AI

Red Hat

추론

시작하기

효율성 향상을 위한 여정 가속화

목차

개요

AI 모델 추론을 최적화하는 것은 인프라 비용을 절감하고 대기 시간을 줄이며 처리량을 높이는 효과적인 방법 중 하나로, 특히 조직이 대규모 모델을 프로덕션 환경에 배포할 때 더욱 유용합니다.

이 E-book에서는 추론 성능 엔지니어링과 모델 최적화의 기본 사항을 소개하며, 컴퓨팅 및 메모리 요구 사항을 줄이는 데 도움이 되는 양자화, 희소화 및 기타 기술에 중점을 둡니다. 또한 효율적인 추론을 지원하는 가상 대형 언어 모델(vLLM)과 같은 런타임 시스템도 함께 살펴봅니다.

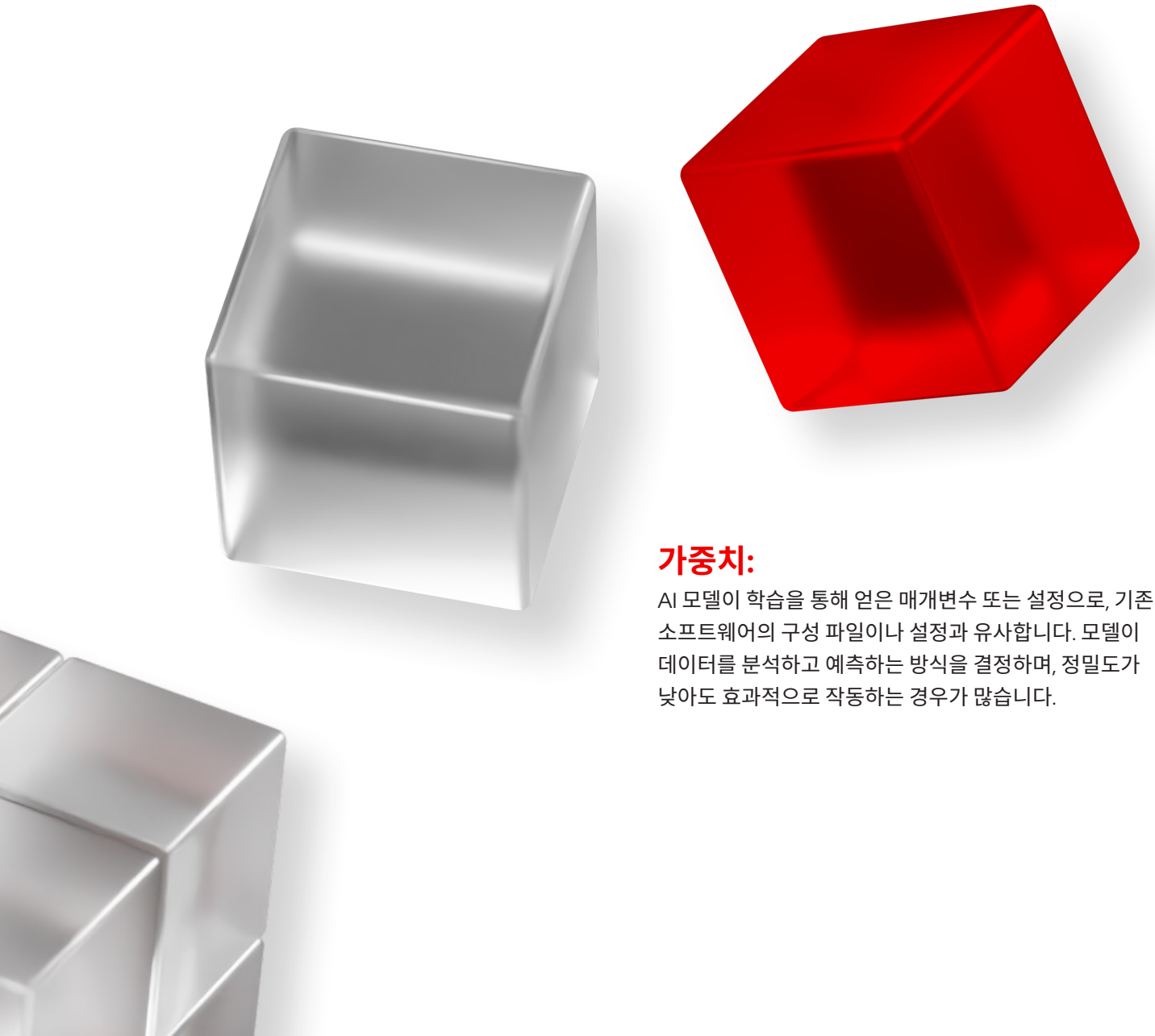
또한 Red Hat의 개방형 접근 방식, 검증된 모델 리포지토리, 그리고 LLM Compressor 및 Red Hat® AI Inference Server와 같은 툴을 활용할 때의 장점도 설명합니다. 그래픽 처리 장치(GPU), 텐서 처리 장치(TPU) 또는 기타 가속기에서 실행할 때에도 이 가이드는 더욱 스마트하고 효율적인 AI 추론 시스템을 구축하는 데 도움이 되는 실질적인 인사이트를 제공합니다.

한눈에 보는 주요 용어

모델 구성 요소 이해

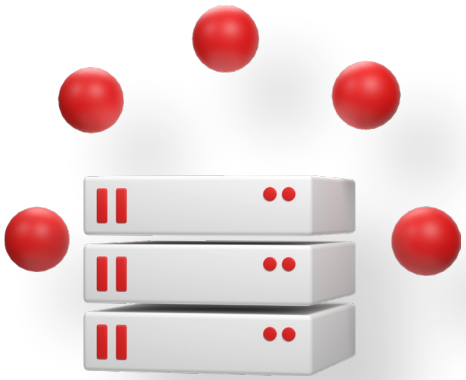
활성화:

모델이 정보(입력 토큰)를 처리하는 과정에서 생성되는 일시적인 데이터로, 계산 중에 생성되는 중간 결과와 유사합니다. 일반적으로 정확한 결과를 내려면 높은 정밀도가 필요합니다.



가중치:

AI 모델이 학습을 통해 얻은 매개변수 또는 설정으로, 기존 소프트웨어의 구성 파일이나 설정과 유사합니다. 모델이 데이터를 분석하고 예측하는 방식을 결정하며, 정밀도가 낮아도 효과적으로 작동하는 경우가 많습니다.



양자화(Quantization)

양자화는 AI 모델의 매개변수(가중치)와 중간 데이터(활성화)를 값당 사용하는 비트 수를 줄여 더 낮은 정밀도의 형식으로 저장함으로써 모델의 크기와 리소스 요구 사항을 줄입니다. 이 기술은 리소스를 효율적으로 관리하는 데 도움이 되며, 컴퓨터에서 파일을 압축하는 것과 유사합니다. 올바르게 적용하면 **모델의 성능이 크게 저하되지 않습니다**.

- **가중치 양자화**는 모델 매개변수의 스토리지 크기를 줄이며, **따라서 추론 중에 메모리를 더 효율적으로 사용할 수 있습니다**.[1]
- **활성화 양자화**는 추론 중 발생하는 중간 출력(임시 데이터)의 메모리 요구 사항을 최소화하여 **더욱 빠르고 효율적인 실행을 구현합니다**.[2]
- **KV 캐시 양자화**는 캐시된 키-값 텐서의 메모리 풋프린트를 줄여 모델이 **긴 프롬프트와 동시 요청을 더 효율적으로 처리하도록 지원합니다**.[3]

16비트, 8비트, 4비트 양자화에서의 정밀도 수준:

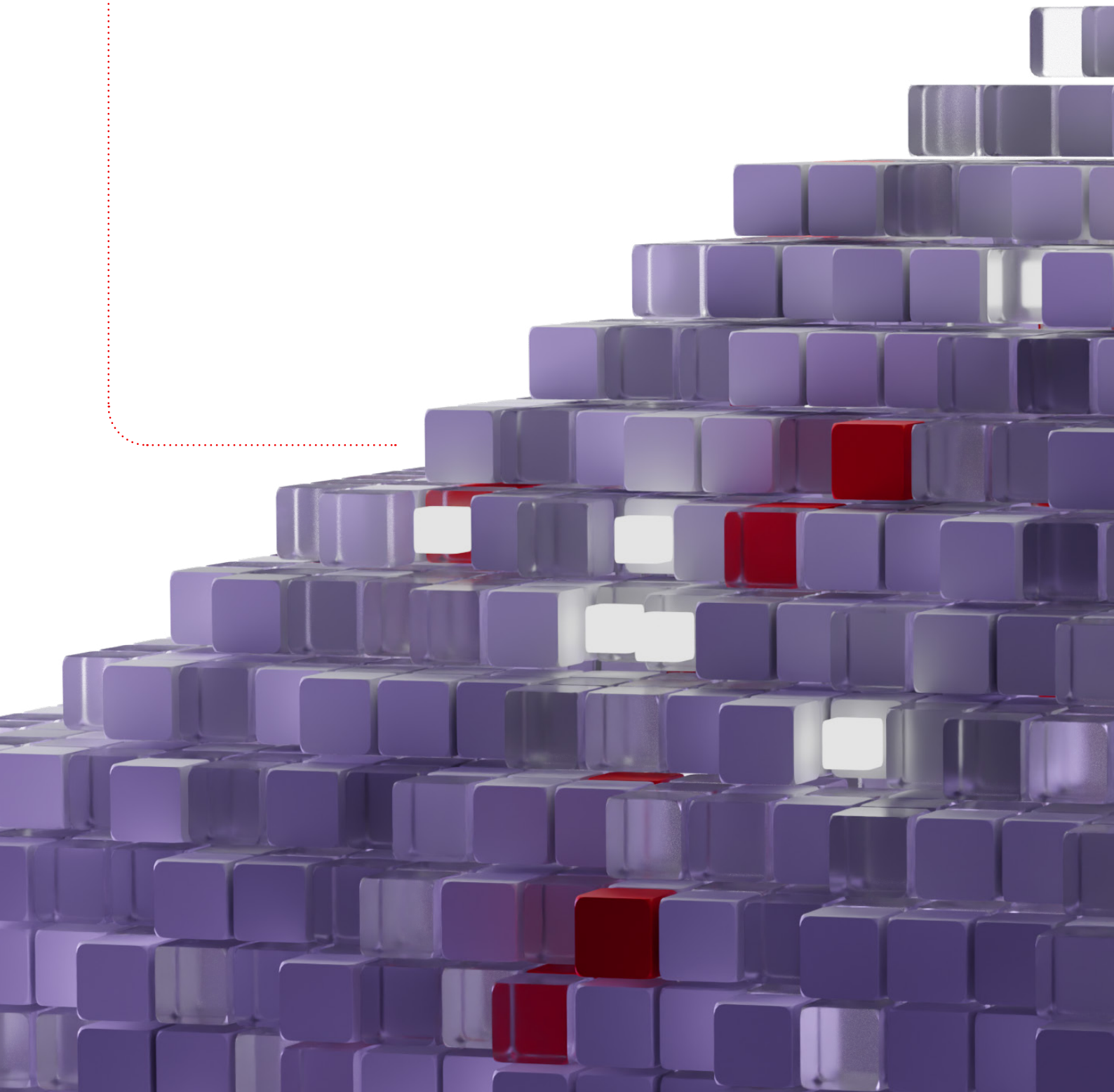
- **16비트(FP16/BF16)**는 표준 정밀도로, 정확도는 유지하지만 상당한 메모리를 요구하므로 매우 대규모의 모델에서는 비용 부담이 커집니다.
- **8비트(FP8/INT8)**는 16비트와 비교하면 메모리 사용량을 약 절반으로 줄이므로, 모델 정확도를 유지하면서도 효율성을 크게 향상합니다.
- **4비트(INT4)**는 모델 크기와 메모리 요구 사항을 대폭 줄이므로 더 적은 리소스로 배포하도록 지원하지만, 고급 양자화 방법으로 주의 깊게 관리하지 않으면 정확도가 눈에 띄게 저하될 수 있습니다.

1 Laboone, Maxime. "**가중치 양자화 개요**." *데이터 사이언스를 지향하며*, 2023년 7월 7일.
2 "**AWQ: LLM 압축 및 가속을 위한 활성화 인식 가중치 양자화**." GitHub, 2025년 8월 8일 액세스.
3 Turganbay, Raushan. "**키-값 캐시 양자화를 통한 더 긴 생성 지원**." *Hugging Face*, 2024년 5월 16일.

희소화를 통한 컴퓨팅 부하 감소

희소화는 모델의 일부 매개변수를 의도적으로 0으로 설정해 시스템이 불필요한 연산을 우회하도록 하는 방식으로 컴퓨팅 요구 사항을 줄입니다. 마치 양식에서 빈칸을 건너뛰는 것과 비슷합니다. 이 방식은 모델을 완전히 재학습시킬 필요 없이 속도와 효율성을 높입니다.

2:4 희소화는 4개의 매개변수 중 정확히 2개를 0으로 설정하는 **구조화된 접근 방식**입니다. 전문적인 하드웨어는 이 방식을 통해 비활성 매개변수 블록을 빠르게 식별하고 효율적으로 우회하여 계산 시간을 단축하여 성능을 더욱 높일 수 있습니다.

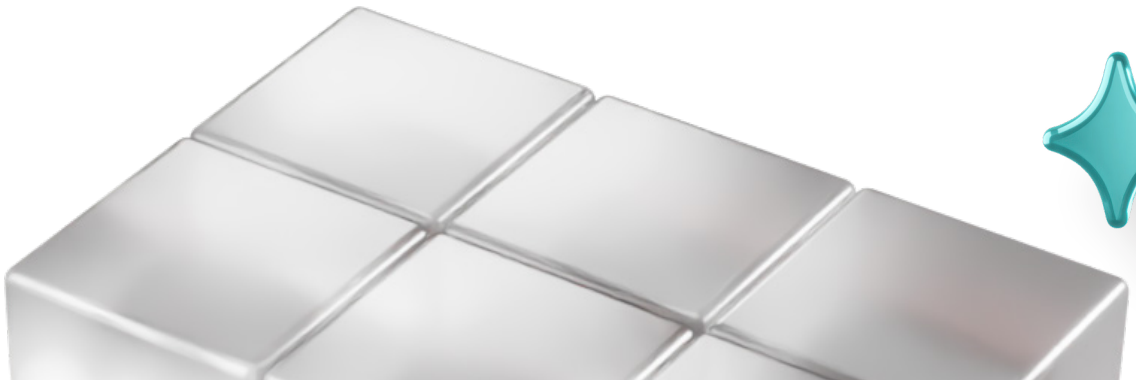


대규모 언어 모델의 진화

대규모 언어 모델(LLM)은 주로 트랜스포머 아키텍처를 기반으로 하며, 연구 단계의 실험에서 실제 애플리케이션을 구동하는 핵심 툴로 진화해 왔습니다. 수십억에서 수천억 개에 이르는 매개변수 규모를 바탕으로 높은 수준의 추론, 창의성, 그리고 도메인 특화 능력을 제공합니다. 이러한 기능은 추론이라는 과정을 통해 실행됩니다.

추론은 학습이 완료된 모델이 새로운 입력 데이터를 처리하여 출력을 생성하는 과정으로, 문장에서 다음 단어를 예측하거나 이미지에서 객체를 식별하는 작업 등이 이에 해당합니다. 대규모 데이터세트를 학습해야 하는 학습과 달리, 추론은 학습된 지식을 적용하여 실시간으로 의사결정을 내리는 데 중점을 둡니다. 따라서 추론은 특히 인터랙티브 애플리케이션, 실시간 분석 또는 대규모 자동화를 지원하기 위해 모델이 프로덕션 환경에 배포될 때 매우 빠르고 효율적이어야 합니다.

추론 모델은 텍스트, 이미지, 오디오와 같은 입력 데이터를 토큰으로 처리하며, 이를 다중 계층 트랜스포머 아키텍처에 통과시켜 예측을 생성합니다. 토큰은 모델이 처리하기 전에 입력 데이터를 나누는 개별 단위입니다. 텍스트 기반 모델에서 토큰은 사용된 토큰화 전략에 따라 개별 문자, 서브워드 또는 전체 단어를 나타낼 수 있습니다.

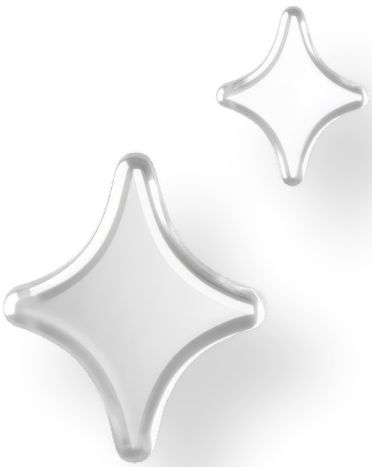


이러한 모델은 입력 토큰을 다중 계층 트랜스포머 아키텍처에 통과시키며, 일련의 수학적 연산을 적용하여 컨텍스트를 분석하고 관계를 평가하며 가능성이 높은 출력을 결정합니다. 각 계층은 입력에 대한 모델의 이해를 점진적으로 정교화하여 최종적으로 한 번에 하나의 토큰씩 예측을 생성합니다. 이 단계별 토큰 생성 방식은 매우 정확하고 컨텍스트에 적합한 출력을 만들어내지만, 특히 계층 수가 많은 대규모 모델에서는 추론 워크로드의 컴퓨팅 부담을 증가시키는 요인이 되기도 합니다.

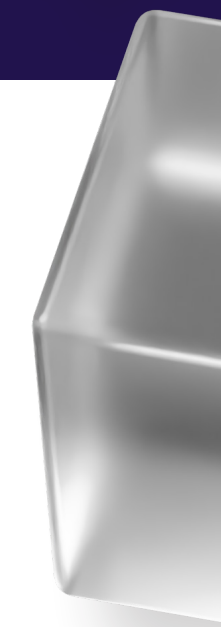
텍스트 기반 LLM을 넘어, 유사한 아키텍처는 이제 비전 모델과 멀티모달 시스템을 포함한 다양한 AI 도메인의 기반이 되고 있습니다. 비전 모델은 토큰 기반 트랜스포머 컴퓨팅의 동일한 원리를 이미지와 영상에 적용합니다. 텍스트를 토큰으로 나누는 대신 픽셀 데이터를 임베딩으로 변환합니다. 이러한 임베딩은 시각적 요소 간의 공간적 패턴, 경계, 텍스처, 관계를 포착하여 모델이 이미지 분류, 객체 감지, 분할, 시각적 질의응답과 같은 태스크를 수행할 수 있도록 합니다. 비전 모델이 프로덕션에 배포되면 자동 검사, 의료 영상, 콘텐츠 조정과 같은 활용 사례를 지원할 수 있습니다.

조직이 AI를 더욱 폭넓게 도입함에 따라 모델 아키텍처는 계속해서 규모와 복잡성이 증가하고 있습니다. Mixture of Experts(MoE)와 같은 새로운 접근 방식은 추론 시 모델의 일부만 활성화하여 필요한 전체 컴퓨팅을 줄임으로써 성능을 확장하는 것을 목표로 합니다. 이러한 혁신은 성능과 비용, 에너지 요구 간 균형을 유지하면서 더욱 강력한 모델로 나아갈 수 있는 기반을 마련합니다.

규모와 관계없이 모든 모델은 프로덕션 환경에서 실질적으로 활용되기 위해 효율적인 서빙과 최적화가 필요합니다. 따라서 모델 배포를 추진하는 조직은 추론 성능 엔지니어링을 매우 중요한 우선순위로 둡니다.



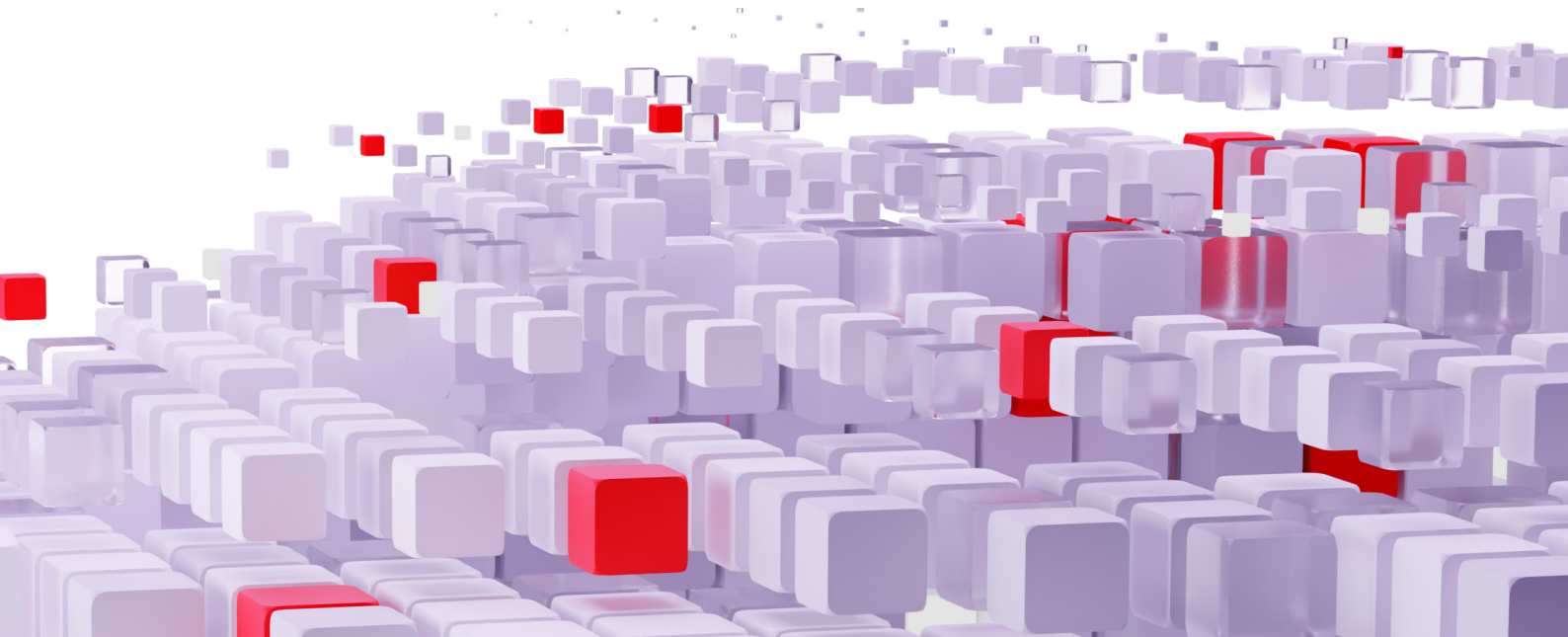
추론 서빙의 과제



대규모 모델의 추론 서빙은 여러 가지 과제를 수반합니다.

수십억 개의 매개변수를 가진 모델은 가중치와 키-값(KV) 캐시와 같은 중간 상태를 저장하기 위해 상당한 GPU 메모리를 필요로 합니다. 동시 요청 수가 증가하거나 입력 길이가 길어질수록 메모리 제약은 주요 장애물로 작용하며, 모델의 처리량과 응답성을 제한합니다. 기본적인 서빙 방식은 비효율적인 배치 기술로 인해 하드웨어 리소스를 충분히 활용하지 못하고 대기 시간을 늘리는 경우가 많습니다.

또한 트랜스포머 아키텍처에서 어텐션 메커니즘 구현은 특히 입력이 길어질수록 컴퓨팅 집약도가 높아져 응답 속도가 크게 느려질 수 있습니다. 이러한 과제를 해결하려면 효율적인 메모리 관리, 고급 배치 전략, 페이지드 어텐션과 같은 최적화된 어텐션 메커니즘 등 정교한 런타임 최적화가 필요합니다. 이러한 요소들이 결합하여 실제 애플리케이션에서 성능과 응답성을 향상합니다.



추론 성능에 대한 풀스택 접근 방식

추론 최적화는 AI 모델이 프로덕션 환경에 배포된 이후 모델이 실행되는 효율성을 개선하는 과정을 의미합니다. LLM을 프로덕션 환경에서 운영하면 특히 높은 토큰 처리량, 긴 프롬프트, 증가하는 사용량 수요를 처리해야 하는 경우 비용이 빠르게 늘어날 수 있습니다. **추론에서 비용 최적화는 정확도나 사용자 경험을 저해하지 않으면서 메모리 사용량을 줄이고 처리량을 높이며 하드웨어 요구 사항을 최소화하는 데 달려 있습니다.**

모델 학습은, 모델 재학습과 같은 예외를 제외하면 단일 인스턴스로 수행되는 태스크인 반면, 추론은 사용자 입력에 대응해 실시간 출력을 생성하며 지속적으로 발생합니다. LLM과 비전 모델의 경우, 특히 하이브리드 또는 글로벌 인프라 전반으로 확장될 때 추론은 AI 배포에서 가장 비용이 많이 들고 리소스 집약적인 요소가 될 수 있습니다.

대규모 LLM을 효과적으로 서빙하려면 모델 자체와 서빙 런타임을 모두 처리하는 포괄적인 풀스택 최적화 전략이 필요합니다. **양자화**와 **희소화**를 통해 모델 매개변수를 최적화하는 데 중점을 두고 있지만, **청크 단위 프리필**,[4] **프리픽스 캐싱**,[5] **추측 디코딩**,[6] **프리필 및 디코딩 분리**[7]와 같은 기술을 사용해 추론 서빙 프로세스를 개선하면 추가적인 성능 향상을 실현할 수 있습니다.

4 "최적화 및 튜닝." *vLLM*, 2025년 8월 7일.

5 "자동 프리픽스 캐싱이란 무엇인가?" *vLLM*, 2025년 8월 8일 액세스.

6 "추측 디코딩으로 vLLM 성능을 최대 2.8배 향상하는 방법." *vLLM*, 2024년 10월 17일.

7 Du, Kuntai. "vLLM 오피스 아워 - vLLM의 프리필 및 KV 캐시 스토리지 분리 - 2024년 11월 14일." *YouTube*, 2024년 11월 18일.

추론 런타임 및 모델 형식 개요

기본 런타임은 장애물로 작용하므로, 대규모 모델을 효율적으로 서빙하려면 적절한 추론 런타임을 선택해야 합니다. 대중적인 런타임은 다음과 같습니다.

- **vLLM**. 가상 대형 언어 모델은 vLLM 커뮤니티에 의해 유지 관리되는 오픈소스 코드 라이브러리입니다. vLLM은 LLM이 계산을 더욱 효율적이고 대규모로 수행할 수 있도록 돕습니다. 구체적으로 설명하면 vLLM은 GPU 메모리를 더욱 효율적으로 활용하여 생성형 AI 애플리케이션의 출력을 가속화하는 추론 서버입니다. GPU 메모리에서 더 많은 토큰을 효율적으로 처리할 수 있게 해주는 페이지드 어텐션과 같은 혁신 덕분에 뛰어난 처리량과 낮은 대기 시간 성능을 제공하며 업계 전반에서 널리 채택되고 있습니다.

- **Triton**. Triton은 종종 독립 실행형 런타임이라는 오해를 받기도 하지만, 실제로는 TensorRT 및 vLLM을 포함한 다양한 백엔드 엔진을 위한 프론트엔드 애플리케이션 프로그래밍 인터페이스(API) 역할을 합니다. TensorRT 기반의 Triton은 NVIDIA GPU에서 약간 더 높은 성능을 제공할 수 있지만, 설정 복잡성이 증가하고 지원되는 모델이 제한적이라는 단점이 있습니다. 고객은 Triton으로 성능 향상을 달성하는 데 vLLM보다 훨씬 더 많은 노력이 필요하다고 보고하는 경우가 많습니다.

- **SGLang**. 비교적 최근에 등장한 SGLang은 vLLM에서 파생되었으며 특정 활용 사례에 최적화되어 있습니다. vLLM과 동일한 기본 구성 요소를 많이 사용하지만 지원하는 모델 아키텍처는 더 적습니다. 특정 환경에서는 vLLM보다 더 나은 성능을 보일 수 있지만 유연성과 커뮤니티 지원이 제한적이어서 기업 전반에서 폭넓게 도입하기에는 장점이 부족합니다.

모델 효율성에 대한 이중 접근 방식

1: 추론 런타임 최적화(vLLM)

런타임의 제한 사항

앞선 장에서 언급했듯이, **LLM을 효율적으로 서빙하는 것은 기본적인 추론 서빙 방식에 내재된 한계로 인해 어려울 수 있습니다**.

이러한 런타임 제한 사항에는 비효율적인 GPU 메모리 사용, 최적화되지 않은 배치 처리, 느린 토큰 생성이 포함됩니다. 런타임은 일반적으로 KV 캐시와 같은 중간 컴퓨팅 데이터를 비효율적으로 저장하여 GPU 메모리를 과도하게 사용하고 동시 요청 처리 용량을 제한합니다. 또한 단순한 배치 전략은 GPU를 유휴 상태로 두거나 충분히 활용하지 못하게 만들어 처리량을 크게 저하시킵니다. 게다가 기본 런타임은 느린 어텐션 메커니즘으로 인해 긴 입력 시퀀스를 처리할 때 대기 시간이 길어지는 문제가 있습니다.

vLLM을 선택해야 하는 이유

vLLM은 추론 성능에 특별히 최적화된 다음과 같은 고급 기술을 제공하여 다양한 런타임 과제를 해결합니다.

- **연속 배치**. vLLM은 여러 유입 요청의 토큰을 동시에 처리하여 GPU 유휴 시간을 최소화합니다. 요청을 한 번에 하나씩 처리하는 대신 여러 시퀀스의 토큰을 배치로 그룹화하여 GPU 활용도와 추론 처리량을 크게 향상시킵니다.

- **PagedAttention**. vLLM은 PagedAttention이라는 새로운 메모리 관리 전략을 사용하여 대규모 KV 캐시를 효율적으로 처리합니다. 이 기술은 GPU 메모리 페이지를 동적으로 할당하고 관리하여 동시에 처리 가능한 요청 수를 크게 늘리고 메모리 병목 현상 없이 훨씬 더 긴 시퀀스를 지원합니다.

자세한 내용은 **vLLM 관련 기술 블로그**를 참고하시기 바랍니다.

vLLM 배포의 장점

포괄적인 통합 기능: vLLM은 Hugging Face와 같은 대중적인 리포지토리에서 모델을 직접 로드할 수 있으며, Triton Inference Server와 같은 프레임워크 내에서 고성능 백엔드 역할을 수행합니다. NVIDIA GPU, AMD GPU, Google TPU를 포함한 다양한 하드웨어 플랫폼과의 호환성을 통해 기업 규모의 배포를 더욱 간소화할 수 있습니다.

표준화 및 벤더 중립성: vLLM과 같이 널리 사용되는 런타임을 활용하면 표준화의 장점을 누릴 수 있으며, 이는 다양한 하드웨어 환경에서 신뢰할 수 있는 성능을 지원하고 독점 솔루션에 종속되는 것을 방지합니다.

vLLM의 병렬 처리 기술에 대해 더 깊이 이해하려면 이 **기술 심층 분석 블로그**를 방문해 보시기 바랍니다.

2: AI 모델 최적화

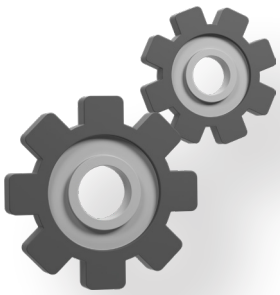
대규모 언어 모델 최적화의 중요성

프로덕션 환경에서의 주요 과제 중 하나는 메모리와 컴퓨팅 효율성을 관리하는 것입니다. 대규모 모델은 특히 긴 프롬프트나 다수의 동시 요청을 처리할 때 매개변수와 KV 캐시 내의 컨텍스트를 저장하기 위해 막대한 GPU 메모리를 필요로 합니다. 모델이 최적화되지 않으면 비효율적으로 실행되어 운영 비용이 증가할 수 있습니다. 대기 시간 또한 중요한 요소입니다. 사용자는 실시간 응답을 기대하기 때문에 대규모 모델 크기나 비효율적인 실행으로 인한 지연은 사용자 경험과 다운스트림 워크플로우의 효능에 부정적인 영향을 미칠 수 있습니다.

모델을 압축하는 이유

모델 압축은 조직이 AI를 대규모로 배포할 때 직면하는 주요 과제인 비용 효율성과 성능 최적화를 해결하는 데 도움이 됩니다.

모델 규모가 수십억 개의 매개변수로 커질수록 이를 프로덕션 환경에서 서빙하는 것은 막대한 메모리와 컴퓨팅 성능을 요구하는 리소스 집약적인 작업이 됩니다. 양자화와 희소화를 포함한 모델 압축 기술은 정밀도와 매개변수 수를 약간 줄이는 대신, 정확도를 크게 저해하지 않으면서도 메모리 풋프린트와 컴퓨팅 요구 사항을 크게 낮춥니다. 모델을 압축함으로써 조직은 더욱 적은 수의 GPU나 다른 가속기를 사용하여 AI 워크로드를 더욱 효율적으로 실행할 수 있으며, 이를 통해 운영 비용을 대폭 절감하고 실시간 응답이 필요한 애플리케이션에 필수적인 빠른 추론을 구현할 수 있습니다.



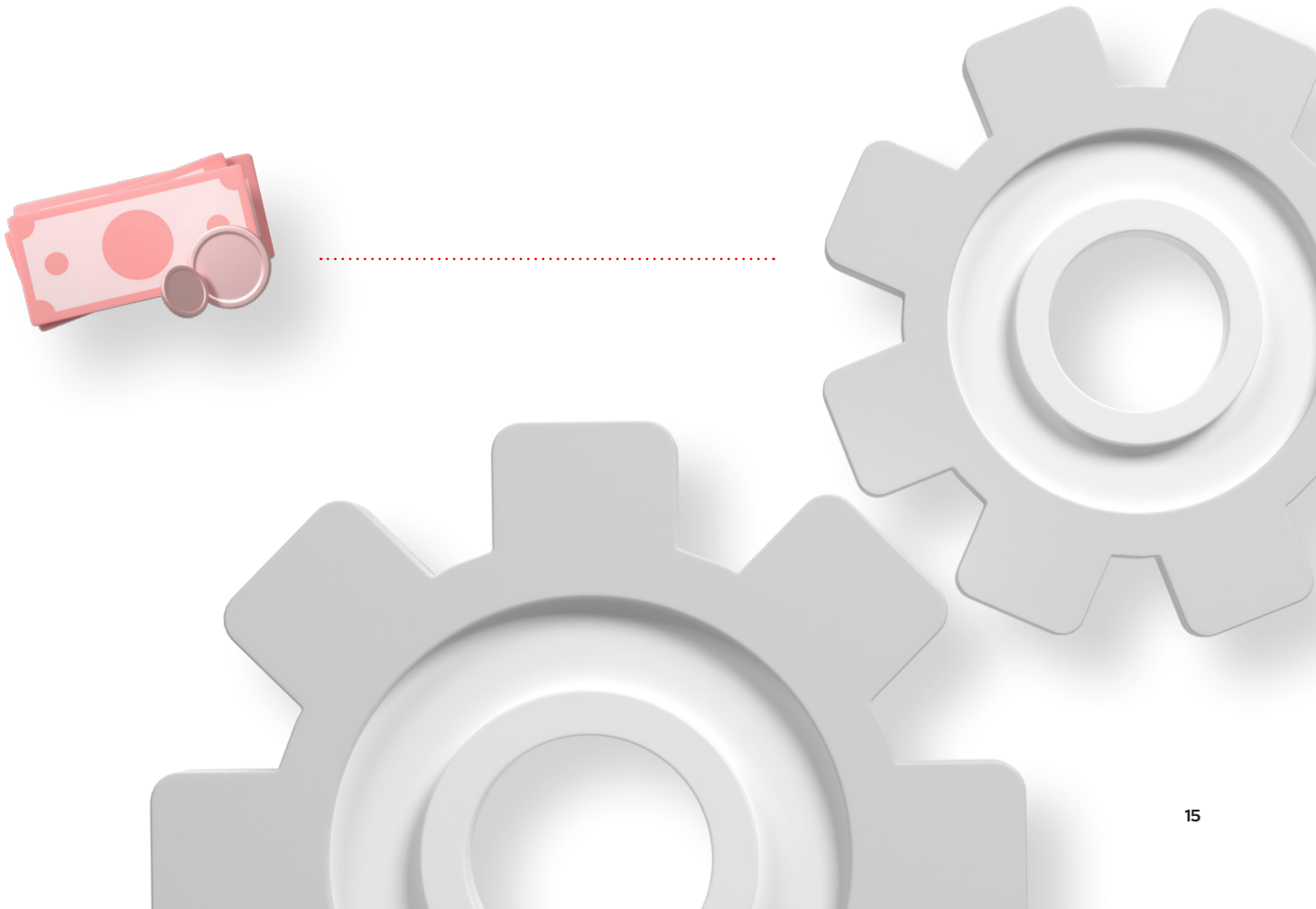
추론 비용을 최적화하려면 모델을 어떻게 해야 하나요?

그러한 비용을 줄이는 가장 효과적인 방법 중 하나는 모델을 압축하는 것입니다. 양자화와 희소화 같은 압축 기술은 모델 크기를 줄이고 컴퓨팅 요구 사항을 낮춰, 더 적은 수의 GPU나 더 작은 GPU에서도 추론 워크로드를 실행할 수 있게 합니다.

양자화는 수치 값, 구체적으로는 모델의 가중치와 활성화 값의 정밀도를 낮춰 모델을 최적화합니다. 일반적으로 모델은 FP16 또는 BF16과 같은 형식을 사용하여 16비트 정밀도(또는 32비트 정밀도)로 작동합니다.

양자화는 이러한 값을 8비트(INT8 또는 FP8) 또는 4비트 정수(INT4)와 같은 더 낮은 정밀도 형식으로 압축합니다. 이 과정은 모델 매개변수를 저장하는 데 필요한 메모리를 크게 줄이며, 이를 통해 700억 개 매개변수를 가진 Llama와 같은 모델을 약 140GB에서 최저 40GB 수준까지 줄일 수 있습니다. 이러한 감소는 추가 컴퓨팅을 위한 메모리를 확보할 뿐만 아니라 특히 메모리 대역폭이 제한된 상황에서 처리량을 향상합니다. 예를 들어 48GB VRAM을 가진 GPU는 140GB 모델보다 40GB 모델을 더 빠르게 처리합니다.

그러나 양자화를 과도하게 하면 정밀도가 하락하여 정확도에 영향을 줄 수 있습니다. 그러한 문제를 완화하기 위해 세밀한 양자화는 모델 정확도를 유지하는 스케일링 계수를 사용하여 보통 정확도 저하를 1% 미만으로 억제합니다. 양자화는 하드웨어 활용을 최적화하여 컴퓨팅 처리량을 두 배로 늘릴 수 있으며, 결과적으로 대기 시간과 운영 비용이 크게 줄어듭니다.



희소화는 매개변수를 구조적으로 줄여 모델을 최적화하는 기술로, 본질적으로 모델 가중치의 상당 부분을 0으로 설정합니다. 이 기술은 중복되거나 중요도가 낮은 가중치를 식별하고 제거하여 추론 시 컴퓨팅을 단순화합니다. 희소화는 모델 복잡성을 크게 줄이고 메모리 사용량과 컴퓨팅 부하를 줄여 추론 속도를 높이고 운영 비용을 절감할 수 있습니다.

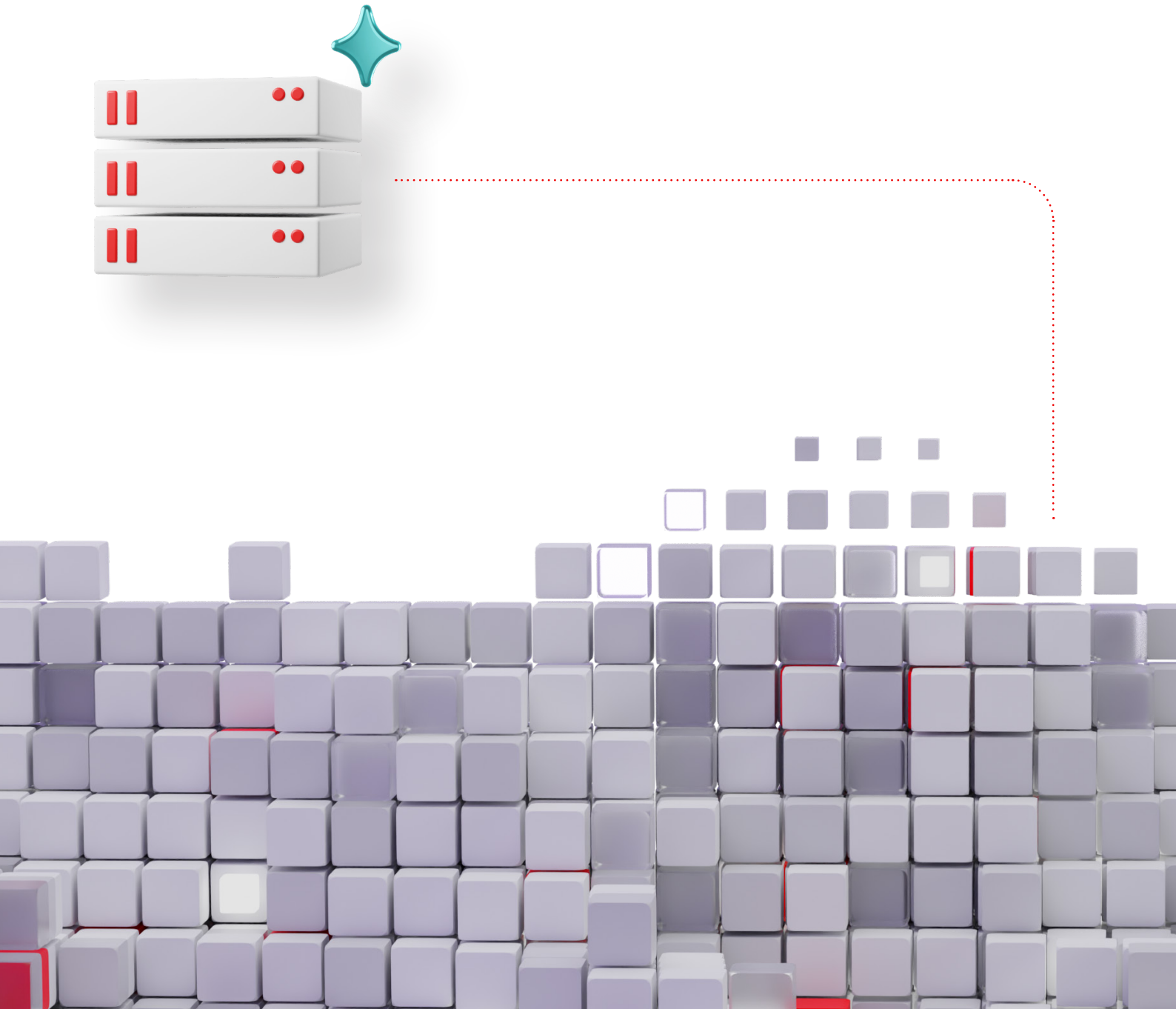
그러나 희소화를 효과적으로 달성하려면 모델을 재학습시켜야 하며, 이는 상당한 초기 리소스가 투입되는 컴퓨팅 집약적인 단계입니다. 희소화의 효율성은 하드웨어 기능에 크게 좌우됩니다. 예를 들어 GPU와 같은 현대적인 가속기에서 지원하는 반정형 희소화의 경우, 0으로 설정된 가중치의 특정 패턴을 활용하여 더 빠른 컴퓨팅이 가능합니다. 주요 장점은 적절히 구현할 경우 컴퓨팅 요구 사항을 크게 줄일 수 있다는 점입니다.

희소화는 특히 양자화와 같은 다른 최적화 방법과 결합하면 유의미한 장점을 제공하지만, 일반적으로 더 복잡한 최적화 과정을 요구합니다. 따라서 대규모 시나리오나 특수 하드웨어 환경에서 사용할 것을 권장합니다. 희소화를 주의 깊게 적용하면 추론 효율성을 개선할 수 있지만, 구현하기가 복잡하기 때문에 통상적으로는 양자화가 1차 최적화 기술로 더욱 널리 권장됩니다.

조직은 압축 워크플로우와 검증된 런타임을 도입함으로써 운영 비용을 더욱 효과적으로 관리하고 확장성을 지원하며 인프라 리소스를 과도하게 투입하지 않고도 향후 AI 사용량 증가에 대비할 수 있습니다.

정확도가 저하될까요?

양자화와 희소화 같은 모델 압축 기술은 메모리와 컴퓨팅 요구 사항을 줄이지만 허용 가능한 수준의 정확도를 유지하도록 설계되어 있습니다. 예를 들어 8비트 양자화는 메모리 사용량을 절반으로 줄이면서도 기준에 근접한 정확도를 제공합니다. 4비트 모델도 가중치 라운딩과 캘리브레이션 같은 고급 양자화 기술을 사용하여 최적화하면 강력한 성능을 유지할 수 있습니다. 2:4 희소화와 같은 구조적 희소화 패턴을 사용하면 하드웨어 가속기가 출력 품질을 저하시키지 않으면서도 중복된 연산을 우회할 수 있습니다. 많은 프로덕션 시나리오에서 팀은 모델 성능 저하를 거의 또는 전혀 발생시키지 않으면서도 상당한 리소스 절감을 실현하고 있습니다. 테스트와 검증은 여전히 필수적이지만, 대부분의 애플리케이션에서 압축을 잘 구현하면 정확도를 유지하면서도 높은 효율의 추론을 실현할 수 있습니다.

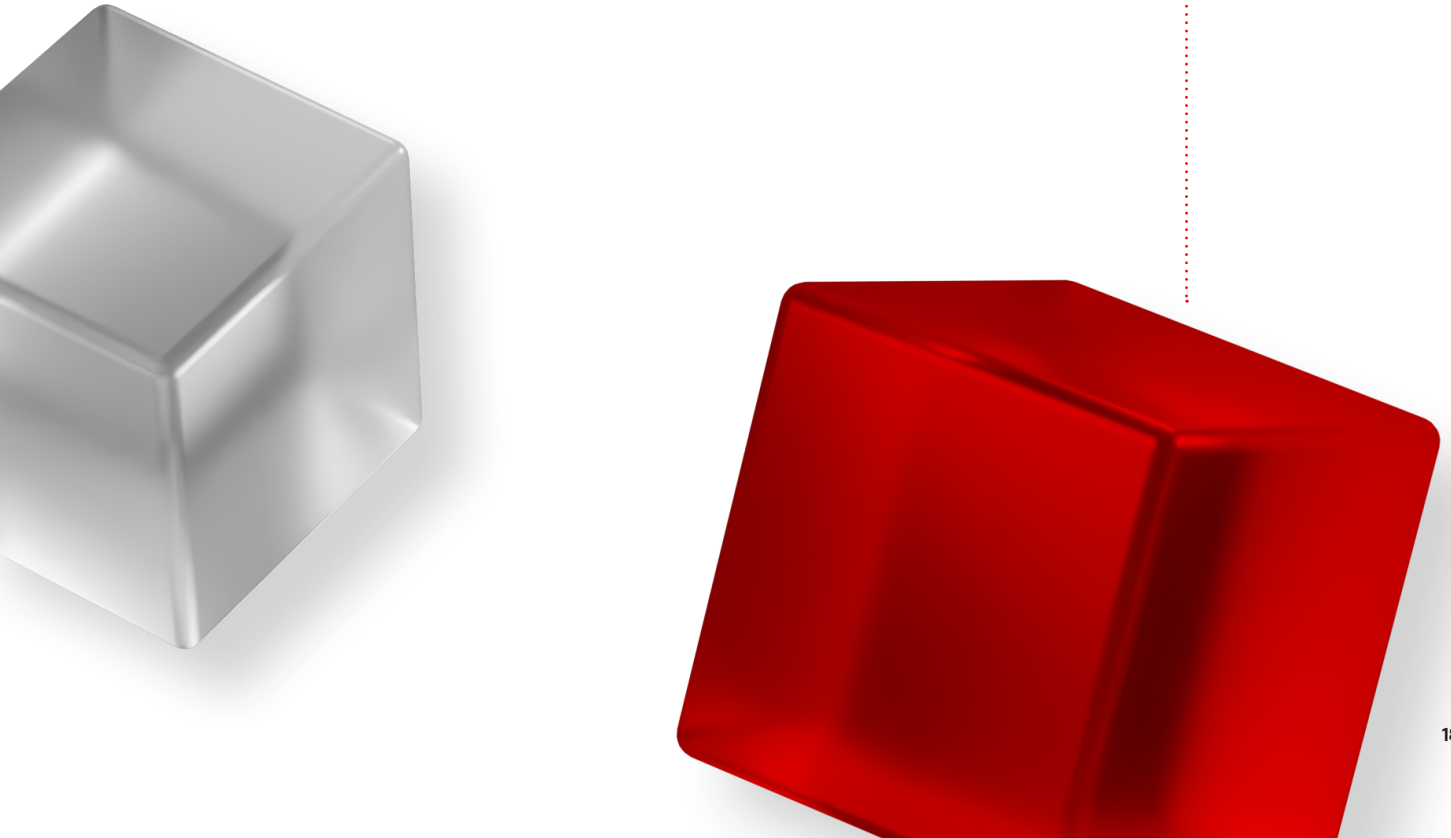


Red Hat AI

Red Hat AI란?

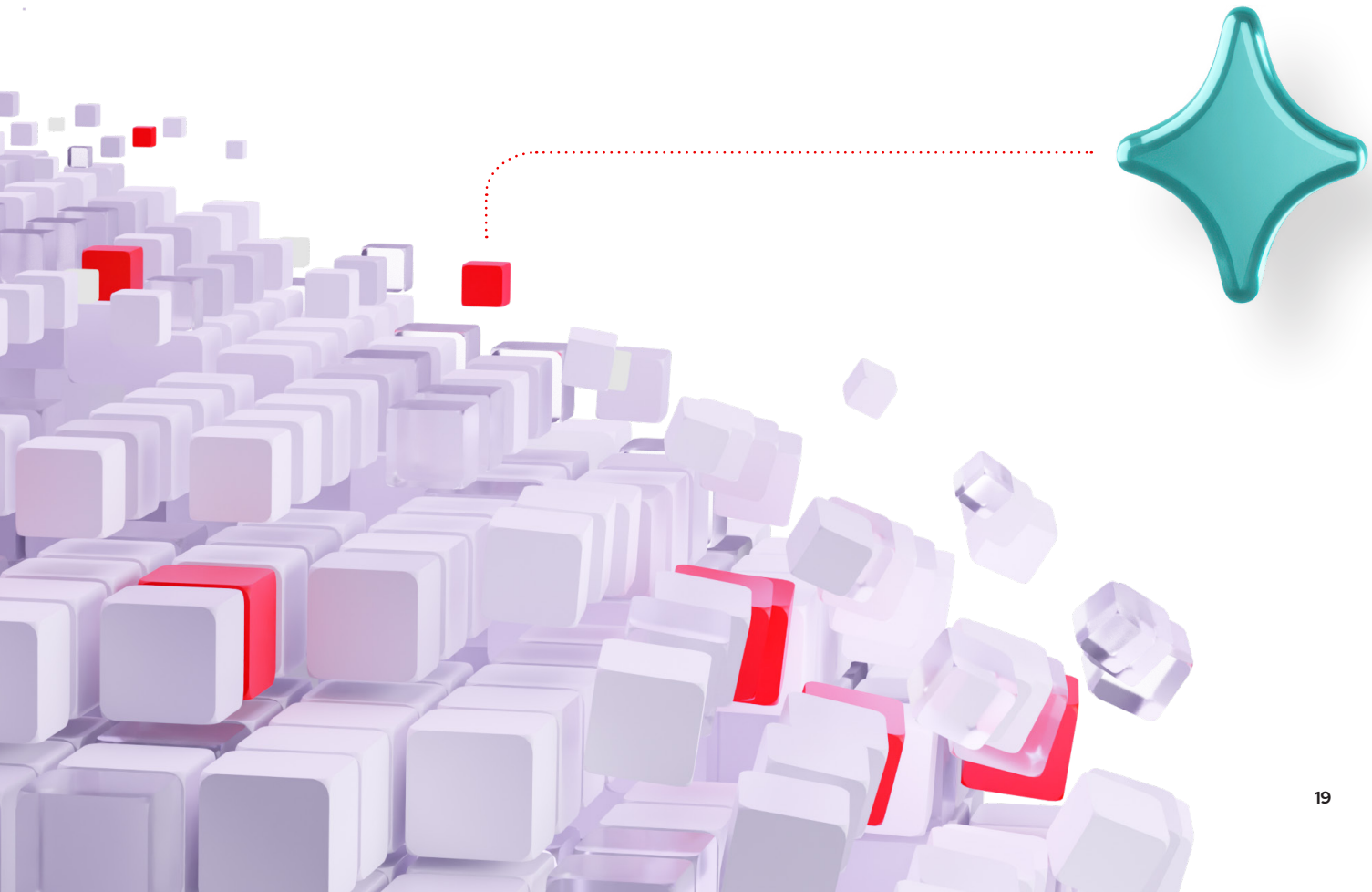
Red Hat AI는 하이브리드 클라우드 환경 전반에서 AI 혁신을 가속화하고 AI 솔루션의 개발 및 제공에 따른 운영 비용을 절감하는 플랫폼입니다. 프라이빗 데이터와의 통합을 간소화하고, 최적화된 모델과 효율적인 추론을 통해 비용을 절감하며, 확장 가능하고 유연한 플랫폼으로 에이전틱(Agentic) AI 워크플로우의 제공을 가속화합니다.

조직은 Red Hat AI를 통해 단일 서버 배포부터 대규모 분산 플랫폼에 이르기까지 예측 모델과 생성형 AI 모델의 라이프사이클을 관리하고 모니터링할 수 있습니다. 이 플랫폼은 오픈소스 기술과 다양한 인프라 전반에서 성능, 안정성, GPU 지원에 초점을 맞춘 파트너 에코시스템을 기반으로 합니다.



Red Hat AI의 구성 요소:

- **최적화 및 검증된 모델** 테스트와 미세 조정(fine-tuning) 부담을 줄이기 위해 사전 평가와 성능 검증을 거친 모델입니다.
- **LLM Compressor** 사용자가 대중적인 모델에 양자화와 압축을 적용할 수 있도록 돕는 툴킷으로, 정확도 저하 없이 추론에 필요한 리소스 요구 사항을 낮춰줍니다.
- **모델 커스터마이징** 파운데이션 모델을 특정 기업의 요구 사항에 맞게 미세 조정하거나 조정할 수 있는 툴입니다.
- **고성능 추론 런타임** 효율적이고 확장 가능하며 안정적인 모델 서빙을 위해 고급 배치 및 메모리 관리 기술을 사용하는 최적화된 vLLM 기반 런타임입니다.
- **LLMOps** 프로덕션 환경에서 LLM의 배포, 모니터링, 관리를 간소화하는 사례와 툴입니다.
- **AI 안전성 및 평가** 모델의 정확성, 공정성, 신뢰성을 평가하여 AI 배포의 책임성과 신뢰성을 확보하기 위한 프레임워크와 방법론입니다.
- **유연하고 일관된 스케일링** 하이브리드 클라우드 환경 전반에서 AI를 스케일링할 때 유연성과 일관성을 확보할 수 있도록 지원하는 인프라입니다.
- **에이전틱(Agentic) AI 배포 가속화** 고도화된 자율 AI 시스템을 신속하게 배포할 수 있도록 설계된 기능으로, 조직이 AI 혁신의 최전선에 설 수 있도록 지원합니다.



Red Hat AI를 통한 모델 최적화

Red Hat AI는 효율성, 정확성, 비용 효율성 간 균형을 맞추도록 설계된 고급 기술을 통해 조직이 AI 모델을 최적화할 수 있도록 지원합니다.

Red Hat AI는 모델 최적화의 두 가지 핵심 요소인 효율적인 런타임과 압축된 모델에 중점을 둡니다. 이러한 접근 방식을 결합하여 Red Hat의 AI 포트폴리오는 필요한 컴퓨팅 리소스를 줄이면서도 빠른 추론 성능을 제공합니다. 특히 Red Hat AI Inference Server는 연속 배치와 메모리 효율적인 방식을 활용합니다. 이를 통해 모델이 초당 더 많은 토큰을 처리할 수 있으며 GPU 사용량을 줄이면서도 더 높은 처리량을 달성합니다.

Red Hat AI LLM Compressor는 이 E-book에서 다룬 압축 기술을 적용할 수 있는 표준화된 접근 방식을 제공하며, 정확도를 99% 유지하는 최적화를 목표로 합니다. 사용자가 vLLM과 같은 추론 런타임에 맞게 튜닝된 대중적인 모델의 최적화 버전을 생성할 수 있도록 지원합니다. 이를 통해 다양한 하드웨어 환경에서 고성능 압축 모델을 더욱 쉽게 실행할 수 있습니다.

Red Hat AI는 조직이 최적화된 모델을 자신 있게 선택, 배포 및 확장할 수 있도록 광범위한 검증을 제공합니다. 사용 가능한 LLM의 범위가 매우 넓기 때문에 조직은 정확성, 성능, 비용 효율성 측면에서 자신의 활용 사례에 가장 적합한 모델을 식별하는 데 어려움을 겪는 경우가 많습니다. 이러한 과제를 해결하기 위해 Red Hat AI는 오픈소스 검증 툴(GuideLLM, Language Model Evaluation Harness, vLLM 등)을 사용하여 다양한 평가 태스크 전반에서 모델 성능을 엄격하게 벤치마킹합니다. 이러한 검증은 재현성과 정보에 기반한 모델 선택을 지원하여 복잡성과 불확실성을 줄여줍니다.

또한 Red Hat AI는 조직이 AI 인프라를 정확하게 계획하고 리소스 사용을 최적화할 수 있도록 **용량 가이드**를 제공합니다. 이를 통해 하드웨어 활용 부족, 높은 컴퓨팅 비용, 추론 시 비효율성과 같은 일반적인 문제를 해결할 수 있습니다. 검증된 모델, 최적화된 배포 설정, 맞춤형 하드웨어 권장 사항을 결합하여 조직은 유연성을 높이고 배포를 가속화하며 비용을 효과적으로 관리하면서 예측 가능한 성능을 확보할 수 있습니다.

Red Hat AI는 압축 기술과 최적화된 런타임을 통해 LLM의 대규모 배포를 현실화하며, 팀이 비용, 복잡성, 컴퓨팅 리소스 사용을 효과적으로 통제하면서 증가하는 수요에 대응할 수 있도록 지원합니다.

다음 단계

LLM 서빙 비용과 복잡성을 줄일 준비가 되셨나요? **Red Hat AI Inference Server**에 대해 자세히 알아보거나, Red Hat 담당자에게 문의하여 시작해 보시기 바랍니다.