알겠습니다. 논문 "A Simulation Study on Reinforcement Learning for Navigation Application" 전체를 한국어로 번역해 드리겠습니다. 내용이 길어 섹션별로 나누어 번역합니다.

페이지 1

Scientific Online

인공지능 및 응용 (ARTIFICIAL INTELLIGENCE AND APPLICATIONS) ISSN(Print): 2374-4979 ISSN(Online): 2374-4987 Volume 1, Number 2, August 2014

인공지능 및 응용

네비게이션 응용을 위한 강화 학습 시뮬레이션 연구

Jaspreet Singh Bal, Nitaigour Premchand Mahalik*

캘리포니아 주립대학교 프레즈노, 조던 농업 과학 기술 대학, 산업 기술 학과 2255 E Barstow Ave., California, 93740, USA. *교신 저자: [이메일 주소 삭제됨]

초록:

[source: 2] 본 논문에서는 비모수적 학습 및 의사 결정 방법인 강화 학습의 일종인 Q-러닝 구현에 대한 연구를 기여했습니다. [source: 3] 우리는 시뮬레이션을 통해 Q-러닝 알고리즘을 공식화하고 시연했습니다. 이 작업에는 의사 코드(pseudo-code) 공식화와 응용 프로그램을 고려한 알고리즘 개발이 포함됩니다. [source: 4] 강화 학습의 응용은 개념적인 농업 필드를 통해 시뮬레이션되며, 여기서 로봇은 나무에 도달한 다음 최종적으로 과일을 저장 지점(목표)으로 전달하도록 명령받습니다. [source: 5] 우리는 $gamma$와 $alpha$의 효과를 연구했습니다. 결과는 학습 매개변수($gamma$)와 학습률($alpha$)이 특정 응용 분야에 대한 Q-러닝 기반 강화 알고리즘을 개발할 때 고려해야 할 두 가지 중요한 매개변수임을 보여줍니다. [source: 6] 우리는 또한 시뮬레이션 연구를 통해 응용 분야의 최적 $gamma$ 및 $alpha$ 값을 설정했습니다.

키워드:

[source: 7] 강화 학습; Q-러닝; 로봇 네비게이션; 소프트 컴퓨팅, 자동화 및 제어

1. 배경 및 소개

의사 결정은 작업을 조직하고, 계획하고, 실행하고, 완수하는 데 유용합니다. [source: 8] 의사 결정은 많은 분야에서 수많은 응용 프로그램을 가지고 있습니다. 마르코프 결정 과정(Markov decision process)은 행동의 효과가 현재 상태에만 의존한다는 마르코프 속성에 전적으로 기반합니다. [source: 9] 마르코프 결정 과정의 행동은 결정론적(deterministic) 행동과 확률론적(stochastic) 행동이라는 두 가지 행동 형태로 나타낼 수 있습니다. [source: 10] 결정론적 행동에서는 모든 행동과 상태에 대해 새로운 상태가 정의됩니다. [source: 11] 얻어진 보상은 합산되지만 최종 결과는 불분명할 수 있습니다. [source: 12] 확률론적 분포에서는 모든 행동과 상태에 대해 다음 상태에 대한 확률 분포가 지정됩니다. [source: 13] 기대값과 최악의 경우에 초점을 맞추면 의사 결정 과정이 근사화 및 샘플링을 적절하게 활용할 수 있습니다 [1, 2]. [source: 14] 베이즈 정리(Baye's theorem)는 모수적 의사 결정 모델의 개념을 이해하는 데 중요한 역할을 합니다. [source: 15] 이 정리는 모델링 및 매개변수 값과 관련된 모든 불확실성을 고려하는 데 유용합니다. [source: 16] 최대 우도(Maximum likelihood)는 매개변수 값을 고정하는 것과 관련되므로 대부분의 의사 결정 분석을 전적으로 책임집니다. [source: 17] 베이즈 접근법은 개별 매개변수의 범위를 지정하므로 사전 정보를 구현하는 이점이 있습니다 [3-5]. [source: 18] 비모수적 의사 결정 모델링은 인공 신경망 및 결정 트리에서 더 나은 성능을 보여줍니다. 통계적 분포가 가정되는데,

---

페이지 2

[source: 19] 이는 다중 소스 데이터와 호환되지 않습니다. [source: 20] 데이터 분포에서 이루어진 가정은 비모수적 의사 결정 모델에서는 이루어지지 않으므로 오류를 피할 수 있습니다 [6, 7]. [source: 21] 강화 학습은 상황을 행동으로 변환하고 보상 점수를 최대화하기 위해 적용되는 방법론적 접근 방식의 한 종류입니다. [source: 22] 에이전트는 스스로 행동을 수행하는 방법과 어떤 행동이 가장 많은 보상을 가져올 것인지를 배워야 합니다. [source: 23] 이를 위해 우리는 선호도를 지정하고, 행동 기록을 남기고, 이러한 행동을 수행할 특정 시간을 명시해야 합니다 [8, 9]. [source: 24] 강화 학습은 컴퓨터가 플레이하는 보드 게임, 엘리베이터 제어, 네트워크 라우팅, 데이터 마이닝, 로봇 제어, 음성 인식, 생물정보학, 웹 및 텍스트 데이터 처리 등에서 훌륭한 응용 분야를 가지고 있습니다. [source: 25] 이는 에이전트에게 보상을 잃고 있을 때 알려주고 승리를 쟁취할 다른 대안적인 방법을 제안합니다 [10]. [source: 26] 본 논문은 강화 학습 알고리즘의 유용성을 광범위하게 연구합니다. [source: 27] 특히, Q-러닝 알고리즘을 연구하고 관련된 매개변수의 최적 값에 대한 제안을 합니다. [source: 28] 결과는 시뮬레이션을 통해 보여줍니다.

2. 강화 학습과 Q-러닝에 대한 검토

[source: 29] 강화 학습은 에이전트가 최적의 방식으로 목표를 수행하고 달성하도록 지시를 받는 기계 학습의 개념입니다. 강화 학습은 매우 많은 응용 분야를 가지고 있습니다. 많은 강화 알고리즘은 동적 프로그래밍 기법과 관련이 있습니다. [source: 30] 강화 학습 알고리즘은 정보가 필요하지 않습니다. 이 학습은 입력 없이 작동한다는 점에서 표준적인 지도 학습(supervised learning)과 다릅니다. [source: 31] 이는 알려지지 않은 영역 탐색과 현재 지식 활용에 관련됩니다. [source: 32] 강화 학습 모델은 (i) 행동 집합 A, (ii) 환경 상태 집합 B, (iii) 에이전트의 관찰을 설명하는 규칙, (iv) 스칼라 즉시 보상 변화를 결정하는 규칙, (v) 상태 간 전환에 관한 규칙으로 설명됩니다. [source: 33] 일반적으로 강화 학습에서는 무작위 규칙이 관찰됩니다. 이는 마지막 전환이 관찰에 따라 즉각적인 보상을 받는다는 원칙에 따라 작동합니다. [source: 34] 환경의 현재 상태 그림은 에이

전트에 의해 완전히 관찰됩니다. [source: 35] 에이전트는 가능한 한 많은 보상 점수를 얻습니다. 에이전트는 관찰 $o(t)$를 수행한 후 보상 $r(t)$를 받습니다. [source: 36] 행동 $a(t)$는 행동 집합에서 선택됩니다. 따라서 행동 맵이 필요합니다. [source: 37] 행동 맵은 주변 환경에 관한 것입니다. 환경은 실제로 에이전트를 새로운 상태로 이동시키고 새로운 보상 $r(t+1)$을 제공합니다. [source: 38] 이 행동은 이력의 함수로 수행되거나 완전히 무작위로 선택될 수 있습니다. [source: 39] 최적으로 행동하는 에이전트는 이전 성능과 비교합니다. 에이전트는 최적의 방식으로 작동하며 장기적인 결과를 고려합니다 [11]. [source: 40] 강화의 강도는 환경의 크기와 정밀한 설명에 따라 달라집니다. [source: 41] 각 상태에서의 행동에 대한 기대값은 지속적으로 업데이트됩니다. 모든 행동은 가능한 모든 상태에 대해 값을 부여받으며, 이는 행동 수행에 대한 즉각적인 보상과 새로운 상태에 기반한 기대 보상 모두에 의존합니다. [source: 42] Q 값의 업데이트는
$Q(s,a) = Q(s,a) +$
$alpha[r +$
$gamma$
$max\_a' Q(s',a') - Q(s,a)]$
를 사용하여 수행됩니다. [source: 43] 여기서, a는 행동 벡터, r은 보상, s는 상태 벡터,
$gamma$
는 할인 계수(discount factor)이며
$alpha$
는 수렴 제어를 위한 학습률(learning rate)입니다. [source: 44] 목표는 Q-값이 상태 $Q(s,a)$를 최대화하는 행동 a를 선택하도록 제안하는 것입니다. [source: 45] 따라서 함수 $Q(s,a)$는 상태 s에서 행동 a를 수행하는 용이성을 평가하기 위해 점진적으로 학습됩니다. [source: 46] 즉각적인 보상 $r(s,a)$와 상태 s'가 해당 단계에서 행동 a가 완료될 때 달성되면, 업데이트된 Q-값은
$Q(s,a) = r(s,a) +$
$gamma$
$max\_a' Q(s',a')$
가 됩니다. 계수
$gamma$
는 보상이 다른 것들보다 더 일찍 와야 하는지 여부를 결정합니다. [source: 47] 그 값은 0과 1 사이에 있습니다. Q-러닝은 다중 에이전트 협조 시스템에 적용될 수 있습니다.

페이지 3

(이미지 1: 강화 학습의 그래픽 표현) [source: 48] 에이전트(AGENT)는 상태 s(t)를 환경(ENVIRONMENT)에 보내고, 환경은 행동 a(t)에 따라 다음 상태 s(t+1)과 보상 r(t+1)을 에이전트에게 반환합니다.

3. Q-러닝 알고리즘 구현

[source: 49] 본 논문에서 우리는 비모수적 의사결정 방법인 강화 학습의 구현에 대한 연구를 기여했습니다. 이 연구에서는 강화 학습의 대표적인 기법인 Q-러닝 기법을 고려합니다. [source: 50] 우리는 시뮬레이션을 통해 Q-러닝 알고리즘을 공식화하고 시연했습니다. 이 작업에는 의사 코드(pseudo-code)의 공식화와 응용을 고려한 알고리즘 개발이 포함됩니다. [source: 51] 결과는 학습 매개변수(
$gamma$
)와 학습률(
$alpha$

)이 특정 응용 분야에 대한 Q-러닝 기반 강화 알고리즘을 개발할 때 고려해야 할 두 가지 중요한 매개변수임을 보여줍니다. [source: 52] 우리는 시뮬레이션 연구를 통해 응용 분야의 최적
$gamma$
및
$alpha$
값을 설정했습니다. [source: 53] 강화 학습의 응용은 개념적인 농업 필드를 통해 시뮬레이션되며, 여기서 로봇은 나무에 도달한 다음 최종적으로 과일을 저장 지점(목표)으로 전달하도록 명령받습니다. [source: 54] 에이전트인 로봇은 학습자이며 경로에 대해 아무것도 모릅니다. [source: 55] 어떠한 감독 없이, 에이전트는 9개의 다른 나무로부터 오렌지 나무에 도달하기 위해 단계별로 학습할 것입니다. [source: 56] 다음으로, (i) 의사 코드, (ii) 환경 모델링, (iii) 상태, 에이전트 및 행동이 제시됩니다.

[source: 57] **의사 코드:** Q-러닝을 위한 의사 코드는 아래에 요약되어 있습니다 [12, 13]. [source: 58] 초기화: 모든 S와 a에 대해 $Q(S, a) = 0$으로 설정 시작 현재 상태 S 결정 상태 S에서 행동 a를 선택하고 실행 즉각적인 보상 r 결정 새로운 상태 S' 찾기 다음 표현식으로 $Q(S, a)$ 업데이트
$Q(S, a) = r +$
$gamma$
$cdot$
$max(Q($
$delta a(S, a), a'))$
S = S' 종료

환경 모델링: 그림 9(->실제 그림 2)와 같이 오렌지 밭에 9개의 오렌지 나무가 있다고 가정합니다. 로봇이 나무에서 오렌지를 수확하기 위해 고용됩니다. [source: 59] 화살표는 로봇이 한 나무에서 다른 나무로 따라가야 할 경로를 나타냅니다. [source: 60] 각 나무에는 1부터 9까지 번호가 매겨져 있으며 목표는 저장 지점(goal)을 나타냅니다. [source: 61] 목적지 지점으로 이어지는 나무는 단 하나뿐이라는 점에 유의하십시오.

페이지 4

(이미지 2 및 3과 해당 캡션) [source: 62] **그림 2.** (a) 1부터 9까지의 나무 아홉 그루와 미래 상태 F로 구성된 오렌지 수확 밭 (b) 상태 1부터 9와 미래 상태 F를 포함하는 노드 그래프 [source: 63]

[source: 64] **그림 3.** (a) 미래 저장 지점과 함께 있는 아홉 개의 다른 상태 (상태 다이어그램) (b) 행은 과거 상태, 열은 미래 상태인 매트릭스 R [source: 65]

다음 그림은 그래프로도 표시될 수 있으며, 각 나무는 노드 또는 정점(vertex)으로, 경로는 링크(link)로 나타냅니다. [source: 66] 로봇은 첫 번째 오렌지 나무 근처에 설정된 에이전트입니다.

[source: 67] **상태, 에이전트 및 행동:** 에이전트가 환경에 대해 아무것도 모르지만 경험을 통해 학습할 수 있다고 가정해 봅시다. [source: 68] 이 경우 에이전트는 오렌지 수집 로봇입니다. [source: 69] 따라야 할 경로의 순서에 대한 아이디어가 없습니다. [source: 70] 에이전트는 현재 1번 나무에 있으며 목표로 표시된 저장 지점에 도달하기를 원합니다. [source: 71] 상태와 행동을 정의해 봅시다. 에이전트는 1부터 9까지의 다른 나무에 도달

합니다. 각 나무는 위에 표시된 노드 그래프의 노드로 표현되며 상태라고 합니다. [source: 72] 화살표는 에이전트가 따르는 행동을 나타냅니다. 다음 상태 다이어그램은 목표 지점에 도달하면 즉시 주어지는 100의 보상을 보여줍니다. [source: 73] 에이전트는 다른 어떤 상태에서도 저장 지점에 직접 도달할 수 없으므로 보상이 0입니다. [source: 74] 즉각적인 보상 값은 각 화살표에 포함되어 있습니다. 에이전트는 추가로 100 포인트를 보상받아 해당 상태에 영원히 머물도록 권장됩니다. [source: 75] 이제 에이전트가 1에 위치하고 최종 지점 F에 도달하기를 원한다고 가정해 봅시다. 하지만 상태 1에서는 상태 4, 5, 6, 7, 8, 9 또는 최종 지점 F로 직접 갈 수 없습니다. 그들 사이에 직접적인 연결이 없기 때문입니다. [source: 76] 상태 2에서 에이전트는 상태 4, 5로 가거나 상태 1로 돌아갈 수 있습니다. 만약 에이전트가 상태 5에 있다면,

---

페이지 5

세 가지 가능한 행동은 상태 6, 2 또는 4로 가는 것입니다. 유사하게, 안팎으로 향하는 화살표는 상태 간의 경로를 나타냅니다. [source: 78] 그림 3에는 단 두 개의 보상만 표시되어 있습니다. 만약 에이전트가 상태 9에서 F로 가거나 F에서 영원히 머무르면, 100 포인트의 보상을 받습니다. [source: 79] 상태 다이어그램은 에이전트의 현재 위치와 미래 위치를 포함하는 보상 테이블 또는 행렬 R로 변환될 수 있습니다. [source: 80] 십자 표시(cross sign)는 한 상태에서 다른 상태로의 경로가 없음을 나타냅니다 [12, 13]. [source: 81] 값 0과 100은 에이전트가 한 위치에서 다른 위치로 이동하여 얻는 보상 포인트를 나타냅니다. [source: 82] 십자 또는 대시 값은 장벽이 존재하여 에이전트가 한 상태에서 다른 상태로 갈 수 없음을 나타냅니다 (그림 4(a)). [source: 83] 더 진행함에 따라, 환경 보상 행렬은 위의 행렬 R 테이블로부터 설계됩니다. [source: 84] 즉, 우리는 작업을 통해 에이전트를 안내할 새로운 행렬 Q를 계산하고 업데이트해야 합니다. [source: 85] R 행렬과 유사하게, 열은 행의 에이전트 현재 상태로부터 도달하는 미래 상태를 상징합니다. [source: 86] 에이전트는 경로를 통과하는 것에 대해 아무것도 모릅니다. 따라서 Q의 초기 값은 0 행렬입니다 (그림 4(b)). [source: 87] 에이전트는 다른 나무보다 큰 나무에서 더 많은 시간을 보낼 것으로 가정됩니다. [source: 88] 장벽은 알고리즘에 명시되지 않은 다른 상태로 에이전트가 가는 것을 방지하는 데 도움이 됩니다. [source: 89] 다음 방정식은 Q 값의 결정을 보여줍니다 [12, 13].

[source: 90]
$$Q(\text{상태}, \text{행동}) = alpha(R(\text{상태}, \text{행동}) + gamma \cdot textMax[Q(\text{다음상태}, \text{모든행동})]) quad(1)$$

괄호 안의 값은 행으로서의 상태와 열로서의 행동을 나타냅니다. 이 경우 학습 매개변수 $gamma$는 0.75로 간주되며, 에이전트의 필요한 효율성과 새로 얻은 정보가 기존 정보를 얼마나 대체할 수 있는지와 같은 요인에 따라 선택됩니다. 에이전트는 학습 인자가 1이면 가장

최신 정보를 이해할 수 있지만, 학습 인자가 0이 되면 작업을 수행할 수 없습니다.

학습: 초기 상태가 1이고 학습 매개변수
$gamma = 0.75$
와 학습률
$alpha = 0.1$
이라고 가정합니다. [source: 91] 초기에 Q 행렬은 0으로 설정됩니다.

에이전트가 상태 1에 있고 상태 2 또는 3으로만 갈 수 있다고 가정합니다. 무작위로 상태 2를 우리의 행동으로 선택합니다. 에이전트가 상태 2에 있다고 간주합니다. 여기서 에이전트는 상태 1, 4 또는 5로 갈 수 있으며 다음과 같이 쓸 수 있습니다.

페이지 6

[source: 92]
$Q(1,2) = 0.1(R(1,2) + 0.75$
$cdot$
$textMax[Q(2,1), Q(2,4), Q(2,5)]) = 0$
$quad(2)$

Q 행렬과 R(1, 2)의 값이 0이므로 Q(1, 2)의 값은 0이 됩니다. [source: 93] 이제 다음과 같이 다른 Q 값들을 결정해 봅시다. [source: 94]
$Q(1,3) = 0.1(R(1,3) + 0.75$
$cdot$
$textMax[Q(3,1), Q(3,4)]) = 0$
$quad(3)$

[source: 95] 이것은 시작점에서 두 번째 상태로 시작된 에이전트 학습의 첫 번째 이벤트를 마칩니다. 에이전트는 1에서 상태 2 또는 3으로 갈 수 있습니다. 유사하게, 경로에 있는 다른 상태들의 Q 값을 찾을 수 있습니다 (아래 참조). [source: 96]
$Q(2,1) = 0.1(R(2,1) + 0.75$
$cdot$
$textMax[Q(1,2), Q(1,3)]) = 0$
$quad(4)$
$Q(2,4) = 0.1(R(2,4) + 0.75$
$cdot$
$textMax[Q(4,2), Q(4,3), Q(4,5)]) = 0$
$quad(5)$
$Q(2,5) = 0.1(R(2,5) + 0.75$
$cdot$
$textMax[Q(5,2), Q(5,4), Q(5,6)]) = 0$
$quad(6)$
$Q(3,1) = 0.1(R(3,1) + 0.75$
$cdot$
$textMax[Q(1,2), Q(1,3)]) = 0$
$quad(7)$

$$Q(3,4) = 0.1(R(3,4) + 0.75$$
$$cdot$$
$$textMax[Q(4,2), Q(4,3), Q(4,5)]) = 0$$
$$quad(8)$$
$$Q(4,2) = 0.1(R(4,2) + 0.75$$
$$cdot$$
$$textMax[Q(2,1), Q(2,4), Q(2,5)]) = 0$$
$$quad(9)$$
$$Q(4,3) = 0.1(R(4,3) + 0.75$$
$$cdot$$
$$textMax[Q(3,1), Q(3,4)]) = 0$$
$$quad(10)$$
$$Q(4,5) = 0.1(R(4,5) + 0.75$$
$$cdot$$
$$textMax[Q(5,2), Q(5,4), Q(5,6)]) = 0$$
$$quad(11)$$
$$Q(5,2) = 0.1(R(5,2) + 0.75$$
$$cdot$$
$$textMax[Q(2,1), Q(2,4), Q(2,5)]) = 0$$
$$quad(12)$$
$$Q(5,4) = 0.1(R(5,4) + 0.75$$
$$cdot$$
$$textMax[Q(4,2), Q(4,3), Q(4,5)]) = 0$$
$$quad(13)$$
$$Q(5,6) = 0.1(R(5,6) + 0.75$$
$$cdot$$
$$textMax[Q(6,5), Q(6,7), Q(6,8)]) = 0$$
$$quad(14)$$
$$Q(6,5) = 0.1(R(6,5) + 0.75$$
$$cdot$$
$$textMax[Q(5,2), Q(5,4), Q(5,6)]) = 0$$
$$quad(15)$$
$$Q(6,7) = 0.1(R(6,7) + 0.75$$
$$cdot$$
$$textMax[Q(7,6), Q(7,8)]) = 0$$
$$quad(16)$$
$$Q(6,8) = 0.1(R(6,8) + 0.75$$
$$cdot$$
$$textMax[Q(8,6), Q(8,7), Q(8,9)]) = 0$$
$$quad(17)$$
$$Q(7,6) = 0.1(R(7,6) + 0.75$$
$$cdot$$
$$textMax[Q(6,5), Q(6,7), Q(6,8)]) = 0$$
$$quad(18)$$

$$Q(7,8) = 0.1(R(7,8) + 0.75$$
$$cdot$$
$$textMax[Q(8,6), Q(8,7), Q(8,9)]) = 0$$
$$quad(19)$$
$$Q(8,6) = 0.1(R(8,6) + 0.75$$
$$cdot$$
$$textMax[Q(6,5), Q(6,7), Q(6,8)]) = 0$$
$$quad(20)$$
$$Q(8,7) = 0.1(R(8,7) + 0.75$$
$$cdot$$
$$textMax[Q(7,6), Q(7,8)]) = 0$$
$$quad(21)$$
$$Q(8,9) = 0.1(R(8,9) + 0.75$$
$$cdot$$
$$textMax[Q(9,8), Q(9,F)]) = 0.1(0 + 0.75$$
$$times100) = 7.5$$
$$quad(22)$$
$$Q(9,8) = 0.1(R(9,8) + 0.75$$
$$cdot$$
$$textMax[Q(8,6), Q(8,7), Q(8,9)]) = 0$$
$$quad(23)$$
$$Q(9,F) = 0.1(R(9,F) + 0.75$$
$$cdot$$
$$textMax[F,F]) = 0.1(100 + 0.75$$
$$times100) = 17.5$$
$$quad(24)$$

페이지 7

[source: 98] *(이미지 5: 에피소드를 통한 Q 행렬 업데이트)* (a) 몇 번의 반복 후 (b) 10억 번의 반복 후

$$Q(F,F) = 0.1(R(F,F) + 0.75$$
$$cdot$$
$$textMax[Q(F,F)]) = 17.5$$
$$quad(25)$$

[source: 99] 언급된 바와 같이, 에이전트의 상태와 행동에 대한 Q-값은 여러 이벤트를 거친 후 방정식 (1)에 값을 넣어 결정됩니다. 새로운 Q-값을 우리 에이전트에 업데이트함으로써 새로운 Q 행렬이 얻어집니다 (그림 5). [source: 100] 이것을 에피소드 업데이트라고 합니다. 그림 5(b)는 에이전트가 10억 번의 반복 또는 에피소드를 통해 학습한 후 얻어진 Q 행렬입니다. [source: 101] 초기에 가장 높은 보상이 100으로 가정되었으므로, 모든 유효한 항목은 가장 높은 값인 331로 나눈 다음 100을 곱하여 상태 다이어그램으로 만듭니다. [source: 102] 끝에 위치한 보상은 로봇이 움직일 때마다 사용자가 값을 업데이트하기 어렵게 만듭니다. [source: 103] 이 한계를 극복하기 위해, 전체 보상 세트를 저장하고 끝에서 반대 순서로 업데이트합니다. [source: 104] 변동은 상태의 과거 행동과

그 보상을 주기적으로 저장하는 대체 기법으로 처리될 수 있습니다 [7b]. [source: 105] 본 논문에서는
$gamma$
의 효과를 연구했습니다. [source: 106] 이 맵은
$alpha = 0.1$
,
$gamma = 0.75$
인 q-러닝 맵 또는 상태 다이어그램입니다 (그림 6). [source: 107] 예를 들어, 동일한 반복 횟수에 대해
$alpha$
와
$gamma$
의 다른 값에 대해 다른 상태 다이어그램이 얻어질 것입니다. [source: 108] 선택된
$alpha$
와
$gamma$
값의 효과는 그림 7에 나와 있습니다.

4. 토론

이 섹션에서는 이 연구와 관련된 중요한 사항에 대한 간략한 토론을 제시합니다. [source: 109] 토론은 (i) 강화 학습의 장단점, (ii) 학습 매개변수
$gamma$
값, (iii) 의사 알고리즘 이해 및 구현의 문제를 다룹니다. [source: 110] 강화 학습은 탐험(exploration)과 활용(exploitation) 사이의 트레이드오프가 발생하는 기계 학습의 한 형태입니다 [14]. [source: 111] 기계 학습의 다른 두 형태는 지도 학습 또는 비지도 학습일 수 있습니다. [source: 112] 강화 학습의 장점은 (i) 응용 프로그램용 알고리즘 개발에 완전한 지식이 필요하지 않다는 것, (ii) 에이전트에 간단하고 쉽게 구현할 수 있다는 것, (iii) Java, C++ 또는 Microsoft Excel 파일로 프로그램을 만들어 모든 작업에 대한 알고리즘을 준비할 수 있다는 것, (iv) 지도 학습이 적합하지 않은 응용 프로그램에 적용될 수 있다는 것입니다 [15-19]. [source: 113]

페이지 8

[source: 114] *(이미지 6:*
$alpha = 0.1$
,
$gamma = 0.75$
값에 대한 상태 다이어그램)

학습 매개변수 값 결정은 응용 프로그램의 중요한 부분입니다. [source: 115] 아홉 개의 다른 나무에서 한 나무에 도달하는 에이전트에 대한
$gamma$
값은 0.75로 선택되었습니다. [source: 116] 먼저, 감마 값이 0.2로 선택되었다고 가정해 봅시다. [source: 117] 에이전트는 한 블록에서 다른 블록으로 더 나아가려고 시도하겠지만, 두 번째 블록에서 에이전트를 찾을 확률은 0이 될 것입니다. [source: 118] 학습 매개변수 값이 낮기 때문에 에이전트가 목적지 지점에 도달하는 데 더 많은 시간이 걸릴 것입니다. [source: 119] 이제 값을 0.8로 선택합니다. 에이전트가 다음 블록에 빠르게 도달하지만 목적지 지점에 도달할 확률은 우리의 기대에 미치지 못하는 것으로 나타났습니

다. [source: 120] 마지막으로, 효율성, 각 상태 간 소요 시간 및 이동 경로의 특성에 따라 학습 매개변수 값은 0.75로 설정됩니다. [source: 121] 유사하게, 학습률 $alpha$ 는 결정의 무작위성을 결정합니다. [source: 122] 에이전트는 한 상태에 있고 다른 상태로 가기 위한 행동을 선택합니다. [source: 123] 따라서 상태-행동 쌍의 Q-값을 설정하기 위해 다음 행동을 보고, 최대 보상을 확인하고, Q-러닝 알고리즘으로 계산을 수행한 다음 Q-행렬에 넣을 값을 얻습니다 (Dar, & Mansour, 2003). [source: 124] 만약 알파를 더 높은 값(예: 0.7, 0.8, 0.9)으로 사용한다고 가정하면, 가장 높은 미래 보상을 제공하는 행동을 선택할 것입니다. [source: 125] 그러나 알파가 낮으면(예: 0.1, 0.2, 0.3), 무작위 행동을 선택하여 새로운 경로를 더 자주 발견하게 되어 겉보기에는 미래 보상을 신경 쓰지 않는 것처럼 보입니다. [source: 126] 처음에는 낮은 알파 값을 선택하는 것이 좋은 습관이 될 것입니다. [source: 127] 값을 증가시킴으로써 에이전트는 먼저 환경에 대해 배우고 나서 최적의 경로를 따르기 시작합니다. [source: 128] 참고: (i) 알파 = 0.9이면 90%는 최적으로, 10%는 무작위 행동을 합니다; [source: 129] (ii) 알파 = 0.4이면 40%는 최적으로, 60%는 무작위 행동을 합니다; [source: 130] (iii) 알파 = 0.2이면 20%는 최적으로, 80%는 무작위 행동을 합니다. [source: 131] 마지막으로, 의사 코드는 실제 알고리즘 구현의 기초를 형성하지만, 여전히 일련의 단계로 정의될 필요가 있습니다. [source: 132] 이해하기 쉽지만, 알파와 감마를 선택하는 방법에 대한 참조가 없기 때문에 실제 문제에 구현하기 어려운 경우가 있습니다 [20, 21]. [source: 133] 이것이 이 연구가 수행된 이유입니다.

낮은 감마는 더 빠른 학습과 낮은 강화를 의미합니다. [source: 134] 또한 낮은 감마는 더 많은 탐험을 의미합니다. 이는 에이전트가 높은 보상 경로에 대한 선호도가 낮다는 것을 의미합니다. [source: 135] 반대로, 높은 감마는 더 느린 학습, 더 높은 강화를 의미합니다. 결과적으로, 높은 감마는 더 적은 탐험을 기대합니다. [source: 136] 결과적으로, 에이전트는 높은 보상 경로에 대한 선호도가 높습니다. Q-러닝 접근 방식은

페이지 9

[source: 137] *(이미지 7: 다양한 감마 값(알파=0.1 고정)에 대한 Q 행렬 결과)* (a) gamma = 0.9 and alpha = 0.1 [source: 138] (b) gamma = 0.8 and alpha = 0.1 (c) gamma = 0.6 and alpha = 0.1 [source: 139] (d) gamma = 0.5 and alpha = 0.1 [source: 140] (e) gamma = 0.4 and alpha = 0.1 (f) gamma = 0.1 and alpha = 0.1 [source: 141][source: 142][source: 143]

유망해 보입니다. Q-러닝을 사용하여, 시뮬레이션 환경에서 농업 필드를 탐색하기 위한 프로토타입 이동 로봇 소프트웨어가 설계되었습니다. [source: 144] C/C++ 언어로 프로그래밍하여 q-러닝을 활용하는 프로그램을 설계하는 것이 가능합니다. [source: 145] 프론트엔드는 LabView로 설계되었지만, 백엔드 계산은 C 컴파일러를 통해 수행되었습니다. [source: 146] 언급된 바와 같이, 다른 감마 값은 다른 유형의 학습으로 이어질 수 있습니다.

페이지 10

[source: 147] **5. 결론**

[source: 148] 본 연구에서는 Q-러닝 알고리즘 구현에 대한 시뮬레이션 연구를 수행했습니다. Q-러닝 매개변수는 에이전트의 학습 행동을 결정하는 데 중요한 역할을 합니다 [22]. [source: 149] 이 연구의 초점은 상황에 따라 다르기 때문에 에이전트의 학습 매개변수에 대한 적절한 값을 선택하는 것이었습니다. [source: 150] 매개변수 학습률은 결정의 무작위성을 결정하며 값이 낮으면 더 많은 무작위 행동을 수행합니다. [source: 151] 학습 매개변수 감마는 미래 보상의 중요성을 결정합니다. [source: 152] 학습률 알파의 값은 0.1이었고 감마 값은 0.75로 실현되었습니다. [source: 153] Q-러닝 알고리즘은 정의된 경로를 가진 아홉 개의 오렌지 나무의 실제 사례를 사용하여 시뮬레이션되며, 에이전트는 해당 경로를 따라 나무에 도달하도록 지시받습니다. [source: 154] 비록 학습 매개변수 값이 각각 0.1과 0.75로 선택되었지만, 이 값들은 최적이 아니며 작동 중에 무작위로 선택되었습니다. [source: 155] 결과적으로, 이 연구 작업은 Q-러닝 알고리즘 개발에서 알파와 감마의 다양한 조합을 제시합니다. [source: 156] 즉, 각 조합의 효과를 비교했습니다. 향후 연구에는 감마와 알파의 최적화 및 Q-행렬을 해결하기 위한 소프트웨어 개발이 포함될 것입니다.

[source: 157] **감사의 글**

저자들은 이 논문을 작성하기 위한 연구 시간(Release Time)과 출판 비용을 제공해주신 조던 농업 과학 기술 대학 학장님들께 감사를 표합니다.

[source: 158] **참고 문헌**

[1] M. D. Awheda and H. M. Schwartz, "Exponential moving average Q-learning algorithm," Adaptive Dynamic Programming And Reinforcement Learning(ADPRL), 2013 IEEE Symposium on, pp. 31-38, IEEE, 2013. [2] B. Givan and R. Parr, An introduction to Markov decision processes. [source: 159] http://www.cs.rice.edu/₩~vardi/dag01/givanl.pdf. [3] M. M. Triola, Bayes' Theorem. [source: 160] http://www.faculty.washington.edu/tamre/BayesTheorem.pdf. [4] B. E. Cline, "Tuning Q-learning parameters with a genetic algorithm," Proceedings of the Journal of Machine Learning Research, vol. 5, 2004. [source: 161] [5] H. Xu, "Robust decision making and its applications in machine learning," Proceedings of the Thesis, pp. 2-5, McGill University, 2009. [6] T. Ishiguro, T. Matsui, N. Inuzuka, and K. Wada, "Reinforcement learning methods to handle actions with differing costs in mdps," Knowledge-Based Intelligent Information and Engineering Systems, pp. 553-560, Springer, 2003. [7] J. Martyna, "Q-Learning algorithm used by secondary users for QoS support in cognitive radio network," Modern Advances in Applied Intelligence, pp. 389-398, Springer, 2014. [8] P. Dayan, Reinforcement Learning. [source: 162] http://www.gatsby.ucl.ac.uk/₩~dayan/papers/dw01.pdf. [9] M. Ahmed, "Optimum short part finder for robot using Q-learning," Diyala Journal of Engineering Sciences, vol. 05, no. 01, pp. 13-24, 2012. [source: 163]

페이지 11

[source: 164] [10] A. Ng, Applications of Reinforcement Learning. http://academicearth.org/lectures/applications-of-reinforcement-learning. [11] D. Pandey and P. Pandey, "Approximate Q-learning," Proceedings of the Second International Conference on Machine Learning and Computing, pp. 317-320, IEEE,

2010. [source: 165] [12] K. Teknomo, Q-learning by examples. 2006. [source: 166] http://people.revoledu.com/kardi/tutorial/ReinforcementLearning/index.html. [13] E. Even-Dar and Y. Mansour, "Learning rates for Q-learning," Proceedings of the Journal of Machine Learning Research, vol. 5, pp. 1-25, 2003. [source: 167] [14] A. R. Cassandra, Partially observable Markov decision processes. 2009. http://www.cassandra.org/pomdp/index.shtml. [source: 168] [15] A. Barto and S. Sutton, "Reinforcement learning: an introduction," [16] J. D. R. Millán, D. Posenato, and E. Dedieu, "Continuous action Q-learning," Machine Learning, vol. 49, no. 2-3, pp. 247-265, 2002. [source: 169] [17] J.-F. Chamberland and V. V. Veeravalli, Wireless Sensor Networks: Signal Processing and Communications Perspectives. Wiley, 2008. [source: 170] [18] A. Chapman, Problem solving and decision making. 2010. http://www.businessballs.com/problemsolving.htm. [source: 171] [19] Y. Chen and Q. Zhao, "Wireless Sensor Networks: Signal Processing and Communications Perspectives," 2008. [20] D. W. Engels and S. E. Sarma, "A hierarchical Q-learning algorithm to solve the reader collision problem," Proceedings of the International Symposium on Applications and Internet Workshops, pp. 1-4, IEEE, 2005. [21] S. Karumanchi, T. Allen, and S. Scheding, "Non-parametric learning to aid path planning over slopes," [22] S. Patnaik and N. Mahalik, "Multiagent coordination utilizing Q-learning," pp. 361-379, 2007.

페이지 12

[source: 172] **이 저널에 대하여**

[source: 173] AIA는 Scientific Online Publishing에서 발행하는 오픈 액세스 저널입니다. 이 저널은 다음 범위에 중점을 둡니다 (이에 국한되지 않음):

➤ 인공 신경망 ➤ 베이즈 네트워크 ➤ 생물정보학 ➤ 인지 과학 ➤ 컴퓨팅과 마음 ➤ 데이터 마이닝 ➤ DNA 컴퓨팅 및 양자 컴퓨팅 ➤ 진화적 영감 컴퓨팅 ➤ AI 기초 ➤ 퍼지 방법 ➤ 지능형 웹 ➤ 기계 학습 ➤ 다중 에이전트 시스템 ➤ 자연 컴퓨팅 ➤ 자연어 처리 ➤ 신경정보학 ➤ 비고전적 컴퓨팅 및 새로운 컴퓨팅 모델 ➤ 퍼베이시브 컴퓨팅 및 앰비언트 인텔리전스 ➤ 철학과 AI ➤ 로보틱스 ➤ 소프트 컴퓨팅 이론 및 응용

여러분의 독창적인 원고 제출을 환영합니다. [source: 174] 더 많은 정보는 저희 웹사이트를 방문해 주십시오: http://www.scipublish.com/journals/AIA/

아래를 클릭하여 저희를 팔로우할 수 있습니다: ◇ 페이스북: https://www.facebook.com/scipublish ◇ 트위터: https://twitter.com/scionlinepub ◇ 링크드인: https://www.linkedin.com/company/scientific-online-publishing-usa ◇ 구글+: https://google.com/+ScipublishSOP

SOP는 저자들이 다음 규칙 하에 연구 결과를 기고하는 것을 환영합니다:

A 모든 독창적이고 새로운 연구 성과를 기꺼이 출판하지만, SOP는 어떠한 부정행위도 용납할 수 없습니다: 표절, 실험 데이터 위조 또는 조작. [source: 175] 국제 출판사로서 SOP는 다양한 문화를 매우 중요하게 생각하며 종교, 정치, 인종, 전쟁 및 윤리에 대해 신중한 태도를 취합니다. [source: 176] SOP는 과학적 결과 전파를 돕지만, 논문과 함께 저자로 인해 발생하는 법적 위험이나 해로운 영향에 대해 어떠한 책임도 공유하지 않습니

다. [source: 177] SOP는 가장 엄격한 동료 검토를 유지하지만, 출판된 모든 논문에 대해 중립적인 태도를 유지합니다. [source: 178] SOP는 연구 발전을 함께 진전시키기 위해 편집 위원회에 봉사할 선임 전문가를 기다리는 열린 플랫폼입니다.

번역이 완료되었습니다.

In [ ]: