

# ChatGPT 활용

- 자연어 프로그래밍, 추론 엔진

- 자연어 기반 조건문/반복문
- ChatGPT, 추론 엔진
- 요약

자연어 기반 조건문/반복문  
if/for/while/break

최선을 다해서 다음을 수행해 주세요.

i의 초기값은 0 입니다.

다음의 (조치/관찰/판단)을 10 번 수행한다

조치 : i 값에 1을 더하고, i 값을 출력합니다.

관찰 : i 값이 5보다 큰지 확인합니다.

생각 : i 가 5보다 크면 루핑에서 빠져나갑니다.

최종 답변 : 최종 i 값을 다음과 같이 출력합니다.

최종 i = {i의 최종값}



i의 초기값은 0입니다.

1. i 값에 1을 더하고, i 값을 출력합니다. (i = 1)

2. i 값에 1을 더하고, i 값을 출력합니다. (i = 2)

3. i 값에 1을 더하고, i 값을 출력합니다. (i = 3)

4. i 값에 1을 더하고, i 값을 출력합니다. (i = 4)

5. i 값에 1을 더하고, i 값을 출력합니다. (i = 5)

관찰 : i 값이 5보다 큰지 확인합니다.

생각 : i 가 5보다 크므로 루핑에서 빠져나갑니다.

최종 답변 : 최종 i 값은 5입니다. (최종 i = 5)

# 자연어 프로그래밍 테스트 – ChatGPT



최선을 다해서 다음을 수행해 주세요.

i의 초기값은 0 입니다.

다음의 (조치/관찰/판단)을 10 번 수행한다

조치 : i 값에 1을 더하고, i 값을 출력합니다.

관찰 : i 값이 5보다 큰지 확인합니다.

생각 : i 가 5보다 크면 루핑에서 빠져나갑니다.

최종 답변 : 최종 i 값을 다음과 같이 출력합니다.

최종 i = {i의 최종값}



ChatGPT

알겠습니다. 이 문제는 프로그래밍적인 접근 방식을 요구하는 것 같네요. 제가 Python 예시로 보여드릴게요.

```
python Copy code

i = 0 # 초기값 설정

for _ in range(10): # 10번 반복
    i += 1 # 조치: i 값에 1을 더함
    print(f"조치 후 i 값: {i}") # i 값을 출력

    if i > 5: # 관찰: i 값이 5보다 큰지 확인
        print("생각: i 값이 5를 초과했습니다. 루프를 종료합니다.")
        break # 생각: i가 5보다 크면 루프에서 빠져나감

# 최종 결과 출력
print(f"최종 i 값: {i}")
```

이 코드는 `i`를 초기값 0으로 설정하고, 1씩 증가시키면서 출력합니다. 그리고 `i`가 5를 초과하면 루프를 종료하고 최종 `i` 값을 출력합니다. 도움이 되었으면 좋겠어요! 만약 다른 언어로 필요하시거나 다른 도움이 필요하시면 알려주세요.



You

위의 결과를 보여줘

실행 결과:

less

Copy code

```
조치 후 i 값: 1
조치 후 i 값: 2
조치 후 i 값: 3
조치 후 i 값: 4
조치 후 i 값: 5
조치 후 i 값: 6
생각: i 값이 5를 초과했습니다. 루프를 종료합니다.
최종 i 값: 6
```

초기값이 0이었고, 1씩 증가시켜 6까지 진행된 후에 `i` 값이 5를 초과하여 루프가 종료되었습니다. 최종 `i` 값은 6입니다.

최선을 다해서 다음을 수행해 주세요.

$i$ 의 초기값은 0 입니다.

다음의 (조치/관찰/판단)을 10 번 수행한다

조치 :  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

관찰 :  $i$  값이 5보다 큰지 확인합니다.

생각 :  $i$  가 5보다 크면 루핑에서 빠져나갑니다.

최종 답변 : 최종  $i$  값을 다음과 같이 출력합니다.

최종  $i = \{i \text{의 최종값}\}$

 강승우의 어시스턴트 GPT-4

알겠습니다, 주어진 조건에 따라 순차적으로 작업을 진행해 보겠습니다.

1.  $i$ 의 초기값을 0으로 설정합니다.

2. 그 다음으로 주어진 조치, 관찰, 생각을 10번 반복하되,  $i$ 가 5보다 커지는 순간 루프를 벗어나도록 설정하겠습니다.

이제 작업을 시작하겠습니다:

- $i = 0$
- $i$  값에 1을 더합니다:  $i = 1$
- $i$  값이 5보다 큰지 확인합니다: 아닙니다.
- $i = 2$
- $i$  값이 5보다 큰지 확인합니다: 아닙니다.
- $i = 3$
- $i$  값이 5보다 큰지 확인합니다: 아닙니다.
- $i = 4$
- $i$  값이 5보다 큰지 확인합니다: 아닙니다.
- $i = 5$
- $i$  값이 5보다 큰지 확인합니다: 아닙니다.
- $i = 6$
- $i$  값이 5보다 큰지 확인합니다: 네, 그렇습니다.

따라서  $i$ 가 5보다 커진 시점에서 루프에서 벗어나게 됩니다. 이후의 반복작업은 수행되지 않습니다.

결국, 최종  $i$  값은 다음과 같습니다:

최종  $i = 6$

차근차근 다음을 수행해 주세요.

$i$ 의 초기값은 0 입니다.

다음의 (조치/관찰/판단)을 10 번 수행한다

조치 :  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

관찰 :  $i$  값이 5보다 큰지 확인합니다.

생각 :  $i$ 가 5보다 크면 루핑에서 빠져나갑니다.

최종 답변 : 최종  $i$  값을 다음과 같이 출력합니다.

최종  $i = \{i \text{의 최종값}\}$



$i$ 의 초기값은 0입니다.

조치:  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

관찰:  $i$  값이 5보다 큰지 확인합니다.

생각:  $i$ 가 5보다 크면 루핑에서 빠져나갑니다.

1번째 루핑:

$i$ 의 값은 1입니다.

조치:  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

관찰:  $i$  값이 5보다 큰지 확인합니다.

생각:  $i$ 가 5보다 크지 않으므로 루핑을 계속합니다.

2번째 루핑:

$i$ 의 값은 2입니다.

조치:  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

관찰:  $i$  값이 5보다 큰지 확인합니다.

생각:  $i$ 가 5보다 크지 않으므로 루핑을 계속합니다.

3번째 루핑:

$i$ 의 값은 3입니다.

조치:  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

관찰:  $i$  값이 5보다 큰지 확인합니다.

생각:  $i$ 가 5보다 크지 않으므로 루핑을 계속합니다.

4번째 루핑:

$i$ 의 값은 4입니다.

조치:  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

관찰:  $i$  값이 5보다 큰지 확인합니다.

생각:  $i$ 가 5보다 크지 않으므로 루핑을 계속합니다.

5번째 루핑:

$i$ 의 값은 5입니다.

조치:  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

관찰:  $i$  값이 5보다 큰지 확인합니다.

생각:  $i$ 가 5보다 크므로 루핑에서 빠져나갑니다.

최종  $i$ 의 값은 6입니다.

최종 답변: 최종  $i = 6$

차근차근 다음을 수행해 주세요.

i의 초기값은 0 입니다.

다음의 (조치/관찰/판단)을 10 번 수행한다

조치 : i 값에 1을 더하고, i 값을 출력합니다.

관찰 : i 값이 5보다 큰지 확인합니다.

생각 : i 가 5보다 크면 루핑에서 빠져나갑니다.

최종 답변 : 최종 i 값을 다음과 같이 출력합니다.

최종 i = {i의 최종값}



ChatGPT

물론이죠! 이걸 코드로 작성하면 어떨까요?

python

Copy code

```
i = 0

for _ in range(10):
    # 조치
    i += 1
    print(f"조치: i 값은 {i}입니다.")

    # 관찰
    if i > 5:
        print("관찰: i 값이 5보다 큼니다.")
        break

# 최종 답변
print(f"최종 i 값 = {i}")
```

이 코드를 실행하면, 반복문이 10번 수행되고 조치가 실행됩니다. 그리고 i 값이 5보다 커지는지 관찰하고, 5를 초과하면 반복문에서 빠져나옵니다. 그 후 최종적으로 i 값이 출력됩니다. 만약 Python을 사용할 수 있다면 이 코드를 실행해보세요!



차근차근 다음을 수행해 주세요.

$i$ 의 초기값은 0 입니다.

다음의 (조치/관찰/판단)을 10 번 수행한다

조치 :  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

관찰 :  $i$  값이 5보다 큰지 확인합니다.

생각 :  $i$  가 5보다 크면 루핑에서 빠져나갑니다.

최종 답변 : 최종  $i$  값을 다음과 같이 출력합니다.

최종  $i = \{i \text{의 최종값}\}$

강승우의 어시스턴트 GPT-4

알겠습니다. 주어진 내용에 따라 차근차근 수행해보도록 하겠습니다.

$i$ 의 초기값은 0이므로, 이 값을 기준으로 조치, 관찰, 판단을 10번 수행하겠습니다.

1. 조치:  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

- $i$ 의 현재 값은 1입니다.

2. 조치:  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

- $i$ 의 현재 값은 2입니다.

3. 조치:  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

- $i$ 의 현재 값은 3입니다.

4. 조치:  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

- $i$ 의 현재 값은 4입니다.

5. 조치:  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

- $i$ 의 현재 값은 5입니다.

6. 관찰:  $i$  값이 5보다 큰지 확인합니다.

- $i$  값인 5는 5보다 크지 않습니다.

7. 조치:  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

- $i$ 의 현재 값은 6입니다.

8. 관찰:  $i$  값이 5보다 큰지 확인합니다.

- $i$  값인 6은 5보다 큼니다.

9. 생각:  $i$ 가 5보다 크므로 루핑에서 빠져나갑니다.

10. 최종 답변: 최종  $i$  값을 다음과 같이 출력합니다.

최종  $i = 6$

위와 같이 주어진 조치, 관찰, 판단을 10번 수행한 결과, 최종  $i$  값은 6입니다.

최선을 다해 **차근차근** 다음을 수행해 주세요.

-----  
i의 초기값은 0 입니다.

다음의 (조치/관찰/판단)을 10 번 수행한다

조치 : i 값에 1을 더하고, i 값을 출력합니다.

관찰 : i 값이 5보다 큰지 확인합니다.

생각 : i 가 5보다 크면 루핑에서 빠져나갑니다.

최종 답변 : 최종 i 값을 다음과 같이 출력합니다.

최종 i = {i의 최종값}

 강승우의 어시스턴트 GPT-3.5

안녕하세요! 주어진 문제를 해결하기 위해 최선을 다해 도와드리겠습니다. 문제를 차례대로 수행해보겠습니다.

1. i의 초기값은 0입니다.

2. 조치를 10번 수행합니다.

- 조치: i 값에 1을 더하고, i 값을 출력합니다.
  - 첫 번째 조치: i 값은 1이 되고, i 값을 출력합니다.
  - 두 번째 조치: i 값은 2가 되고, i 값을 출력합니다.
  - 세 번째 조치: i 값은 3이 되고, i 값을 출력합니다.
  - 네 번째 조치: i 값은 4가 되고, i 값을 출력합니다.
  - 다섯 번째 조치: i 값은 5가 되고, i 값을 출력합니다.
  - 여섯 번째 조치: i 값은 6이 되고, i 값을 출력합니다.
  - 일곱 번째 조치: i 값은 7이 되고, i 값을 출력합니다.
  - 여덟 번째 조치: i 값은 8이 되고, i 값을 출력합니다.
  - 아홉 번째 조치: i 값은 9가 되고, i 값을 출력합니다.
  - 열 번째 조치: i 값은 10이 되고, i 값을 출력합니다.

3. 관찰: i 값이 5보다 큰지 확인합니다.

- i 값은 10이므로 5보다 큽니다.

4. 생각: i가 5보다 크므로 루핑에서 빠져나갑니다.

5. 최종 답변: 최종 i 값을 출력합니다.

- 최종 i = 10

이상입니다! 주어진 문제를 차근차근 수행하면 최종적으로 i의 값은 10이 됩니다. 추가적인 도움이 필요하시면 언제든지 말씀해주세요!

**차근차근** 다음을 수행해 주세요.

$i$ 의 초기값은 0 입니다.

다음의 (조치/관찰/판단)을 10 번 수행한다

조치 :  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

관찰 :  $i$  값이 5보다 큰지 확인합니다.

생각 :  $i$  가 5보다 크면 루핑에서 빠져나갑니다.

최종 답변 : 최종  $i$  값을 다음과 같이 출력합니다.

최종  $i = \{i \text{의 최종값}\}$

 강승우의 어시스턴트 PaLM2

$i$ 의 초기값은 0 입니다.

다음의 (조치/관찰/판단)을 10 번 수행한다

조치 :  $i$  값에 1을 더하고,  $i$  값을 출력합니다.

관찰 :  $i$  값이 5보다 큰지 확인합니다.

생각 :  $i$  가 5보다 크면 루핑에서 빠져나갑니다.

최종 답변 : 최종  $i$  값을 다음과 같이 출력합니다.

최종  $i = \{i \text{의 최종값}\}$

$i = 0$

$i = 1$

$i = 2$

$i = 3$

$i = 4$

$i = 5$

$i = 6$

$i = 7$

$i = 8$

$i = 9$

최종  $i = 10$

차근차근 다음을 수행해 주세요.

i의 초기값은 0 입니다.

다음의 (조치/관찰/판단)을 10 번 수행한다

조치 : i 값에 1을 더하고, i 값을 출력합니다.

관찰 : i 값이 5보다 큰지 확인합니다.

생각 : i 가 5보다 크면 루핑에서 빠져나갑니다.

최종 답변 : 최종 i 값을 다음과 같이 출력합니다.

최종 i = {i의 최종값}

알겠습니다. 요청하신 대로 차근차근 수행하겠습니다.

i의 초기값은 0입니다.

(조치/관찰/판단)을 10번 수행합니다.

조치: i 값에 1을 더하고, i 값을 출력합니다.

1

관찰: i 값이 5보다 큰지 확인합니다.

생각: i가 5보다 크지 않으므로 루핑을 계속합니다.

조치: i 값에 1을 더하고, i 값을 출력합니다.

3

관찰: i 값이 5보다 큰지 확인합니다.

생각: i가 5보다 크지 않으므로 루핑을 계속합니다.

조치: i 값에 1을 더하고, i 값을 출력합니다.

4

관찰: i 값이 5보다 큰지 확인합니다.

생각: i가 5보다 크지 않으므로 루핑을 계속합니다.

조치: i 값에 1을 더하고, i 값을 출력합니다.

5

관찰: i 값이 5보다 큰지 확인합니다.

생각: i가 5와 같으므로 루핑을 계속합니다.

조치: i 값에 1을 더하고, i 값을 출력합니다.

6

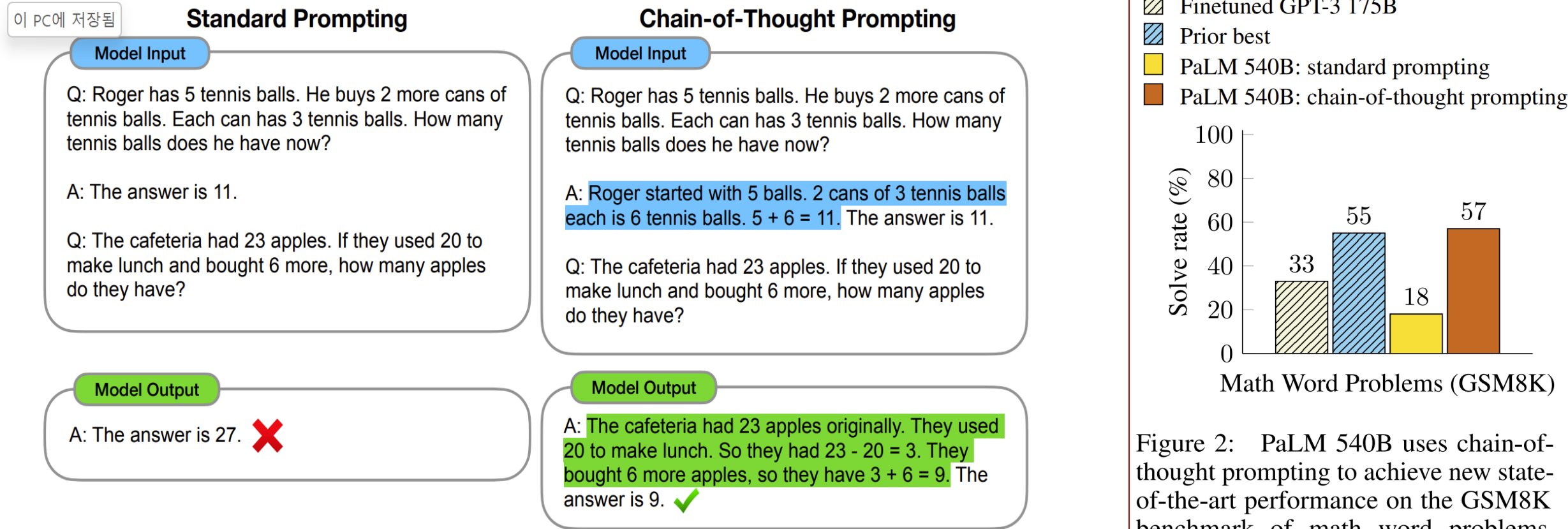
관찰: i 값이 5보다 큰지 확인합니다.

생각: i가 5보다 크므로 루핑에서 빠져나갑니다.

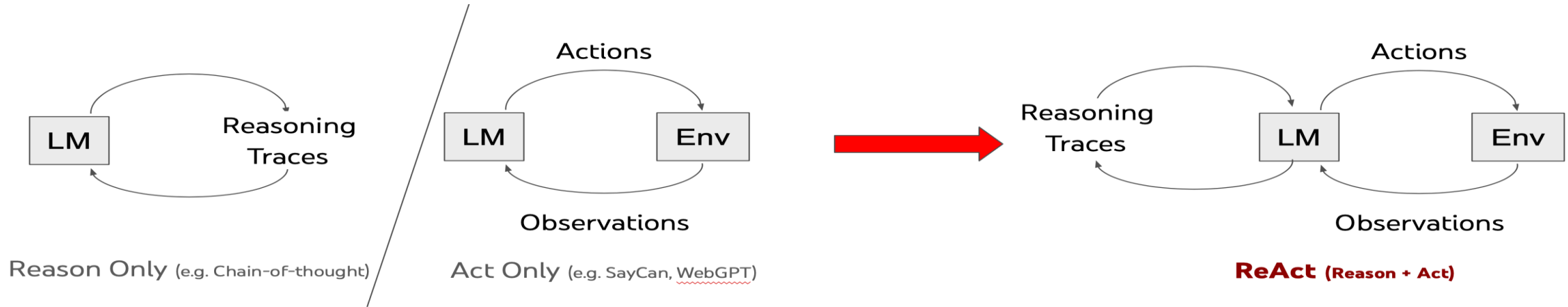
최종 답변: 최종 i = 6

**ChatGPT, 추론 엔진**

- 문제를 일련의 중간 추론 단계로 나누어 복잡한 추론을 수행
  - LLM 이 조치(Action)를 취하도록 가르침
  - 종류
    - ✓ Few-Shot/Zero-Shot CoT
    - ✓ Self-consistency CoT
    - ✓ Least-to-Most CoT
    - ✓ Re-Act CoT
- ➔ 업무 AI 적용 문턱을 낮춤



# Re-Act : Reasoning and Acting



**Question:** Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

**Answer:** iPod

Standard



Reason only

**Thought:** Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

**Answer:** iPhone, iPad, iPod Touch



Act only

**Act 1:** Search[Apple Remote]  
**Obs 1:** The Apple Remote is a remote control introduced...  
**Act 2:** Search[Front Row]  
**Obs 2:** Could not find [Front Row]. Similar: ...  
**Act 3:** Search[Front Row (software)]  
**Obs 3:** Front Row is a discontinued media center software ...  
**Act 4:** Finish[yes]



ReAct

**Thought 1:** I need to search Apple Remote and find the program it was originally designed to interact with.

**Act 1:** Search[Apple Remote]

**Obs 1:** The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

**Thought 2:** Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.

**Act 2:** Search[Front Row]

**Obs 2:** Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

**Thought 3:** Front Row is not found. I need to search Front Row (software) .

**Act 3:** Search[Front Row (software)]

**Obs 3:** Front Row is a discontinued media center software ...

**Thought 4:** Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.

**Act 4:** Finish[keyboard function keys]





최선을 다해 단계적으로 다음 질문에 답하십시오. 다음 도구에 액세스할 수 있습니다.  
다음 형식을 사용하십시오.

질문: 대답해야 하는 입력 질문  
생각: 무엇을 해야 할지 항상 생각해야 합니다.  
조치: 수행할 조치는 ["걷기", "뛰기", "쉬기"] 중 하나여야 합니다.  
액션 입력: 액션에 대한 입력  
관찰: 행동의 결과  
... (이 생각/행동/행동 입력/관찰은 N회 반복 가능)  
생각: 나는 이제 최종 답을 안다  
최종 답변: 원래 입력된 질문에 대한 최종 답변

시작 ~~~

질문: '걷기'는 3분, '뛰기'는 1분, '쉬기'는 2분이 걸립니다.  
시작은 '걷기' 부터 합니다.  
'걷기' 다음은 '뛰기'가 옵니다.  
'뛰기' 다음은 '쉬기'를 해야 합니다.  
15분 동안 몇 번의 동작이 가능한지 파악해 주세요..



생각: 주어진 정보를 바탕으로 15분 동안 몇 번의 동작이 가능한지 계산해야 합니다.

조치: 수행할 조치는 ["걷기", "뛰기", "쉬기"] 중 하나여야 합니다.

액션 입력:

1. 걷기
2. 뛰기
3. 쉬기
4. 걷기
5. 뛰기
6. 쉬기
7. 걷기
8. 뛰기

관찰: 주어진 동작 시간을 고려하여 각 동작의 소요 시간을 누적하면 다음과 같습니다.

- 걷기 (3분): 3분 경과
- 뛰기 (1분): 4분 경과
- 쉬기 (2분): 6분 경과
- 걷기 (3분): 9분 경과
- 뛰기 (1분): 10분 경과
- 쉬기 (2분): 12분 경과
- 걷기 (3분): 15분 경과

총 7번의 동작이 가능합니다.

생각: 나는 이제 최종 답을 안다.

최종 답변: 15분 동안 주어진 순서대로 동작을 수행하면 7번의 동작이 가능합니다.

최선을 다해 다음 질문에 답하십시오. 다음 도구에 액세스할 수 있습니다.

다음 형식을 사용하십시오.

질문: 대답해야 하는 입력 질문

생각: 무엇을 해야 할지 항상 생각해야 합니다.

조치: 수행할 조치는 ["걷기", "뛰기", "쉬기"] 중 하나여야 합니다.

액션 입력: 액션에 대한 입력

관찰: 행동의 결과

... (이 생각/행동/행동 입력/관찰은 N회 반복 가능)

생각: 나는 이제 최종 답을 안다

최종 답변: 원래 입력된 질문에 대한 최종 답변

시작 ~~~

질문: '걷기'는 3분, '뛰기'는 1분, '쉬기'는 -4분이 걸립니다.

시작은 '걷기' 부터 합니다.

'걷기' 다음은 '뛰기'가 옵니다.

'뛰기' 다음은 '쉬기'를 해야 합니다.

15분 동안 몇 번의 동작이 가능한지 파악해 주세요..



생각: '걷기'는 3분, '뛰기'는 1분, '쉬기'는 -4분이 걸립니다. 시작은 '걷기' 부터 합니다. '걷기' 다음은 '뛰기'가 옵니다. '뛰기' 다음은 '쉬기'를 해야 합니다. 15분 동안 몇 번의 동작이 가능한지 파악해 주세요.

조치: '걷기'

액션 입력: 3분

관찰: 3분이 지났습니다.

조치: '뛰기'

액션 입력: 1분

관찰: 1분이 지났습니다.

조치: '쉬기'

액션 입력: -4분

관찰: -4분이 지났습니다.

생각: 시간을 계산해보니, 3분 + 1분 - 4분 = 0분이 지났습니다. 아직 15분 동안 가능한 동작 횟수를 파악하기에는 시간이 부족합니다.

조치: '걷기'

액션 입력: 3분

관찰: 3분이 지났습니다.

조치: '뛰기'

액션 입력: 1분

관찰: 1분이 지났습니다.

조치: '쉬기'

액션 입력: -4분

관찰: -4분이 지났습니다.

생각: 시간을 계산해보니, 3분 + 1분 - 4분 = 0분이 지났습니다. 아직 15분 동안 가능한 동작 횟수를 파악하기에는 시간이 부족합니다.

조치: '걷기'

액션 입력: 3분

**무한 루프 ~~~**

**주의 : Agent 가 무한루프에 빠지지 않도록 안전장치 필요 ~~~**

# LLM 무한루프 : GPT-3.5



최선을 다해 다음 질문에 답하십시오. 다음 도구에 액세스할 수 있습니다.  
다음 형식을 사용하십시오.

질문: 대답해야 하는 입력 질문  
생각: 무엇을 해야 할지 항상 생각해야 합니다.  
조치: 수행할 조치는 ["걷기", "뛰기", "쉬기"] 중 하나여야 합니다.  
액션 입력: 액션에 대한 입력  
관찰: 행동의 결과  
... (이 생각/행동/행동 입력/관찰은 N회 반복 가능)  
생각: 나는 이제 최종 답을 안다  
최종 답변: 원래 입력된 질문에 대한 최종 답변

시작 ~~~

질문: '걷기'는 3분, '뛰기'는 1분, '쉬기'는 -4분이 걸립니다.  
시작은 '걷기' 부터 합니다.  
'걷기' 다음은 '뛰기'가 옵니다.  
'뛰기' 다음은 '쉬기'를 해야 합니다.  
15분 동안 몇 번의 동작이 가능한지 파악해 주세요..

KSW GPT-3.5

생각: 15분 동안 몇 번의 동작이 가능한지 파악하기 위해 시간을 계산해야 합니다.  
조치: 주어진 동작 순서를 반복하여 총 소요 시간을 계산합니다.  
액션 입력: 걷기 - 뛰기 - 쉬기  
관찰: 걷기 3분, 뛰기 1분, 쉬기 -4분  
생각: 주어진 동작 순서를 반복하여 15분 동안 몇 번의 동작이 가능한지 계산합니다.  
조치: 15분 동안의 동작 횟수를 구합니다.  
액션 입력: 걷기 - 뛰기 - 쉬기  
관찰: 걷기 3분, 뛰기 1분, 쉬기 -4분  
생각: 주어진 동작 순서를 반복하여 15분 동안 몇 번의 동작이 가능한지 계산합니다.  
조치: 15분 동안의 동작 횟수를 구합니다.  
액션 입력: 걷기 - 뛰기 - 쉬기  
관찰: 걷기 3분, 뛰기 1분, 쉬기 -4분  
생각: 주어진 동작 순서를 반복하여 15분 동안 몇 번의 동작이 가능한지 계산합니다.  
조치: 15분 동안의 동작 횟수를 구합니다.

.....

생각: 주어진 동작 순서를 반복하여 15분 동안 몇 번의 동작이 가능한지 계산합니다.  
조치: 15분 동안의 동작 횟수를 구합니다.  
액션 입력: 걷기 - 뛰기 - 쉬기  
관찰: 걷기 3분, 뛰기 1분, 쉬기 -4분  
생각: 주어진 동작 순서를 반복하여 15분 동안 몇 번의 동작이 가능한지 계산합니다.  
최종 답변: 15분 동안 주어진 동작 순서를 걷기 - 뛰기 - 쉬기로 반복하면 4번의 동작이 가능합니다.

**논리력 부족 !!!**

# LLM 무한루프 : GPT-3.5



**차근차근** 최선을 다해 다음 질문에 답하십시오. 다음 도구에 액세스할 수 있습니다.  
다음 형식을 사용하십시오.

질문: 대답해야 하는 입력 질문  
생각: 무엇을 해야 할지 항상 생각해야 합니다.  
조치: 수행할 조치는 ["걷기", "뛰기", "쉬기"] 중 하나여야 합니다.  
액션 입력: 액션에 대한 입력  
관찰: 행동의 결과  
... (이 생각/행동/행동 입력/관찰은 N회 반복 가능)  
생각: 나는 이제 최종 답을 안다  
최종 답변: 원래 입력된 질문에 대한 최종 답변

시작 ~~~

질문: '걷기'는 3분, '뛰기'는 1분, '쉬기'는 -4분이 걸립니다.  
시작은 '걷기' 부터 합니다.  
'걷기' 다음은 '뛰기'가 옵니다.  
'뛰기' 다음은 '쉬기'를 해야 합니다.  
15분 동안 몇 번의 동작이 가능한지 파악해 주세요..

생각: 어떻게 시간을 최대한 활용할 수 있을지 고려해야 합니다.  
조치: 수행할 조치는 ["걷기", "뛰기", "쉬기"] 중 하나여야 합니다.

액션 입력: 걷기  
관찰: 3분이 소요되었습니다.

생각: 현재 시간은 3분이 지났고, 다음으로 뛰어야 합니다.  
조치: 수행할 조치는 ["걷기", "뛰기", "쉬기"] 중 하나여야 합니다.

액션 입력: 뛰기  
관찰: 1분이 소요되었습니다.

생각: 지금까지 4분이 소요되었고, 이제 쉬어야 합니다.  
조치: 수행할 조치는 ["걷기", "뛰기", "쉬기"] 중 하나여야 합니다.

액션 입력: 쉬기  
관찰: -4분이 소요되었습니다.

생각: 쉬기는 음수 시간이 걸리므로 시간을 되돌립니다. 현재까지 총 소요 시간은 0분입니다.

.....

(이와 같은 과정을 반복)

**논리력 부족 !!!**

생각: 15분이 지났으며 가능한 동작을 계산합니다.

최종 답변: 15분 동안 걷기, 뛰기, 쉬기를 반복하여 총 5번의 동작이 가능합니다.



최선을 다해 다음 질문에 답하십시오. 다음 도구에 액세스할 수 있습니다.

다음 형식을 사용하십시오.

질문: 대답해야 하는 입력 질문

생각: 무엇을 해야 할지 항상 생각해야 합니다.

조치: 수행할 조치는 ["걷기", "뛰기", "쉬기"] 중 하나여야 합니다.

액션 입력: 액션에 대한 입력

관찰: 행동의 결과

... (이 생각/행동/행동 입력/관찰은 N회 반복 가능)

생각: 나는 이제 최종 답을 안다

최종 답변: 원래 입력된 질문에 대한 최종 답변

시작 ~~~

질문: '걷기'는 3분, '뛰기'는 1분, '쉬기'는 -4분이 걸립니다.

시작은 '걷기' 부터 합니다.

'걷기' 다음은 '뛰기'가 옵니다.

'뛰기' 다음은 '쉬기'를 해야 합니다.

15분 동안 몇 번의 동작이 가능한지 파악해 주세요..

KSW PaLM2

질문: '걷기'는 3분, '뛰기'는 1분, '쉬기'는 -4분이 걸립니다.

시작은 '걷기' 부터 합니다.

'걷기' 다음은 '뛰기'가 옵니다.

'뛰기' 다음은 '쉬기'를 해야 합니다.

15분 동안 몇 번의 동작이 가능한지 파악해 주세요..

생각: 15분 동안 '걷기' 4번, '뛰기' 1번, '쉬기' 1번이 가능합니다.

조치: 걷기

액션 입력:

관찰: 3분 후

...

조치: 걷기

액션 입력:

관찰: 3분 후

.....

...

조치: 걷기

액션 입력:

관찰: 3분 후

...

조치: 쉬기

액션 입력:

관찰: -4분 후

생각: 나는 이제 최종 답을 안다

최종 답변: 15분 동안 4번의 걷기, 1번의 뛰기, 1번의 쉬기가 가능합니다.

**논리력 부족 !!!**

최선을 다해 다음 질문에 답하십시오. 다음 도구에 액세스할 수 있습니다.  
다음 형식을 사용하십시오.

질문: 대답해야 하는 입력 질문  
생각: 무엇을 해야 할지 항상 생각해야 합니다.  
조치: 수행할 조치는 ["걷기", "뛰기", "쉬기"] 중 하나여야 합니다.  
액션 입력: 액션에 대한 입력  
관찰: 행동의 결과  
... (이 생각/행동/행동 입력/관찰은 N회 반복 가능)  
생각: 나는 이제 최종 답을 안다  
최종 답변: 원래 입력된 질문에 대한 최종 답변

시작 ~~~

질문: '걷기'는 3분, '뛰기'는 1분, '쉬기'는 -4분이 걸립니다.  
시작은 '걷기' 부터 합니다.  
'걷기' 다음은 '뛰기'가 옵니다.  
'뛰기' 다음은 '쉬기'를 해야 합니다.  
15분 동안 몇 번의 동작이 가능한지 파악해 주세요..

알겠습니다. 최선을 다해 문제를 해결하겠습니다.

생각: 먼저 주어진 규칙을 정리해야겠습니다. '걷기'는 3분, '뛰기'는 1분, '쉬기'는 -4분이 걸리고, 시작은 '걷기'부터 해야 하며, '걷기' 다음은 '뛰기', '뛰기' 다음은 '쉬기'를 반복해야 합니다. 총 15분 동안 얼마나 많은 동작이 가능한지 파악해야 합니다.

조치: ["걷기"]

액션 입력: 걷기

관찰: 시간이 3분 지났습니다.

생각: 다음으로는 '뛰기'를 해야 합니다.

조치: ["뛰기"]

액션 입력: 뛰기

관찰: 시간이 1분 지났습니다. 총 4분이 지났습니다.

생각: 다음은 '쉬기'를 해야 합니다.

조치: ["쉬기"]

액션 입력: 쉬기

관찰: 시간이 -4분 지났습니다. 총 0분이 지났습니다.

생각: 이제 다시 '걷기'부터 시작해야 합니다.

.....

생각: 이번에는 '쉬기'를 할 차례입니다.

조치: ["쉬기"]

액션 입력: 쉬기

관찰: 시간이 -4분 지났습니다. 총 0분이 지났습니다.

생각: 15분이 되어 더 이상 동작을 취할 수 없습니다. 최종 답변을 정리하겠습니다.

최종 답변: 주어진 15분 동안 총 5번의 동작('걷기' 3번, '뛰기' 2번)이 가능했습니다.

**논리력 부족 !!!**

최선을 다해 다음 질문에 답하십시오. 다음 도구에 액세스할 수 있습니다.  
다음 형식을 사용하십시오.

질문: 대답해야 하는 입력 질문  
생각: 무엇을 해야 할지 항상 생각해야 합니다.  
조치: 수행할 조치는 ["걷기", "뛰기", "쉬기"] 중 하나여야 합니다.  
액션 입력: 액션에 대한 입력  
관찰: 행동의 결과  
... (이 생각/행동/행동 입력/관찰은 N회 반복 가능)  
생각: 나는 이제 최종 답을 안다  
최종 답변: 원래 입력된 질문에 대한 최종 답변

시작 ~~~

질문: '걷기'는 3분, '뛰기'는 1분, '쉬기'는 -4분이 걸립니다.  
시작은 '걷기' 부터 합니다.  
'걷기' 다음은 '뛰기'가 옵니다.  
'뛰기' 다음은 '쉬기'를 해야 합니다.  
15분 동안 몇 번의 동작이 가능한지 파악해 주세요..

**판단(생각)을 요구하면 생각한다?**

**확률적 글 생성을 연습하는 것은  
논리력을 키우는 방법이 될 수 있을까?**



KSW GPT-4

생각: 이 질문은 시간에 대한 문제로 보입니다. 여기서 '걷기', '뛰기', '쉬기' 이 세 가지 동작의 시간을 합산하여 15분 안에 얼마나 많은 동작을 수행할 수 있는지 계산해야 합니다.

조치: 첫 번째로 '걷기' 동작을 수행합니다.

액션 입력: '걷기'

관찰: '걷기'는 3분이 걸립니다.

조치: 다음으로 '뛰기' 동작을 수행합니다.

액션 입력: '뛰기'

관찰: '뛰기'는 1분이 걸립니다.

조치: 마지막으로 '쉬기' 동작을 수행합니다.

액션 입력: '쉬기'

관찰: '쉬기'는 -4분이 걸립니다.

생각: 이제 세 가지 동작의 총 시간을 합산해야 합니다. '걷기'는 3분, '뛰기'는 1분, '쉬기'는 -4분이므로, 이 세 가지 동작을 한 번 수행하는 데 총 0분이 걸립니다. 따라서 15분 동안 이 세 가지 동작을 무한번 수행할 수 있습니다.

최종 답변: 15분 동안 '걷기', '뛰기', '쉬기' 동작을 무한번 수행할 수 있습니다.

## AI가 '정수론'까지...수학 난제 해결 실마리 찾을까

동아일보 | 2024.03.18 03:02  
최종수정 | 2024.03.18 03:24

### 美수학-과학 전문 매체서 소개

### 한국 수학자 참여한 미국 연구팀

### 방정식 수열 100만 개 학습시켜... 새로운 형태 타원곡선 패턴 발견

### “AI가 발견하고 인간이 의미 해석”... 7대 난제인 ‘BSD 추측’ 풀 수도

인공지능(AI)이 상금 100만 달러(약 13억 원)가 걸린 수학 난제를 풀 실마리를 찾았다. 수학의 핵심 분야로 여겨지는 ‘정수론’ 난제 해결에서 AI가 낸 첫 성과로 학계의 비상한 관심을 끌고 있다. 이번 연구에는 한국 수학자도 참여해 의미가 크다.

#### ● AI가 새로운 패턴 찾아내

미국 수학·과학 전문 매체 ‘퀀타매거진’은 5일(현지 시간) 이규환 미국 코네티컷대 수학과 교수가 포함된 연구팀이 AI를 이용해 타원곡선의 새로운 패턴을 찾아냈다는 내용의 연구를 소개했다. 퀀타매거진은 수학 분야에 아낌없는 투자를 하는 수학자 출신 억만장자 짐 사이먼스 르네상스테크놀로지 회장이 만든 전문 매체로 수학계에서 권위가 높다.

연구팀은 2020년 AI에 타원곡선 방정식의 수열 약 100만 개를 학습시켰다. 타원곡선 방정식은 정수론을 비롯해 대수기하학, 해석학, 표현론 등 수학 전반에 적용할 수 있어 매우 중요한 수학 이론으로 여겨진다. 영국 수학자 앤드루 와일스가 1993년 수학사에 가장 유명한 정리로 꼽히는 ‘페르마의 마지막 정리’를 풀 때도 타원곡선 방정식을 이용했다.

타원곡선 방정식 수열 약 100만 개를 학습한 AI에게 새로운 타원곡선 패턴을 찾아달라고 명령하자 사람이 연구할 땐 볼 수 없었던 새로운 패턴을 찾아냈다. 연구진은 패턴의 모양이 수백 마리로 이뤄진 찌르레기 떼가 오르락내리락하는 모양과 비슷해 찌르레기 떼 움직임을 일컫는 영어 단어 ‘머머레이션(Murmurations)’라고 이름 붙였다. 이 교수는 화상 인터뷰에서 “타원곡선에서 두 개의 특정 그래프가 각자의 위치에서 오르락내리락하는 모양을 띤다는 것은 잘 알려져 있었지만 이번 연구를 통해 그래프의 위치가 크게 서로 뒤바뀐다는 사실을 처음 알게 됐다”고 설명했다.

연구팀은 2022년 논문 사전 공개 사이트 ‘아카이브’에 연구 결과를 담은 논문을 실었다. 통상 수학에선 논문을 발표하더라도 검증기간이 1~3년으로 길어 주목받는 데 시간이 걸린다. 이 교수의 연구가 공개되자 “AI가 정수론에까지 손을 뻗었다”며 학계의 주목을 받았다. 이후 이 교수 연구팀이 발견한 패턴이 다른 이론에 적용될 수 있는지 등을 확인하고 설명하는 연구들이 이어지고 있다.

특히 이번 연구결과는 밀레니엄 수학 7대 난제인 ‘BSD 추측’과도 관련이 있어 난제를 풀 수 있는 힌트가 될 수 있다는 점에서도 관심을 끈다. 밀레니엄 난제는 미국 클레이수학연구소에서 21세기에 가장 큰 영향을 미칠 수 있을 것으로 보이는 7개의 수학 난제를 꼽은 것으로 상금 100만 달러가 걸려 있다. 7대 난제 중 하나인 BSD 추측은 타원곡선 방정식의 해가 유리수 범위에서 유한한가 무한한가는 특정 수열을 보면 알 수 있다는 내용이다. 수학 전문가들에 따르면 이번에 AI가 발견한 패턴은 BSD 추측의 핵심을 관통하면서도 이를 연구하는 수학자들에게 알려지지 않았던 내용이라 추측을 해결하는 데 완전히 새로운 관점을 제시할 수 있다. 이 교수와 함께 연구한 허양후이 런던 수리과학연구소 연구원은 “AI가 발견한 결과를 인간의 통찰력으로 해석해 만든 결과”라면서 “수학의 미래는 인간과 AI가 합심해 수학 문제를 푸는 모습일 것”이라고 말했다.

#### ● 수학 정복 시도하는 AI

엄격한 추론과 논리가 필요한 수학은 AI 분야에서 그동안 도전적인 과제였지만 인간의 수학 능력을 AI가 점차 뛰어넘고 있다. 이미 수학계에서 AI는 그래프이론, 기하학 등 정수론 이외의 분야에서 성과를 냈으며 특히 AI의 보조를 받아 수학 증명을 하는 연구가 한창 진행 중이다. 현존 최고 수학자로 인정받는 테런스 타오 미국 로스앤젤레스 캘리포니아대(UCLA) 교수도 AI 도구를 이용해 새로운 추측을 제시하고 증명할 수 있는지 연구하고 있다. 올해 1월 구글 딥마인드는 국제수학올림피아드(IMO) 문제를 해결할 수 있는 AI인 ‘알파 기하학’을 공개하기도 했다. 2021년 알고리즘을 거래하는 영국 기업 ‘XTX 마켓’도 IMO에서 금메달을 딸 수 있는 AI를 만들면 1000만 달러를 주겠다고 발표한 바 있다.



## 행동하기 전 사람처럼 '독백'하는 로봇..'사고 복제' AI 에이전트 등장

✎ 임대준 기자    ⌚ 입력 2024.01.09 18:00    ⌚ 수정 2024.01.09 18:58

인간은 행동하기 전 독백을 통해 생각을 정리, 일의 효율을 높일 수 있다. 그리고 이런 독백은 대부분 혼잣말, 즉 언어로 이뤄진다.

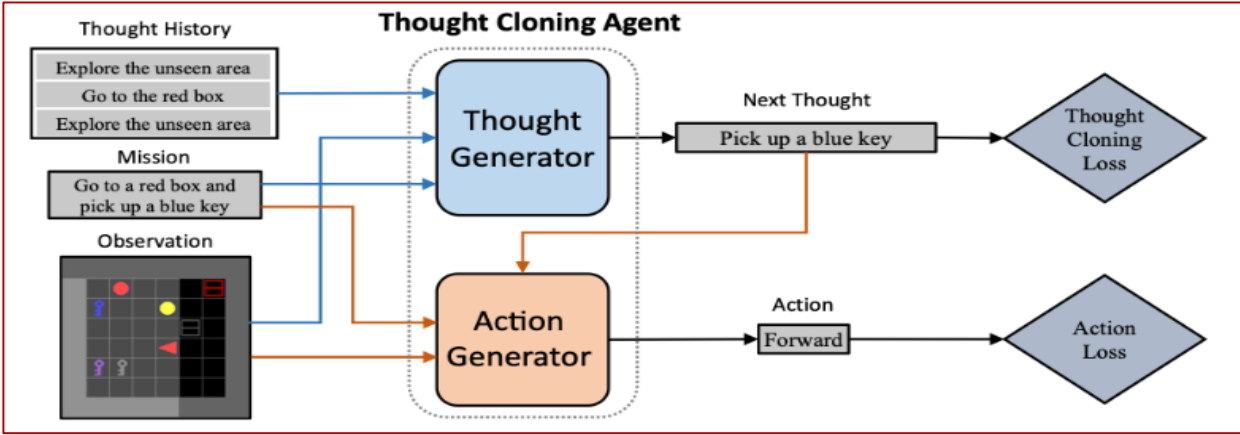
이런 점을 로봇에 적용, 임무 수행 성공률을 획기적으로 높였다는 연구 결과가 나왔다. 즉 언어모델을 활용해 '독백하는 로봇'을 만들었다는 설명이다.

사이언스는 8일(현지시간) 브리티시 컬럼비아 대학교 연구진이 '사고 복제(Thought Cloning)' 프레임워크를 도입한 AI 에이전트를 개발했다고 소개했다. 이 연구는 지난해 10월 열린 머신러닝 분야 세계 최대 학회 '뉴립스(NeurlPS) 2023'에서 발표했다.

이에 따르면 연구진은 20x20 그리드로 구성된 가상 2D 세계에서 일종의 '방 탈출 게임'과 같은 임무를 수행할 수 있는 AI 에이전트를 설계했다. AI는 방안에 배치된 다양한 색깔의 열쇠를 찾아 정확한 문을 열고 나와야 하는 임무가 주어졌다.

AI 에이전트는 자신이 향하고 있는 방향의 일부를 볼 수 있고, 이를 통해 얻은 시각적 정보에 미션 및 에이전트의 학습치를 더해 '계속 탐색하기 위해 파란색 문을 열어라' '보라색 상자로 이동하라' 등과 같은 명령을 내리도록 했다.

이는 기존 AI 에이전트의 움직임을 학습한 '행동 복제(Behavioural cloning)'를 적용한 경우다. 이는 기존 로봇에 사용되는 방식으로, 유튜브 영상을 보고 단순하게 따라 하는 식이다.



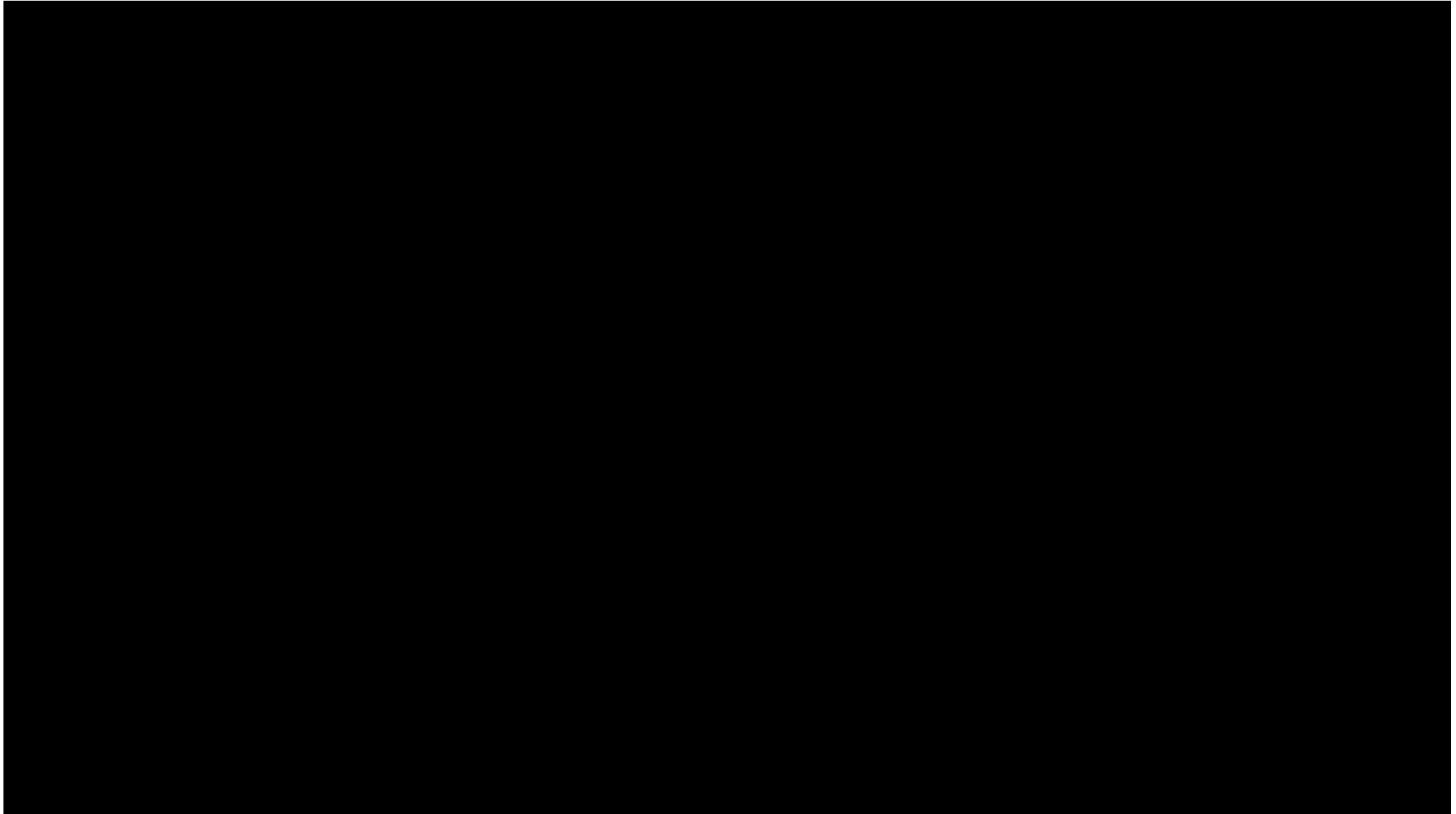
여기에 연구진은 사고 복제를 적용한 에이전트를 추가했다. 이는 앞선 에이전트와 달리, 행동을 결정하기 위해 시각 정보와 임무, 이에 따른 결정 사항 등을 '언어'로 재정리하도록 만든 것이다. 이를 통해 축적한 데이터로 다시 AI 에이전트를 재학습, 정확도를 끌어올린다는 의도다.

연구진은 복잡한 미로탈출 임무에서 사고 복제 AI 에이전트가 무려 80%의 성공률을 보였다고 전했다. 단순 행동 복제 에이전트는 30%에 그쳤다.

예를 들어 이미지나 비디오 입력이 가능한 'GPT-4 V'와 같은 모델에 내부 독백 구성 요소를 추가, 이를 통해 학습하는 내용을 더 유용하게 만들겠다고 밝혔다. 요리법이나 타이어 수리법, 포토샵으로 사진을 편집하는 법, 마인크래프트를 플레이하는 법 등에 적용, AI가 더 효과적인 솔루션을 찾아낼 수 있다는 말이다.

이에 대해 인지 및 언어학자인 안나 보르키 로마 사피엔자 대학교 심리학과 교수는 "가장 흥미로운 점은 언어의 존재가 AI에 유연성을 부여한다는 것"이라며 "복잡한 작업도 더 쉽게 수행할 수 있기 때문에 정말 효율적인 방식"이라고 밝혔다.

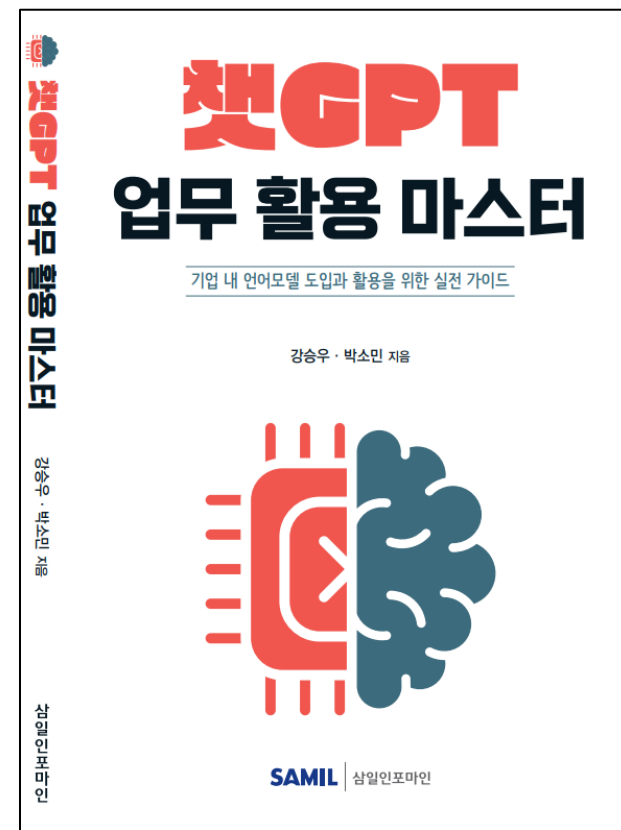
연구를 진행한 제프클론 브리티시 컬럼비아 대학교 컴퓨터 과학자는 "인간 지능의 강력한 사고 과정을 AI 에이전트에도 도입할 수 있다"라며 "독백은 인간에게는 단순한 작업이지만 AI에는 큰 도약이라고 할 수 있다"라고 말했다.





요약

- 임베딩 기술 : 자연어와 기계언어 통역 기술
- 자연어를 이해하는 ChatGPT
  - ✓ 자연어 기반 프로그래밍
- 추론 능력으로 발현되는 NLU
- 현실화 되는 지능화된 로봇
  - ✓ LLM + HW Robot



2024년 3월 출간 예정



감사합니다.