계층적 모방 및 강화 학습

호앙 엠 레 [1] **난 장** [2] **알렉 아가르왈** [2] **미로슬라브 두딕** [2] **이송 위** [1] **할 다우메 3** [3 2]

초록

전문가의 피드백을 효과적으로 활용하여 순차적 의사 결정 정책을 학습하는 방법을 연구합니다. 우리는 일반적으로 강화 학습에서 심각한 문제를 야기하는 희박한 보상과 긴 기간의 문제에 초점을 맞춥니다. 우리는 기본 문제의 계층 구조를 활용하여 다양한 전문가 상호 작용 모드를 통합하는 *계층적 가이드라고 하는* 알고리즘 프레임워크를 제안합니다. 이 프레임워크는 다양한 수준에서 모방 학습(IL)과 강화 학습(RL)의 다양한 조합을 통합하여 전문가의 노력과 탐색 비용을 획기적으로 절감할 수 있습니다. 몬테주마의 리벤지를 포함한 긴 지평선 벤치마크를 사용하여 우리의 접근 방식이 계층적 RL보다 훨씬 빠르게 학습할 수 있고 표준 IL보다 훨씬 더 레이블 효율적이라는 것을 입증합니다. 또한 프레임워크의 특정 순간에 대한 라벨링 비용을 이론적으로 분석합니다.

1. 소개

강화 학습(RL)의 목표인 보상 신호만으로 좋은 에이전트 행동을 학습하는 것은 계획 기간이 길고 보상이 드물 때 특히 어렵습니다. 이러한 긴 지평을 다루는 성공적인 방법 중 하나는 에이전트가 전문가를 관찰하고 질문하여 학습하는 모방 학습(IL)입니다(Abbeel & Ng, 2004; Daume´ et al., 2009; Ross et al., 2011; Ho & Ermon, 2016). 기존 모방 학습 접근법의 한 가지 한계는 장시간 문제에서 많은 양의 데모 데이터가 필요하다는 점입니다.

이 백서에서 다루는 핵심 질문은 '*전직 전문가가 있을 때 그들의 피드백을 가장 효과적으로 활용할 수 있는 방법은 무엇인가'입니다.* 샘플 유효성을 개선하기 위한 일반적인 전략

[1]캘리포니아 공과대학교, 패서디나, 캘리포니아[2] Microsoft 리서치, 뉴욕, 뉴욕[3] 메릴랜드 대학교 칼리지 파크,

오랜 기간에 걸쳐 RL에 능숙해지는 방법은 문제의 계층적 구조를 활용하는 것입니다(Sutton et al., 1998; 1999; Kulkarni et al., 2016; Dayan & Hinton, 1993; Vezhnevets et al., 2017; Dietterich, 2000). 우리의 접근 방식은 모방 학습에서 고도의 계층 구조를 활용합니다. 우리는 근본적인 문제가 계층적이며 전문가로부터 하위 과제를 쉽게 도출할 수 있는 경우를 연구합니다. 우리의 핵심 설계 원칙은 *계층적 가이드라는* 알고리즘 프레임워크로, 상위 전문가의 피드백(레이블)을 하위 학습자에게 집중(안내)하는 데 사용됩니다. 상위 수준 전문가는 하위 수준 학습자가 필요한 경우(하위 작업이 매스 태스크화되지 않은 경우)에만 상태 공간의 관련 부분에 대해서만 학습할 수 있도록 합니다. 이는 단순히 하위 작업을 세분화하는 단순한 계층적 접근 방식과는 다릅니다. 상태 공간의 관련 부분에 집중하면 학습 속도가 빨라지고(샘플 효율성 향상), 이미 숙달된 하위 작업에 대한 피드백을 생략하면 전문가의 노력이 줄어듭니다(라벨 효율성 향상).

먼저 계층적 모방 학습의 문제를 공식화하고(섹션 3), 전

MD. 대응: Hoang M. Le <hmle@caltech.edu>.

35th 국제 기계 학습 컨퍼런스, 스웨덴 스톡홀름, PMLR 80, 2018.

문가가 여러 추상화 수준에서 피드백을 제공할 때 자연스럽게 발생하는 비용 구조를 신중하게 분리합니다. 먼저 IL에 계층적 가이드를 적용하고, 계층적 가이드를 적용한 행동 복제 및 DAgger의 변형을 도출한 후(Ross et al., 2011), 그 이점을 이론적으로 분석합니다(섹션 4). 다음으로 계층적 안내를 높은 수준의 IL과 낮은 수준의 RL이 있는 하이브리드 설정에 적용합니다(섹션 5). 이 아키텍처는 특히 높은 수준의 의미론적 지식에 액세스할 수 있고 하위 작업 범위가 충분히 짧지만 낮은 수준의 전문가가 너무 비싸거나 구할 수 없는 환경에 적합합니다. 간단하지만 매우 까다로운 미로 영역과 몬테주마의 복수(섹션 6)에서 접근 방식의 효율성을 입증합니다. 실험 결과, 약간의 전문가 피드백을 통합하면 순수한 계층적 RL에 비해 성능이 크게 향상될 수 있음을 보여줍니다.[1]

2. 관련 작업

간결성을 위해 여기서는 관련 작업에 대한 간략한 개요를 제공하고, 추가 논의는 부록 C를 참조하시기 바랍니다.

[1]코드 및 실험 설정은 https:// 에서 확인할 수 있습니다. sites.google.com/view/hierarchical-il-rl

모방 학습. 모방 학습은 크게 수동적인 모방 수집(행동 복제)과 능동적인 모방 수집으로 이분화할 수 있습니다. 전자의 설정(Abbeel & Ng, 2004; Ziebart et al., 2008; Syed & Schapire, 2008; Ho & Ermon, 2016)은 데모를 선험적으로 수집한다고 가정하고, IL의 목표는 데모를 모방하는 정책을 찾는 것이라고 가정합니다. 후자의 설정(Daume´ et al., 2009; Ross et al., 2011; Ross & Bagnell, 2014; Chang et al., 2015; Sun et al., 2017)은 현재 정책에 의해 취해진 조치에 대응하여 데모를 제공하는 능동적인 전문가가 있다고 가정합니다. 우리는 두 접근법을 계층적 환경으로 확장하는 방법을 모색합니다.

계층적 강화 학습. 계층적 정책을 학습하기 위한 몇 가지 RL 접근 방식이 제안되었는데, 그중에서도 옵션 프레임워크가 가장 많이 사용되었습니다(Sut-ton et al., 1998; 1999; Fruit & Lazaric, 2017). 이 프레임워크에서는 유용한 옵션 집합이 선험적으로 완전히 정의되어 있고, (세미 마르코프) 상위 레벨의 옵션만 계획하고 학습한다고 가정합니다. 이에 비해 에이전트는 이러한 하위 목표를 달성하는 정책에 직접 액세스할 수 없으며 전문가 또는 강화 피드백을 통해 학습해야 합니다. 우리와 가장 유사한 계층적 RL 연구는 Kulkarni 등(2016)의 연구로, 유사한 계층 구조를 사용하지만 상위 수준의 전문가가 없으므로 계층적 지침이 없습니다.

강화 학습과 모방 학습의 결합. 강화 학습과 모방 학습을 결합하는 아이디어는 새로운 것이 아닙니다(Nair et al., 2017; Hester et al., 2018). 하지만 이전 연구에서는 IL을 '사전 훈련' 단계로 사용하는 평면 정책 클래스에 초점을 맞췄습니다(예: 데모로 리플레이 버퍼를 미리 채우는 방식). 이와는 대조적으로, 우리는 계층적 정책 클래스에 대해 여러 수준의 피드백을 고려하며, 각 수준은 잠재적으로 다른 유형의 피드백을 받을 수 있습니다(예: 한 수준에서는 모방, 다른 수준에서는 강화). 계층적 전문가 감독과 어느 정도 관련이 있는 접근 방식은 An-dreas 등(2017)의 접근 방식으로, 하위 목표의 상징적 디스크립션에 대한 접근을 가정하지만 해당 상징이 무엇을 의미하는지 또는 어떻게 실행할지 모른다고 가정합니다. 최근 Sun 등(2017)의 연구를 제외하면, 이전 문헌에서는 IL과 RL 간의 샘플 복잡성 비교에 크게 초점을 맞추지 않았습니다.

3. 계층적 형식주의

간단하게 설명하기 위해 자연스러운 2단계 계층 구조를 가진 환경을 고려하며, HI 수준은 하위 작업 선택에 해당하고 LO 수준은 해당 하위 작업 실행에 해당합니다. 예를 들어 상담원의 전체 목표가 건물에서 나가는 것일 수 있습니다. HI 수준에서 상담원은 먼저 하위 작업인 *"엘리베이터로 이동"을* 선택한 다음 *"엘리베이터를 타고 내려가기"*, 마지막으로 *"걸어 나가기"를* 선택할 수 있습니다. 이러한 각 하위 작업은 LO 수준에서 실제로 내비게이션을 통해 실행되어야 합니다.

환경 정리하기, 엘리베이터 버튼 누르기 등입니다.[2] *하위 목표라고도* 하는 하위 작업은 다음과 같이 표시됩니다.
로 표시되며, 기본 액션은 a 로 표시됩니다. 에이전트(학습자라고도 함)는 하위 목표 g*를* 반복적으로 선택하고, 완료될 때까지 일련의 작업 a*를* 실행하여 이를 수행한 다음, 새로운 하위 목표를 선택하는 방식으로 행동합니다. 에이전트의 선택은 관찰된 상태 s 에 따라 달라질 수 있습니다.[3] HI 수준의 지평선은 H_{HI} 즉, 궤적이 최대 H_{HI} 하위 목표를 사용한다고 가정하고, LO 수준의 지평선은 H_{LO} 즉, 최대 H_{LO} 원시 액션을 수행한 후 에이전트가 하위 목표를 달성하거나 새로운 하위 목표를 결정해야 한다고 가정해 봅시다. 따라서 궤적에서 원시 행동의 총 개수는 최대 $H_{\text{FULL}} := H\, H_{\text{HILO}}$ 입니다.

계층적 학습 문제는 *메타 컨트롤러라고* 하는 상위 수준 정책 μ *하위* 목표 정책 π_g : 하위 정책이라고 하는 각 g*에* 대해 동시에 학습하는 것입니다. 학습자의 목표는 메타 컨트롤러와 하위 정책을 함께 실행할 때 높은 보상을 얻는 것입니다. 각 하위 목표 g*에* 대해 (아마도 학습된) 종료 함수 β_g 도 있습니다:
참, 거짓, π_g 의 실행을 종료합니다.
계층 에이전트는 다음과 같이 작동합니다:

1: **for** $h_{\text{HI}} = 1 \ldots H_{\text{HI}}$ **do**
2: 상태 S 관찰 및 하위 목표 G 선택 $\mu(s)$
3: **FOR** $H_{\text{LO}} = 1 \ldots H_{\text{LO}}$ **do**
4: 상태 S 관찰
5: **IF** $\beta_g(S)$ **THEN BREAK**
6: CHOOSE $A \leftarrow \pi_g(S)$

각 하위 정책 π_g 의 실행은 $\tau = (s_1, a_1, \ldots, s_H, a_H, s_{H+1})$의 LO 수준 *궤적을* 생성하며, H는 $_{\text{LO}}$ 입니다.[4] 전체 동작은 *계층적 궤적* $\sigma = (s_1, g_1, \tau_1, s_2, g_2, \tau_2, \ldots)$ 를 생성하며, 각 LO 수준 궤적 τ_h 의 마지막 상태는 σ 의 다음 상태 s_{h+1} 와 다음 LO 수준 궤적 τ_{h+1} 의 첫 번째 상태와 일치합니다 . LO 레벨 궤적 τ_h 을 제외한 σ 의 후속 궤적 $\tau_{\text{HI}} := (s_1, g_1, s_2, g_2, \ldots)$를 HI 레벨 *궤적이라고* 합니다. 마지막으로, *전체 궤적* τ_{FULL} 은 모든 LO 레벨 궤적의 연결입니다.

메타데이터를 보유한 *전문가에게* 접근한다고 가정합니다.

[2]중요한 실제 적용 사례는 목표 지향적 대화형 시스템입니다 . 예를 들어, 사용자의 항공편 및 호텔 예약 및 예약을 지원하는 챗봇은 여러 차례의 대화 과정을 거쳐야 합니다(Peng et al., 2017; El Asri et al., 2017). 챗봇 개발자는 *사용자 목표 물어보기*, *날짜 물어보기*, *항공편 제안하기*, 확인하기 등과 같은 하위 작업의 계층 구조를 설계합니다. 각 하위 작업은 여러 차례의 대화로 구성됩니다. 일반적으로 글로벌 상태 추적기는 계층적 대화 정책과 함께 존재하여 하위 작업 간 제약 조건이 충족되도록 보장합니다.

[3]단순화를 위해 상태라는 용어를 사용하지만, 환경이 완전히 관찰 가능하거나 마르코프적일 필요는 없습니다.

[4]궤적에는 선택적으로 각 원시 행동 후 보상 신호가 포함될 수 있으며, 이는 환경에서 제공되거나 섹션 5에서 살펴볼 것처럼 의사 보상일 수 있습니다.

컨트롤러 $\mu^{\wedge}$, 하위 정책 $\pi^{\wedge}$, 종료 함수 $\beta^{\wedge}$,
하나 또는 여러 유형의 감독을 제공할 수 있는 사람입니다 :

- *계층* 데모: *계층 데모*. 전 퍼트는 s*부터*시작하여 계층 정책을 실행합니다.
  를 반환하고 결과 계층적 궤적 σ = $(s^{\wedge}_1, g^{\wedge}_1, s^{\wedge}_2, g^{\wedge}_2, \tau^{\wedge}, \ldots)$, 여기서 $s^{\wedge}_1 = s$입니다.
- 라벨$_{\text{HI}}$ *(*τ_{HI} *)*: HI 레벨 *라벨링*. 전문가는 주어진 HI 레벨 궤적의 각 상태에서 좋은 다음 하위 목표 $\tau_{\text{HI}} = (s_1, g_1, s_2, g_2, \ldots)$ 를 제공하여 la-.
  벨드 데이터 세트 $\{(s_1, g^{\wedge}_1), (s_2, g^{\wedge}_2), \ldots\}$.
- 라벨$_{\text{LO}}$ *(*τ ; g*)*: *LO* 레벨 *라벨링*. 전문가는 주어진 LO 수준 궤적 $\tau = (s_1, a_1, s_2, a_2, \ldots)$의 각 상태에서 주어진 하위 목표 g에 대한 좋은 다음 기본 동작을 제공하여 라벨링된 데이터 세트를 생성합니다.
  $\{(s_1, a^{\wedge}_1), (s_2, a^{\wedge}_2), \ldots\}$.
  니다.
- $_{\text{LO}}$ *(*τ ; g*)* 검사 : LO 수준 *검사*. 궤적의 모든 상태에 주석을 달지 않고 하위 목표 g*가* 달성되었는지 여부만 확인하여 *합격* 또는 *불합격* 중 하나를 반환합니다.
- 라벨$_{\text{FULL}}$ *(*τ_{FULL} *)*: *전체 라벨링*. 전문가 라벨
  에이전트의 전체 궤적 $\tau_{\text{FULL}} = (s_1, a_1, s_2, a_2, \ldots)$,
  계층 구조를 무시하고 처음부터 끝까지입니다,
  레이블이 지정된 데이터 세트 $\{(s_1, a^{\wedge}_1), (s_2, a^{\wedge}_2), \ldots\}$.
  쉽습니다.
- 검사$_{\text{FULL}}$ *(*τ_{FULL} *)*: *전체 검사*. 전문가가 상담원의 전체 목표 달성 여부를 확인하여 *합격* 또는 *불합격* 중 하나를 반환합니다.

에이전트가 하위 정책 π_g 뿐만 아니라 종료 함수 β_g 도 학습하면 $\texttt{Label}_{\text{LO}}$ 도 양호한 값을 반환합니다.
의 각 상태에 대한 종료 값 $\omega^{\wedge} \in \{True,$ False$\}$입니다.
$\tau = (s_1, a_1 \ldots)$, 데이터 세트 $\{(s_1, a^{\wedge}, \omega^{\wedge}), .1. \}$를 산출합니다. }.
HierDemo와 Label을 모두 생성할 수 있지만
전문가의 계층 정책($\mu^{\wedge}$, $\{\pi^{\wedge}\}$)에 따라 다릅니다.
전문가 상호 작용 모드에서. HierDemo는

수동적 IL에 필요한 *전문가에 의해* 실행되는 계층적 궤적을 제공하며, 행동 복제의 계층적 버전을 가능하게 합니다

알고리즘 1 계층적 비헤어비어 클로닝(**h-BC**)

1: 데이터 버퍼 $_{\text{DHI}} \leftarrow \emptyset$ 및 $_{Dg} \leftarrow \emptyset$ 초기화, $g \in$ G
2: **for** t = 1, . . . , T **do**
3:시작 상태 s로새 환경 인스턴스 가져오기
4: $\sigma^s \leftarrow$ HierDemo(s)
5: **FOR ALL** $(S^s, G^s, T^s) \in \sigma^s$ **DO**
6: D 추가$_g$? ← D$_g$? ∪ $_{th}$
7: 7: 추가 $_{\text{DHI}} \leftarrow _{\text{DHI}} \cup \{(S^s, G^s)\}$
8: 모든 g에대해 하위 정책 $_{\pi g} \leftarrow$ Train($_{\pi g}$, $_{Dg}$)을 훈련합니다.
9: 훈련 메타 컨트롤러 $\mu \leftarrow$ Train(μ, $_{\text{DHI}}$)

검사, 즉 검사 $_{\text{LO}}(\tau_h; g_h)$ = *통과*, 재

전체 궤적도 전체 검사를 통과해야 합니다 ($_{\text{FULL}}$ *(*τ_{FULL} *)* = *통과*). 즉, 계층적 정책이 전체 작업에서 성공하기 위해 항상 LO 수준에서 전문가의 실행과 일치할 필요는 없습니다.

알고리즘적인 이유 외에도 분리하려는 동기는 다음과 같습

피드백의 유형은 전문가 쿼리마다 일반적으로 다른 양의 노력이 필요하며, 이를 *비용이라고 합니다*. 라벨 작업의 비용은 C^L , C^L 및 C^L 이며, 각 검사 작업의 비용은 다음과 같다고 가정합니다.

eration은 C^{l} 및 C^{l}. 많은 설정에서 LO 레벨 인-

검사는 LO 레벨 라벨링(예: C^l C^L)보다 훨씬 적은 노력이 필요합니다. 예를 들어, 로봇이 엘리베이터로 성공적으로 이동했는지 식별하는 것은 추정입니다.
엘리베이터로 가는 전체 경로에 라벨을 붙이는 것보다 훨씬

실험 환경에 적합한 합리적인 비용 모델 중 하나는 검사 작업은 궤적의 최종 상태를 확인하는 데 $O(1)$의 시간이 걸리는 반면, 라벨 작업은 궤적 길이에 비례하는 시간, 즉 세 가지 라벨 작업의 경우 $O(H_{\text{HI}})$, $O(H_{\text{LO}})$, $O(H\,H_{\text{HILO}})$이 걸린다고 가정하는 것입니다.

4. 계층적 가이드 모방 학습

계층적 안내는 다음과 같은 알고리즘 설계 원칙입니다.

높은 수준의 전문가의 피드백이 낮은 수준의 전문가를 안내합니다.

(Abbeel & Ng, 2004; Syed & Schapire, 2008). 레이블 연산은 대화형 IL에 필요한 *학습 에이전트의 궤적에 대한* 레

이블을 제공합니다. $\texttt{Label}_{\text{FULL}}$ 은 플랫 정책 학습에 대한 이전 연구에서 사용된 표준 쿼리이며(Daume´ et al., 2009; Ross et al., 2011), $\texttt{Label}_{\text{HI}}$ 과 $\texttt{Label}_{\text{LO}}$ 은 그 계층적 확장입니다.

이 백서에서 새롭게 소개된 검사 작업은 라벨 효율성을 크게 절감할 수 있는 대화형 계층적 가이드 프로토콜의 초석을 형성합니다. 이는 반응형 라벨 작업의 "지연" 버전으로 볼 수 있으며, 더 적은 노력이 필요합니다. 우리의 기본 가정은 주어진 계층적 트래킹이
젝토리 $\sigma = \{(s, g, \pi)\}$는 HI에 대한 전문가와 동의합니다.

레벨, 즉 $g_h = \mu(s_h)$, LO 레벨 궤적은

(i) 상위 레벨 전문가는 하위 레벨 전문가가 필요할 때만(하위 작업을 아직 숙달하지 않은 경우) 쿼리하도록 하고
(ii) 저수준 학습은 상태 공간의 관련 부분으로 제한됩니다. 먼저 데모로부터의 패시브 학습 내에서 이 프레임워크를 인스턴스화하여 *계층적 행동 복제*(알고리즘 1)를 얻은 다음, 능동적 모방 학습 내에서 *계층적으로 유도된 DAgger*(알고리즘 2)를 얻어 최고 성능의 알고리즘을 얻습니다.

4.1. 계층적 행동 복제(h-BC)

우리는 행동 복제를 계층적 설정으로 자연스럽게 확장하는 것을 고려합니다(알고리즘 1). 전문가 프로
계층적 데모 세트 σ 를 제공하며, 각 데모는 LO 수준 궤적 $\tau = \{(s, a)\}$HLO로 구성됩니다. $A \quad A \quad A=1$
를 HI 레벨 궤적 $\tau = \{(s, g)\}$로 정의합니다. 그런 다음 다음을 실행합니다.

알고리즘 2 계층적으로 유도된 DAgger(**hg-DAgger**)

데이터 버퍼 $\mathcal{D}_{HI} \leftarrow \emptyset$ 및 $\mathcal{D}_g \leftarrow \emptyset$ 초기화, $g \in \mathcal{G}$
계층적 동작 복제 실행(알고리즘 1)
  최대 t = 트윔 시작
for $t = T_{warm\text{-}start} + 1, \ldots, T$ **do**
시작 상태 s로새 환경 인스턴스 가져오기
  초기화 $\sigma \leftarrow \emptyset$
  6: 반복
$g \leftarrow \mu(s)$
π_g를 실행하여 LO 레벨 궤적 τ를구합니다.
σ에 (s, g, τ)를 추가합니다
$s \leftarrow \tau$의 마지막 상태
에피소드 종료 **시까지**
σ에서 τ_{FULL}과 τ_{HI}를 추출합니다.
  $\text{InspectFULL}(\tau_{FULL})$ = **실패하면**
    $\mathcal{D}^s \leftarrow \text{Label}_{HI}(\tau_{HI})$
    다음 조건에 따라 $(s_h, g_h, \tau_h) \in \sigma$를 순서대로 처리합니다.
      g_h 전문가의 선택에 동의 g^s in $\mathcal{D}^s$: h
      if $\texttt{Inspect}(\tau_h; g_h)$ = *Fail* **then**
        Append $\mathcal{D}_g$ h $\leftarrow \mathcal{D}_g$ h $\cup$ $\text{Label}_{LO}(\tau_h; g)_h$
        18: break
$\mathcal{D}_{HI} \leftarrow \mathcal{D}_{HI} \cup \mathcal{D}^s$ 추가
모든 g에대해 하위 정책 $\pi_g \leftarrow \texttt{Train}(\pi_g, \mathcal{D}_g)$을 업데이트합니다.
메타 컨트롤러 μ 업데이트 $\leftarrow \texttt{Train}(\mu, \mathcal{D}_{HI})$

훈련(8~9행)을 통해 각각 $s^{\wedge}$ 에서 $a^{\wedge}$ 를 가장 잘 예측하는 하위 정책 $\pi_{g_h}^{A}$ 와 $s^{\wedge}$ 에서 $g^{\wedge}$ 를 가장 잘 예측하는 메타 컨트롤러 μ를 찾습니다. 훈련은 일반적으로 신경망에 대한 확률적 최적화 또는 일부 배치 훈련 절차와 같은 모든 퍼베이시브 학습 서브루틴이 될 수 있습니다. 종료 함수 β_g 를 계층적 정책의 일부로 학습해야 하는 $\{h^{A}\}$ 경우, 라벨 $\omega^{\wedge}$ 는 $\tau^{\wedge} = (s^{\wedge}, a^{\wedge}, \omega^{\wedge})$의 일부로 전문가에 의해 제공될 것입니다.[5] 이 설정에서는 하위 정책 데몬스트레이션이 상태 공간의 관련 부분에서만 발생하기 때문에 계층적 안내가 자동으로 이루어집니다.

4.2. 계층적으로 유도된 DAgger(hg-DAgger)

행동 복제와 같은 수동적 IL은 학습과 실행 배포 사이의 배포 불일치로 인해 어려움을 겪습니다. 이러한 불일치는 전문가가 레이블 (Label_{FULL}) 작업을 통해 학습자의 궤적을 따라 올바른 행동을 제공하는 SEARN(Daume´ et al., 2009) 및 DAgger(Ross et al., 2011)와 같은 대화형 IL 알고리즘으로 해결할 수 있습니다. 순진한 계층적 구현은 레이블$_{HI}$과 레이블$_{LO}$ 을 통해 전체 계층적 궤적을 따라 올바른 레이블을 제공할 것입니다. 다음으로 계층적 안내를 사용하여 LO 수준의 전문가 비용을 줄이는 방법을 보여 드리겠습니다.

우리는 두 가지 HI 수준 쿼리 유형을 활용합니다: $\texttt{Inspect}_{LO}$ 와 $\texttt{Label}_{HI}$ 입니다. $\texttt{Inspect}_{LO}$ 를 사용하여 하위 작업이 성공적으로 완료되었는지 확인하고 $\texttt{Label}_{HI}$ 을 사용하여 상태 공간의 해당 부분에 머물고 있는지 확인합니다. 자세한 내용은 알고리즘 2에 나와 있습니다.

DAgger를 두 레벨의 학습자로 사용하지만, 이 방식은 다른 대화형 모방 학습자에게도 적용할 수 있습니다.

각 에피소드에서 학습자는 하위 목표 선택(7줄), LO 수준 궤적 실행(즉, 선택한 하위 목표에 대한 하위 정책 π_g 롤아웃), β_g 에 따른 실행 종료(8줄)를 포함하여 계층적 정책을 실행합니다. 전문가는 $\texttt{Inspect}_{\text{FULL}}$ (13줄)에서 확인된 대로 에이전트가 전체 작업을 실행하지 못한 경우에만 피드백을 제공합니다. 검사$_{\text{FULL}}$ 가 실패하면 전문가는 먼저 레이블$_{\text{HI}}$ (14줄)을 통해 올바른 하위 목표에 레이블을 지정하고 학습자의 메타 컨트롤러가 올바른 하위 목표 g_h (15줄)를 선택했지만 해당 하위 정책이 실패한 경우(즉, 16줄의 검사$_{\text{LO}}$ 가 실패한 경우)에만 LO 수준의 레이블을 지정합니다. 이전의 모든 하위 목표가 올바르게 선택 및 실행되었고 현재 하위 목표도 올바르므로 LO 수준 학습은 상태 공간의 "관련" 부분에 있습니다. 그러나 하위 정책 실행이 실패했기 때문에 아직 학습이 완료되지 않았습니다. 다음으로 계층적 안내로 인한 전문가 비용 절감 효과를 분석합니다.

이론적 분석. 다소 정형화된 가정 하에 플랫 DAgger와 비교하여 hg-DAgger의 비용을 분석합니다. 학습자가 어떤 정책 클래스에서 메타 컨트롤러 μ를, 어떤 클래스 Π에서 하위 폴리시 π를 학습하는 것을 목표로 한다고 가정합니다. 클래스와 Π는 유한하고(그러나 기하급수적으로 클 수도 있음), 과제는 실제화할 수 있습니다. 즉, 전문가의 정책은 연관된 클래스에서 찾을 수 있습니다: μ^*, $\pi^* \in \Pi_{\text{LO}}$, g^*. 이를 통해 두 수준 모두에서 온라인 학습자로 *반감기 알고리즘*(Shalev-Shwartz et al., 2012)을 사용할 수 있습니다. (알고리즘의 구현에는 이러한 가정이 필요하지 않습니다.)

반감 알고리즘은 폴리시에 대한 버전 공간을 유지하고 다수결에 따라 행동하며, 실수를 할 경우 버전 공간에서 모든 잘못된 정책을 제거합니다. 따라서 계층적 설정에서는 HI 수준에서 최대 $\log_2|\mathcal{M}|$ 로그의 실수를 저지르고, 각 π_g 를 학습할 때 최대 $\log_2|\Pi_{\text{LO}}|$ 로그의 실수를 저지릅니다. 실수 한계는 hg-DAgger와 플랫 DAgger 모두에서 총 전문가 비용의 상한을 설정하는 데 추가로 사용될 수 있습니다. 사과 대 사과를 활성화하려면

비교를 위해, 플랫 DAgger가 정책 클래스 $\Pi_{\text{FULL}} = \{\mu, \pi_g : \mu \in \mathcal{M}, \pi_g \in \Pi_{\text{LO}}, g \in \mathcal{G}\}$ 를 학습하지만 계층적 작업 구조에 대해서는 알지 못한다고 가정합니다.

경계는 섹션 3의 마지막에 정의된 대로 다양한 유형의 연산을 수행하는 비용에 따라 달라집니다. 먼저 $\texttt{Inspect}_{\text{FULL}}$ 를 호출하고 검사에 실패한 경우에만 레이블($\texttt{Label}_{\text{FULL}}$)을 요청하는 플랫 DAgger의 수정된 버전을 고려해 보겠습니다. 증명은 부록 A로 미루겠습니다.

정리 1. *유한 클래스와 Π_{LO} 및 실현 가능한 전문가 정책이 주어졌을 때, 라운드 T까지 전문가가 hg-DAgger에서 발생하는 총 비용은 다음과 같이 경계가 지정됩니다.*

$$C^I_{\text{전체}} \le \log_2|\mathcal{M}| + |G_{opt}|\log_2|\Pi_{\text{LO}}|\,(C^L_{\text{안내}} + H\,C^I_{\text{HI}})_{\text{LO}} + |G_{opt}|\log_2|\Pi_{\text{LO}}|\,C^L_{\text{LO}} \quad (1)$$

[5] 계층적 모방 학습 실험에서는 용어-국가 함수를 모두 학습합니다. 공식적으로 종결 신호 ω_g 는 LO 레벨에서 증강된 동작의 일부로 볼 수 있습니다.

여기서 $G_{opt} \subseteq G$*는 전문가가 실제로 사용한 하위 목표 집합이며,* $G_{opt} := \mu^{\wedge}(S)$ *입니다.*

정리 2. *전체 정책 클래스* $\Pi_{\text{FULL}} = \{\mu, \pi_{gg\in G}) : \mu, \pi_g \in \Pi_{\text{LO}}\}$ *및 실현 가능한 문제 정책이 주어질 때, 전문가가 플랫 DAg-에서 발생하는 총 비용은 다음과 같습니다.*

라운드 T*에 의한 ger는 다음과 같이 경계가 지정됩니다.*

$$TC^{I}_{\text{전체}} + \log_2 |M| + |G| \log_2 |\Pi_{\text{LO}} / C^{L}_{\text{전체}}. \quad (2)$$

두 바운드의 선행 용어는 동일한 $TC^{I}_{\text{전체}}$ 입니다.

매 라운드마다 발생하며 "모니터링 비용"으로 볼 수 있습니다. 이와 대조적으로, 두 설정에서 재사용 용어는 "학습 비용"으로 볼 수 있으며, 각각의 실수 범위에서 발생하는 용어를 포함합니다. 그러면 평면 학습에 대한 계층적 안내 학습 비용의 비율은 다음과 같이 제한됩니다.

$$\frac{\text{방정식 (1) - } TC\ C^{I}_{\text{전체}}}{\text{방정식 (2) - } TCI} \leq \frac{^{L} + H\ C_{\text{HI}}{}^{I} + C^{L}_{\text{HI}}}{C^{L}_{\text{전체}}}, \quad (3)$$

상한선을 적용한 $|G| \leq |G|$. 계층적 안내로 인한 절감 효과는 특정 비용에 따라 달라집니다. 일반적으로 최종 상태를 확인하는 것으로 충분하다면 검사 비용은 $O(1)$이 될 것으로 예상하지만, 라벨링 비용은 궤적의 길이에 따라 선형적으로 확장됩니다. 그러면 비용 비율은 $\frac{H_{\text{HI}}+H_{\text{LO}}}{H}$입니다. 따라서 각 개별 레벨의 지평선이 전체 지평선보다 하위적으로 짧을 때 가장 큰 비용 절감을 실현할 수 있습니다. 특히 다음과 같은 경우 $H_{\text{HI}} = H_{\text{LO}} = \sqrt{H_{\text{FULL}}}$, 계층적 가이드 접근 방식를 사용하면 전체 라벨 제작 비용이 $\sqrt{H}$ 만큼 감소합니다 $_{\text{FULL}}$.

일반적으로 H_{FULL} 가 클수록 학습 비용이 최소한 일정하게 감소하므로 도메인 전문가의 노력을 절약할 수 있다면 상당한 이득이 됩니다.

5. 계층적 안내 IL/RL

계층적 안내는 하이브리드 설정에서도 적용되며, 하이 레벨에는 대화형 IL이, 로우 레벨에는 RL이 있습니다. HI 레벨 전문가는 각 하위 목표에 대한 의사 보상 기능을 포함하여 계층적 분해를 제공합니다,[6] 각 단계에서 올바른 하위 목표를 선택할 수 있습니다. hg-DAgger와 유사하게, HI 레벨 전

알고리즘 3 계층적 가이드 DAgger / Q-러닝 (**hg-DAgger/Q**)

의사 보상을 제공하는 의사 함수 pseudo(s; g) **입력**
입력 술어 터미널(s), g의 종결을 나타내는 g
입력 어닐링된 탐색 확률 $_{gg} > 0$, $g \in G$ 1: 데이터 버퍼 $_{\text{DHI}} \leftarrow \emptyset$ 및 $_{Dg} \leftarrow \emptyset$ 초기화, $g \in G$ 2: 서브목표 Q-함수 $_{Qg}$ 초기화, $g \in G$
3: **for** $t = 1, \ldots, T$ **do**

4:시작 상태 s로새 환경 인스턴스 가져오기
5: 초기화 $\sigma \leftarrow \emptyset$
6: **6: 반복**
7: $_{\text{sHI}} \leftarrow s$, $g \leftarrow \mu(s)$를 초기화하고 $\tau \leftarrow \emptyset$를 초기화합니다.
8: **반복**
9: $a \leftarrow {}_{gg}$-greedy($_{Qg}$, s)
10: a, 다음 상태 $\tilde{s}$, $\tilde{r} \leftarrow$ pseudo($\tilde{s}$; g)를 실행합니다.
11: 업데이트 $_{Qg}$: (확률론적) 경사 하강 단계 $_{g}$미니 배치에서
12: τ에(s, a, $\tilde{r}$, $\tilde{s}$)를 더하고 $s \leftarrow \tilde{s}$를 업데이트합니다.
13: **때까지** 터미널(S; G)
14: *14:* σ에($_{\text{sHI}}$, g, τ)를 더합니다.
15: 에피소드 종료 **시까지**

16: *16:* σ에서 τFULL과 $_{\tau\text{HI}}$를 추출합니다.
17: 검사하면 $_{\text{전체}}$ ($_{\tau\text{FULL}}$) = *Fail* **then**

문가의 레이블은 메타 컨트롤러 μ를 훈련하는 데 사용될 뿐만 아니라 LO 레벨 학습을 상태 공간의 관련 부분으로 제한하는 데에도 사용됩니다. 알고리즘 3에서는 HI 레벨의 DAgger와 LO 레벨의 Q-러닝을 통해 세부 사항을 제공합니다. 이 방식은 다른 대화형 IL 및 RL 알고리즘에도 적용할 수 있습니다.

학습 에이전트는 메타 컨트롤러와 함께 *롤인하여* 진행합니다(7줄). 선택된 각 하위 목표 g에 대해 하위 정책 π_g 은 다음을 통해 기본 작업을 선택하고 실행합니다.

[6]이는 많은 계층적 RL 접근법, 포함 옵션(Sutton et al., 1999), MAXQ(Dietterich, 2000), UVFA(Schaul et al., 2015a), h-DQN(Kulkarni et al., 2016)과 일치합니다.

18: $D^s \leftarrow$ LabelHI(τ_{HI})
19: 다음과 같은 경우 $(s_h, g_h, \tau_h) \in \sigma$ 를 순서대로 처리합니다.
g_h 전문가의 선택에 동의 g^s in D^s : 20: 추가 $D_{g_h} \leftarrow D_{g_h} \cup \tau_h$
$D_{HI} \leftarrow D_{HI} \cup D$ 추가^s
21: **else**
22: 모든 $(s_h, g_h, \tau_h) \in \sigma$에대해 $D_{g_h} \leftarrow D_{g_h} \cup \tau_h$ 를 추가합니다.
23:메타 컨트롤러 μ 업데이트 $\leftarrow$ Train(μ, D_{HI})

욕심 규칙(9~10줄)에 따라 종료 조건이 충족될 때까지 기다립니다. 에이전트는 내재적 보상이라고도 하는 의사 보상을 받습니다(Kulkarni et al., 2016)(10줄). 하위 목표가 종료되면 에이전트의 메타 컨트롤러 μ는 다른 하위 목표를 선택하고 이 과정은 전문가의 개입이 시작되는 에피소드가 끝날 때까지 계속됩니다. hg-DAgger에서와 마찬가지로 전문가는 학습자의 전반적인 실행을 검사하고(17줄), 성공하지 못한 경우 메타 컨트롤러의 학습을 위해 축적된 HI 레벨 레이블을 제공합니다.

계층적 안내는 LO 수준의 학습자가 경험을 축적하는 방식에 영향을 미칩니다. 메타 컨트롤러의 하위 목표 g가 전문가의 하위 목표와 일치하는 한, 에이전트의 하위 목표 g 실행 경험은 경험 리플레이 버퍼($_g$)에 추가됩니다. 메타 컨트롤러가 "나쁜" 하위 목표를 선택하면 현재 에피소드의 경험 누적이 종료됩니다. 이렇게 하면 경험 버퍼에 상태 공간의 관련 부분의 데이터만 포함되도록 할 수 있습니다.

알고리즘 3은 실제 값 함수 의사(s; g)에 대한 액세스를 가정하고, 의사 보상을 실행할 때 상태 s에서 제공하고, 술어 터미널(s; g)은 하위 목표 g의 종료(반드시 성공할 필요는 없음)를 나타냅니다. 이 설정은 계층적 RL에 대한 선행 연구와 유사합니다(Kulkarni et al., 2016). 하나의 자연스러운 정의

하위 목표 g의 성공적인 완료를 나타내는 추가 술어 성공 $(s; g)$을 기반으로 한 의사 보상의 개념은 다음과 같습니다 :

$$r_{\text{pseudo}}(s, a, s', g) = \begin{cases} 1 & \text{if 성공}(S; G) \\ -1 & \text{if } \neg\text{성공}(s; g) \text{ 및 터미널}(s; g) \\ -\kappa, & \text{그렇지 않으면} \end{cases}$$

여기서 $\kappa > 0$은 작은 페널티로, 짧은 트레일을 장려합니다. 술어 성공과 종결은 전문가가 제시하거나 감독 또는 강화 피드백을 통해 학습합니다. 실험에서는 이러한 술어를 hg-DAgger/Q와 계층적 RL 모두에 명시적으로 제공하여 하위 정책을 종료할 시점을 학습해야 하는 hg-DAgger보다 유리하도록 했습니다.

6. 실험

(i) 단순하지만 도전적인 미로 탐색 도메인, (ii) 아타리 게임인 몬테주마의 복수라는 두 가지 도메인에서 알고리즘의 성능을 평가합니다.

6.1. 미로 탐색 도메인

작업 개요. 그림 1(왼쪽)은 미로 탐색 도메인의 스냅샷을 보여줍니다. 각 에피소드에서 에이전트는 다양한 레이아웃으로 구성된 대규모 컬렉션에서 미로의 새로운 인스턴스를 탐색합니다. 각 미로는 4×4 격자로 배열된 16개의 방으로 구성되지만, 에이전트와 타깃의 초기 위치와 마찬가지로 방 사이의 간격은 인스턴스마다 다릅니다. 에이전트(흰색 점)는 미로의 한 구석에서 노란색으로 표시된 타겟까지 이동해야 합니다. 빨간색 셀은 에이전트가 생존을 위해 피해야 하는 장애물(용암)입니다. 에이전트가 받는 컨텍스트 정보는 방문한 위치를 나타내는 부분적인 흔적(녹색으로 표시됨)을 포함하여 환경을 조감도로 픽셀로 표현한 것입니다.

이 도메인은 무작위 환경 인스턴스가 많기 때문에 표 형식 알고리즘으로는 해결할 수 없습니다. 방이 항상 연결되어 있는 것은 아니며 복도의 위치가 항상 벽의 중앙에 있는 것은 아니라는 점에 유의하세요. 기본 동작에는 *위*, *아래*, *왼쪽* 또는 *오른쪽으로* 한 단계 이동하는 것이 포함됩니다. 또한 환경의 각 인스턴스는 초기 위치에서 목표까지 경로가 있도록 설계되었습니다,

에이전트가 실패한 경우 추가 단계를 수행합니다. 하위 정책과 메타 컨트롤러는 유사한 신경망 아키텍처를 사용하며 동작 출력의 수만 다릅니다. (네트워크 아키텍처의 디테일은 부록 B에 나와 있습니다.)

계층적 안내 IL. 먼저 계층적 IL 알고리즘을 플랫 버전과 비교합니다. 알고리즘 성능은 훈련에 사용되지 않은 무작위 환경 인스턴스에서 이전 100개의 테스트 에피소드에 대한 평균 작업 완료율로 정의되는 성공률로 측정됩니다. 각 레이블 작업의 비용은 레이블이 지정된 궤적의 길이와 같으며, 각 검사 작업의 비용은 1입니다.

h-BC와 hg-DAgger는 모두 평면 모방 학습자보다 성능이 뛰어납니다(그림 2, 왼쪽). 특히 hg-DAgger는 1000회 미만의 에피소드에서 100%에 근접하는 등 지속적으로 가장 높은 성공률을 달성합니다. 그림 2(왼쪽)는 알고리즘을 무작위로 5회 실행한 결과의 중앙값과 최소 성공률에서 최대 성공률까지의 범위를 표시합니다.

전문가 비용은 두 계층적 알고리즘 간에 큰 차이가 있습니다. 그림 2(가운데)는 동일한 성공률을 전문가 비용의 [illegible]

[illegible]로 설정하면 최단 경로는 최소 45걸음($H_{\text{FULL}} = 100$)이 걸립니다.
에이전트가 용암에 부딪히면 보상 -1의 불이익을 받습니다
,

를 누르면 에피소드가 종료됩니다. 상담원은 노란색 블록을 밟을 때만 긍정적인 보상을 받습니다.

환경의 계층적 분해는 *북쪽*, *남쪽*, *서쪽*, *동쪽으로* 방으로 이동하는 네 가지 가능한 하위 목표와 다섯 번째 가능한 하위 목표인 *타겠으로 이동*(방 구성에서만 유효)과 상관관계가 있습니다.
타겟에 연결). 이 설정에서는 $H_{\text{LO}} \approx 5$단계, 그리고 $H_{\text{HI}} \approx 10\text{-}12$ 단계. 에피소드는 100초 후에 종료됩니다.

함수로 표시한 것입니다. hg-DAgger는 계층적 안내를 통해 LO 수준의 전문가를 보다 효율적으로 사용하기 때문에 다른 모방 학습 알고리즘에 비해 전문가 비용을 크게 절감할 수 있습니다. 그림 1(가운데)은 hg-DAgger가 학습 초기에 대부분의 LO 레벨 레이블을 필요로 하며, 하위 목표가 매스터링된 후에는 주로 HI 레벨 레이블을 요청한다는 것을 보여줍니다. 결과적으로 hg-DAgger는 플랫 DAgger에 비해 LO 레벨 레이블의 일부만 필요로 합니다(그림 2, 오른쪽).

계층적 안내 IL/RL. *딥러닝은* 딥 더블 Q-러닝(DDQN, Van Has- selt 외., 2016)과 우선순위 경험 재생(Schaul 외., 2015b)을 기본 RL 절차로 사용하여 hg- DAgger/Q를 평가합니다. 각 하위 정책 학습자는 실행에 성공할 때마다 올바른 문(예: 하위 목표가 *북쪽인* 경우 북쪽 문)을 통과하면 1의 의사 보상을 받고 용암에 들어가거나 다른 문을 통과하면 음의 보상을 받습니다.

그림 1(오른쪽)은 hg- DAgger/Q의 학습 진행 상황을 보여주며, 두 가지 주요 관찰 사항을 암시합니다. 첫째, 하이 레벨 레이블의 수는 초기에 급격히 증가하다가 학습자의 성공률이 높아진 후 평평해집니다.
풀, `Inspect` 전체 작동합니다
.

하이브리드 알고리즘이 발전하고 학습 에이전트가 $\texttt{Inspect}_{\text{FULL}}$ 작업을 점점 더 많이 통과함에 따라 알고리즘은 전문가 피드백을 크게 절약하기 시작합니다. 둘째, HI 레벨 레이블의 수가 hg-DAgger 및 h-BC보다 더 많습니다. FULL 검사 특히 훈련 초기에는 *실패를* 자주 반환합니다. 이는 주로 LO 수준에서 Q러닝의 학습 속도가 느리기 때문에 HI 수준에서 더 많은 전문가 피드백이 필요하기 때문입니다.

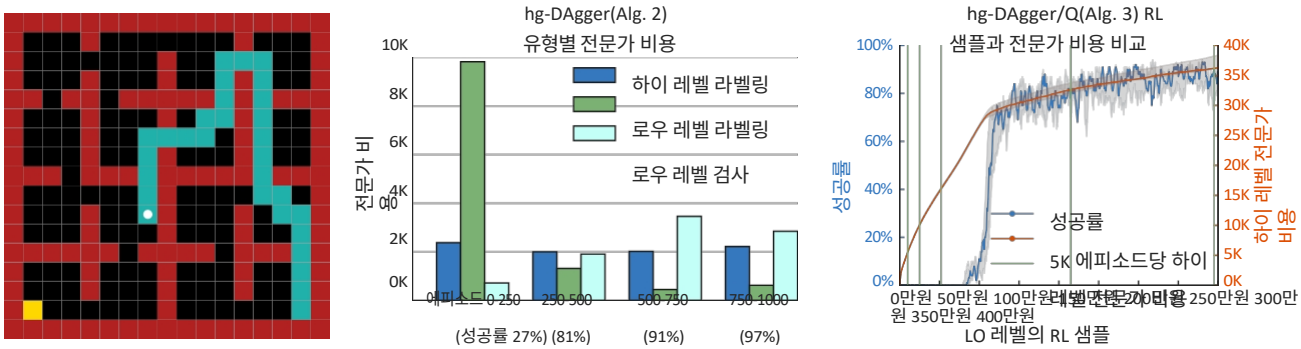

그림 1. 미로 탐색. (왼쪽) 하나의 샘플링된 환경 인스턴스, 에이전트는 오른쪽 하단에서 왼쪽 하단으로 탐색해야 합니다. *(가운데)* 시간 경과에 따른 전문가 비용, 라벨 작업 비용은 라벨링된 궤적의 길이와 같고, 검사 작업 비용은 1입니다. *(오른쪽)* LO 수준 RL 샘플 수의 함수로서 hg-DAgger/Q의 성공률 및 HI 수준 라벨 비용.

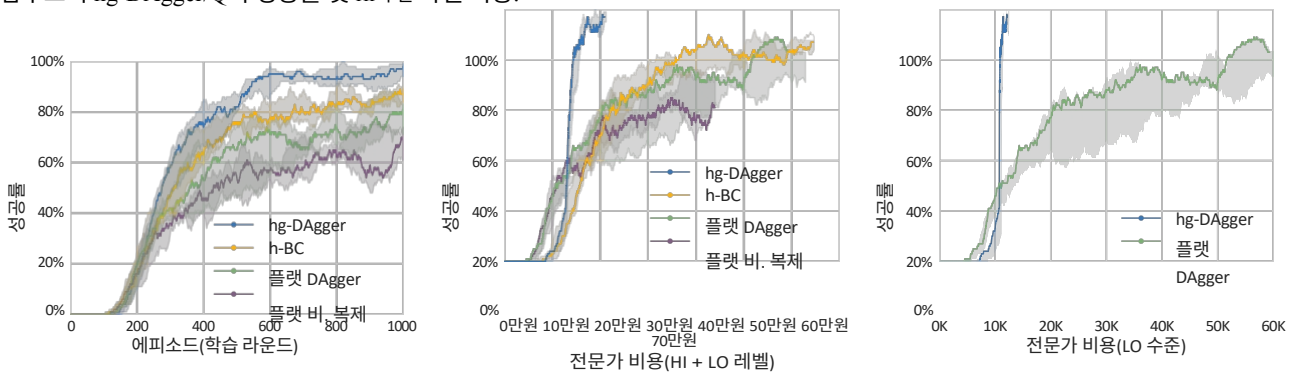

그림 2. 미로 탐색: 계층적 모방 학습과 평면 모방 학습 비교. 각 에피소드는 한 라운드의 훈련과 한 라운드의 테스트가 이어집니다. 성공률은 이전 100개의 테스트 에피소드에 대해 측정되며, 전문가 비용은 그림 1과 같습니다. *(왼쪽)* 에피소드당 성공률. *(가운데)* 전문가 비용 대비 성공률. *(오른쪽)* 성공률과 LO 수준의 전문가 비용 비교.

레벨. 즉, 하이브리드 알고리즘은 LO 수준의 전문가 레이블을 사용할 수 없거나 HI 수준의 레이블보다 더 비싼 설정에 적합합니다. 다음 섹션에서 분석할 설정이 바로 여기에 해당합니다.

부록 B.1에서는 hg-DAgger/Q와 계층적 RL(h-DQN, 쿨카니 외., 2016)을 비교하여 h-DQN이 훨씬 더 많은 LO 수준의 샘플을 사용하더라도 hg-DAgger/Q와 비슷한 성공률에 도달하지 못한다는 결론을 내렸습니다. 플랫 Q-러닝도 긴 계획 기간과 희소한 보상으로 인해 이 설정에서는 실패합니다(Mnih et al., 2015).

6.2. 계층적 유도 IL/RL과 계층적 RL: 몬테주마의 복수에 대한 비교

작업 개요. 몬테주마의 리벤지는 현존하는 딥 RL 알고리즘으로는 가장 어려운 아타리 게임 중 하나로, 하위 작업의 순차적 순서로 인해 계층적 접근 방식에 적합한 게임입니다. 그림 3(왼쪽)은 환경과 주석이 달린 하위 목표의 순서를 보여줍니다. 지정된 4개의 하위 목표는 오른쪽 계단 아래로 이동하기, 열쇠를 얻기, 경로를 반대로 돌아가서 오른쪽 계단으로 돌아가기, 문을 열기(전체적으로 장애물을 피하면서)입니다.

상담원에게는 각 하위 목표에 대해 1의 가상 보상이 주어집니다.

완료, 생명력 손실 시 -1이 됩니다. 에이전트가 에피소드당 하나의 생명력만 가질 수 있도록 하여 에이전트가 열쇠를 수집한 후 지름길을 택하는 것을 방지했습니다(자신의 생명을 빼앗고 시작 위치에서 새로운 생명력으로 다시 초기화하여 작업 지평을 효과적으로 축소). 이 설정의 경우 실제 게임 환경에는 열쇠를 줍는 것(하위 목표 2, 보상 100)과 열쇠를 사용하여 문을 여는 것(하위 목표 4, 보상 300)에 해당하는 두 가지 긍정적 외부 보상이 있습니다. 이러한 일련의 하위 목표를 최적으로 실행하려면 200개 이상의 기본 동작이 필요합니다. 당연히 플랫 RL 알고리즘은 이 영역에서 0점을 받는 경우가 많습니다(Mnih et al., 2015; 2016; Wang et al., 2016).

hg-DAgger/Q와 h-DQN 비교. 미로 영역과 유사하게, 저희는 우선순위가 지정된 경험 재생을 hg-DAgger/Q의 LO 수준에서 DDQN을 사용합니다. Kulka- rni 등(2016)과 동일한 신경망 아키텍처를 사용하여 h-DQN과 성능을 비교합니다. 그림 3(가운데)은 하이브리드 알고리즘의 학습 진행 과정을 보여줍니다. HI 수준의 수평선 $H_{\text{HI}} = 4$이므로 메타 컨트롤러는 상당히 적은 수의 샘플에서 학습됩니다. 각 에피소드는 대략 하나의 $\texttt{Label}_{\text{HI}}$ 쿼리에 해당합니다. 하위 정책은 전문가가 지정한 하위 목표 실행 순서대로 학습됩니다.

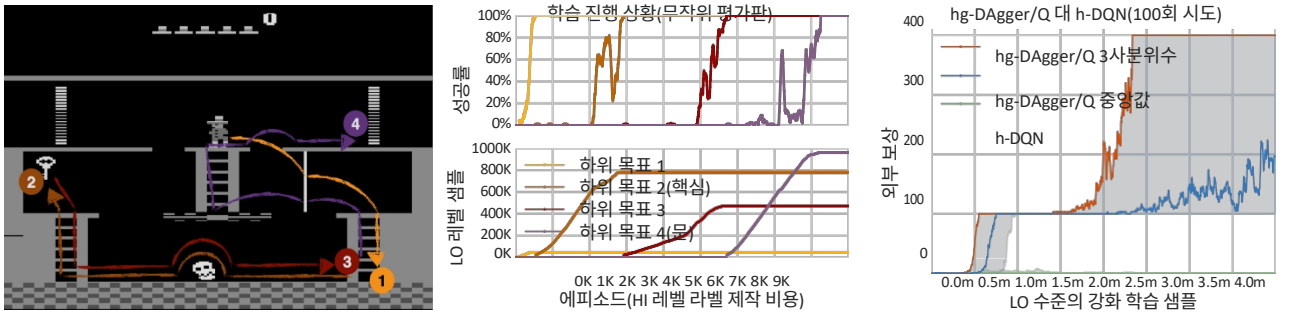

그림 3. 몬테주마의 복수: hg-DAgger/Q와 h-DQN 비교. (왼쪽) 컬러로 구분된 하위 목표가 있는 흑백의 몬테주마의 복수 스크린샷. *(가운데)* 몬테주마의 복수의 첫 번째 방을 풀었을 때 hg-DAgger/Q의 학습 진행률과 일반적인 성공 사례. 하위 목표 색상은 왼쪽 창과 일치하며, 성공률은 LO 수준의 RL 학습자가 이전 100번의 시도 중 하위 목표를 달성한 횟수의 백분율입니다. *(오른쪽)* hg-DAgger/Q 대 h-DQN(중앙값 및 사분위수 범위)의 학습 성능.

학습 속도를 높이기 위해 LO 수준에서 *Q러닝에* 간단한 수정 사항을 도입했습니다. 에이전트가 처음으로 긍정적인 의사 보상을 접할 때까지 경험 재생 버퍼의 축적이 시작되지 않습니다. 이 피리어드 동안에는 사실상 메타 컨트롤러만 학습됩니다. 이 수정은 강화 학습자가 최소한 일부 긍정적인 의사 보상에 대응하도록 보장하여 긴 기간 설정에서 학습을 향상시키며, 기본적으로 모든 정책 외 학습 체계(DQN, DDQN, 결투-DQN)와 함께 작동해야 합니다. 공정한 비교를 위해 h-DQN 학습자에게도 동일한 수정 사항을 적용했습니다(그렇지 않은 경우, h-DQN은 보상을 얻지 못했습니다).

DQN의 불안정성을 완화하기 위해(예를 들어 그림 3 가운데의 하위 목표 2와 4의 학습 진행 상황 참조), 한 가지 추가 수정 사항을 도입했습니다. 성공률이 90%를 초과하면 하위 정책의 학습을 종료하며, 이 시점에서 하위 목표를 학습한 것으로 간주합니다. 하위 목표 성공률은 이전 100번의 시도 중 하위 목표를 성공적으로 완료한 비율로 정의됩니다.

그림 3(오른쪽)은 100회 실행에 걸친 hg-DAgger/Q 및 hg-DQN의 중앙값과 사분위수 간 범위를 보여줍니다.[7] LO 수준의 샘플 크기는 여러 번의 실행에 대한 집계가 아닌 무작위로 성공한 실행에 대한 학습 진행률을 표시하는 중간 패널과 직접 비교할 수 없습니다. 모든 실험에서 모방 학습 구성 요소의 성능은 여러 번의 다양한 시도에서 안정적이었지만, 강화 학습 구성 요소의 성능은 상당히 다양했습니다. 하위 목표 4(문)는 학습 기간이 길기 때문에 가장 학습하기 어려운 반면, 하위 목표 1~3은 매우 빠르게 마스터할 수 있으며, 특히 h-DQN과 비교하면 더욱 그렇습니다. 저희 알고리즘은 계층적 안내의 이점을 활용하고 각 하위 목표가 최적 궤적의 일부인 상태 공간의 관련 부분 내에서만 각 하위 목표에 대한 경험을 축적합니다. 반면, h-DQN

[7]부록 B에서는 각 알고리즘의 최고 실행 횟수 10회, 100회 이상 시도한 하위 목표 완료율, 추가 무작위 인스턴스에 대한 그림 3(가운데)의 버전을 포함한 추가 플롯을 제공합니다.

는 잘못된 하위 목표를 선택할 수 있으며, 그 결과로 생성된 LO 레벨 샘플은 하위 목표 경험 리플레이 버퍼를 '손상'시켜 컨버전스 속도를 크게 떨어뜨립니다.[8]

그림 3(가운데)의 HI 레벨 레이블 수는 LO 레벨에서 DDQN보다 더 효율적인 RL 절차를 사용하여 더 줄일 수 있습니다. 몬테주마의 복수의 구체적인 예에서, 인간 전문가는 일련의 하위 목표를 한 번만 제공하면(간단한 하위 목표 감지기와 함께) 하이레벨 라벨링을 자동으로 수행할 수 있기 때문에 실제 인간의 노력은 훨씬 더 적습니다. 인간 전문가는 높은 수준의 의미만 이해하면 되고 게임을 플레이할 수 없어도 됩니다.

7. 결론

계층적 모방 학습과 하이브리드 모방-강화 학습에서 학습 속도를 높이고 전문가 피드백 비용을 절감하는 데 계층적 *지도* 프레임워크를 어떻게 사용할 수 있는지 보여드렸습니다.

우리의 접근 방식은 여러 가지 방식으로 확장될 수 있습니다. 우선순위 또는 그라디언트 스타일의 피드백(Fu¨rnkranz 외., 2012; Loftin 외., 2016; Christiano 외., 2017)과 같이 약한 형태의 모방 피드백을 고려할 수도 있고, 모방 학습의 산적 변형에 해당하는 에이전트 행동이 옳은지 틀린지만 말하는 더 약한 형태의 모방 피드백(Ross 외., 2011)도 고려할 수 있습니다.

하이브리드 IL/RL 접근 방식은 하위 목표가 달성되는 시점을 나타내는 하위 목표 종료 술어의 가용성에 의존했습니다. 많은 설정에서 이러한 종료 술어는 비교적 쉽게 지정할 수 있지만, 다른 설정에서는 이 술어를 학습해야 합니다. 강화 피드백을 통해 행동하는 법을 학습하면서 종료 술어를 학습하는 문제는 향후 연구를 위해 남겨두었습니다.

[8]실제로 h-DQN의 하위 목표 수를 초기 하위 목표 두 개로 줄였지만 에이전트는 여전히 두 번째 하위 목표조차 학습하지 못했습니다(자세한 내용은 부록 참조).

감사

이 작업의 대부분은 HML이 Microsoft Research에서 인턴으로 근무할 때 수행되었습니다. HML은 부분적으로 Amazon AI 펠로우십의 지원을 받기도 합니다.

참조

Abbeel, P. 및 Ng, A. Y. 구절 내 강화 학습을 통한 도제식 학습. In *ICML*, pp. ACM, 2004.

Andreas, J., Klein, D. 및 Levine, S. 모듈식 멀티태스크 정책 스케치를 사용한 강화 학습. In *ICML*, 2017.

Chang, K.-W., Krishnamurthy, A., Agarwal, A., Daume III, H., Langford, J. 선생님보다 더 잘 검색하는 법 배우기. In *ICML*, 2015.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D. 인간 선호도를 통한 심층 강화 학습. In *NIPS*, 2017.

Daume´, H., Langford, J., Marcu, D. 검색 기반 구조적 예측. *기계 학습*, 75(3):297-325, 2009.

다얀, P. 과 힌튼, G. E. 봉건 강화 학습. In *NIPS*, 1993.

디터리히, T. G. MAXQ 값 함수 분해를 사용한 계층적 강화 학습. *J. Artif. Intell. Res.(JAIR)*, 13(1):227-303, 2000.

El Asri, L., Schulz, H., Sharma, S., Zumer, J., Harris, J., Fine, E., Mehrotra, R., Suleman, K. 프레임: 목표 지향 대화 시스템에 메모리를 추가하기 위한 코어. tems. *담화와 대화에 관한 제18회 연례 SIGdial 회의록*, 207-219쪽, 2017.

과일, R. 및 라자릭, A. 옵션이 있는 mdps 의 탐색-탐색. *arXiv preprint arXiv:1703.08667*, 2017.

Fu¨rnkranz, J., Hu¨llermeier, E., Cheng, W., Park, S.-H. 선호도 기반 강화 학습: 공식적인 프레임워크와 정책 반복 알고리즘. *기계 학습*, 89(1-2):123-156, 2012.

Hausknecht, M. 및 Stone, P. 심층 강화 학습 매개 변수화된 작업 공간에서. In *ICLR*, 2016.

He, R., Brunskill, E. 및 Roy, N. Puma: 거시적 행동으로 불확실성 하에서 계획하기. In *AAAI*, 2010.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Sendonaris, A., Dulac-Arnold, G., Osband, I., Agapiou, J. 등. 데모를 통한 딥 큐러닝(Deep q-learning). In *AAAI*, 2018.

Ho, J. 및 Ermon, S. 생성적 적대적 모방 학습. In *NIPS*, 4565-4573, 2016.

쿨카르니, T. D., 나라심한, K., 사에디, A., 테넨바움, J. 계층적 심층 강화 학습: 격자형 시간 추상화 및 내재적 동기 부여. In *NIPS*, pp. 3675-3683, 2016.

Loftin, R., Peng, B., MacGlashan, J., Littman, M. L., Tay- lor, M. E., Huang, J. 및 Roberts, D. L. 인간이 제공하는 개별 피드백을 통한 학습 행동: 학습 속도를 높이기 위한 암묵적 피드백 전략 모델링. In *AAMAS*, 2016.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. 외. 심층 강화 학습을 통한 인간 수준의 제어. *Nature*, 518 (7540):529, 2015.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D. 및 Kavukcuoglu, K. 심층 강화 학습을 위한 비동기적 방법. In *ICML*, pp. 1928-1937, 2016.

Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W. 및 Abbeel, P. 데모를 통한 강화 학습의 탐색 극복. In *ICRA*, 2017.

Peng, B., Li, X., Li, L., Gao, J., Celikyilmaz, A., Lee, S., and Wong, K.-F. 계층적 심층 강화 학습을 통한 복합 작업 완료 다이어로그 정책 학습. *2017 자연어 처리의 경험적 방법에 관한 컨퍼런스 논문집*, pp. 2231-2240, 2017.

Ross, S. and Bagnell, J. A. 대화형 무후회 학습을 통한 강화 및 모방 학습. *arXiv preprint arXiv:1406.5979*, 2014.

Ross, S., Gordon, G. J. 및 Bagnell, D. 모방 학습의 감소 및 후회 없는 온라인 학습을 위한 구조화된 예측. In *AISTATS*, pp. 627-635, 2011.

Schaul, T., Horgan, D., Gregor, K. 및 Silver, D. 범용 가치 함수 근사치. *기계 학습에 관한 국제 컨퍼런스*, 1312-1320, 2015a.

Schaul, T ., Quan, J . , Antonoglou, I . 및 Silver, D. 우선 순위가 지정된 경험 재생. *arXiv preprint arXiv:1511.05952*, 2015b.

온라인 학습 및 온라인 컨벡스 최적화. *기계 학습의 기초와 동향* R, 4(2):107-194, 2012.

Sun, W., Venkatraman, A., Gordon, G. J., Boots, B. 및 Bagnell, J. A. 심하게 악화: 순차적 예측을 위한 차별적 이미테이션 학습. *arXiv 사전 인쇄물 arXiv:1703.01030*, 2017.

Sutton, R. S., Precup, D., Singh, S. P. 시간적으로 추상적인 행동에 대한 옵션 내 학습. *ICML*, 98권, 556-564쪽, 1998.

서튼, R. S., 프레컵, D., 싱, S. MDPS와 세미 MDPS 사이: 강화 학습에서 시간적 추상화를 위한 프레임워크. *인공지능*, 112(1- 2):181-211, 1999.

Syed, U. and Schapire, R. E. 도제식 학습에 대한 게임 이론적 접근. In *NIPS*, pp. 1449-1456, 2008.

Van Hasselt, H., Guez, A., Silver, D. 심층 강화 학습과 이중 Q-러닝. *AAAI*, 16권, 2094-2100쪽, 2016.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D. 및 Kavukcuoglu, K. 계층적 강화 학습을 위한 봉건 네트워크. *arXiv preprint arXiv:1703.01161*, 2017.

왕, Z., 쇨, T., 헤셀, M., 하셀, H., 란토트, M., 프레이타스, N. 심층 강화 학습을 위한 결투 네트워크 아키텍처. In *ICML*, pp. 1995-2003, 2016.

Zheng, S., Yue, Y., Lucey, P. 심층 계층적 네트워크를 이용한 장기 궤적 생성. In *NIPS*, 2016.

Ziebart, B. D., Maas, A. L., Bagnell, J. A. 및 Dey, A. K. 최대 엔트로피 역 강화 학습. In *AAAI*, 2008.

A. 증명

정리 2의 증명. 첫 번째 용어 $TC^{I}_{\text{전체}}$ 는 ob-
전문가가 각 에피소드에서 에이전트의 전반적인 행동을 검사하기 때문입니다. 에피소드에서 문제가 발생할 때마다 전문가는 전체 궤적에 레이블을 지정하여 매번 C^{L} 를 발생시킵니다. 남은 작업은 에이전트가 한 번 이상 실수를 하는 에피소드의 수를 제한하는 것입니다. 이 양은 반감기 알고리즘에 의한 총 오테이크 횟수에 의해 제한되며, 이는 최대
후보 함수(정책) 수의 로그입니다,
로그 $|\Pi_{\text{FULL}}| = \text{로그}|\mathbb{M}|\,|\mathbb{G}| = \text{로그}|\mathbb{M}|+|\mathbb{G}|$ 로그 $|\Pi_{\text{LO}}|$.
이것으로 증명이 완료되었 □
습니다.

정리 1의 증명. 정리 2의 증명과 유사하게
첫 번째 용어 $TC^{I}_{\text{전체}}$ 는 분명합니다. 두 번째 용어는 다음과 같습니다.
를 $\texttt{Inspect}_{\text{FULL}}$ 가 문제를 발견하는 상황으로 설정합니다. 그런 다음 알고리즘 2에 따라 전문가는 하위 목표에 레이블을 지정하고
는 또한 각 하위 목표가 성공적으로 달성되었는지 검사하며, 매번 $C^{L} + H\,C^{I}_{\text{HI}}$ 비용이 발생합니다. The
이 상황이 발생하는 횟수는 다음과 같이 제한됩니다.
(a) 잘못된 하위 목표가 선택된 횟수와 (b) 모든 하위 목표는 양호하지만 하위 정책 중 하나 이상이 하위 목표 달성에 실패한 횟수를 더한 값입니다. 상황 (a)는 $|\mathbb{M}|$ 대부분의 로그 시간에 발생합니다. 상황 (b)에서 에피소드에서 선택한 하위 목표는 $_{\text{opt}}$ 에서 가져와야 하며, 이러한 각 하위 목표에 대해 반감 알고리즘은 최대 로그 Π_{LO} 실수를 합니다. 마지막 항은 $\texttt{Label}_{\text{LO}}$ 연산 비용에 대응합니다. 이는 메타 컨트롤러가 올바른 하위 목표를 선택했지만 해당 하위 정책이 실패한 경우에만 발생합니다. 이전 분석과 유사하게
의 경우, 이 상황은 각 "좋은" 하위 목표($g \in \mathrm{G}_{\text{opt}}$)에 대해 최대 로그 $|\Pi_{\text{LO}}|$에서 발생합니다. 이것으로 증명이 완료되었습니다.

B. 추가 실험 세부 정보

실험에서 성공률과 *외부 보상은* 이전 100회의 훈련에 대한 후행 평균으로 보고됩니다. 미로 탐색 영역의 계층적 모방 학습 실험의 경우, 성공률은 훈련에 사용되지 않은 별도

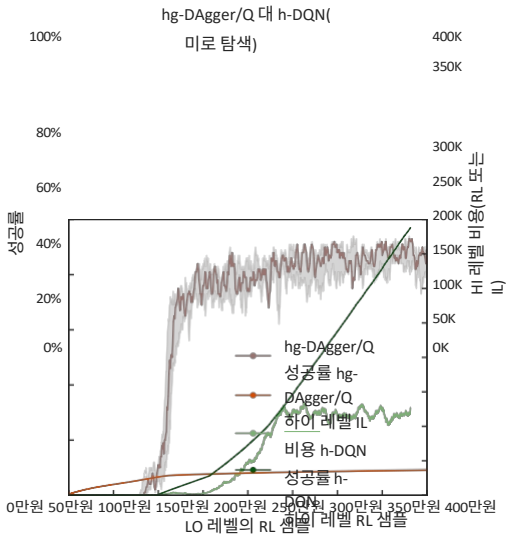

그림 4. 미로 탐색: 하이브리드 IL-RL(전체 작업) vs. DQN(50% 선취점).

몬테주마의 복수 도메인과 마찬가지로, 미로 도메인에서는 h-DQN이 잘 작동하지 않습니다. HI 수준에서는 10-12의 계획 기간과 4-5개의 가능한 하위 목표가 있습니다.

의 테스트 환경에서만 측정됩니다.

이 섹션에서는 실험 결과와 더불어 몬테주마의 리벤지에 대한 하위 목표 탐지/단말 예측 메커니즘과 미로 탐색 환경이 생성되는 방식에 대해 설명합니다. 실험에서 사용한 네트워크 아키텍처는 표 1과 표 2에 나와 있습니다.

B.1. 미로 탐색 도메인

메타 컨트롤러 및 하위 정책에 대한 네트워크 아키텍처가 hg-DAgger/Q와 동일하고 Q 학습 절차가 유사하게 강화된 계층적 강화 학습 기준선(h-DQN, Kulkarni et al., 2016)과 hg-DAgger/Q를 비교합니다.

각 단계는 HI 수준의 강화 학습자에게는 엄청나게 어려우며, 어떤 실험에서도 0이 아닌 리와드를 달성할 수 없었습니다. 비교를 위해 계층적 학습자에게 궤적의 전반부를 최적으로 실행하도록 함으로써 h-DQN 알고리즘에 추가적인 이점을 제공하려고 시도하여 수평선을 50% 줄인 상태에서 h-DQN을 실행했습니다. 그 결과 성공률은 그림 4에 나와 있습니다. 하이브리드 IL-RL은 50%의 이점을 얻지 못하지만, 여전히 30%의 성공률로 평탄해지는 h-DQN을 빠르게 능가한다는 것을 알 수 있습니다.

B.1.1. 미로 탐색 환경 만들기

2000개의 미로 탐색 환경을 생성하고, 이 중 1000개는 훈련에, 1000개의 맵은 테스트에 사용합니다. 미로 탐색 비교 결과(예: 그림 2)는 모두 1000개의 테스트 맵 중 무작위로 선택된 환경을 기반으로 합니다. 생성된 환경의 추가 테스트 사례는 그림 5를 참조하세요. 각 맵(환경 인스턴스)은 17개의 그리드로 시작하여 4개의 방 구조로 나뉩니다. 처음에는 방과 방 사이에는 문이 존재하지 않습니다. 미로 탐색 환경의 인스턴스를 생성하기 위해 목표 블록(노란색)과 시작 위치가 무작위로 선택됩니다(동일하지 않은 경우 허용됨). 다음으로, 두 개의 다른 방을 구분하는 벽을 무작위로 선택하고 이 벽을 따라 무작위로 빨간색 블록(용암)을 문(검은색 셀)으로 대체합니다. 이 과정은 두 가지 조건이 충족될 때까지 계속됩니다:

- 시작 위치와 목표 블록(노란색) 사이에 가능한 경로가 있습니다.
- 출발지에서 목적지까지의 최소 거리는 40보 이상이어야 합니다. 최적의 경로는 다음을 사용하여 구성할 수 있습니다.

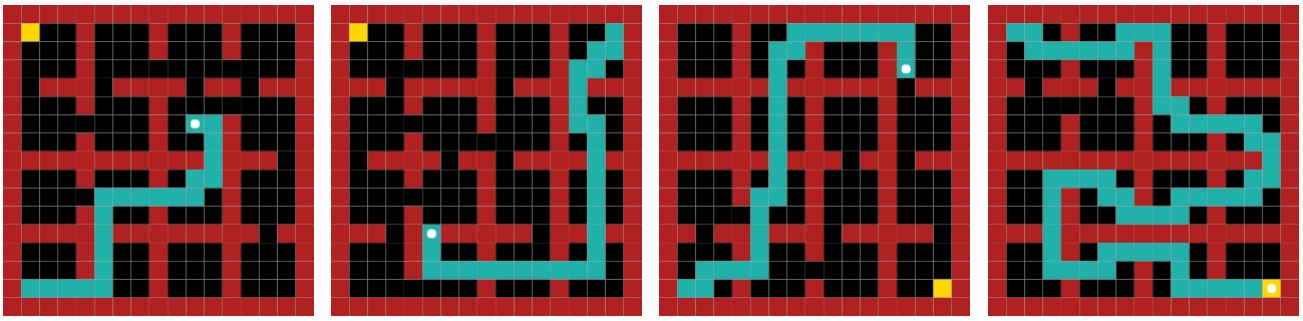

그림 5. 미로 탐색. 미로 도메인의 무작위 샘플 인스턴스(본문과는 다름). 미로의 17×17픽셀 표현은 신경망 정책의 입력으로 사용됩니다.

그래프 검색

생성되는 2000개의 환경은 각각 두 가지 조건을 모두 충족해야 합니다. 각 환경에 대한 전문가 레이블은 값 반복을 통해 계산된 최적의 정책(주어진 그리드 월드의 표 형식 표현을 기반으로 빠르게 계산됨)에서 비롯됩니다.

B.1.2. 미로 탐색을 위한 하이퍼파라미터

미로 탐색에 사용되는 네트워크 아키텍처는 표 1에 설명되어 있습니다. 하위 목표 정책 네트워크와 메타컨트롤러 네트워크의 유일한 차이점은 출력 클래스의 수입니다(4개의 액션 대 5개의 하위 목표). 계층적 모방 학습 알고리즘의 경우, 하위 목표 분류를 위해 각 하위 목표 정책을 따라 작은 네트워크를 유지합니다(하위 목표 분류기를 하위 목표 정책 네트워크의 추가 헤드로 볼 수도 있습니다).

정책 네트워크에 대한 컨텍스트 입력(상태)은 미로 환경의 3채널 픽셀 표현으로 구성됩니다. 목표 블록, 에이전트 양이온, 에이전트의 흔적 및 용암 블록에 서로 다른(고정된) 값을 할당합니다. 계층적 모방 학습 구현에서 기본 정책 학습자(DAgger 및 행동 복제)는 확률적 최적화를 사용하여 100단계마다 정책을 업데이트합니다. 아담 옵타이마이저와 0.0005의 학습 속도를 사용합니다.

표 1. 네트워크 아키텍처-미로 도메인

1: 컨볼루션	레이어32 필터, 커널 크기 3, 스트래이드 1
2: 컨볼루션 레이어	32개필터, 커널 크기 3, 스트레이드3: 최대 풀링 레이어풀 크기2
4: 컨볼루션	레이어64 필터, 커널 크기 3, 스트레이드 1
5: 컨볼루션 레이어	64개필터, 커널 크기 3, 스트레이드6: 최대 풀링 레이어풀 크기2
7: 완전 연결 레이어	256개노드, relu활성화 8:
출력 레이어	소프트맥스 활성화 (하위 정책의 경우 차원 4, 메타 컨트롤러의 경우 차원 5)

B.2. 몬테주마의 복수

모방 학습 구성 요소는 안정적이고 일관된 경향이 있지만, 강화 학습에 필요한 샘플은 동일한 하이퍼파라미터를 사용하는 실험마다 다를 수 있습니다. 이 섹션에서는 몬테주마의 복수 도메인에 대한 하이브리드 알고리즘의 추가 결과를 보고합니다.

몬테주마의 복수라는 게임에서 하이브리드 알고리즘을 구현하기 위해 LO 수준의 강화 학습자에 대해 4백만 프레임(4개의 하위 정책 모두 합산)으로 통신을 제한하기로 결정했습니다. 100번의 실험 중 81번이 처음 3개의 하위 정책을 성공적으로 학습했고, 100번 중 89번이 처음 2개의 하위 정책을 성공적으로 학습했습니다. 마지막 하위 목표(계단 아래에서 문을 열기)가 가장 어려웠으며, 실험의 거의 절반이 4백만 프레임 제한 내에 네 번째 하위 정책 학습을 완료하지 못했습니다(그림 7 가운데 창 참조). 그 이유는 주로 다른 세 가지 하위 목표에 비해 4번째 하위 목표의 학습 기간이 더 길기 때문입니다. 물론 이것은 하위 목표 설계의 함수이며 중간 하위 목표를 도입하여 언제든지 기간을 단축할 수 있습니다.

그러나 h-DQN 기준선을 2개의 하위 목표(키 획득까지)로만 제한하더라도 일반적으로 h-DQN 기준선은 제안한 하이브리드 알고리즘보다 큰 폭으로 성능이 저하되는 경향이 있다는 점을 지적할 필요가 있습니다. h-DQN 구현에 주어진 이점이 있음에도 불구하고 모든 h-DQN 실험은 두 번째 하위 목표(키 획득)를 성공적으로 마스터하는 데 실패했습니다. 키 획득(첫 번째 긍정적 외부 보상, 그림 7 오른쪽 창 참조)과 관련된 샘플 복잡성을 살펴보는 것도 도움이 됩니다. 여기에서는 하이 레벨에서 전문가 피드백을 받는 것과 강화 학습에만 의존하여 메타 컨트롤러를 훈련하는 것의 차이를 평가하기에 충분히 짧은 기간입니다.

학습 성과의 현저한 차이(그림 7 오른쪽 참조)는 HI 수준의 전문가 조언이 LO 수준의 강화 학습자가 다음을 효과적으로 방지한다는 사실에서 비롯됩니다.

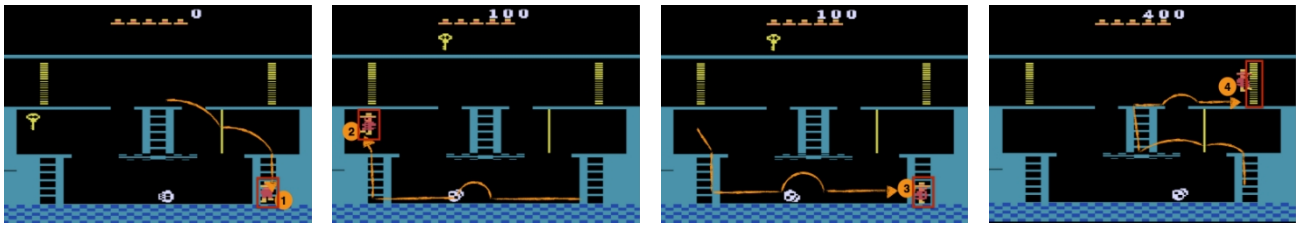

그림 6. 몬테주마의 복수: 4개의 하위 목표가 순차적으로 지정된 환경 스크린샷.

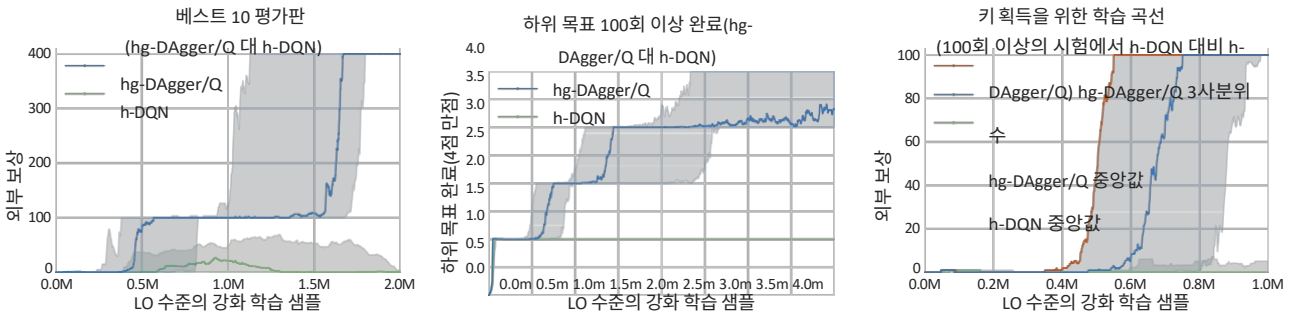

그림 7. 몬테주마의 복수: 하이브리드 IL-RL과 계층적 RL 비교. (왼쪽) 가장 좋은 10번의 실험에 대한 보상 평균, 최소 및 최대. 에이전트가 2백만 개 미만의 샘플로 첫 번째 방을 완료합니다. 음영 처리된 영역은 가장 좋은 10번의 시도 중 최소값과 최대값에 해당합니다. *(가운데)* 100번의 시도에서 하위 목표 완료율의 중앙값, 1사분위수 및 3사분위수. 음영 처리된 영역은 1사분위수 및 3사분위수에 해당합니다. *(오른쪽)* 100번의 시도에서 보상의 중앙값, 첫 번째 및 세 번째 사분위수. 음영 처리된 영역은 첫 번째 및 세 번째 사분위수에 해당합니다. h-DQN은 학습 과제를 단순화하기 위해 처음 두 개의 하위 목표만 고려합니다.

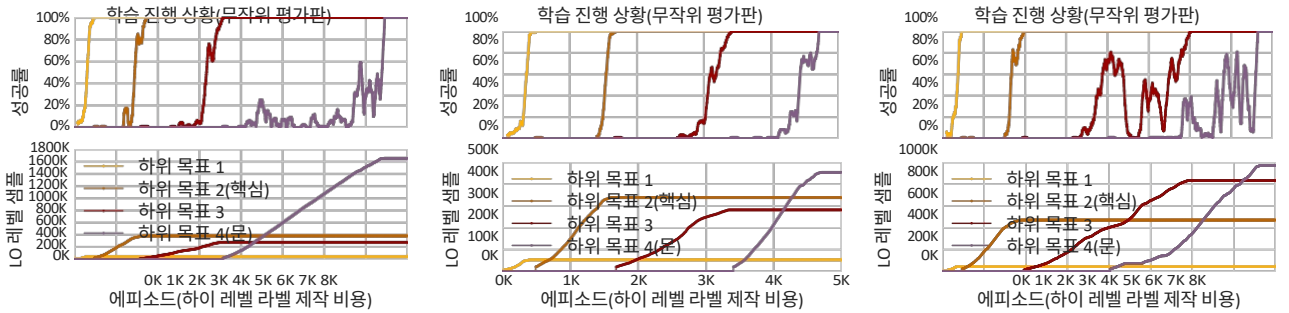

그림 8. 몬테주마의 복수: 첫 번째 방 전체를 푸는 알고리즘 3의 학습 진행 상황. 그림은 무작위로 선택된 세 번의 성공적인 시도를 보여줍니다.

나쁜 경험을 축적하는 경우가 많으며, 이는 h-DQN에서 자주 발생합니다. 경험 재생 버퍼의 잠재적 손상은 또한 우리가 고려한 환경에서 계층적 DQN을 사용한 학습이 플랫 DQN 학습에 비해 쉽지 않다는 것을 의미합니다. 따라서 계층적 DQN은 플랫 학습 버전으로 축소되기 쉽습니다.

B.2.1. 몬테주마의 복수를 위한 서브 골 탐지기

원칙적으로 시스템 설계자는 피드백을 제공하기에 가장 편리한 계층적 분해를 선택합니다. 몬테주마의 리벤지에서는 전문가의 피드백을 쉽게 받을 수 있도록 4개의 하위 목표와 자동 감지기를 설정했습니다. 하위 목표는 다음과 같이 설명되는 랜드마크입니다.

작은 직사각형. 예를 들어, 문 하위 목표(하위 목표 4)는 오른쪽 문 주위의 픽셀 패치로 표시됩니다(그림 6 오른쪽 참조). 이 하위 목표의 정확한 용어/국가/달성도를 감지하려면 미리 지정된 상자 안의 픽셀 수에서 값이 변경된 픽셀 수를 세기만 하면 됩니다. 특히 우리의 경우, 랜드마크의 검출기 상자에서 최소 30%의 픽셀이 변경되면 하위 목표 달성을 감지합니다.

B.2.2. 몬테주마의 복수를 위한 하이퍼파라미터

사용된 신경망 아키텍처는 (Kulkarni et al., 2016)과 유사합니다. 한 가지 차이점은 메인 목표 정책 대신 각 하위 목표 정책에 대해 별도의 신경망을 훈련한다는 것입니다.

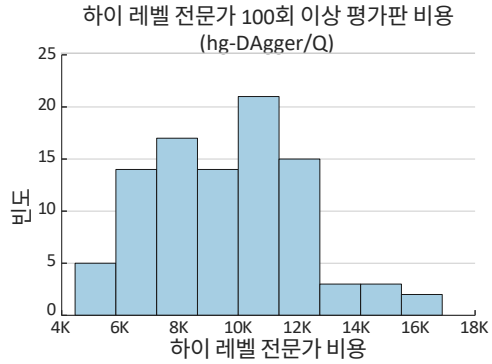

그림 9. 몬테주마의 복수: 하이레벨 전문가 라벨 수. 100개의 실험에 필요한 하이레벨 전문가 라벨의 분포, 히스토그램은 시각화를 쉽게 하기 위해 라벨 수가 20,000개를 초과하는 6개의 이상값을 제외했습니다.

표 2. 네트워크 아키텍처-몬테즈마의 복수 1: Conv.

	Layer32 필터, 커널 크기 8, 스트레
이드 4, relu	
2: Conv.	Layer64 필터, 커널 크기 4, 스트라이드
2, relu	
3: Conv.	Layer64 필터, 커널 크기 3, 스트라이드
1, relu	
4: 완전	연결됨512 노드, relu,
	표준 0.01을 사용한 레이어 노멀 초기화
5: 출력	레이어선형(하위 정책의 경우 차원 8,
	메타 컨트롤러의 경우 차원 4)

여러 하위 목표에 대한 표현을 공동으로 공유하는 다층 신경망에 대한 입력의 일부로 하위 목표 인코딩을 확보합니다. 경험적으로 여러 하위 목표에 걸쳐 표현을 공유하면 학습된 하위 목표에서 다음 하위 목표로 이동할 때 정책 성능이 저하됩니다(딥 러닝 문학에서 치명적인 망각 현상이라고도 함). 각 하위 목표에 대해 별도의 신경망을 유지하면 하위 목표 순서 전반에 걸쳐 안정적인 성능을 보장할 수 있습니다. 메타컨트롤러 정책 네트워크도 비슷한 아키텍처를 가지고 있습니다. 유일한 차이점은 출력의 수입니다(메타컨트롤러의 경우 4개의 출력 클래스, 각 LO 수준 정책의 경우 8개의 클래스(액션)).

Q러닝으로 LO 수준 정책을 훈련하기 위해 우선순위가 지정된 경험 재생(Schaul et al., 2015b)*이 포함된* DDQN(Van Hasselt et al., 2016)을 사용합니다(우선순위 *지정* 요소 $\alpha = 0.6$, 중요도 샘플링 지수 $\beta_0 = 0.4$). 아타리 게임에 적용된 이전의 딥 강화 학습 작업과 유사하게, 컨텍스트 입력(상태)은 4개의 연속 프레임으로 구성되며, 각 프레임은 회색조로 변환되어 84픽셀 크기로 축소됩니다. 아케이드 학습 환경의 일부인 프레임 건너뛰기 매개변수는 기본값인 4로 설정되어 있습니다. 반복 동작 확률은 0으로 설정되어 있으므로 아타리 환경은 대부분 결정론적입니다. 경험 메모리 용량은 500K입니다. 목표 순

Q러닝에 사용되는 작업은 2000단계마다 업데이트됩니다. 확률론적 최적화의 경우 학습률이 0.0001이고 미니 배치 크기가 128인 rmsProp을 사용합니다.

C. 추가 관련 작업

모방 학습. 강화 학습과 마찬가지로 모방 학습의 또 다른 이분법은 가치 함수 학습과 정책 학습의 이분법입니다. 가치함수 학습은 미지의 가치 함수를 최대화함으로써 최적의 (입증된) 행동을 유도하는 것으로 가정합니다 (Abbeel & Ng, 2004; Ziebart et al., 2008). 그런 다음 목표는 정책 클래스에 특정 구조를 부과하는 해당 가치 함수를 학습하는 것입니다. 후자의 설정(Daume´ et al., 2009; Ross et al., 2011; Ho & Ermon, 2016)은 이러한 구조적 가정을 하지 않으며, 의사 결정이 데모를 잘 모방하는 정책을 직접적으로 맞추는 것을 목표로 합니다. 후자의 설정은 일반적으로 더 일반적이지만 표본의 복잡성이 높을 때 더 적합합니다. Facebook의 접근 방식은 이러한 이분법에 구애받지 않으며 두 가지 학습 스타일을 모두 수용할 수 있습니다. 프레임워크의 일부 인스턴스에서는 정책 학습 설정에 의존하는 이론적 보증을 도출할 수 있습니다. 모방 학습과 강화 학습 간의 샘플 복잡도 비교는 최근 AggreVaTeD의 분석을 제외하고는 문헌에서 많이 연구되지 않았습니다(Sun et al., 2017).

계층적 강화 학습. 봉건적 강화학습은 우리가 과제를 계층적으로 구성하는 방식과 유사한 또 다른 계층적 프레임워크입니다(Dayan & Hinton, 1993; Dietterich, 2000; Vezhnevets et al., 2017). 특히 봉건적 시스템에는 관리자(우리의 하이 레벨 학습자와 유사)와 여러 명의 하위 관리자(우리의 로 레벨 학습자와 유사)가 있으며, 하위 관리자에게는 하위 목표를 정의하는 의사 보상이 주어집니다. 봉건적 RL의 기존 연구에서는 두 수준 모두에 재인포메이션 학습을 사용했는데, 이는 수준 중 하나가 긴 계획 기간을 갖는 경우 많은 양의 데이터를 필요로 할 수 있으며, 이는 실험에서 입증되었습니다. 반면, 적절한 수준의 전문가 피드백만 있다면 모방 학습자가 강화 학습을 대체하여 학습 속도를 크게 높일 수 있는 보다 일반적인 프레임워크를 제안합니다. 계층적 정책 클래스에 대한 추가 연구는 He 등(2010), Hausknecht & Stone(2016), Zheng 등(2016), An-dreas 등(2017)에 의해 수행되었습니다.

약한 피드백을 통한 학습. 우리의 작업은 약한 전문가 피드백 하에서 효율적인 학습에 동기를 부여합니다. 높은 수준의 데모 데이터만 받고 낮은 수준에서 강화 학습을 활용해야 하는 경우, 우리의 설정은 약한 데모 피드백 하에서 학습하는 사례로 볼 수 있습니다. 약한 데모 피드백을 유도하는 다른 주요 방법은 선호도 기반 또는 기울기 기반 학습으로, Fu¨rnkranz 외(2012), Loftin 외(2016), Christiano 외(2017)가 연구한 바 있습니다.