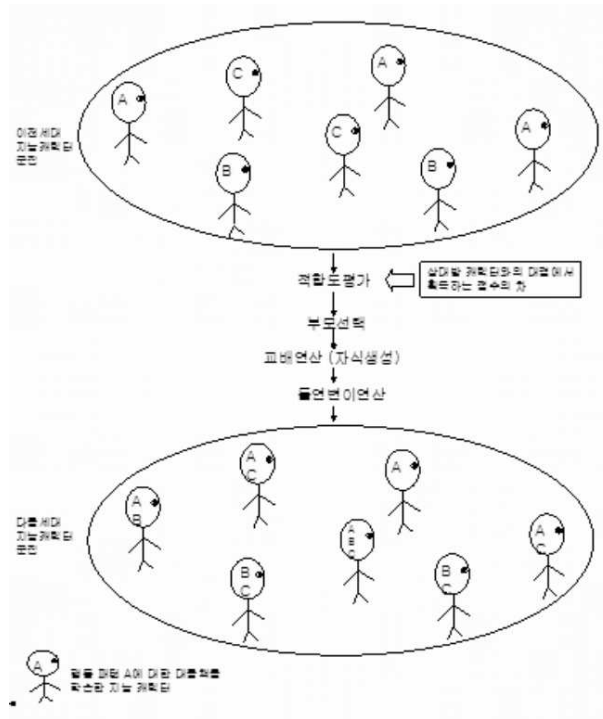
적응 방법

Slide 25

- 개체군 적응방법
 - 유전자 알고리즘을 이용하여 개체군 수준에서 진화적으로 적응
 - 특정 행동 패턴에 학습한 지능 캐릭터군 → 모든 행동 패턴에 대해 적응하는 지능 캐릭터군으로 진화
 - 점수체계가 변화하면 점수획득에 유리한 새로운 공격형태를 갖게 진화
- 진화 방법
 - 여러 개체가 상대 캐릭터와 게임하면서 점수를 획득하고
 - 점수를 적합도로 하여 개체를 평가한 후
 - 유전자 알고리즘의 선택(selection), 자식생성(reproduction)을 통해 진화
 - 자식생성의 주요 연산은 교배(crossover)와 돌연변이(mutation)임

적응 방법 (Cont'd)

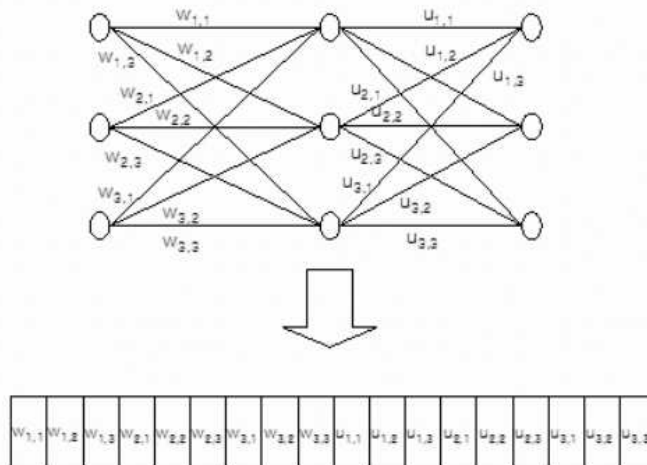
Slide 26



적응 방법 (Cont'd)

- 염색체 표현 방법
 - 신경망의 가중치를 유전자로 하여 염색체를 표현

Slide 27



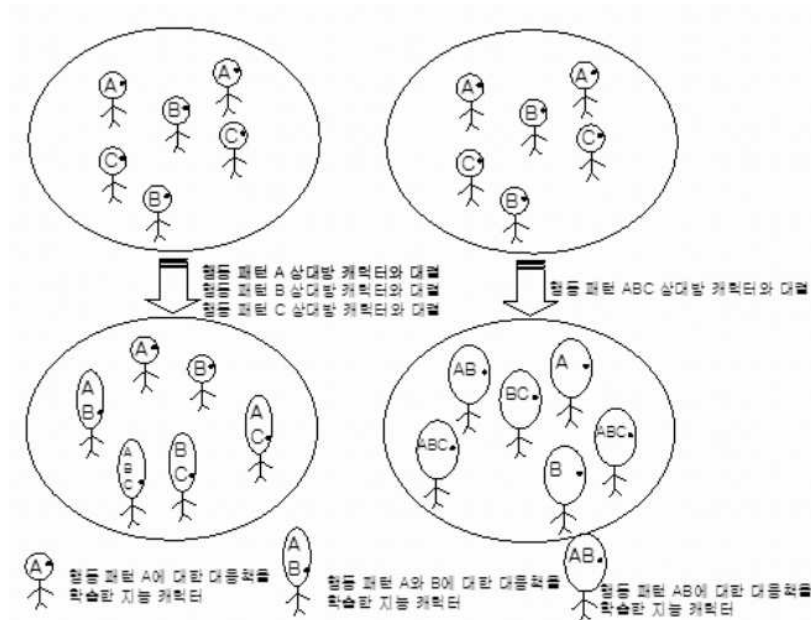
적응 방법 (Cont'd)

Slide 28

- 실험
 - 개체수: 30, 교배확률: 0.8, 돌연변이 확률: 0.1, 최대 1000세대 후 종료
 - 상대 캐릭터와 일정 횟수 대결한 후 점수차로 적합도 계산
 - 초기 개체는 A, B, C 행동패턴을 학습한 지능 캐릭터로 초기화
 - 행동 패턴 적응
 - * 개별: A,B,C 각각 행동하는 캐릭터와 대결하게 함
 - * 결과: A,B 혹은 B,C등 조합에 대하여 대응할 수 있는 캐릭터로 진화
 - * 연결: A,B,C 행동을 연결해서 행동하는 캐릭터와 대결하게 함
 - * 결과: AB, BC 등 연결 조합에 대하여 대응할 수 있는 캐릭터로 진화

적응 방법 (Cont'd)

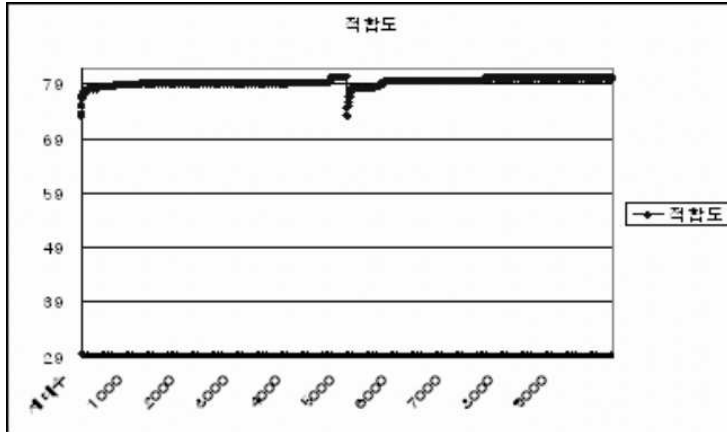
Slide 29



적응 방법 (Cont'd)

- 행동 패턴 적응 실험 결과
 - * 1 ~ 5,000: A,B,C 로 각각 행동하는 캐릭터와 대결하게 함
 - * 5000 세대에 A,B,C 로 행동하는 캐릭터와 대결하게 함 (적합도가 급격히 낮아짐)
 - * 초기 빠르게 적응하고 이후 점차적으로 적응함

Slide 30



적응 방법 (Cont'd)

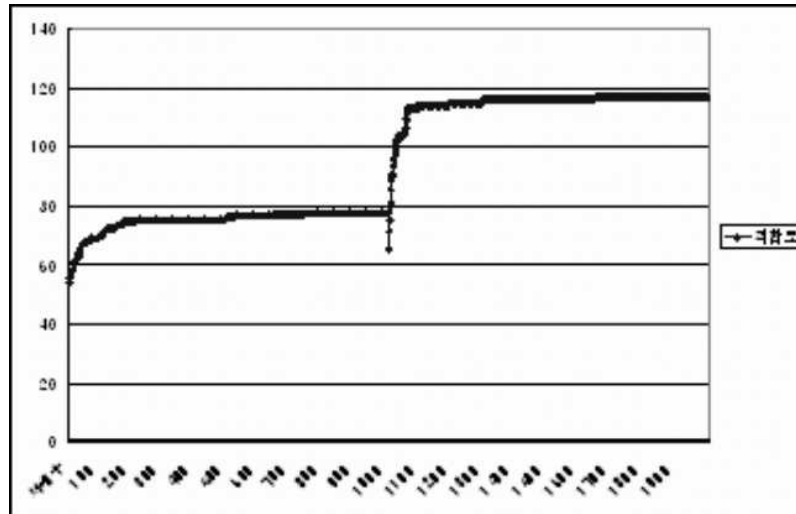
- 게임 규칙 변화 실험 결과
 - * 30개의 개체수
 - * 각각 10개씩 A,B,C패턴에 적응한 개체
 - * 1000 세대에서 게임 규칙 (점수)를 다음 표처럼 변화시킴

Slide 31

공격 행동	변경 전 공격점수	변경 후 공격점수
아래주먹	1	5
위주먹	2	4
아래발	3	3
위발	4	2
필살기	5	1

적응 방법 (Cont'd)

* 결과: 빠르게 적응



연구 결론

- Slide 33
- 기존 연구
 - 지능 캐릭터에 인공지능을 도입하는 시도가 적었음
 - 인공지능 시도도 대부분 단순 규칙기반 추론 수준임
 - 미리 입력된 행동으로 게이머의 흥미를 반감시킴
 - 환경 변화에 대한 적응적 요소가 적음
 - 새로운 규칙을 적용하면 새로 코딩해야함
 - 제안 방법
 - 장점
 - * 게임 규칙을 전혀 모르는 상태에서 학습을 통하여 규칙을 스스로 학습
 - * 여러 특성을 갖는 개체에 대하여 시간이 경과함에 따라서 스스로 적응할 수 있음
 - * 게이머가 예측할 수 없는 행동도 가능함
(단, 해당 행동이 엉뚱한 것이면 흥미를 반감시킬 수 있음)
 - * 환경변화 및 게임규칙 변화에 스스로 적응할 수 있음

연구 결론 (Cont'd)

Slide 34

- 제한점
 - * 강화학습을 위하여 행위에 대하여 잘했는지 못했는지를 판단해 주는 제 3자가 있어야함
 - * 지능 캐릭터의 외부 인터페이스(즉, 입출력)이 너무 복잡한 경우 학습에 오랜 시간이 걸릴 수 있음, 특히 학습되지 않은 영역의 경우 엉뚱한 행위를 할 수 있음
 - * 대전 게임과 같이 즉각적으로 반응하는 지능 캐릭터의 경우 유용하나 전략적인 판단이나 계획을 수행해야하는 캐릭터의 경우 불가능할 수 있음
 - * 게임이 복잡한 경우 전체 캐릭터의 일부 기능으로 제한해야함