8장 sLLM 서빙하기

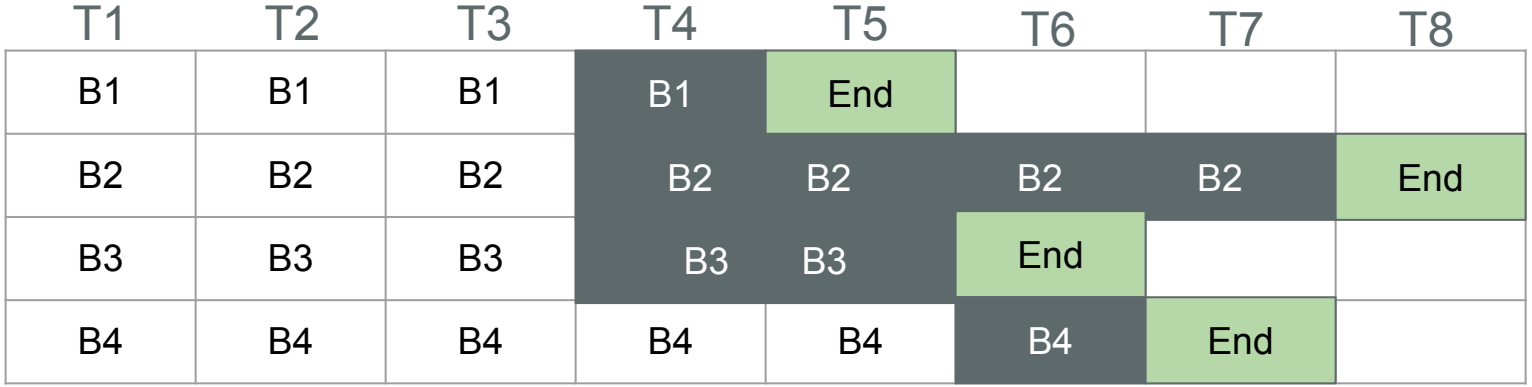
2024.11.05

최유진

PseudoLab study AI Application with LLM

8.1 효율적인 배치 전략

- 일반 배치(naive batching) 또는 정적 배치(static batching)
  - 한번에 N개의 입력을 받아 모두 추론이 끝날 때 까지 기다리는 방식
  - 가장 기본적인 방식



T8이 지나가야 새로운 문장 추론 시작!

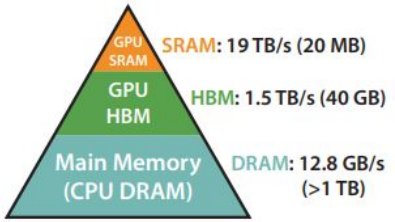
8.1 효율적인 배치 전략

- 연속 배치(continuous batching)
  - 한번에 들어온 배치 데이터의 추론이 모두 끝날때까지 기다리지 않고 하나의 토큰 생성이 끝날 때마다 생성이 종료된 문장은 제거하고 새로운 문장을 추가한다.

T1	T2	T3	T4	T5	T6	T7	T8
B1	B1	B1	B1	End	B5	B5	B5
B2	B2	B2	B2	B2	B2	B2	End
B3	B3	B3	B3	B3	End	B6	B6
B4	B4	B4	B4	B4	B4	End	B7

8.2 효율적인 트랜스포머 연산

- 플래시어텐션(FlashAttention)
  - 트랜스포머가 더 긴 시퀀스를 처리하도록 만들기 위해 개발되었다.
  - 어텐션 연산과정을 변경해 학습 과정에서 필요한 메모리를 시퀀스 길이에 비례하도록 개선
  - 블록 단위로 어텐션 연산을 수행하고 전체 어텐션 행렬을 쓰거나 읽지 않는 방식으로 어텐션 연산의 속도를 높였다.

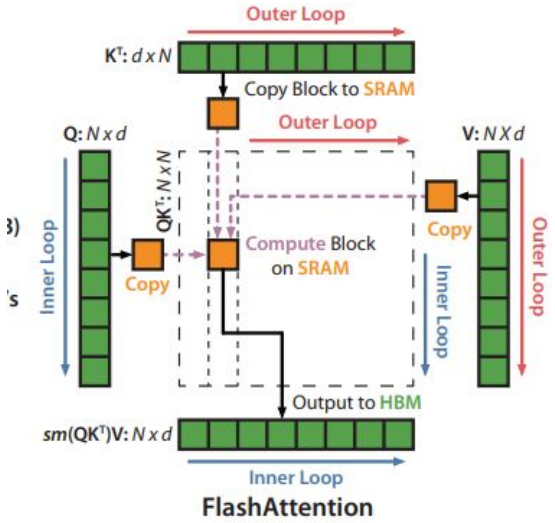


Memory Hierarchy with Bandwidth & Memory Size

어텐션 행렬에 취하는 연산

8.2 효율적인 트랜스포머 연산

- 플래시어텐션(FlashAttention)
  - 트랜스포머가 더 긴 시퀀스를 처리하도록 만들기 위해 개발되었다.
  - 어텐션 연산과정을 변경해 학습 과정에서 필요한 메모리를 시퀀스 길이에 비례하도록 개선
  - 블록 단위로 어텐션 연산을 수행하고 전체 어텐션 행렬을 쓰거나 읽지 않는 방식으로 어텐션 연산의 속도를 높였다.



8.2 효율적인 트랜스포머 연산

- 플래시어텐션2
  - 플래시어텐션 개선해 2배 정도 속도 향상
    - 순전파에서 최대 GPU 처리량의 73%
    - 역전파에서 63% 사용하도록 하ㅏㅁ
  - 개선한 부분
    - 행렬 곱셈이 아닌 연산 줄이기
    - 시퀀스 길이 방향의 병렬화 추가

8.2 효율적인 트랜스포머 연산

- 플래시어텐션2

FP16 BF16
행렬 곱셈 연산

최대 312 TFLOPS

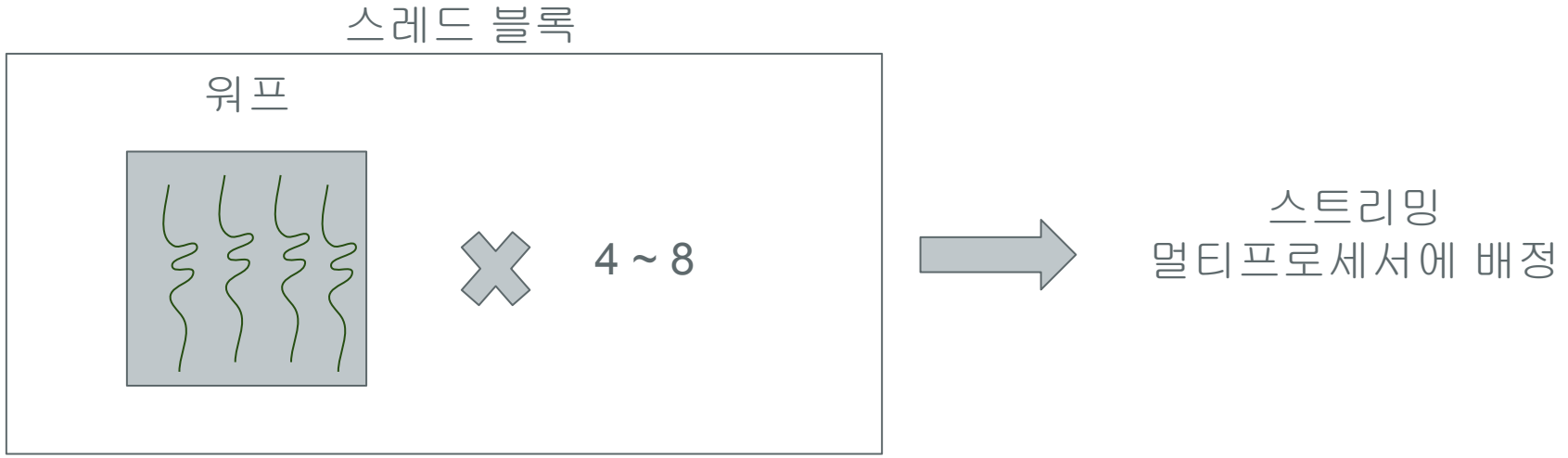
FP32
비 행렬 곱셈 연산

최대 19.5 TFLOPS

이 연산을 최대한 효율적으로 수행한다.

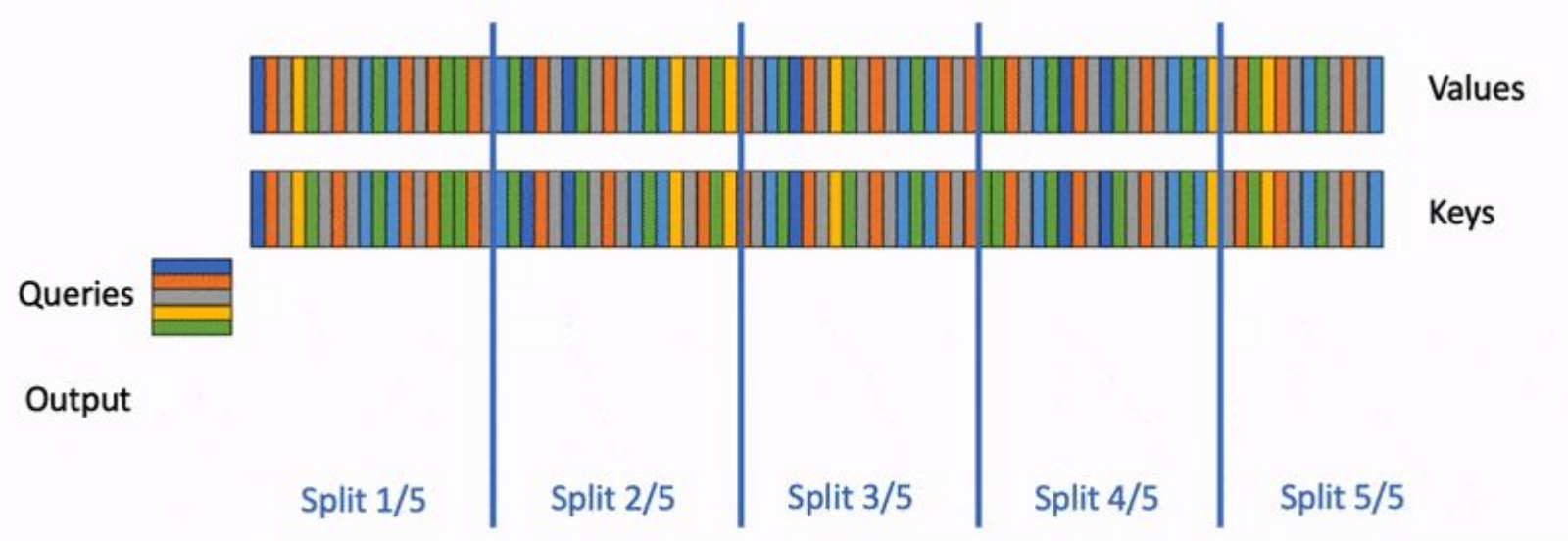
8.2 효율적인 트랜스포머 연산

- 플래시어텐션 2



8.2 효율적인 트랜스포머 연산

- 플래시어텐션 2



8.2 효율적인 트랜스포머 연산

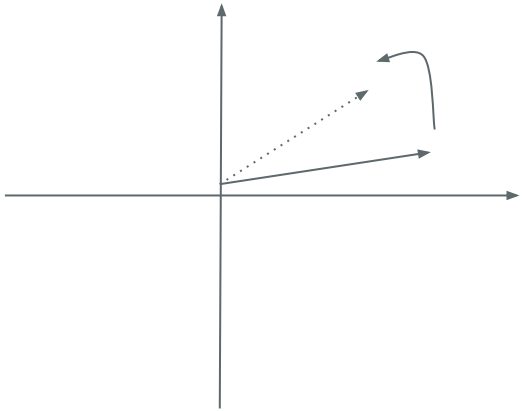
- 상대적 위치 인코딩
  - 토큰의 절대적인 위치에 따라 임베딩을 더하는 것이 아니라 토큰과 토큰 사이의 상대적인 위치 정보를 추가
  - 토큰의 상대적 위치에 따라 입력 문장의 의미가 달라지는지 학습할 수 있도록 한다.

작은 강아지가 공을 물고 뛰어다닌다

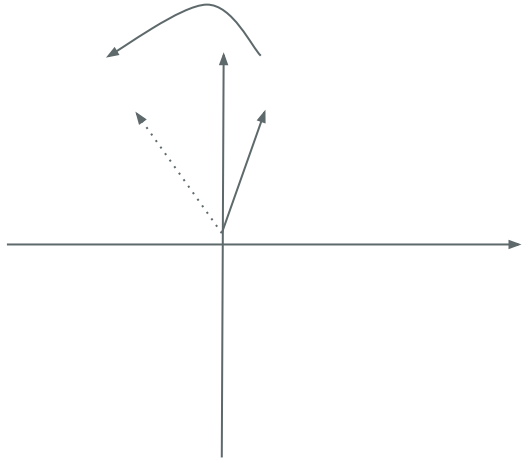
어제 저녁에 공원에서 작은 강아지가

8.2 효율적인 트랜스포머 연산

- 상대적 위치 인코딩
  - RoPE(Rotary Positional Encoding)
    - 토큰 사이의 위치 정보가 두 임베딩 사이의 각도를 통해 모델에 반영
  - ALiBI(Attention with Linear Biases)
    - 어텐션 행렬에 오른쪽에서 왼쪽으로 갈수록 더 작은 값을 더하는 방식



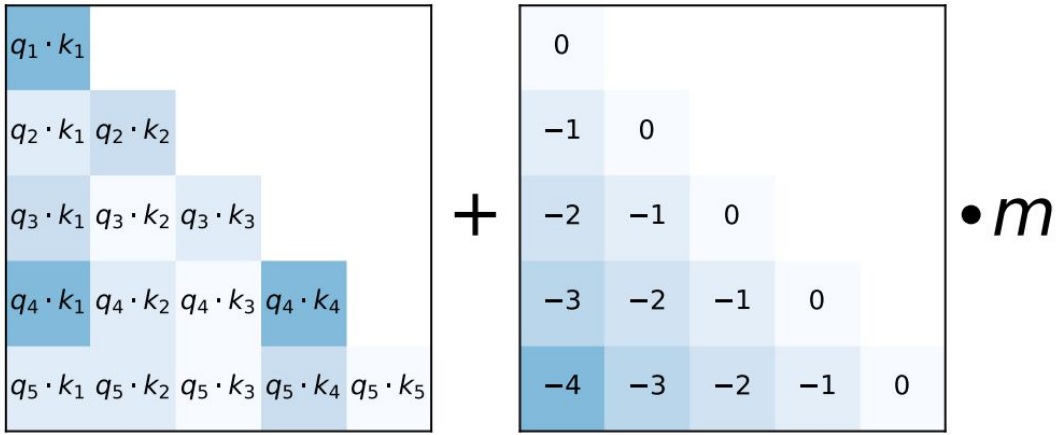
작은 강아지가 공을 물고 뛰어다닌다



어제 저녁에 공원에서 작은 강아지가

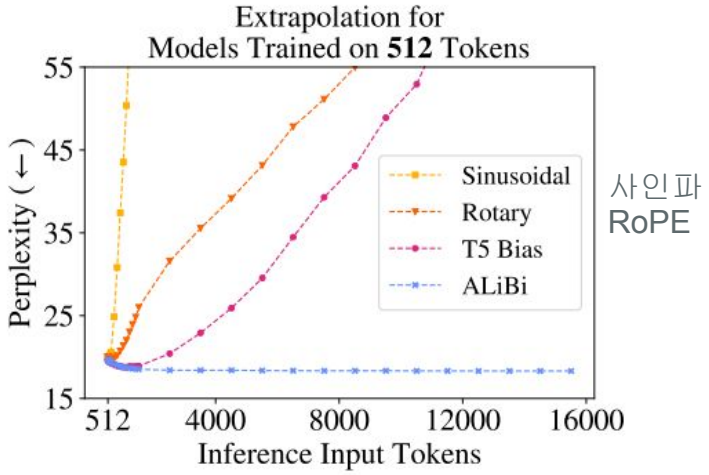
8.2 효율적인 트랜스포머 연산

- 상대적 위치 인코딩
  - RoPE(Rotary Positional Encoding)
    - 토큰 사이의 위치 정보가 두 임베딩 사이의 각도를 통해 모델에 반영
  - ALiBI(Attention with Linear Biases)
    - 어텐션 행렬에 오른쪽에서 왼쪽으로 갈수록 더 작은 값을 더하는 방식

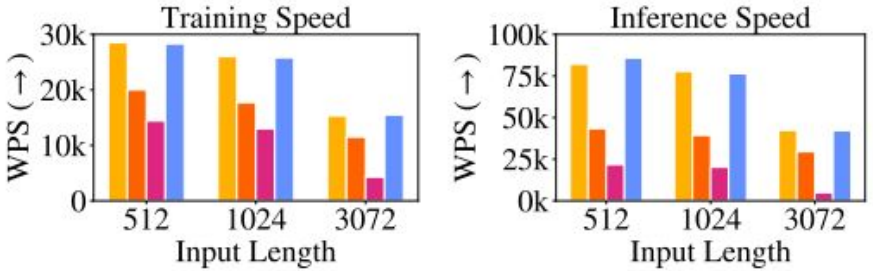


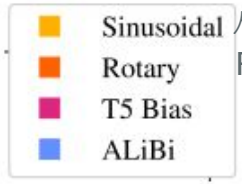
8.2 효율적인 트랜스포머 연산

- 상대적 위치 인코딩
  - RoPE(Rotary Positional Encoding)
    - 토큰 사이의 위치 정보가 두 임베딩 사이의 각도를 통해 모델에 반영
  - ALiBI(Attention with Linear Biases)
    - 어텐션 행렬에 오른쪽에서 왼쪽으로 갈수록 더 작은 값을 더하는 방식



사인파
RoPE

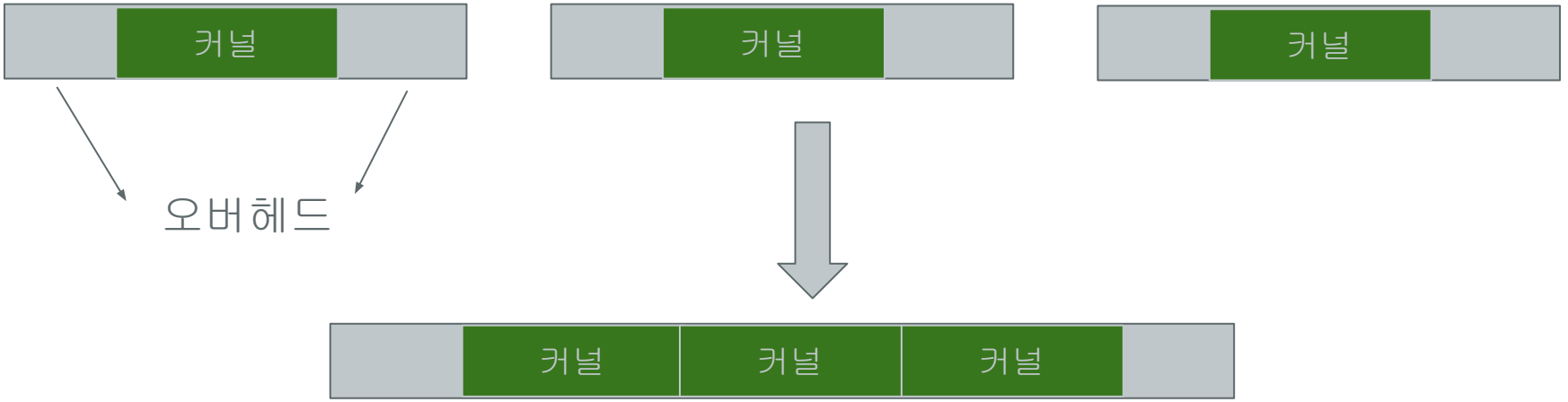




사인파
RoPE

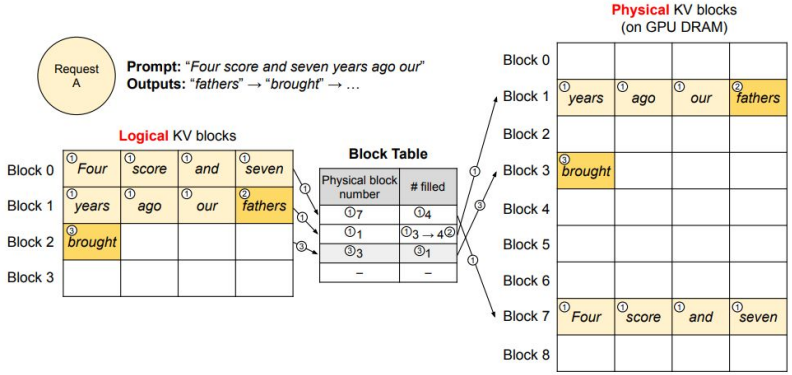
8.3 효율적인 추론 전략

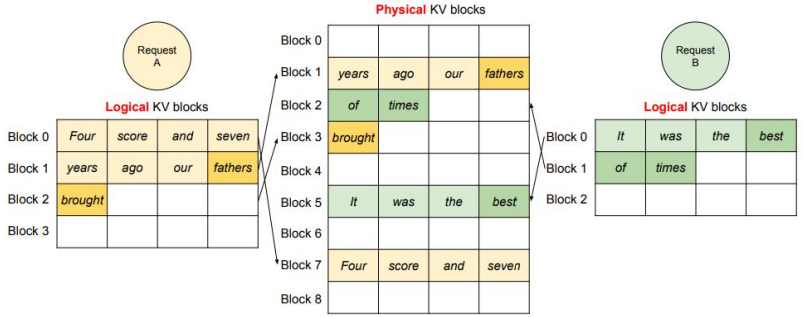
- 커널 퓨전
  - 반복적으로 수행하는 연산에 대해 연산을 하나로 묶어 오버헤드를 줄인다.



8.3 효율적인 추론 전략

- 페이지 어텐션
  - 중간에서 논리적 메모리와 물리적 메모리를 연결하는 블록테이블을 관리
  - 실제로는 물리적으로 연속된 메모리를 사용하지 않으면서도 논리적 메모리에서는 서로 연속적이도록 만든다.
  - 동일한 입력 프롬프트에서 여러 개의 출력을 생성하는 병렬 샘플링에서 입력 프롬프트에 대한 메모리를 공유함으로써 메모리 절약



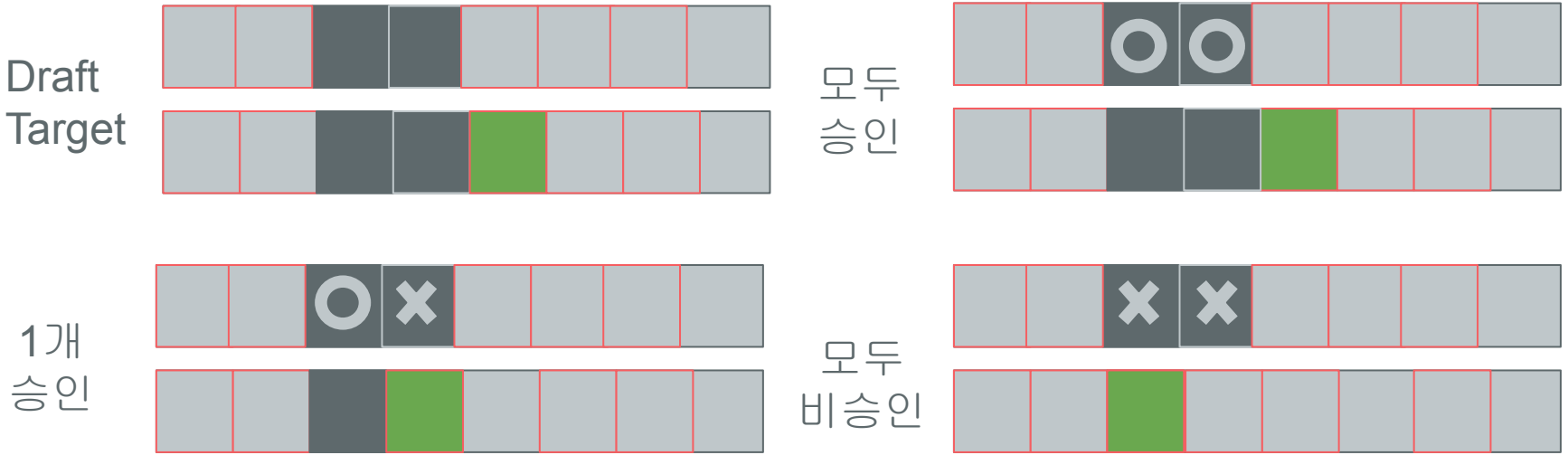


8.3 효율적인 추론 전략

- 추측 디코딩
  - 쉬운 단어는 더 작고 효율적인 모델이 예측하고 어려운 단어는 더 크고 성능이 좋은 모델이 예측하는 방식
  - 추론 수행 모델
    - 작은 드래프트 모델(draft model)
      - K개의 토큰을 먼저 생성
    - 큰 타깃 모델(target model)
      - 드래프트 모델이 생성한 K개의 토큰이 타깃 모델이 추론했다면 생성했을 결과와 동일한지 계산해 동일하다면 승인하고 동일하지 않다면 거절한다.

8.3 효율적인 추론 전략

- 추측 디코딩 추론 과정



8.3 효율적인 추론 전략

- 장점
  - 원본 모델에 비해 훨씬 작은 드래프트 모델의 추가 만으로 원본 모델의 성능을 그대로 유지하면서 속도를 2배 이상 높일 수 있다.
- 단점
  - 2개의 모델을 이용하기에 시스템 복잡도가 올라간다.
  - 이를 해결하기 위해 하나의 원본 모델 내에서 여러 토큰을 예측하는 Medusa가 있다.