의사결정 트랜스포머를 사용한 추상화와 추론

박재현[1O] 임재균[1] 이영도[3] 신동현[1] 김세진[2] 김선동[2ψ]
광주과학기술원 전자전기컴퓨터공학부[1] 광주과학기술원 AI 대학원[2] 한국과학기술원[3]
jaehyun00518@gmail.com, jaegyun999@gmail.com, leeyoungdo20876@gmail.com,
shindong97411@gmail.com, sjkim7822@gmail.com, sdkim0211@gmail.com

Abstraction and Reasoning Challenge with Decision Transformer

Jaehyun Park[1O], Jaegyun Im[1], Youngdo Lee[3], Donghyeon Shin[2], Sejin Kim[2], Sundong Kim[2ψ]
GIST EECS[1] GIST AI[2] KAIST[3]

요 약

Abstraction and Reasoning Challenge (ARC)는 범용적 인공지능 개발을 위한 중요한 과제 중 하나이다. ARC 문제는 인간에게는 쉬운 문제이지만 인공지능 모델의 경우 상당히 낮은 정확도를 보여준다. 본 연구는 인간의 풀이 과정을 따라 하는 모방 학습 접근법을 이용하고자 Decision Transformer (DT)를 사용했고, 학습된 모델이 ARC의 특정 문제에 대해서 높은 성능을 보이는 것을 실험을 통해 보여주었다. 그러나 DT 기반 모델은 충분한 양의 학습 데이터가 필요하다는 한계가 있으며, 향후 이를 보완하기 위해 Prompt Decision Transformer와 같은 방법이 도움이 될 것으로 예상된다. 또한, Domain Specific Language (DSL)의 충분한 확보와 함축된 표현의 개발 역시 도움이 될 것으로 예상된다.

1. 서 론

현재 인공지능은 이미지 인식, 음성 인식, 자연어 처리 등 다양한 분야에서 인간을 능가하는 성능을 보여주고 있다. 특히 딥러닝의 등장 이후 인공지능 분야는 급속한 발전을 이루었으나, 대부분의 인공지능은 학습 데이터셋에 성능이 국한되거나, 학습 데이터에 지나치게 의존하는 공통적인 약점이 나타난다. 이러한 문제점을 해결하기 위해, 2020년 Abstraction and Reasoning Challenge (ARC) [1] 가 개최되었다.

ARC 대회에서 범용적 인공지능을 평가하기 위한 데이터셋을 공개했다. 이 ARC 데이터셋은 인간의 경우 대부분 간단하게 해결할 수 있는 다양한 문제로 이루어져 있다. 하지만 딥러닝을 비롯한 기존의 인공지능 기술들은 ARC 데이터셋의 문제들을 정확하게 해결하지 못했다 [2].

ARC 데이터셋은 총 400개의 학습 데이터, 200개의 검증 데이터, 그리고 200개의 비공개 평가 데이터로 이루어져 있다. 그림 1은 ARC 데이터셋의 문제 중 하나의 예시를 보여주고 있다. 다른 문제들 또한 그림 1과 같이 입력 격자와 출력 격자로 이루어진 2~4개의 학습 예시와 입력 격자로만 이루어진 평가 예시로 이루어져 있다. 모델은 각 문제마다 학습 예시를 통해 문제의 목적을 파악해서 평가 예시의 출력 격자를 올바르게 생성해야 한다.

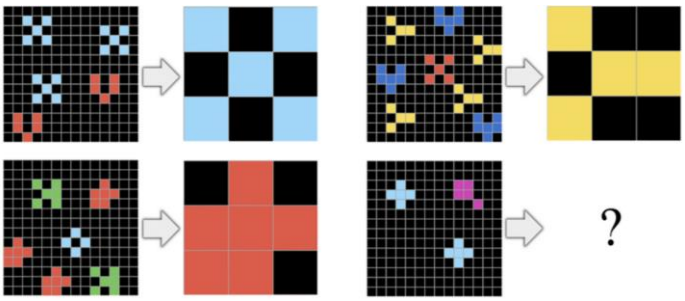

그림 1 ARC 문제 예시

기존의 ARC 데이터셋은 문제에 따라 입력 격자와 출력 격자의 크기가 일정하지 않아 모델의 학습에 어려움이 있다. 그 대신, 문제의 다양성은 유지하되 5×5 크기의 일정한 격자 크기를 가지는 Mini-ARC 데이터셋 [3]이 제안됐고, 사람의 문제 풀이 과정을 수집할 수 있는 O2ARC tool [3]이 함께 개발됐다. 본 연구는 인공지능 모델이 인간의 사고 과정을 학습하기 위해, O2ARC tool로 수집한 Mini-ARC 데이터셋의 풀이 과정을 학습 데이터셋으로 사용했다.

ARC 문제는 모델의 객체 및 패턴 탐지 능력, 객체들 사이의 관계를 파악하는 능력 등 일반적이고 폭넓은

[1] 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (RS-2023-00216011, 사람처럼 개념적으로 이해/추론이 가능한 복합인공지능 원천기술 연구)

사전지식을 요구하고 있다. 따라서 ARC 문제를 풀기 위해서는 인간처럼 생각하고 학습하며 이를 응용할 수 있는 범용적 인공지능을 개발하는 것에 집중할 필요가 있다. 현재까지도 범용적 인공지능을 개발하기 위한 다양한 방법들이 연구되고 있는데, 최근에는 오프라인 강화학습(offline reinforcement learning) 기법 중 하나인 Decision Transformer (DT) [4] 가 주목받고 있다. 온라인 강화학습은 에이전트가 직접 환경과 상호작용하면서 보상을 최대화하는 방법을 탐색하는 방법이다. 이와 다르게 오프라인 강화학습은 사전에 수집된 데이터셋을 바탕으로 각각의 상태(state)에 대해서 어떤 행동(action)을 선택해야 하는지를 학습하는 강화학습의 한 방법론이다. DT는 자연어 처리 분야에서 최고 성능으로 인정받는 모델로서, 의사결정 과정에 활용되며, 전문가의 경로를 학습하는 과정을 통해 기존의 오프라인 강화학습 최고 성능 모델의 성능과 일치하거나 이를 능가하는 결과를 보여준다 [4]. 이러한 오프라인 데이터셋을 이용한 학습 방법은 프로게이머와 같은 전문가의 기록을 학습하여 사람과 유사한 결정을 하도록 학습 할 수 있다.

2. 모델 설명

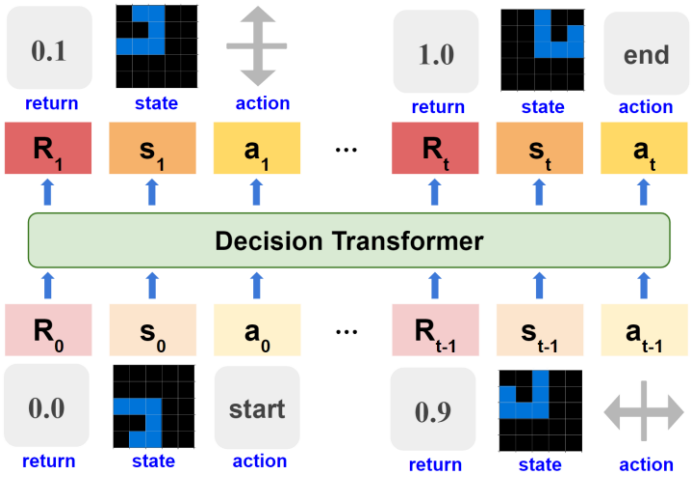

그림 2 Mini-ARC 문제를 적용한 Decision Transformer

DT 모델에서는 각 시간 단계에 해당하는 return-to-go, state, action 3가지를 입력 값으로 사용한다 [4]. Return-to-go는 시작 state를 0, 종료 state를 1로 설정한 후 각 state 사이의 간격을 등간격으로 분할하여 사용했다. State는 Mini-ARC 데이터셋의 한 픽셀 당 0~9까지의 색을 가지는 5×5 크기의 입력 격자를 사용했다. Action의 경우 O2ARC tool에서 제공하는 시계방향 회전, 색칠하기, 좌우 반전, 상하 반전 등의 14개의 기능들을 action으로 정의하여 사용했다. 이 3가지 입력 값은 모두 각각의 임베딩 층을 통해 임베딩 벡터로 변환하여 사용됐다.

ARC 문제에서는 전체 이미지를 보는 것 보다 픽셀 간의 관계를 파악하는 것이 더 중요하다. 따라서 state를 단일 임베딩 벡터가 아닌 ViT [5] 와 유사하게 픽셀 단위의 벡터로 생성하고 입력한다. 그림 2는 ARC 문제에 적합하게 변형된 DT 모델의 구조이다. 모델은 입력 값에 기반하여, 다음 단계의 return-to-go, state, action을 예측하며 전체 풀이 과정을 예상할 수 있다.

3. 실험 설정

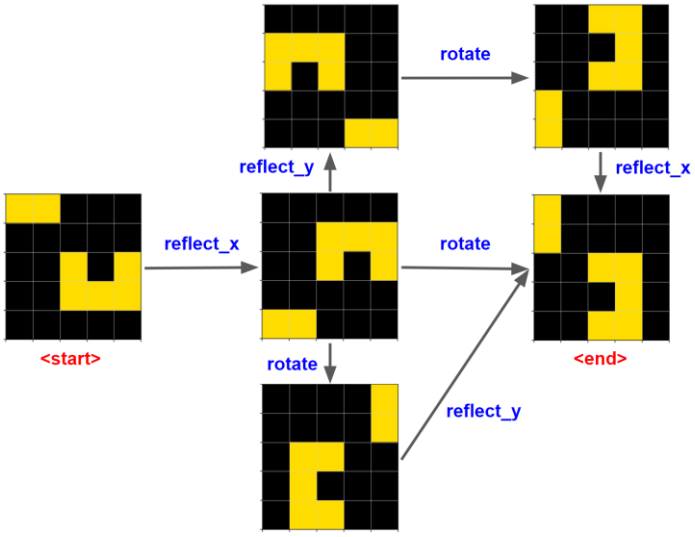

그림 3 동일한 입력 값에 대한 풀이 과정 그래프

DT 모델을 학습하기 위해선 다양한 종류의 입력 격자와 풀이 방법이 필요하다. Mini-ARC tool으로 수집한 데이터셋은 문제당 1개의 입력 격자에 대한 풀이 과정만 존재한다. 따라서 인간의 풀이 과정을 유지하면서, 임의의 입력 격자를 사용하여 데이터를 증강했다. 인간의 풀이 과정은 O2ARC tool을 통해 전문가(expert) 풀이 과정에 해당하는 최적의 풀이 방법 3~10가지를 수집했다. 이후 랜덤하게 생성된 입력 격자와 인간의 풀이 과정을 조합하여 다양한 예시들을 생성했다. 전체 풀이 과정은 JSON 파일로 저장하여 데이터셋을 구성했다. 그림 3은 5×5 크기의 ARC 문제 중 하나인 대각선 뒤집기 문제의 다양한 해결 방법을 보여준다.

학습 데이터셋은 문제마다 10,000개의 풀이 과정 전체가 포함되어 있으며, 평가 데이터셋은 2,000개의 입력과 정답 쌍으로 구성되어 있다. 학습 과정에서는 풀이 과정을 5개의 시간 단계로 잘라 학습하며 입력 값의 길이가 부족한 경우, 시작 격자와 동일한 값으로 패딩하여 데이터의 길이를 동등하게 조정했다. State와 action의 경우 cross-entropy loss를 사용했다. Return-to-go의 경우 0부터 1 사이의 연속적인 실숫값을 가지므로 MSE loss를 사용했다.

평가 단계에서는 처음 입력 값만 주어진다. 이후 다음 시간 단계에 해당하는 return-to-go, state, action을 생성하는 과정을 “end” action이 나올 때까지 반복했다.

생성 과정이 일정 시간 단계 이상 진행되는 경우, 모델이 문제를 해결하지 못했다고 판단하여 정답 예측을 중단 한다. 마지막으로, 모델의 최종 결과를 기반으로 정답과 비교하여 정확도를 계산했다.

4. 결과

대각선 뒤집기 문제와 테트리스 문제에 대한 2,000개의 평가 데이터셋을 통해 모델의 문제 해결 과정을 분석했다. 그림 4는 대각선 뒤집기 문제, 그림 5는 테트리스 문제에 대한 DT의 결과이다. 같은 대각선 뒤집기 문제를 입력하더라도 DT는 고유한 풀이 방법을 찾아가므로, 그림 4의 두 풀이 과정에서 서로 다른 action으로 문제를 해결하는 것을 확인할 수 있다. 대각선 뒤집기 문제와 테트리스 문제에 대한 정확도는 평가 데이터셋에서 각각 77.2%, 71.5%의 정확도를 보여준다.

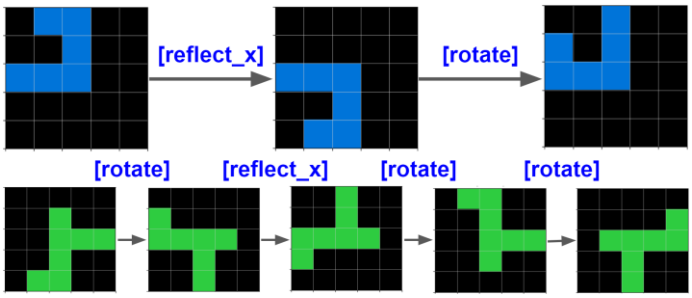

그림 4 대각선 뒤집기 문제에 대한 DT가 예측한 풀이 과정

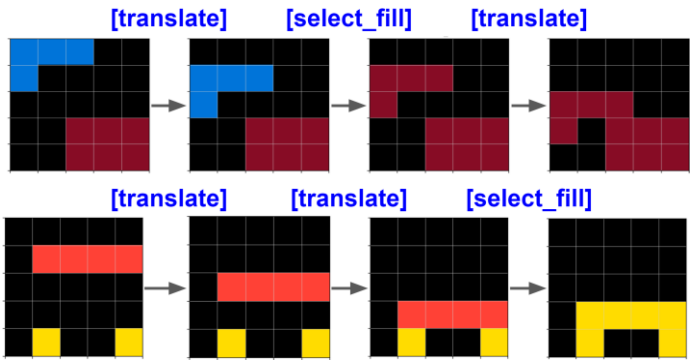

그림 5 테트리스 문제에 대한 DT가 예측한 풀이 과정

5. 결론 및 향후 연구

본 연구에서는 사람의 풀이 과정과 지능을 따라 하는 방법으로 문제 해결을 시도하였으며, 주목 받고 있는 트랜스포머 구조와 다양한 데이터셋을 활용하여 인공지능이 해결하기 어려워하는 ARC 문제 해결에 대한 가능성을 보여준다. 또한, 인간과 인공지능 간의 격차를 줄이기 위해 DT를 사용한 모방 학습 방법론을 제시했다.

의사결정 트랜스포머 (DT)는 오프라인 데이터셋을 이용하여 정책(policy)을 학습한다. 이로 인해, 학습 데이터셋에 없는 입력이 주어질 경우에 대해 적응력이 부족할 수 있다. 테트리스 문제의 경우 대각선 뒤집기 문제 보다 긴 풀이 과정에 의해 많은 학습 데이터셋이 요구되어 보다 낮은 성능을 보여준다. 이러한 문제는 프롬프트 의사결정 트랜스포머[6] 와 충분한 학습 데이터셋을 통해 해결할 수 있을 것으로 예상한다. 기존의 학습을 통해 기본적인 패턴을 인지할 수 있는 상태에서, 프롬프트와 같은 추가적인 입력을 통해 새로운 ARC 문제가 주어져도 스스로 문제를 분석할 수 있을 것으로 예상한다.

ARC 문제를 해결하는 인간의 풀이 과정은 특정 action이 반복되는 것을 알 수 있다. 인간은 기초적인 개념을 조합하여 DSL에 없는 새로운 action을 스스로 만들어 낸다. 예를 들어, 시계 방향으로 90도 회전하는 action 3번과 반시계 방향으로 90도 회전하는 action 1번은 같은 방법이다. 반복되는 action을 Dreamcoder [7]와 같은 방법을 사용하여 함축된 표현으로 만들면 더욱 일반적인 문제 해결이 가능할 것으로 기대한다.

본 연구에서는 O2ARC tool에서 제공되는 기능으로 action의 종류를 제한했다. 하지만 인간의 사고 과정을 제한된 action을 통해 표현하는 것은 제약이 많다. 이에 충분한 DSL을 확보하는 것이 중요하며[8], 이를 토대로 충분한 action이 제공될 때, DT가 더욱 범용적이면서도 효율적인 문제 해결이 가능할 것으로 예상된다.

6. 참고 문헌

[1] François Chollet, “On the measure of intelligence,” arXiv:1911.01547, 2019.
[2] Aysja Johnson et al., “Fast and flexible: Human program induction in abstract reasoning tasks,” in CogSci, 2021.
[3] Subin Kim et al., “Playgrounds for abstraction and reasoning,” in NeurIPS nCSI, 2022
[4] Lili Chen et al., “Decision transformer: Reinforcement learning via sequence modeling,” in NeurIPS, 2021.
[5] Alexey Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in ICLR, 2021
[6] Mengdi Xu et al., “Prompting decision transformer for few-shot policy generalization,” in ICML, PMLR 162:24631-24645, 2022
[7] Kevin Ellis et al., “Dreamcoder: Bootstrapping inductive program synthesis with wake-sleep library learning,” in PLDI, pages 835-850, 2021.
[8] Samuel Acquaviva et al., “Communicating natural programs to humans and machines,” arXiv:2106.07824, 2021.