

DeepSpeed-MoE: 전문가 혼합 추론 및 교육을 발전시켜 차세대 AI 규모 강화

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani
Aminabadi, Ammar Ahmad Awan, Jeff Rasley, Yuxiong He

마이크로소프트

추상적인

거대한 고밀도 모델의 교육이 오늘날 하드웨어 리소스의 가용성과 기능의 경계에 도달함에 따라 전문가 혼합(MoE) 모델은 품질과 동등한 밀집 모델. 교육 비용 절감은 인코더-디코더 모델(이전 작업)에서 자동 공격 언어 모델의 경우 5배 절감(병렬 탐색과 함께 작동)으로 입증되었습니다. 그러나 훨씬 더 큰 모델 크기와 고유한 아키텍처로 인해 빠른 MoE 모델 추론을 제공하는 방법은 여전히 어렵고 해결되지 않아 실제 사용이 제한됩니다. 이를 해결하기 위해 MoE 모델 크기를 최대 3.7배까지 줄이는 새로운 MoE 아키텍처 설계 및 모델 압축 기술을 포함하여 DeepSpeed 라이브러리의 일부로 종단 간 MoE 교육 및 추론 솔루션인 DeepSpeed-MoE를 제시합니다. 기존 MoE 추론 솔루션에 비해 7.3배 더 나은 대기 시간과 비용을 제공하는 고도로 최적화된 추론 시스템입니다. DeepSpeed-MoE는 전례 없는 확장성과 효율성을 제공하여 동일한 품질의 고밀도 모델에 비해 최대 4.5배 빠르고 9배 저렴한 추론으로 대규모 MoE 모델을 제공합니다. 우리는 우리의 혁신과 시스템이 더 적은 리소스로 더 높은 품질의 모델을 교육하고 배포하는 것이 더 광범위하게 가능해지는 대규모 모델 환경에서 조밀한 MoE 모델에서 희소한 MoE 모델로 전환하는 새로운 방향으로 유망한 길을 여는 데 도움이 되기를 바랍니다.

1. 소개

지난 3년 동안 가장 큰 훈련된 모델의 크기가 수억 개의 매개변수에서 5억 개의 매개변수로 1000배 이상 증가했습니다(Megatron-Turing NLG 530B).

크기에 따른 모델 품질의 개선은 모델 크기가 클수록 모델 품질이 좋아지면서 이러한 추세가 계속될 것임을 시사합니다.

그러나 증가하는 컴퓨팅 요구 사항으로 인해 모델 크기의 성장을 유지하는 것이 점점 더 어려워지고 있습니다. 예를 들어, 2021년 12월 현재 존재하는 가장 큰 단일 밀도 모델인 Megatron-Turing NLG 530B 모델은 NVIDIA Selene 슈퍼컴퓨터에서 2000개 이상의 A100 GPU를 훈련하는 데 약 3개월이 걸렸으며 3백만 GPU 시간 이상을 소비했습니다[2].

조밀한 모델 크기의 또 다른 3~5배 증가는 합리적인 시간 내에 실행 불가능합니다.

최신 모델을 교육하는 데 엄청난 컴퓨팅 리소스가 필요하다는 점을 감안할 때 자연스러운 질문은 다음과 같습니다.

이 문서는 ICML 2022[1]: <https://proceedings.mlr.press/v162/rajbhandari22a>에 게시되었습니다. 저자 기여는 종이 끝에 나열됩니다.

컴퓨팅 비용이 증가합니까?" 또는 "3~5배 적은 리소스를 사용하여 유사한 품질의 모델을 생산할 수 있습니까?"

모델 품질을 희생하지 않고 대형 모델을 교육하기 위한 컴퓨팅 요구 사항을 줄이기 위한 수많은 노력이 있었습니다. 이를 위해 MoE(Mixture-of-Experts) 기반 아키텍처 [3, 4, 5]는 모델 매개변수와 관련하여 준선택형 컴퓨팅 요구 사항을 가능하게 하고 교육 비용을 늘리지 않고도 모델 품질을 개선할 수 있는 유망한 경로를 열었습니다. 그러나 MoE 기반 모델에는 다양한 실제 시나리오에서의 사용을 제한하는 고유한 일련의 문제가 있습니다.

- 제한된 범위 NLP 영역에서 MoE 기반 모델의 범위는 기본적으로 인코더-디코더 모델 및 시퀀스-시퀀스 작업으로 제한되며 다른 도메인에서 해당 응용 프로그램을 탐색하는 작업이 제한됩니다. GPT-3 및 MT-NLG 530B와 같은 자동 회귀 자연어 생성(NLG)에 MoE를 적용하면 최첨단 언어 모델을 교육하는 데 드는 컴퓨팅 비용이 인코더 디코더 모델보다 훨씬 더 높을 수 있으므로 적습니다. 탐험했다.
- 대규모 메모리 요구 사항 MoE 모델은 밀도가 높은 대응 모델과 동일한 모델 품질을 달성하기 위해 컴퓨팅이 덜 필요하지만 훨씬 더 많은 매개 변수가 필요합니다. 예를 들어, MoE 기반 Switch-Base 모델은 T5-large(7.4B 대 0.74B)보다 매개변수가 10배 더 많지만 다양한 다운스트림 작업에서 비교할 때 여전히 동일한 모델 품질을 갖지 못합니다[5]. 즉, MoE 기반 모델은 품질과 동등한 고밀도 모델에 비해 "매개변수 효율성"이 훨씬 낮습니다.

더 큰 모델 크기와 더 낮은 매개변수 효율성은 훈련과 추론 모두에서 문제를 일으킵니다.

학습을 위해 모델 크기가 이렇게 크게 증가하려면 기기 메모리도 이에 비례하여 증가해야 합니다. 위에서 언급한 T5-large(0.74B)는 단일 32GB V100 GPU의 메모리에 맞을 수 있는 반면, Switch-Base(7.4B)를 교육하려면 장치 메모리에 모델을 맞추려면 이러한 GPU가 최소 8-10개 필요합니다. 훈련을 위해. 조밀한 모델 크기를 500B 매개변수가 있는 MT-NLG로 확장하는 경우 MoE 기반 모델과 유사한 품질을 달성하려면 5조 개 이상의 매개변수가 있는 모델이 필요할 수 있습니다(10배 확장이 여전히 유지된다고 가정). 모델은 훈련을 위해 상태를 나타냅니다. 이 거대한 모델 크기는 필요한 장치 메모리와 수천 개의 GPU에서 장치 메모리를 효과적으로 활용하는 데 필요한 시스템 지원으로 인해 MoE 기반 모델 훈련을 대규모로 어렵게 만듭니다.

- 제한된 추론 성능 위에서 언급한 큰 모델 크기와 열악한 매개변수 효율성으로 인해 MoE 기반 모델의 빠른 추론은 훨씬 더 어렵습니다. 한편으로 매개변수 크기가 클수록 적합하려면 더 많은 GPU가 필요하며 다중 GPU 추론 기술은 MoE 기반 모델에서 작동하도록 설계되지 않았습니다. 반면에 추론은 종종 메모리 대역폭에 제한을 받기 때문에 고밀도 모델보다 10배 더 클 수 있는 MoE 기반 모델은 고밀도 모델과 유사한 추론 대기 시간을 달성하기 위해 10배 더 높은 달성 가능한 메모리 대역폭이 필요할 수 있습니다.

교육 비용의 유망하고 적지 않은 감소에도 불구하고 위에서 언급한 이러한 문제는 MoE의 실제 적용 가능성을 심각하게 제한합니다. MoE를 실용적이고 접근 가능하며 적용 가능하도록 만들기 위해 이 백서에서는 세 가지 해당 솔루션을 제공하여 이러한 문제를 해결합니다.

- MoE 기반 모델의 범위를 자동 회귀 NLG 작업으로 확장하여 GPT-3 및 MT NLG와 같은 모델에 대해 동일한 모델 품질을 달성하기 위해 5배의 교육 비용 절감을 보여줍니다. 이러한 결과는 교육 비용을 절감할 수 있는 명확한 기회를 보여줄 뿐만 아니라

대규모 NLG 모델뿐만 아니라 현재 세대 하드웨어 리소스의 제한 하에서 훨씬 더 높은 차세대 모델 품질에 도달할 수 있는 가능성을 열어줍니다.

- PR-MoE(Pyramid-Residual MoE)라고 하는 새로운 MoE 아키텍처를 개발하여 MoE 기반 모델의 매개변수 효율성을 개선합니다. PR-MoE는 잔류 연결을 사용하여 생성된 하이브리드 밀도 및 MoE 모델이며 가장 효과적인 곳에만 전문가를 적용합니다. PR-MoE는 모델 품질을 변경하지 않고 컴퓨팅 요구 사항을 최소한으로 변경하면서 MoE 모델 매개변수 크기를 최대 3배까지 줄일 수 있습니다. 또한 단계별 지식 종류를 통해 MoS(Mixture-of-Students)라고 하는 PR-MoE의 종류 버전을 만듭니다. MoS는 유사한 모델 품질을 유지하면서 MoE 모델 크기를 최대 3.7배까지 줄입니다.
- 수백 개의 GPU에서 추론 워크로드를 효율적으로 확장할 수 있는 고도로 최적화된 MoE 추론 시스템인 DeepSpeed-MoE 추론 시스템을 개발하여 기존 MoE 추론 솔루션과 비교할 때 추론 대기 시간 및 비용을 최대 7.3배 감소시킵니다. 1조 매개변수 MoE 모델에 대해 초고속 추론 대기 시간(25ms 미만)을 제공합니다. 또한 DeepSpeed-MoE는 시스템 및 모델 최적화를 결합하여 품질과 동등한 고밀도 모델에 비해 MoE 모델에 대해 최대 4.5배 빠르고 9배 저렴한 추론을 제공합니다.

우리의 혁신과 시스템은 MoE가 고밀도 모델에 비해 훨씬 더 효과적이고 경제적인 대안이 될 수 있도록 하여 동일한 모델 품질을 얻는 동시에 교육 및 추론 비용을 크게 낮춥니다. 또는 이와 동등하게 컴퓨팅 리소스를 늘리지 않고도 차세대 AI 규모의 모델을 강화하는 데 사용할 수 있습니다.

우리는 DeepSpeed-MoE가 대규모 모델 환경에서 조밀한 MoE 모델에서 희소 MoE 모델로 전환하는 새로운 방향으로의 유망한 경로를 여는 데 도움이 되기를 바랍니다.

백서 개요 이 백서의 나머지 부분은 필요한 배경 및 관련 작업과 함께 독립적인 섹션으로 제시된 위의 세 가지 솔루션으로 시작합니다.

소프트웨어 우리는 여러 단계에 걸쳐 DeepSpeed 라이브러리의 일부로 DeepSpeed-MoE를 오픈 소싱하는 과정에 있습니다. DeepSpeed GitHub(<https://github.com/microsoft/DeepSpeed>) 및 웹사이트 (<https://www.deepspeed.ai/>)에서 코드, 자습서 및 문서를 찾으십시오 . 이 백서의 실험은 DeepSpeed를 사용하여 모델을 교육하고 제공하기에 가장 적합한 Microsoft Azure AI 플랫폼에서 수행되었습니다. 자습서에는 Azure에서 DeepSpeed를 시작하고 Azure ML을 사용하여 다양한 모델을 실험하는 방법이 포함되어 있습니다.

2 관련 업무

2.1 대규모 고밀도 NLP 모델

매개변수 수와 관련하여 모델 용량에 대한 스케일링 법칙 [6] 의 상한을 테스트하고 확인하기 위해 사전 훈련된 자연어 처리 모델 크기는 지난 몇 년 동안 매년 10배씩 증가했습니다. 초기 작업은 일반적으로 BERT [7], XLNet [8], RoBERTa [9], ALBERT [10] 및 GPT [11] 등을 포함하여 수억 개의 매개 변수 규모입니다. GPT-2 [12], TuringNLG [13], Megatron-LM [14], T5 [15] 등과 같은 수십억 개의 모델이 도입되었으며 다양한 자연어 이해 및 생성 작업에서 더 나은 일반화 성능을 보이는 것으로 나타났습니다 . [16, 17, 18, 19, 20, 21]. GPT-3 [22] 는 상한을 1,750억 개의 매개변수로 추가로 확장하고

제로/퓨 샷 학습을 통해 미세 조정을 통해 이전의 소규모 모델과 비슷하거나 더 나은 성능을 달성할 수 있습니다. 최근에는 초대형 Megatron-Turing NLG 530B 모델 [2] 이 DeepSpeed [23] 및 Megatron-LM [14] 의 소프트웨어 지원을 통해 5300억 개의 매개변수를 달성 하고 제로라는 새로운 최첨단 결과를 달성했습니다. 하나의 조밀한 모델 내에서 /few-shot learning. 그러나 2000개 이상의 A100 GPU에서 훈련하는 데 3개월이 걸리므로 극복할 수 없는 컴퓨팅 요구 사항으로 인해 단순히 모델 크기를 늘리는 것만으로는 더 나은 모델 품질을 달성할 수 없습니다.

2.2 MoE 아키텍처를 통한 교육 비용 절감

교육 비용을 줄이는 유망한 방법 중 하나는 MoE(Mixture of Expert)를 사용하는 것입니다 [24]. [3] 에서 작성자는 언어 모델링 및 기계 번역을 위해 적응된 LSTM 레이어 [25] 사이에 컨벌루션 MoE 레이어를 적용하여 LSTM 기반 모델을 127B로 확장합니다. MoE의 희소성 특성은 계산 비용을 증가시키지 않고 모델 크기의 스케일링을 크게 향상시킵니다. GShard [4] 는 MoE를 활용하여 변압기 기반 모델 [26] 을 다국어 번역을 위한 600B 매개변수로 교육하며, 이 600B MoE 모델의 교육 비용이 100B 밀집 모델보다 훨씬 저렴함을 보여줍니다. Switch Transformer [5] 는 T5 모델을 기반으로 이것을 계속하고 모델을 1.6조로 확장합니다. 동일한 정확도 성능을 달성하기 위해 [5]는 대형 밀집 모델에 비해 MoE 모델의 훈련 속도가 2.5배 더 빠름을 보여줍니다. 이에 따라 보다 최근의 연구 [27, 28, 29] 는 MoE의 희소성 이점을 보여줍니다.

몇몇 최근 및 병렬 작업 [30, 31] 은 MoE 모델이 여러 다운스트림 평가 작업에 대한 예비 결과를 포함하여 자동 회귀 자연어 생성 작업에도 적용될 수 있음을 보여줍니다. 그러나 우리 작업 은 이러한 병렬 탐색과 비교하여 큰 차이점이 있습니다. (2) 섹션 3.3에서 설명한 대로 더 나은 MoE 매개변수 효율성과 동일/더 나은 제로 샷 평가 품질을 달성하기 위해 PR-MoE 아키텍처 및 MoS 지식 증류를 제안합니다. (3) 우리는 높은 처리량과 낮은 대기 시간으로 대규모 MoE 모델을 효율적으로 제공하기 위해 DeepSpeed-MoE 추론 시스템을 개발합니다. [30] 및 [31] 과 같은 최근 연구에서는 FLOP의 감소에 대해 논의하지만 훈련과 달리 추론 대기 시간 및 비용이 계산에만 의존하지 않는다는 점을 언급하는 것이 적절합니다.

효율적인 추론은 모델 크기, 메모리 대역폭, 메모리에서 데이터를 효율적으로 읽는 시스템의 기능에 따라 달라집니다. DeepSpeed-MoE 추론 시스템은 모델 아키텍처 혁신(섹션 4)과 무수한 최적화 및 기술(섹션 5)을 결합하여 처리량을 위한 초저지연 및 초선형 속도 향상을 제공하는 엔드 투 엔드 시스템입니다.

2.3 MoE 교육 및 추론 시스템 최근 전 세계 많은 연구자들이 MoE 모델

을 채택함에 따라 MoE 모델 교육을 지원하기 위해 오픈 소스 시스템 및 프레임워크가 개발 또는 확장되고 있습니다. 우리가 아는 한, 추론을 위해 특별히 설계되고 최적화된 MoE 시스템은 아직 없습니다. 여러 MoE 교육 시스템이 최근에 소개되었습니다. 아래에서 우리는 그들 중 몇 가지를 논의합니다.

DeepSpeed MoE 훈련 시스템 [32] 은 주로 MoE 모델의 대규모 훈련을 최적화하는 것을 목표로 했습니다. 이 작업의 주요 목표는 효율적이고 확장 가능한 MoE 모델 교육을 위해 기존 교육 코드를 확장하기 위한 유연한 사용자 대면 API를 설정하는 것이었습니다. 텐서 슬라이싱, 데이터 병렬 처리, ZeRO [23] 기반 데이터 병렬 처리 및 전문가 병렬 처리를 포함한 다양한 병렬 처리 유형의 유연한 조합을 활용하여 최대 8배 더 큰 모델 크기를 지원합니다. 5배 적은 교육 시간으로 50개 언어로 100억 매개변수 멀티태스킹 다국어 MoE 모델(변압기 인코더-디코더 아키텍처)인 Z 코드 MoE를 교육하는 데 사용되었습니다.

동일한 품질의 조밀한 모델과 비교됩니다.

FastMoE [33] 는 MoE 모델이 데이터 및 전문가(모델) 병렬 처리 하에서 훈련될 수 있는 방법을 보여주어 위해 개발된 연구 소프트웨어입니다. 다양한 평행도 차원의 조합은 완전히 지원되지 않습니다. Fast-MoE의 예에는 Transformer-XL 및 Megatron-LM이 포함되지만 대규모 종단 간 교육에 대한 결과는 아직 사용할 수 없습니다. Fairseq-MoE [31]는 MoE API와 일반 언어 모델을 위한 교육 파이프라인을 제공합니다. Fairseq 시스템은 Tutel [34] 에 의해 더욱 최적화되어 Fairseq보다 최대 40% 개선되었습니다.

3 NLG용 DeepSpeed-MoE: 교육 비용 절감 언어 모델의 5배

자동화 변환기 기반 자연어 생성(NLG) 모델은 문서 요약, 헤드라인 생성, 질문 및 답변에서부터 다양한 프로그래밍 언어로 코드 생성까지 광범위한 언어 작업에 대한 설득력 있는 솔루션을 제공합니다.

이러한 모델의 일반적인 적용 가능성으로 인해 품질 개선은 학계와 산업계 모두에게 큰 관심을 불러일으켰습니다. NLG 모델군을 교육하기 위한 엄청난 컴퓨팅 및 에너지 요구 사항을 감안할 때 우리는 MoE가 교육 비용을 줄이기 위해 제공하는 기회를 탐색합니다. MoE를 NLG 모델 제품군에 적용하여 동일한 교육 비용으로 모델 품질을 크게 향상시킬 수 있음을 보여줍니다. 또는 조밀한 NLG 모델과 동일한 모델 품질을 달성하기 위해 교육 비용을 5배 절감할 수 있습니다.

3.1 MoE 기반 NLG 모델 아키텍처

MoE 기반 NLG 모델을 생성하기 위해 변압기 기반 NLG 모델과 같은 GPT를 연구했습니다.

합리적인 시간 내에 교육을 완료하기 위해 다음 모델이 선택됩니다. 4096 숨겨진 크기, 32 관심 머리).

"350M+MoE-128"은 350M dense 모델을 기본 모델로 사용하고 다른 모든 피드포워드 레이어에 128명의 전문가를 추가하는 MoE 모델을 나타냅니다. 즉, 350M+MoE-128 및 1.3B+MoE-128 모두에 대해 총 12개의 MoE 레이어가 있습니다.

우리는 게이팅 기능을 사용하여 각 토큰에 대해 MoE 계층의 전문가 하위 집합을 활성화합니다.

구체적으로 실험에서는 상위 1위 전문가만 선택합니다. 따라서 교육 및 추론 중에 MoE 모델은 각 토큰에 대해 밀도가 높은 부분과 동일한 수의 매개 변수를 활성화합니다(그림 3(왼쪽) 참조). 예를 들어 1.3B+MoE 128은 토큰당 1.3B 매개변수만 활성화하고 토큰당 훈련 계산량은 1.3B 밀집 모델과 유사합니다. 우리는 또한 top-2 게이팅 기능을 테스트했고 그것이 약간의 수렴 개선을 제공한다는 것을 발견했지만, 그것은 감소하는 수익이며 top-1 게이팅에 비해 큰 훈련 계산/통신 오버헤드가 발생합니다.

3.2 교육 및 평가 설정

128 Ampere A100 GPU(Azure ND A100 인스턴스)에서 DeepSpeed를 사용하여 위 모델의 밀도 및 MoE 버전을 사전 훈련했습니다. 이러한 Azure 인스턴스는 완전히 최적화된 환경과 NCCL, Mellanox OFED, Sharp 및 CUDA의 최고 성능 라이브러리 버전을 제공하는 최신 Azure HPC 도커 이미지로 구동됩니다. DeepSpeed는 데이터 병렬 및 전문가 병렬 교육의 조합을 사용하여 MoE 모델 교육을 효과적으로 확장합니다. MT-NLG 모델[2]에 대해 동일한 학습 데이터를 사용했습니다. 공정한 비교를 위해 300B 토큰을 사용하여 밀집 모델과 MoE 모델을 모두 교육합니다.

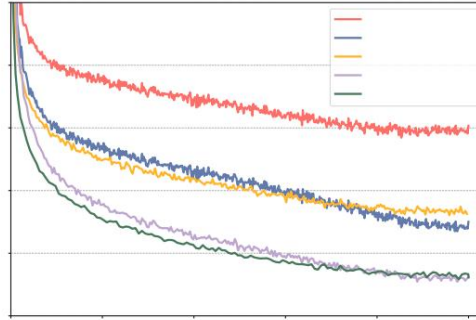


그림 1: 모델 크기가 다른 고밀도 및 MoE NLG 모델에 대한 토큰별 검증 손실 곡선.

3.4 5배 적은 교육 비용으로 동일한 품질 위의 결과에서 보았듯이 128명의 전문가가

있는 MoE를 NLG 모델에 추가하면 NLG 모델의 품질이 크게 향상됩니다. 그러나 이러한 전문가는 각 토큰이 단일 전문가에 의해서만 처리되기 때문에 모델의 컴퓨팅 요구 사항을 변경하지 않습니다. 따라서 밀도가 높은 모델과 동일한 기반의 해당 MoE 모델에 대한 컴퓨팅 요구 사항은 비슷합니다. 보다 구체적으로, 1.3B+MoE-128 모델 교육에는 훨씬 더 나은 모델 품질을 제공하면서 1.3B 밀도 모델과 대략 동일한 양의 컴퓨팅 작업이 필요합니다.

또한 MoE를 적용하면 1.3B 매개변수 밀집 모델의 훈련 비용으로 6.7B 매개변수 밀집 모델의 모델 품질을 달성할 수 있어 효과적인 훈련 컴퓨팅 감소를 5배로 줄일 수 있습니다. 이러한 컴퓨팅 비용 절감은 효율적인 DeepSpeed MoE 교육 시스템을 활용하여 처리량 증가, 교육 시간 및 교육 비용 절감으로 직접 변환될 수 있습니다. 표 3은 128 NVIDIA A100 GPU에서 6.7B 밀도 모델과 비교한 1.3B+MoE-128 모델의 훈련 처리량을 보여줍니다.

결론적으로, 이 섹션은 NLG 모델에서 MoE를 사용하여 상당한 훈련 비용 절감을 보여줍니다. 환경부 확장이 유지된다고 가정하면 결과는 현재 사용 가능한 하드 리소스를 사용하여 더 저렴한 시간과 비용으로 대규모 모델 교육 환경을 변환하고 훨씬 더 큰 모델 규모를 강화할 수 있는 잠재력을 가지고 있습니다. 예를 들어, 1조 매개변수 고밀도 모델과 유사한 정확도를 가진 모델은 잠재적으로 200B 매개변수(예: GPT-3) 크기의 고밀도 모델 비용으로 훈련될 수 있으므로 훈련 비용 절감 및 에너지 절약에서 수백만 달러로 변환됩니다[22].

4 PR-MoE 및 MoS: 모델 크기 및 Im 줄이기 매개변수 효율성 입증

MoE 기반 모델은 NLG 예에서 5배의 교육 비용 절감으로 동일한 품질을 달성하지만 결과 모델은 해당 밀도 모델의 매개변수가 대략 8배입니다(예: 6.7B 밀도 모델에는 67억 개의 매개변수가 있고 1.3B+MoE-128은 520억 매개변수). 이러한 대규모 MoE 모델은 훈련 중에 훨씬 더 많은 메모리를 필요로 하며, 모델 가중치를 읽는 데 소비되는 메모리 대역폭이 추론의 주요 성능 병목이기 때문에 추론 중에 이러한 모델에 대한 대기 시간 요구 사항을 충족하는 것이 어렵습니다. MoE의 매개변수 수를 줄이고 매개변수 효율성을 향상시키기 위해

표 2: 다양한 고밀도 및 MoE NLG 모델에 대한 제로 샷 평가 결과(마지막 6개 열). 모든 제로 샷 평가 결과는 정확도 메트릭을 사용합니다. PR MoE NLG 및 Mixture-of-Students NLG의 결과에 대한 자세한 내용은 섹션 4에 설명되어 있습니다. 마지막 6개 행은 Google [30] 및 Meta(Facebook) [31]의 관련 작업입니다.

모델(숫자 매개변수) LAMBADA PIQA BoolQ RACE-h TriviaQA WebQs Dense NLG: 350M(350M)							
		52.03	69.31	53.64	31.77	3.21	1.57
	1.3B(1.3B)	63.65	73.39	63.39	35.60	10.05	3.25
	6.7B(6.7B)	71.94	76.71	67.03	37.42	23.47	5.12
표준 환경부 NLG:							
	350M+MoE-128 (13B)	62.70	74.59	60.46	35.60	16.58	5.17
	1.3B+MoE-128 (52B)	69.84	76.71	64.92	38.09	31.29	7.19
PR-MoE NLG: 350M+PR-							
	MoE-32/64(4B)	63.65	73.99	59.88	35.69	16.30	4.73
	1.3B+PR-MoE-64/128(31B)	70.60	77.75	67.16	38.09	28.86	7.73
학생들의 혼합 NLG: 350M+PR-							
	MoE+L21+MoS (3.5B)	63.46	73.34	58.07	34.83	13.69	5.22
	1.3B+PR-MoE+L21+MoS (27B)	70.17	77.69	65.66	36.94	29.05	8.22
관련 MoE 작품: 0.1B+MoE-64							
	(1.9B) [30]	36.9	69.0	53.6	29.1	15.2	5.9
	1.7B+MoE-64 (27B) [30]	63.7	76.6	64.4	40.7	42.0	8.5
	8B+MoE-64 (143B) [30]	67.3	78.6	72.2	43.4	55.1	10.7
	125M+MoE-512 (15B) [31]	해당 없음	74.3	60.9	해당 없음	해당 없음	해당 없음
	355M+MoE-512 (52B) [31]	해당 없음	76.8	56.0	해당 없음	해당 없음	해당 없음
	1.3B+MoE-512 (207B) [31]	해당 없음	78.2	54.2	해당 없음	해당 없음	해당 없음

표 3: 동일한 모델 품질을 달성할 수 있는 MoE 기반 모델과 밀집 모델을 비교한 교육 처리량(128 A100 GPU에서).

초당 훈련 샘플 처리량 증가 / 비용 절감 1x 5x	
6.7B 밀도	70
1.3B+MoE-128	372

모델 품질에 영향을 주지 않고 전체 모델 크기를 최대 3배까지 줄이는 MoE 모델 아키텍처(PR-MoE라고 함)의 혁신을 제시합니다. 또한 새로운 MoE-to-MoE 지식 분류 기술을 설계하여 MoE 모델 크기를 더욱 줄이고 추론 시간과 비용을 최적화하는 MoS(Mixture-of-Students)라고 하는 PR-MoE의 증류 버전을 생성합니다. . 아래에서 우리는 새로운 PR-MoE 아키텍처로 시작한 다음 MoS에 대해 논의합니다.

4.1 PR-MoE: 더 작은 모델 크기를 위한 피라미드-잔차-MoE 및 빠른 추론

4.1.1 두 가지 관찰과 직관

현상-I 첫째, 표준 MoE 아키텍처는 모든 MoE 계층에서 전문가의 수와 구조가 동일합니다. 이것은 기계 학습 커뮤니티의 근본적인 질문을 상기시킵니다. 심층 신경망의 모든 계층이 동일한 표현을 학습합니까? 이 질문은 컴퓨터 비전(CV)에서 잘 연구되었습니다. 얇은 레이어(입력에 가까운)는 일반 표현을 학습하고 깊은 레이어(출력에 가까운)는 보다 객관적인 특정 표현을 학습합니다[37] . 이것은 또한 미세 조정을 위해 얇은 레이어를 동결하기 위해 CV의 전이 학습에 영감을 주었습니다 [38].

그러나 이 현상은 자연어 처리, 특히 MoE 아키텍처에서 잘 탐구되지 않았습니다.

이 질문을 조사하기 위해 서로 다른 두 Half-MoE 아키텍처의 성능을 비교합니다.

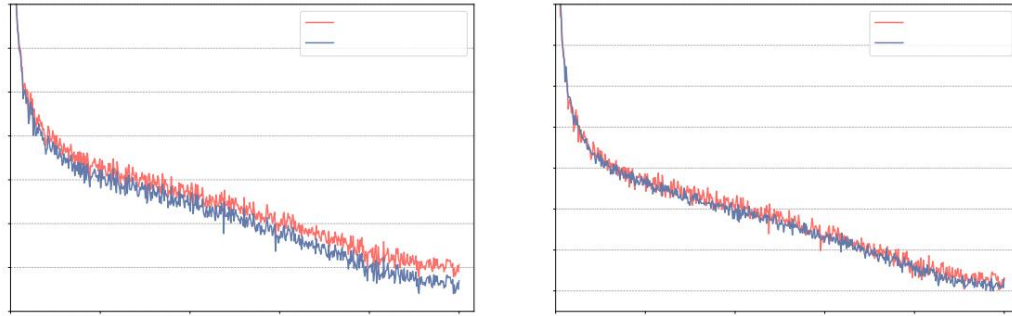


그림 2: 350M+MoE 모델을 기반으로 한 First-Half-MoE/Second-Half-MoE(왼쪽) 및 Top2-MoE/Residual-MoE(오른쪽)의 검증 손실.

350M+MoE 모델을 기반으로 한 강의. 보다 구체적으로, a) MoE 레이어를 모델의 전반부 레이어에 배치하고 레이어의 후반부를 고밀도 모델과 동일하게 둡니다(First-Half-MoE라고 함). b) MoE 레이어를 두 번째 레이어로 전환합니다. 전반부에서 밀도를 사용합니다(Second-Half-MoE라고 함). 결과는 그림 2(왼쪽)에 나와 있습니다. 보시다시피, Second-Half-MoE는 그에 상응하는 것보다 훨씬 더 나은 성능을 가지고 있습니다. 이는 모든 MoE 레이어가 동일한 수준의 표현을 학습하는 것은 아님을 확인합니다. 더 깊은 레이어는 많은 수의 전문가로부터 더 많은 이점을 얻습니다. 간단히 하기 위해 이 현상을 Phenomenon-I라고 합니다.

현상-II 둘째, MoE 모델의 일반화 성능을 개선하기 위해 두 가지 일반적인 방법이 있습니다. 같은; (2) 동일한 수의 전문가를 유지하면서 약간 더 많은 계산(33%)을 희생하여 전문가 용량을 두 배로 늘립니다. 그러나 (1)의 경우 전문가 수가 많아짐에 따라 교육 리소스에 대한 메모리 요구 사항이 증가해야 합니다. (2)의 경우 용량이 커지면 통신량이 두 배가 되어 훈련 및 추론 속도가 크게 느려질 수 있습니다. 일반화 성능 향상을 얻으면서 교육/추론 효율성을 유지하는 방법이 있습니까?

더 큰 전문가 역량이 정확성에 도움이 되는 이유에 대한 한 가지 직관은 추가 전문가가 첫 번째 전문가의 "표현"을 수정하는 데 도움을 줄 수 있다는 것입니다. 그런데 이 첫 번째 전문가를 매번 바꿔야 할까요? 아니면 첫 번째 전문가를 수정하고 다른 추가 전문가만 다른 토큰에 할당할 수 있습니까? 이 알려지지 않은 속성을 조사하기 위해 두 가지 방법으로 비교를 수행합니다. (1) 용량을 두 배로 늘리고(Top2-MoE라고 함) (2) 전문가 한 명을 고정하고 두 번째 전문가를 여러 전문가에 걸쳐 변경합니다(Residual-MoE라고 함). 특히 (2)의 경우 토큰은 항상 MLP 모듈과 MoE 모듈의 전문가를 통과하므로 잔여 네트워크의 특수한 경우로 볼 수 있습니다. 그런 다음 이 두 분기의 출력을 함께 추가하여 최종 출력을 얻습니다. 주요 직관은 MoE 모듈의 전문가를 dense MLP 모듈의 오류 수정 항으로 취급하는 것입니다. 따라서 Top-1 게이팅 기능과 동일한 양의 통신량으로 레이어당 2명의 전문가를 사용하는 이점을 얻을 수 있습니다.

32명의 전문가와 함께 3억 5천만 + MoE 모델에 대한 비교를 수행하고 검증 곡선은 그림 2(오른쪽)에 나와 있습니다. 우리는 이 두 가지(일명 Top2-MoE 및 Residual-MoE)의 일반화 성능이 서로 동등하다는 것을 알게 되었습니다. 그러나 새로운 설계인 Residual-MoE의 교육 속도는 통신량 감소로 인해 Top2-MoE보다 10% 이상 빠릅니다. 이 현상을 Phenomenon-II라고 합니다.



그림 3: 표준 MoE(왼쪽) 및 PR-MoE(오른쪽) 그림.

4.1.2 피라미드 잔류 MoE 아키텍처

위의 내용을 바탕으로 새로운 MoE 아키텍처를 제안합니다. 섹션 4.1.1의 Phenomenon-I에서 후기 레이어에서 MoE를 활용하면 더 많은 이점을 얻을 수 있다고 제안한 것처럼 새로운 아키텍처는 이전 레이어와 비교하여 마지막 몇 레이어에서 더 많은 전문가를 활용합니다. 이것은 Pyramid-MoE 설계를 제공하며 그림 3(오른쪽)에서 예를 보여줍니다. 마지막 두 레이어에는 이전 레이어와 같이 $2x$ 전문가가 있습니다. 한편, Phenomenon II를 고려하여 그림 3(오른쪽)과 같이 각 토큰이 하나의 고정 MLP 모듈과 하나의 선택된 전문가를 개별적으로 전달하는 Residual-MoE 아키텍처를 제안합니다. 여기서 주황색 블록은 고정 MLP입니다.

Pyramid-MoE와 Residual-MoE를 함께 결합함으로써 모든 표준 MoE 레이어가 새로운 PR-MoE 레이어로 대체되는 Pyramid-Residual MoE 모델(간단히 PR-MoE)이 있습니다. 그림 3은 표준 MoE 및 PR-MoE 아키텍처의 그림을 보여줍니다.

4.1.3 시스템 설계

이 섹션에서는 전문가 병렬 처리를 사용하여 MoE 모델을 효율적으로 교육하는 방법에 대해 논의하는 것으로 시작합니다. 그런 다음 PR-MoE 모델에 적용할 때 이러한 접근 방식의 한계에 대해 논의합니다. 마지막으로 PR-MoE 모델을 효율적으로 훈련하기 위해 기존 전문가 병렬 기반 훈련 시스템을 확장하는 방법에 대해 논의합니다.

효율적인 MoE 모델 교육 MoE 모델을 효율적으로 교육하려면 우수한 컴퓨팅 효율성을 달성하기 위해 MoE 모듈의 각 전문가에 대해 충분히 큰 배치 크기가 필요합니다. 이는 MoE에 대한 입력 토큰 수가 모든 전문가에 걸쳐 분할되어 그러한 분할이 수행되지 않은 나머지 모델과 비교할 때 전문가 수에 비례하여 전문가당 토큰 수가 감소하기 때문에 어렵습니다. 전문가당 토큰 감소를 피하는 가장 간단한 방법은 전문가 수와 동일한 전문가 병렬 처리 [32]와 데이터 병렬 처리로 모델을 훈련하는 것입니다. 이것은

표 4: 표준 MoE와 PR MoE 간의 제로 샷 평가 비교(마지막 6개 열).

모델(숫자 매개변수)	LAMBADA	PIQA	BoolQ	RACE-h	TriviaQA	WebQs	350M+MoE-128 (13B)
	62.70	74.59	60.46				35.60
350M+PR-MoE-32/64 (4B)	63.65	73.99	59.88	76.71			35.69
1.3B+MoE-128 (52B)	69.84	64.92	77.75	67.16			38.09
1.3B+PR-MoE-64/128 (31B)	70.60						38.09

전문가 간에 분할될 MoE 복제당 배치의 집계 토큰으로, 모델의 나머지 부분과 비교하여 전문가당 토큰이 감소하지 않습니다.

PR-MoE의 과제 PR MoE 모델을 효율적으로 교육할 수 있는 교육 인프라를 설계하는 것은 모델의 여러 단계에 다양한 수의 전문가가 존재하기 때문에 쉽지 않습니다. 위에서 논의한 바와 같이 MoE 기반 모델을 교육하는 가장 효율적인 접근 방식은 전문가 병렬 처리를 전문가 수와 동일하게 만들어 전문가당 입력 토큰이 감소하지 않도록 하는 것입니다. 그러나 PR-MoE의 전문가 수에 차이가 있기 때문에 모든 MoE 계층에 최적인 단일 전문가 병렬도는 없습니다. 또한 전문가 병렬 처리가 모델에서 가장 적은 수의 전문가로 설정되면 더 많은 수의 전문가가 있는 MoE 레이어에 대해 GPU당 여러 전문가가 필요하므로 전문가당 배치 크기가 줄어들고 효율성이 떨어집니다. GPU당 필요한 메모리 증가. 반면 전문가 병렬 처리를 모델에서 가장 많은 수의 전문가로 설정하면 로드 밸런싱 문제가 발생하여 일부 GPU가 다른 GPU보다 더 많은 전문가를 처리하여 궁극적으로 훈련 처리량 효율성이 제한됩니다.

다중 전문가 및 다중 데이터 병렬 처리를 지원하는 DeepSpeed-MoE 이러한 문제를 해결하기 위해 우리는 DeepSpeed-MoE 위에 유연한 다중 전문가 및 다중 데이터 병렬 설계를 개발하고 구현합니다. 다른 전문가 및 데이터 병렬 처리 수준. 예를 들어, 서로 다른 MoE 계층에서 32, 64, 128명의 전문가가 있는 128개의 GPU에서 실행되는 PR-MoE 모델은 비전문가 병렬 처리를 위한 128방향 데이터 병렬 처리로 훈련할 수 있으며 {32, 64, 128} MoE 매개변수에 대한 전문가 병렬 처리 및 {4, 2, 1} 데이터 병렬 처리. 이제 각 GPU는 전문가 수에 관계없이 MoE 레이어당 정확히 1명의 전문가를 교육할 수 있으므로 전문가당 입력 토큰 감소, 부하 불균형 또는 GPU당 메모리 요구 사항 증가가 없습니다.

이 유연한 확장을 통해 DeepSpeed-MoE [32] 는 교육 효율성이나 메모리 요구 사항을 손상시키지 않고 모델의 여러 단계에서 다른 전문가가 필요할 수 있는 다른 미래 MoE 변형과 함께 PR-MoE 모델을 교육할 수 있습니다.

4.1.4 PR-MoE 평가

PR-MoE와 Standard-MoE의 비교 우리는 새로운 아키텍처인 PR-MoE를 두 가지 다른 크기의 모델, 즉 기본 크기 350M 및 1.3B에서 평가하고 더 큰 Standard-MoE 아키텍처와 성능을 비교합니다. 보다 구체적으로, 350M+PR MoE-32/64를 350M+MoE-128과 비교하고 1.3B+PR-MoE-64/128을 1.3B+MoE와 비교합니다.

128.

결과는 표 4에 나와 있습니다. 350M 및 1.3B 사례 모두에서 PR-MoE는 훨씬 적은 매개변수를 사용하지만 Standard-MoE 모델과 비슷한 정확도를 달성합니다. 특히, (1) 350M의 경우 PR-MoE는 Standard-MoE로 파라미터의 1/3 미만만 사용합니다. (2) 1.3B 사례의 경우 PR-MoE는 Standard-MoE로 파라미터의 약 60%만 사용하면서 유사한 정확도를 달성합니다.

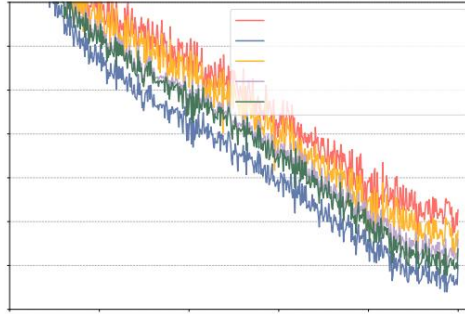


그림 4: 3억 5천만 + MoE를 기반으로 한 다양한 MoE 모델의 검증 곡선.

다양한 MoE 아키텍처의 절제 연구 다양한 MoE 아키텍처의 성능, 특히 Standard-MoE와 Residual MoE/Pyramid-MoE/PR-MoE 간의 비교를 완전히 연구하기 위해 350M+MoE 32를 포함하여 5가지 MoE 기반 모델을 평가합니다. 350M+MoE-128, 350M+Pyramid-MoE-32/64(32명의 전문가를 사용하는 10개의 MoE 레이어와 64명의 전문가를 사용하는 2개의 MoE 레이어가 있음), 350M+Residual-MoE-32 및 350M+PR-MoE-32/ 64(350M+Pyramid-MoE-32/64와 동일한 전문가 설정).

전체 훈련에 대한 검증 곡선은 그림 4에 나와 있습니다. 볼 수 있듯이 350M+MoE-128과 350M+MoE-32 사이의 검증 손실 격차는 Pyramid-MoE 및 Residual-MoE에 의해 크게 감소될 수 있습니다. Pyramid-MoE와 Residual-MoE의 조합인 PR-MoE를 사용하면 손실 격차를 약 0.01로 더 줄일 수 있어 품질 영향을 최소화하면서 PR MoE의 뛰어난 매개변수 효율성을 보여줍니다.

4.2 학생들의 혼합: 더 작은 모델 크기를 위한 증류

더 빠른 추론

모델 압축 및 지식 증류는 참조 성능을 더욱 향상시킬 수 있는 추가 기회를 제공합니다. 양자화 [39, 40, 41] 및 가지치기 [42, 43]와 같은 모델 압축을 위한 다양한 방법이 있지만 현재 우리의 노력은 지식 증류 [44] (KD)를 통한 계층 축소에 중점을 두고 있습니다. 모델 계산 및 학생 모델에서 MoE 구조를 유지합니다.

KD는 큰 모델을 작은 모델로 압축하는 성공적인 방법으로 입증되었습니다. 이 모델에는 매개변수와 계산이 훨씬 적지만 여전히 경쟁력 있는 결과를 얻을 수 있습니다. KD를 사전 훈련된 큰 LM을 작은 모델로 작업별 증류에 적용하는 작업이 있었습니다 [45, 46, 47, 48]. 그러나 그들은 작은 변환기(수백 개의 매개변수)와 고밀도 인코더 기반 LM 모델(예: BERT)만 고려합니다. 대조적으로, 우리는 수십억 개의 매개변수 규모에서 희박한 MoE 기반 자동 생성 LM 모델에 대한 KD 연구에 중점을 둡니다. 우리가 알고 있는 MoE 증류에 대한 유일한 다른 분석은 [49, 31]의 MoE 증류를 조밀한 모델로 연구하는 것입니다. 그러나 그렇게 함으로써 증류된 학생 모델은 MoE가 제공하는 드문 미세 조정 및 추론 이점을 잃습니다. 대조적으로, 우리의 연구는 지식 증류로 사전 훈련된 더 작은 MoE 모델에 대해 많은 다운스트림 작업에 대한 제로 샷 평가와 같은 유사한 성능에 도달할 수 있음을 보여 주므로 추론 시간 동안 모델이 더 가볍고 빠릅니다.

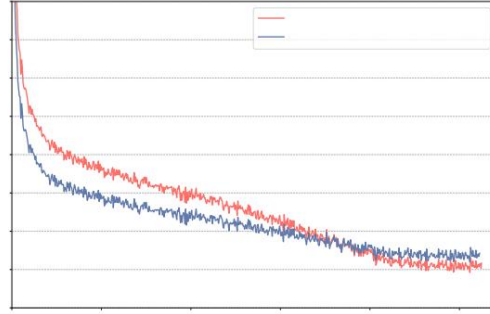


그림 5: 처음부터 증류하지 않은 훈련과 전체 사전 훈련 프로세스에 대해 지식 증류를 수행한 검증 곡선. 그림에서 학생 PR-MoE는 21계층으로 훈련되었습니다. KD는 처음에는 도움이 되지만 훈련이 끝날 무렵 정확도가 떨어지기 시작합니다.

4.2.1 단계별 KD를 통한 학생 혼합

아키텍처 선택 및 최적화 목표 MoE에 대한 지식 추출을 적용하기 위해 먼저 교사 MoE 모델을 교육합니다. 교사 모델에서 각 전문가 분기의 깊이를 줄여 해당 학생을 얻습니다. 그렇게 함으로써, 각 전문가 분기가 더 작은 깊이를 갖는다는 점을 제외하고 교사 MoE와 동일한 성가게 게이트된 아키텍처를 갖는 최종 학생 모델입니다. 이러한 이유로 결과 모델을 MoS(Mixture-of-Students)라고 합니다.

MoE 구조는 희소한 훈련 및 추론을 가능하게 하여 상당한 이점을 제공하므로 작업에 구애받지 않는 증류된 학생들의 혼합은 이러한 이점을 상속하면서 품질 등가 밀도 모델에 대한 추론 이점을 유지합니다. MoS를 생성한 후 MoS가 교육 데이터 세트에서 교사 MoE의 출력을 모방하도록 강제합니다. KD 손실 [50]의 일반적인 공식은 다음과 같습니다.

$$\min_{\theta} E(x,y) \quad D[L(x; \theta) + \alpha LKD(x; \theta)], \quad (1)$$

여기서 예측과 주어진 하드 레이블 간의 교차 엔트로피 손실과 예측과 교사의 소프트 레이블 간의 KL 발산 손실의 가중치 합입니다. 또한 PR-MoE의 우수한 성능을 고려하여 PR-MoE를 KD(PR-MoS)와 결합하여 MoE 모델 크기를 더욱 줄입니다. 즉, 교사 모델과 학생 모델을 모두 PR-MoE로 선택합니다.

단계적 지식 증류로 학생 정확도 향상 MoE 모델 증류 중 한 가지 흥미로운 관찰은 교사 PR-MoE를 사용하면 처음부터 훈련된 PR-MoE 학생보다 학생 정확도가 낮다는 것입니다(표 5의 2행 및 4행). KD는 종종 학생 일반화를 개선하므로 KD가 생성 언어 모델에 대한 MoE 사전 교육에 대한 정확도를 개선하지 않는 이유에 대한 의문이 제기됩니다. 이전 실험에서 증류된 MoE에 대한 증류 실험 결과가 보고되지 않았기 때문에 결과를 더 깊이 파헤칩니다. 그림 5는 처음부터 훈련된 PR-MoE와 교사와 함께 지식 증류를 사용하여 검증 손실을 비교한 것입니다. 우리는 KD 손실이 처음에는 유효성 검사 정확도를 향상시키지만 훈련이 끝날 무렵(예: 400K 단계 후) 정확도가 떨어지기 시작한다는 것을 발견했습니다.

우리는 PR-MoE가 아키텍처 변경(예: 하위 계층의 전문가 감소)을 활용하여 표준 MoE에 비해 용량을 이미 줄였기 때문에,

모델의 깊이를 더 줄이면 학생의 용량이 부족하여 과소적합 체제에 빠지게 됩니다. 따라서 학생 PR-MoE는 교육 손실과 지식 증류 손실을 모두 최소화할 수 있는 용량이 충분하지 않을 수 있으며, 다른 손실(교차 엔트로피 손실)을 희생하여 하나의 손실(KD 손실)을 최소화할 수 있습니다. 훈련 종료. 앞서 언급한 가설은 우리가 KD의 영향을 점진적으로 감소시키거나 교육 프로세스 초기에 KD를 중지하고 나머지 교육에 대한 표준 언어 모델링 손실에 대해서만 최적화를 수행하기를 원할 수 있음을 시사합니다.

4.2.2 혼합 학생 평가

우리는 350M+PR-MoE-32/64 및 1.3B+PR-MoE-64/128의 두 가지 다른 PR-MoE 모델 구성에서 접근 방식을 평가합니다. 두 경우 모두 교사의 깊이를 21(12.5%)로 줄임으로써 학생 모델을 구축하고 이전 섹션에서 설명한 방법을 사용하여 결과 MoS 모델을 교사와 비교합니다.

먼저 제안된 stage-KD가 사전 훈련 수련에 어떤 영향을 미치는지 평가합니다. 그림 6은 400K 단계에서 중지 KD의 검증 곡선 비교 및 Teacher 모델과의 비교를 보여줍니다. 우리는 KD의 이 단계적 버전이 이제 우리에게 지식 증류의 약속된 이점을 제공한다는 것을 발견했습니다. 학생 모델은 이제 교사와 유사한 검증 곡선을 갖습니다.

다운스트림 작업에 대한 평가 결과는 다음과 같이 단계별 KD가 전체 교육 프로세스에 KD를 적용하는 것보다 훨씬 더 나은 제로샷 평가 정확도를 달성함을 보여줍니다.

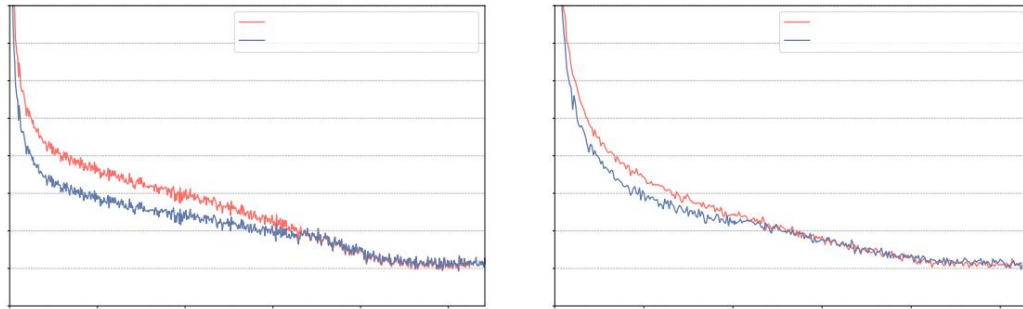


그림 6: 단계별 지식 추출을 통해 학생 PR-MoE 교육의 검증 곡선은 350M+PR-MoE 및 1.3B+PR-MoE 모두에서 교사 PR-MoE와 거의 동일한 검증 손실을 얻습니다.

다음으로 일련의 NLP 작업에 대해 제로샷 평가를 수행합니다. 각 작업에 대한 결과는 표 5에 나와 있습니다. 우리는 몇 가지 관찰을 합니다. 첫째, 같은 양의 깊이 감소가 있지만 MoS 기반 KD(2행)가 없는 PR-MoE 모델은 LAMBADA(1.3포인트) 및 BoolQ(7.5포인트)와 같은 여러 작업에서 눈에 띄는 정확도 저하를 경험합니다. 전문가 깊이를 줄이면 모델 정확도가 떨어질 수 있습니다. 둘째, 단계적 KD(4행)를 사용하여 학생 PR-MoE의 성능을 개선하고 6개 작업 중 5개에서 정확도 향상을 관찰할 수 있습니다. 특히 LAMBADA는 1.1포인트, BoolQ는 6.5포인트, RACE-h는 1.7포인트, TriviaQA는 4.5포인트 향상되었습니다. 한 가지 예외는 PIQA이며, 이 경우 학생 PR-MoE는 약간의 정확도 저하를 경험합니다. 이러한 결과는 MoE 모델에 대한 새로운 KD 기술로서 우리가 제안한 Mixture-of-Students 방법의 효율성을 나타냅니다. 셋째, 전체 교육 프로세스(전체 KD, 행 3)에 대해 KD를 수행하면 LAMBADA(0.8포인트 낮음) 및 PIQA(0.7포인트 낮음)의 다운스트림 작업 정확도가 떨어집니다. 이전 부분에서 설명했듯이 학생 모델에는

과소적합으로 인해 교육이 끝날 때까지 KD 손실과 표준 LM 손실을 모두 최적화할 수 있는 충분한 용량입니다. 대조적으로, 우리가 제안한 단계적 KD는 이 문제를 해결할 수 있고 KD로부터 약속된 이점을 가져올 수 있습니다. 전반적으로 단계적 KD를 통한 증류된 MoE 모델은 42.87 및 47.96의 평균 정확도를 달성하여 레이어가 12.5% 적음에도 불구하고 350M(43.08) 및 1.3B 교사 모델(48.37)의 99.5% 및 99.1% 성능을 유지합니다. 이를 통해 추론을 위한 추가 대기 시간 감소 및 처리량 감소가 가능하며 다음 섹션에서 설명합니다.

표 5: PR-MoE와 PR-MoE + MoS 간의 제로 샷 평가 비교(마지막 6개 열).

모델(숫자 매개변수)	LAMBADA	PIQA	BoolQ	RACE-h	TriviaQA	WebQs	350M+PR-MoE+L24(4B)	73.99	350M+PR-MoE+L21(3.5B)	73.88	350M+PR-MoE+L21+KD만(3.5B)	73.40	350M+PR-MoE+L21+MoS(3.5B)	73.34	1.3B+PR-MoE	74.73
			62.33			52.35	32.54	8.81	4.48							
			61.56			57.89	33.78	12.13	4.87							
			63.46			58.07	34.83	13.69	5.22							
			70.60	77.75		67.16	38.09	28.86	7.73							
1.3B+PR-MoE+L21+KD 전용(27B)			69.73	76.93		64.16	36.17	26.17	6.25							
1.3B+PR-MoE+L21+MoS (27B)			70.17	77.69		65.66	36.94	29.05	8.22							

4절의 결론으로 PR-MoE를 학습하는 동안 원래 표준 MoE 모델에 비해 메모리 소비가 최대 3배 감소합니다. 추론하는 동안 PR-MoE와 MoS는 모델 정확도를 유지하면서 MoE 모델 크기를 최대 3.7배까지 줄입니다.

이는 다음 섹션에서 설명하는 것처럼 대기 시간과 비용의 추론에 상당한 이점이 있습니다.

5 DeepSpeed-MoE 추론: MoE 모델 제공 전례 없는 규모와 속도

MoE 모델이 실제로 유용하려면 추론 대기 시간과 비용을 최적화하는 것이 중요합니다. 추론하는 동안 배치 크기는 일반적으로 작기 때문에 MoE 모델의 추론 대기 시간은 기본적으로 메인 메모리에서 모델 매개변수를 로드하는 데 걸리는 시간에 따라 달라지며, 계산량이 적으면 추론이 빨라진다는 기존의 믿음과 대조됩니다. 따라서 MoE 추론 성능은 전체 모델 크기와 전체 달성 가능한 메모리 대역폭이라는 두 가지 주요 요인에 따라 달라집니다.

이전 섹션에서는 모델 정확도를 유지하면서 MoE 모델 크기를 줄이기 위해 PR-MoE 및 MoS를 제시했습니다. 이 섹션에서는 추론 속도를 높이기 위해 수십 개의 분산 GPU에서 집계된 메모리 대역폭을 활용하는 다중 GPU MoE 추론 시스템을 생성하여 달성 가능한 메모리 대역폭을 최대화하는 시스템 최적화 솔루션을 제시합니다. 함께 DeepSpeed-MoE(간단히 DS-MoE)는 기본 MoE 시스템에 비해 대기 시간이 7.3배 향상되고 비용이 저렴하며 MoE 추론에 비해 최대 4.5배 빠르고 9배 저렴한 대규모 MoE 모델을 지원하는 전례 없는 확장성과 효율성을 제공합니다. 품질 등과 밀도 모델. DS-MoE는 [51]에 제시된 더 큰 DeepSpeed 추론 노력의 일부입니다 .

5.1 DeepSpeed-MoE 추론 시스템 설계

MoE 추론 성능은 흥미로운 역설입니다.

- 최상의 사례 보기에서 MoE 모델(top-1 게이팅 포함)의 각 입력 토큰은 각 MoE 계층에서 단일 전문가만 활성화하여 다음과 같은 중요한 데이터 경로를 생성합니다.

기본 밀집 모델 크기와 동일하며 실제 모델 크기보다 작은 자릿수입니다. 예를 들어 13억 + MoE-128 모델을 추론할 때 각 입력 토큰에는 전체 매개변수 크기가 520억 개인데도 매개변수가 13억 개만 필요합니다.

- 최악의 경우 토큰 배치(예: 문장 또는 텍스트 단락)를 처리하는 데 필요한 집계 매개변수는 전체 모델 크기만큼 클 수 있습니다(다른 토큰이 다른 전문가를 활성화할 수 있으므로). 앞의 예에서 520억 개의 매개변수로 인해 짧은 대기 시간과 높은 처리량을 달성하기가 어렵습니다.

DeepSpeed-MoE 추론 시스템의 설계 목표는 성능을 최상의 경우 보기. 이는 세 가지 잘 조정된 최적화 세트를 통해 달성됩니다.

- 신중하게 모델을 분할하고 다양한 유형의 병렬 처리를 수용합니다. 중요한 데이터 경로가 동일한 모든 토큰을 함께 그룹화하고 라우팅하여 장치당 데이터 액세스를 줄이고 최대 총 대역폭을 달성합니다.

- 효과적으로 토큰을 그룹화하고 라우팅하기 위해 병렬성 조정으로 통신 일정을 최적화합니다. 그리고

- 변환기 및 MoE 관련 커널을 최적화하여 장치별 성능을 개선합니다.

다음 세 섹션에서 이러한 최적화에 대한 심층적인 논의를 제공합니다.

5.2 Tensor-Slicing, Expert-Slicing, Data Parallelism 및 Expert Parallelism

MoE를 위한 전례 없는 규모로 낮은 대기 시간과 높은 처리량을 달성하기 위해 장치당 중요한 데이터 경로를 최소화하고 달성 가능한 총 메모리 대역폭을 최대화하며 동시에 충분한 총 메모리를 제공하여 (1) 전문가 병렬 처리 [32] 및 전문가 매개변수에 대한 슬라이싱 및 (2) 비전문가 매개변수에 대한 데이터 병렬 처리 및 텐서 슬라이싱. 그림 7은 전문가(예: MLP) 및 비전문가 매개변수(예: 주의)를 모두 포함하는 단일 MoE Transformer 레이어와 각 구성 요소를 처리하기 위해 병렬 전략 조합을 사용하는 방법을 보여줍니다.

아래에서는 모델과 데이터가 분할되는 방식과 각 조각을 처리하는 데 사용하는 각 병렬 처리 형식에 대한 세부 정보를 설명합니다.

전문가 매개변수에 대한 전문가 병렬화 및 전문가 슬라이싱 MoE 성능 패러독스에서 설명한 것처럼 각 토큰은 각 MoE 계층에서 단일 전문가만 활성화하지만 여러 토큰을 사용한 일괄 추론의 경우 모든 토큰에 필요한 집계 매개변수는 다음과 같이 커질 수 있습니다. 짧은 대기 시간과 높은 처리량을 모두 달성하기가 어렵습니다. 이 문제를 해결하기 위해 장치 간에 전문가를 분할하고, 동일한 중요 데이터 경로에서 동일한 전문가에게 할당된 모든 입력 토큰을 그룹화하고, 전문가 병렬 처리를 사용하여 서로 다른 장치 간에 서로 다른 중요 경로를 가진 토큰 그룹 처리를 병렬화합니다.

1.3B+MoE-128의 예에서 전문가 병렬 처리가 128이면 각 GPU는 해당 장치의 전문가에 해당하는 단일 토큰 그룹만 처리합니다. 그 결과 장치당 13억 개의 매개변수인 순차 경로가 생성되며, 이는 67억 개의 매개변수가 있는 품질과 동등한 고밀도 모델보다 5배 작습니다. 따라서 이론적으로 MoE 기반 모델은 통신 오버헤드가 없다고 가정하고 전문가 병렬 처리를 사용하는 품질과 동등한 고밀도 모델보다 최대 5배 더 빠르게 실행할 수 있는 잠재력을 가지고 있습니다. 이에 대해서는 다음 섹션에서 논의합니다.

또한 전문가 매개변수를 수평/수직으로 분할하는 전문가 내의 매개변수에 대한 텐서 슬라이싱 개념을 활용하는 "전문가 슬라이싱"을 제안합니다.

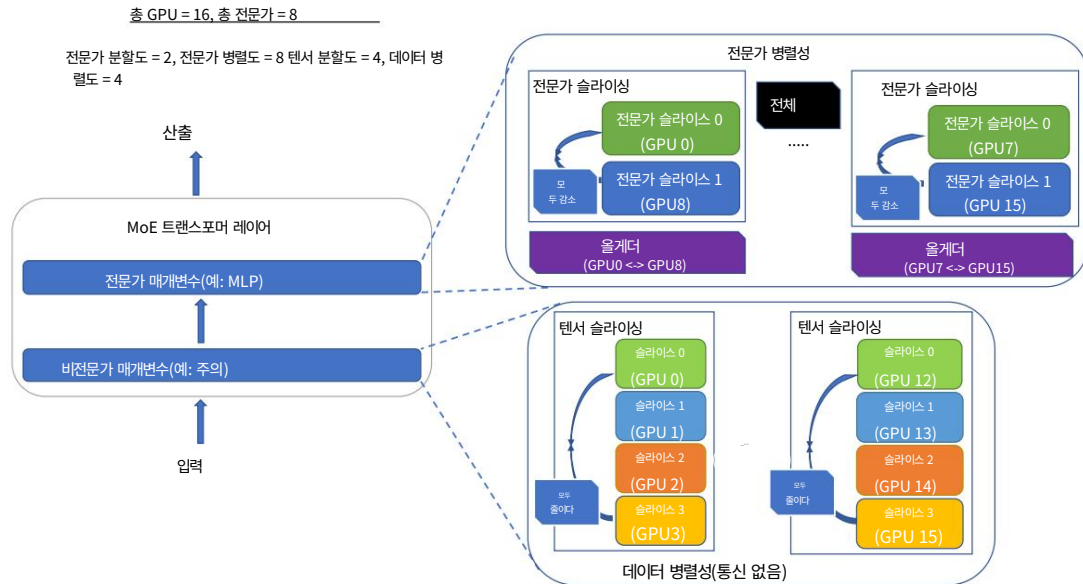


그림 7: 모델의 다양한 파티션(전문가 및 비전문가)에 대한 다차원 병렬 처리의 복잡성을 수용하는 DS-MoE 설계.

여러 GPU에서, 이 추가 병렬 처리 차원은 전문가 수보다 더 많은 장치로 확장되는 대기 시간이 엄격한 시나리오에 유용합니다.

비전문가 매개변수에 대한 데이터 병렬화 및 텐서 슬라이싱 전문가 병렬화는 장치당 임계 경로의 전문가 매개변수 수를 줄이는 반면 임계 경로의 비전문가 매개변수는 줄이지 않습니다. 이로 인해 두 가지 제한 사항이 발생합니다. (1) MoE 모델에서 추론할 수 있는 비전문가 매개변수의 최대 크기는 단일 장치 메모리에 의해 제한되고 (2) 모델의 비전문가 구성 요소의 실행 대기 시간은 제한됩니다. 단일 장치 메모리 대역폭 기준.

우리는 이러한 병목 현상을 해결하기 위해 노드 내에서 텐서 슬라이싱을 사용하여 집계 GPU 메모리를 활용하여 수천억 개의 비전문가 매개변수를 허용하는 동시에 노드 내의 모든 GPU에서 집계 GPU 메모리 대역폭을 활용합니다. 노드 전체에서 텐서 슬라이싱을 수행할 수 있지만 텐서 슬라이싱의 통신 오버헤드와 감소된 계산 세분화로 인해 일반적으로 노드 간 텐서 슬라이싱이 불가능합니다. 여러 노드에서 비전문가 매개변수를 확장하기 위해 통신 오버헤드 또는 계산 세분성 감소가 발생하지 않는 노드에서 서로 다른 배치를 처리하는 비전문가 매개변수 복제본을 생성하여 데이터 병렬성을 사용합니다.

다차원 병렬 처리의 시너지 전문가 병렬 처리 및 전문가 슬라이싱을 텐서 슬라이싱 및 데이터 병렬 처리와 결합함으로써 DS-MoE 추론은 수조 개의 매개변수 MoE 모델(수조 개의 전문가 매개변수와 수천억 개의 비전문가 매개변수 포함)을 확장할 수 있습니다.) 노드 전체에서 수십 또는 수백 개의 장치로. 이러한 장치 전반의 총 대역폭과 장치당 최소화된 중요 데이터 경로는 전혀 없는 규모로 짧은 대기 시간과 높은 처리량 추론을 가능하게 하는 기회를 열어줍니다. 그러나 거기에 도달하려면 여전히 고성능 통신 집합체와 단일 장치 커널이 필요합니다.

우리는 다음에 대해 이야기합니다.

5.3 최적화된 통신 하위 시스템: 토큰을 보다 효율적으로 그룹화 및 라우팅

전문가 병렬 처리에는 모든 전문가 병렬 장치 간의 전면 통신이 필요합니다. 기본적으로 DS-MoE는 "torch.distributed" 인터페이스를 통한 이 통신에 NCCL을 사용하지만 대규모로 사용될 때 주요 오버헤드가 관찰됩니다(자세한 내용은 섹션 5.5 참조). 이를 최적화하기 위해 Microsoft SCCL [52]을 사용하는 맞춤형 통신 인터페이스를 개발하고 NCCL보다 더 나은 성능을 달성합니다. 플러그인 최적화에도 불구하고 대기 시간이 장치 증가에 따라 선형적으로 증가하므로 전문가 병렬 처리를 많은 장치로 확장하는 것은 어렵습니다. 이 중요한 확장 문제를 해결하기 위해 기본 지점 간 NCCL 작업과 사용자 정의 CUDA 커널을 활용하여 필요한 데이터 레이아웃 변환을 수행하는 두 가지 새로운 통신 최적화 전략을 설계했습니다.

계층적 All-to-all 계층적 트리 기반 알고리즘은 종종 통신 홉 수를 줄이기 위해 allreduce, broadcast 등과 같은 통신 집산과 함께 사용됩니다. 우리는 계층적 율투윌을 데이터 레이아웃 변환, 노드 내 율투윌, 두 번째 데이터 레이아웃 변환 및 최종 노드 간 변환을 포함하는 2단계 프로세스로 구현합니다. 모두에게. 이렇게 하면 통신 홉이 $O(p)$ 에서 $O(G + p/G)$ 로 줄어듭니다. 여기서 G 는 노드의 GPU 수이고 p 는 총 GPU 장치 수입니다. 그림 8은 이 구현의 설계 개요를 보여줍니다. 통신 볼륨이 2배 증가했음에도 불구하고 이 계층적 구현을 통해 이 메시지 크기의 통신은 대역폭 제한보다 대기 시간 제한이 더 크기 때문에 작은 배치 크기에 대해 더 나은 확장이 가능합니다.

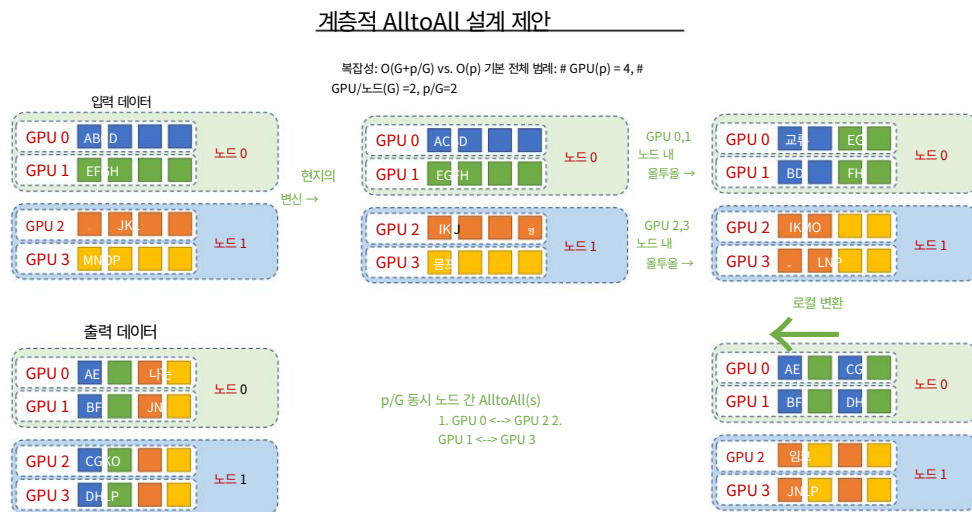


그림 8: 제안된 계층적 all-to-all 디자인의 그림.

병렬성 조정 통신 최적화 단일 모델 내에서 전문가 병렬화 및 텐서 슬라이싱을 데이터 병렬화와 결합하는 것은 통신을 효과적으로 처리하는 측면에서 쉽지 않습니다. 텐서 슬라이싱은 개별 연산자를 여러 GPU로 분할하며

전문가 병렬 처리는 GPU를 분할하지 않고 전문가 연산자를 GPU 전반에 배치하고 그들 사이에 모두를 필요로 합니다. 이러한 통신을 처리하는 순진한 접근 방식은 각 병렬 처리를 블랙 박스로 취급하여 필요한 통신을 독립적으로 수행하는 것입니다. 그러나 이것은 차선의 성능으로 이어질 것입니다.

텐서 슬라이싱의 전체 축소 작업은 관련 장치 간에 데이터를 복제합니다. 텐서 병렬 연산자 다음에 전문 병렬 연산자를 실행할 때 이 복제를 통해 모든 전문 병렬 프로세스 간의 통신이 필요하지 않은 전체 연산자에 대해 최적화된 통신 일정을 만들 수 있습니다. 대신, 텐서 병렬 랭크 전체의 데이터가 복제되기 때문에 동일한 텐서 슬라이싱 랭크를 공유하는 장치의 하위 집합 내에서만 유틸리티가 발생할 수 있습니다(그림 9). 결과적으로 all-to-all의 대기 시간은 $O(p)$ 대신 $O(p/L)$ 로 제한됩니다. 여기서 L 은 텐서 슬라이싱 병렬도이고 p 는 총 GPU 장치 수입니다.

유사하게 텐서 슬라이싱 연산자가 뒤따르는 전문 병렬 연산자를 실행할 때 최종 유틸리티는 같은 방식으로 수행될 수 있지만 이번에는 텐서 슬라이싱에 필요한 데이터를 복제하기 위해 텐서 병렬 랭크 사이에 allgather 연산자가 뒤따릅니다. (그림 9). 이렇게 하면 대기 시간 오버헤드가 $O(p)$ 에서 $O(p/L) + O(L)$ 로 줄어듭니다.

대기 시간 오버헤드가 감소하여 많은 수의 장치로 더 잘 확장할 수 있습니다. 예를 들어, 8방향 텐서 슬라이싱 및 128방향 전문가 병렬 처리를 사용하여 128개의 GPU로 확장할 때 이 접근 방식은 8-웨이 텐서 슬라이싱, 여기서 $C1$ 과 $C2$ 는 점대점 대기 시간, 메시지 크기 및 대역폭에 의해 결정되는 일부 상수입니다.

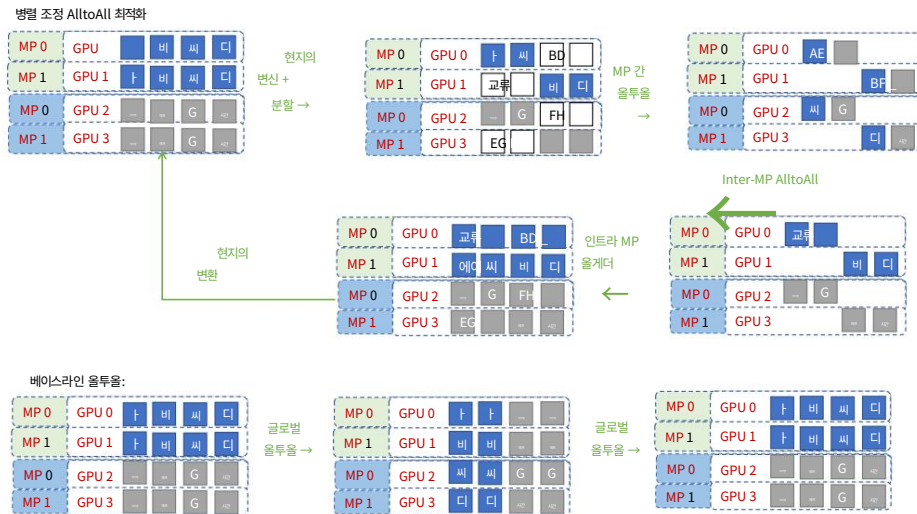


그림 9: 병렬성 조정 통신의 그림.

5.4 고도로 최적화된 변압기 및 MoE 관련 커널

DS-MoE 추론 시스템은 고도로 최적화된 다중 GPU 변환기 커널과 고도로 최적화된 MoE 관련 커널로 구성됩니다. 우리는 비전문가 변압기 계층의 대역폭 활용을 최대화하기 위해 DeepSpeed 추론 커널을 사용합니다. 자세한 내용은 [53]을 참조하십시오.

이 섹션에서는 게이팅을 수행하기 위한 MoE 관련 연산자와 다양한

문헌에서 sparse-dense einsum을 사용하여 전통적으로 구현된 데이터 레이아웃 변환. 높은 수준에서 우리는 계산 복잡성을 3차에서 2차로 줄이기 위해 고도로 희소한 einsum 대신 명시적 데이터 레이아웃 변환으로 구현하여 이러한 연산자를 최적화합니다. 또한 이러한 연산자 대부분을 단일 커널에 융합합니다.

보다 구체적으로 MoE 관련 계산은 세 가지 주요 구성 요소로 구성됩니다.

- 결과가 희소 텐서(시퀀스의 각 토큰에 할당된 전문가를 나타내는 원-핫 벡터)로 표시되는 전문가에 대한 토큰 할당을 결정하는 게이팅 기능.
- 원-핫 텐서와 모든 토큰 사이의 스parse einsum 연산자는 다음을 정렬합니다.
할당된 전문가 ID를 기반으로 토큰 순서 지정.
- 토큰을 확장하고 다시 정렬하는 MoE 계산 종료 시 최종 einsum
원래 주문으로 돌아옵니다.

게이팅 함수 및 희소 einsum 연산자의 희소 텐서 표현은 상당한 대기 시간 오버헤드를 도입합니다. 첫째, 게이팅 기능에는 토큰 마스크를 생성하고, top-k 전문가를 선택하고, 각 전문가에게 가는 토큰 ID를 찾기 위해 누적 합계(cumsum)를 수행하고 희소 행렬 곱하기를 수행하는 수많은 작업이 포함됩니다. 드문 테너 표현으로 인해 낭비적이지만 많은 커널 호출 호출로 인해 매우 느립니다.

또한 희소 einsum은 $S \times E \times M \times ce$ 의 복잡도를 가지며, 여기서 S는 총 토큰 수, E는 전문가 수, M은 모델 숨겨진 차원, ce는 전문가 용량(S, E 및 M은 주요 복잡성 요소인 반면 ce는 일반적으로 매우 작습니다. 이 방정식에서 각 토큰에 대한 E 연산자 중 (E - 1)은 ce 토큰을 처리하기 위해 일반적으로 한 명의 전문가만 선택되기 때문에 0이 있는 곱셈 및 덧셈입니다.

이것은 게이팅 작업을 일반화하면 각 전문가에 대한 올바른 토큰을 선택하기 위해 0으로 많은 불필요한 계산을 생성하는 여러 마스크 행렬 또는 원-핫 벡터에 대한 einsum이 발생한다는 사실에서 비롯됩니다.

조밀한 표현과 커널 융합을 사용하여 이러한 연산자를 최적화합니다.

먼저 게이팅 기능을 단일 커널에 융합하고 조밀한 토큰-전문가 매핑 테이블을 사용하여 토큰에서 전문가로의 할당을 나타내어 커널 시작 오버헤드와 메모리 및 희소 표현으로 인한 컴퓨팅 오버헤드를 크게 줄입니다..

보다 구체적으로 게이팅 커널은 각 전문가에게 올바른 토큰을 분배하기 위해 top-k, cumsum 및 scatter 작업을 포함합니다. top-k 연산자는 각 입력 토큰에 대해 k-가장 높은 로짓을 가진 k명의 전문가를 선택하고 MoE 모델의 경우 k는 일반적으로 작기 때문에(예: 1 또는 2) 최고의 전문가 인덱스를 매핑 테이블에 저장합니다. 나머지 게이팅 기능 작업을 위해 마스크를 만드는 것보다 Cumsum은 각 전문가가 처리한 토큰의 ID를 계산합니다. 이는 MoE 구성의 용량 계수로 정의됩니다. 소위 Blelloch 스캔 알고리즘을 사용하여 GPU 아키텍처에서 cumsum을 병렬화합니다. 마지막으로 MoE 전문가에게 올바른 토큰을 라우팅하기 위해 매핑 테이블과 토큰 ID를 사용합니다.

둘째, 나머지 두 개의 sparse einsum을 최적화하기 위해 위에서 언급한 매핑 테이블을 사용하여 데이터 레이아웃 변환으로 구현하여 먼저 전문가 ID를 기반으로 정렬한 다음 sparse einsum을 요구하지 않고 원래 순서로 되돌려 $S \times E \times M \times ce$ 에서 $S \times M \times ce$ 까지 이러한 작업의 복잡성. 데이터 변환과 함께 해당 게이팅 로짓(확률 영역에서)을 사용하여 전문가 출력을 업데이트합니다.

이러한 최적화를 결합하면 MoE 커널 관련 대기 시간이 6배 이상 감소합니다.

5.5 DS-MoE 추론의 성능 평가

최신 생산 환경에서 강력한 DL 모델은 종종 트래픽 수요를 충족하고 짧은 대기 시간을 제공하기 위해 수백 개의 GPU 장치를 사용하여 제공됩니다. 이 섹션에서는 대규모 MoE 모델 추론을 위해 높은 처리량과 낮은 대기 시간이라는 두 가지 광범위한 목표를 실현할 수 있는 방법을 살펴봅니다. 또한 MoE 모델 추론이 밀집된 대응 모델과 어떻게 다른지 살펴봅니다.

우리가 탐구하려는 주요 질문은 다음과 같습니다. 1)

MoE 추론의 고유한 속성은 무엇입니까?

2) 모델 크기와 리소스가 증가함에 따라 성능과 확장성은 어떻게 됩니까?

3) PR-MoE 및 MoS와 같은 모델 최적화는 MoE 모델 추론에 어떤 이점을 제공합니까?

4) 대기 시간 및 비용과 관련하여 MoE 모델은 품질과 동등한 고밀도 모델과 어떻게 비교됩니까?

표 6에 표시된 다양한 모델 구성을 사용하여 대규모(Azure에서 최대 256개의 A100 GPU)에 대한 이러한 질문에 답하려고 합니다.

표 6: DS MoE 추론 시스템의 성능 평가에 사용되는 다양한 MoE 모델의 구성. 이러한 구성은 표준 MoE 아키텍처 사례를 나타내며 크기는 더 작지만 품질은 동일한 PR-MoE 및 PR-MoE+MoS 사례도 테스트합니다.

모델 크기(십억)	#Layers 숨겨진 크기	MP 정도	EP 정도	1	
1.3B+MoE-128	52	24	2048년		128
2.4B+MoE-128	107.7	16	3584	1	128
8B+MoE-128	349.0	30	4096	4	128
24B+MoE-128	1064.9	40	8192	8	128
47B+MoE-128	2024.0	58	8192	8	128

5.5.1 낮은 대기 시간 및 초선형 처리량 증가 동시 달성 하게

밀도가 높은 모델의 경우 여러 GPU와 데이터 병렬 처리(GPU 간 통신이 없는 종속 복제본에서)를 사용하여 처리량을 늘릴 수 있는 반면, 여러 GPU에 걸쳐 모델을 분할하는 텐서 슬라이싱과 같은 기술을 통해 대기 시간을 줄일 수 있습니다[53]. 총 처리량 측면에서 최상의 확장 사례는 GPU 수가 증가함에 따라 선형적입니다. 즉, GPU당 일정한 처리량입니다. 이는 GPU 간에 통신이 없기 때문에 순수한 데이터 병렬 추론 시나리오에서 가능합니다. 대기 시간을 줄이기 위해 텐서 슬라이싱 스타일의 모델 병렬 처리가 유익한 것으로 입증되었지만 [53] 비용(GPU 간 통신 오버헤드)이 수반되며, 이는 종종 GPU당 처리량을 낮추고 총 처리량의 준선형 확장을 초래합니다. 즉, 밀도가 높은 모델의 경우 병렬 처리를 활용하여 대기 시간과 처리량을 동시에 최적화할 수 없습니다. 그들 사이에는 장단점이 있습니다. 그러나 MoE 추론은 최적화된 대기 시간과 처리량을 동시에 제공하는 동시에 많은 수의 장치로 확장할 수 있는 고유한 기회를 제공합니다.

이러한 기회를 연구하기 위해 우리는 52B MoE 모델(1.3B 기본 모델 및 128명의 전문가)을 8 GPU에서 64 GPU로 확장하고 DeepSpeed MoE 추론 시스템의 대기 시간 및 처리량 추세를 강력한 기준인 완전한 기능을 갖춘 분산형 PyTorch와 비교하여 관찰합니다. 텐서 슬라이싱과 전문가 병렬 처리가 모두 가능한 구현 [32]. 그림 10에서 볼 수 있듯이 DeepSpeed와 PyTorch 모두 예상대로 GPU 수가 증가함에 따라 추론 대기 시간이 감소하는 것을 관찰할 수 있습니다. PyTorch는 DeepSpeed에 비해 훨씬 느리고 최대 32 GPU까지만 확장됩니다.

반면 처리량 추세는 흥미롭고 추가 분석이 필요합니다.

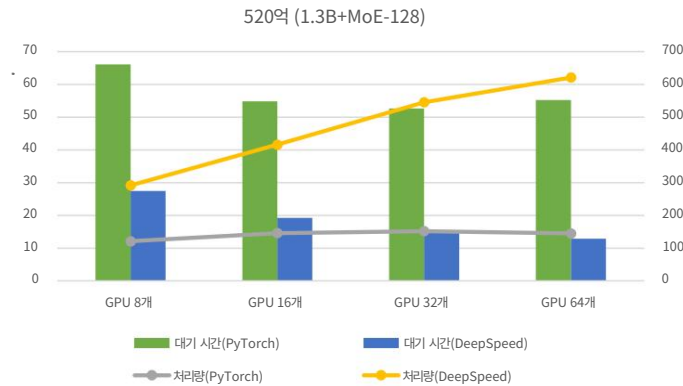


그림 10: 8~64개의 GPU를 사용하여 128명의 전문가가 있는 520억 MoE 모델에 대해 PyTorch(기준선)보다 DeepSpeed-MoE 추론 시스템(최적화됨)이 제공하는 지연 시간 및 처리량 개선.

앞서 언급했듯이 밀집 모델에 대한 최상의 처리량 확장은 GPU 수와 관련하여 선형입니다. 그러나 그림 10의 결과는 GPU 수를 8개에서 64개로 늘릴 때 DeepSpeed가 GPU당 처리량을 증가시켜 총 처리량이 초선형으로 증가함을 보여줍니다. 이것은 조밀한 모델과 극명한 대조를 이루며 조밀한 모델에 비해 MoE 모델을 확장하는 것의 주요 이점을 보여줍니다.

조금 더 깊이 들어가면 전문가 병렬 처리의 두 가지 주요 속성이 작동하는 것을 볼 수 있습니다. 1) 전문가 병렬 처리를 사용할 때 GPU 수가 증가함에 따라 GPU당 전문가 수가 감소합니다. 예를 들어 이 52B MoE 모델에는 총 128명의 전문가가 있습니다. 8개의 GPU를 사용하여 이를 제공하는 경우 GPU당 16명의 전문가가 필요한 반면, 64개의 GPU에서는 GPU당 2명의 전문가만 필요합니다. GPU당 전문가의 감소는 각 GPU가 이제 메모리에서 더 적은 데이터(전문가 매개변수에 필요함)를 읽고 있기 때문에 데이터 지역성에 좋습니다. to-all)은 여러 GPU에 상주합니다. 전문가 병렬 처리의 이러한 속성은 PyTorch와 DeepSpeed 모두에 적용됩니다. 그러나 DeepSpeed-MoE는 전문가 병렬 처리의 이점을 최대한 활용할 수 있지만 PyTorch는 그렇게 할 수 없습니다. DeepSpeed는 고급 통신 최적화(섹션 5.3)를 활용하므로 전문가 병렬 처리의 통신 병목 현상을 크게 극복합니다. 동시에 GPU당 전문가가 줄어들 때 증가된 데이터 지역성을 활용할 수 있는 고도로 최적화된 커널(섹션 5.4)이 있습니다. PyTorch에 대한 이러한 주요 승리는 DeepSpeed-MoE를 대규모 확장으로 처리량을 선형적으로 증가시키고 짧은 대기 시간과 높은 처리량을 동시에 달성하기 위한 프레임워크로 만듭니다.

5.5.2 전례 없는 규모의 낮은 대기 시간 및 높은 처리량

모델 규모가 MoE 추론에 미치는 영향을 연구하기 위해 이제 그림 11에서 PyTorch 및 DeepSpeed를 사용하여 1,070억 개의 매개변수에서 2조 개의 매개변수까지 MoE 모델을 탐색합니다.

- DeepSpeed-MoE는 대기 시간을 최대 7.3배 줄이고 최대 7.3배 더 높은 성능을 달성합니다. 기준선과 비교한 처리량.

그림 6

- 수백 개의 GPU를 병렬로 효율적으로 활용함으로써 DeepSpeed-MoE는 믿을 수 없을 정도로 짧은 대기 시간에서 전례 없는 추론 규모를 달성합니다. 엄청난 수의 매개변수 MoE 모델을 25ms 미만에서 추론할 수 있습니다.



그림 11: 다양한 MoE 모델 크기(1070억에서 2조 매개변수)에 대해 PyTorch(기준선)보다 DeepSpeed-MoE(최적화됨)가 제공하는 대기 시간 및 처리량 개선. 우리는 베이스라인의 모든 구성에 128개의 GPU를 사용하고 DeepSpeed-MoE의 경우 128/256개의 GPU를 사용합니다(1조 규모 모델의 경우 256개의 GPU). 여기에 표시된 처리량은 GPU당이며 클러스터의 총 처리량을 얻으려면 GPU 수를 곱해야 합니다.

5.5.3 PR-MoE 및 MoS의 향상된 이점

그림 7

DeepSpeed-MoE 추론 시스템이 제공하는 시스템 최적화와 PR-MoE 및 MoS의 모델 혁신을 결합함으로써 DeepSpeed-MoE는 두 가지 이점을 더 제공합니다.

- 이러한 모델에서 추론을 수행하는 데 필요한 최소 GPU 수를 줄입니다.

그림 12와 같이.

- 다양한 MoE 모델 크기의 대기 시간과 처리량을 모두 개선합니다(그림 13 참조).

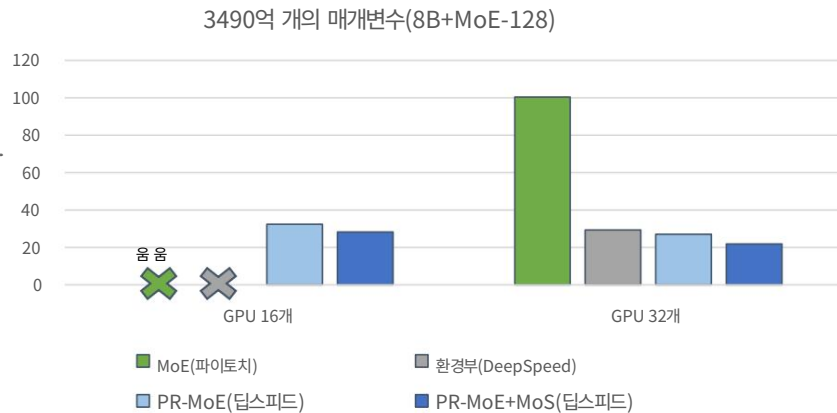


그림 12: PR-MoE+MoS를 사용할 때 MoE 추론에 필요한 리소스가 2배 적습니다. 그림 제목은 표준 MoE 아키텍처에서 모델 크기를 나타내며 PR-MoE 및 PR-MoE+MoS는 크기는 작지만 동일한 투영 품질을 갖습니다.

그림 8

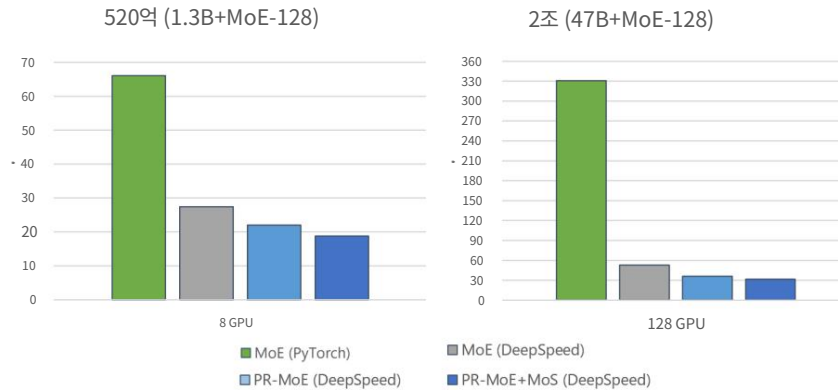


그림 13: 다양한 GPU 수 및 모델 크기에서 표준 MoE를 PR-MoE 및 PR-MoE+MoS 압축과 비교한 추론 대기 시간. 그림 제목은 표준 MoE 아키텍처에서 모델 크기를 나타내며 PR-MoE 및 PR-MoE+MoS는 크기는 작지만 품질은 동일합니다(2조의 경우 예상).

그림 12와 13 모두에 대해 기본 버전(PyTorch의 표준 MoE)과 함께 세 가지 모델 변형의 비교를 보여줍니다. (i) MoE(DeepSpeed)로 표시된 표준 MoE 모델, (ii) PR-MoE(DeepSpeed) 및 (iii) PR-MoE+MoS(DeepSpeed). 결과는 PR-MoE+MoS 모델이 가장 낮은 대기 시간을 제공하고 32개의 GPU 대신 16개의 GPU만 사용하여 모델을 제공할 수 있음을 보여줍니다.

5.5.4 품질과 동등한 고밀도 모델보다 더 나은 대기 시간 및 처리량

품질과 동등한 고밀도 모델과 비교하여 MoE 모델의 추론 성능을 더 잘 이해하려면 MoE 모델이 훈련하는 데 5배 더 빠르고 저렴하지만 추론에는 그렇지 않을 수 있다는 점에 유의해야 합니다. 추론 성능에는 다양한 병목 현상이 있으며 주요 요인은 계산 대신 메모리에서 읽은 데이터의 양입니다.

우리는 두 가지 표준 MoE 모델에 대한 추론 대기 시간과 처리량을 품질과 동등한 밀도 모델과 비교하여 보여줍니다. (1) 67억 매개변수 밀도 모델과 비교한 520억 매개변수 MoE 모델(1.3B-MoE-128)) 그림 14 및 15의 1,750억 매개변수 고밀도 모델과 비교하여 1조 5,000억 매개변수 MoE 모델. 또한 품질과 동등한 PR-MoE+MoS 모델을 테스트했습니다.

PyTorch를 사용하는 경우 MoE 모델 추론은 품질과 동등한 고밀도 모델에 비해 더 비싸고 느립니다. 이는 두 모델 크기 모두에 해당됩니다. 그러나 DeepSpeed의 최적화는 이러한 추세를 역전시키고 MoE 모델 추론을 품질과 동등한 고밀도 모델에 비해 더 빠르고 저렴하게 만듭니다. 이것은 매우 중요한 결과입니다. 훈련뿐만 아니라 실제 배포가 가장 중요한 추론 대기 시간 및 비용 측면에서 고밀도 모델에 비해 MoE 모델의 이점을 보여줍니다.

그림 14의 결과를 그림 15와 비교하면 모델 크기가 커질수록 MoE 모델이 밀집 모델보다 더 큰 이점이 있음을 알 수 있습니다. 그림 14에서 10억 규모의 PR-MoE+MoS 모델(DeepSpeed-MoE에서 제공됨)은 67억 개의 매개변수 고밀도 모델(PyTorch에서 제공됨)보다 2.4배 빠르고 저렴하지만, 그림 15에서는 1조 규모의 PR-MoE+MoS 모델은 1,750억 매개변수 밀집 모델보다 4.5배 빠르고 9배 저렴합니다. DeepSpeed는 텐서를 사용할 때 통신 오버헤드를 줄이기 위해 병렬성 조정 최적화를 활용하기 때문에 더 큰 모델의 경우 이점이 증가합니다.

우레 9

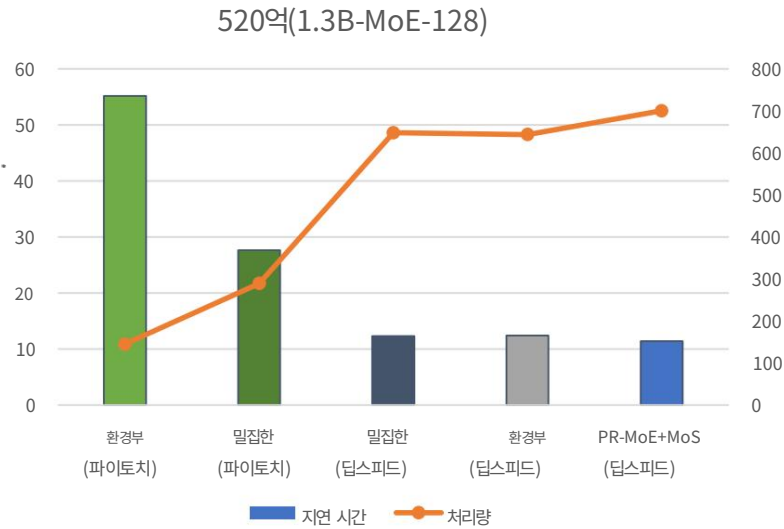


그림 14: MoE 모델과 품질에 상응하는 67억 매개변수 밀집 모델의 추론 대기 시간 비교. 67억 개의 매개변수 모델에 가장 낮은 대기 시간을 제공하므로 1개의 GPU를 사용합니다. MoE 모델에는 128개의 GPU를 사용합니다. 품질 동등성은 훈련 섹션에 제시된 실험에 의해 검증되었습니다. 그림 제목은 표준 MoE 아키텍처에서 모델 크기를 나타내며 PR-MoE 및 PR-MoE+MoS는 크기는 작지만 품질은 동일합니다.

우레 10

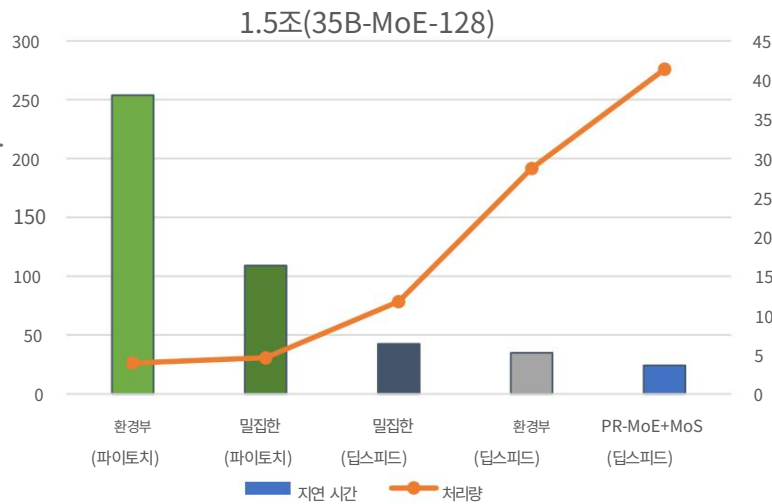


그림 15: MoE 모델과 품질에 상응하는 1,750억 밀도 모델의 측정된 추론 대기 시간 비교. 우리는 출판된 문헌의 관찰뿐만 아니라 그림 14의 소규모 실험의 스케일링 법칙이 유지된다는 가설로 이 두 모델의 품질 동등성을 가정합니다. 그림 제목은 표준 MoE 아키텍처에서 모델 크기를 나타내며 PR-MoE 및 PR-MoE+MoS는 크기는 작지만 동일한 두영 품질을 갖습니다.

모델의 비전문가 부분을 슬라이싱합니다. 또한 이 규모에서 전문가 슬라이싱을 활용할 수 있으므로 PyTorch 기준선에 비해 더 많은 수의 GPU로 확장할 수 있습니다. 또한 더 큰 1조 5000억 개의 매개 변수 MoE 모델의 경우 그림 15와 같이 대기 시간에 비해 처리량이 2배 더 향상되는 것을 관찰했습니다. 이는 MoE 모델이 고밀도 모델(8방향 vs. 16-way) 따라서 배치 크기가 2배 더 큼니다.

전반적으로 DeepSpeed-MoE는 PyTorch를 사용하여 품질과 동등한 고밀도 모델을 제공하는 것과 비교하여 최대 4.5배 빠르고 최대 9배 저렴한 MoE 모델 추론을 제공합니다. 이러한 결과에서 볼 수 있듯이 모델 크기 및 하드웨어 리소스에 따라 확장되는 이점을 통해 MoE 모델이 AI 규모의 차세대 발전을 가져오는 데 중요할 것이라고 믿게 됩니다.

6 차세대 AI 스케일 시대

최근 모델 크기가 기하급수적으로 증가함에 따라 최신 슈퍼컴퓨팅 클러스터가 대형 모델을 교육하고 제공할 수 있는 한계에 도달했습니다. 하드웨어 리소스에 대한 탑재 가능한 요구 사항으로 인해 단순히 모델 크기를 늘리는 것만으로는 더 나은 모델 품질을 달성하기가 점점 더 어려워지고 있습니다. 우리가 가진 선택은 차세대 하드웨어를 기다리거나 현재 하드웨어를 사용하여 교육 및 추론 효율성을 혁신하고 개선하는 것입니다.

우리는 최근 문헌 [30, 31] 과 함께 MoE 기반 모델이 어떻게 대규모 NLG 모델의 훈련 비용을 품질과 동등한 조밀한 모델에 비해 몇 배로 줄여 다음 규모의 AI를 훈련할 수 있는 가능성을 제공하는지 입증했습니다. 현재 세대의 하드웨어에 대한 모델. 그러나 이 작업 이전에 우리가 아는 한, 밀집 모델과 비슷하거나 더 나은 대기 시간 및 비용으로 MoE 모델(더 많은 매개 변수 포함)을 제공하는 방법에 대한 기존 작업이 없었습니다. 이는 대규모 MoE 모델의 실제 배포를 차단하는 어려운 문제입니다.

MoE 모델에 대한 실용적이고 효율적인 추론을 가능하게 하기 위해 새로운 PR-MoE 모델 아키텍처와 MoS 증류 기술을 제공하여 이러한 모델의 메모리 요구 사항을 크게 줄입니다. 또한 MoE 추론 프레임워크를 제공하여 전혀 없는 모델 규모에서 놀랍도록 낮은 대기 시간과 비용을 달성합니다. 이러한 혁신을 결합하여 이러한 MoE 모델을 제공할 수 있을 뿐만 아니라 품질이 동등한 고밀도 모델보다 낮은 대기 시간과 비용으로 추론에 사용할 수 있습니다.

전체적으로 새로운 혁신과 인프라는 컴퓨팅 리소스를 늘리지 않고도 차세대 AI 규모의 교육 및 추론을 향한 유망한 경로를 제공합니다. 조밀한 MoE 모델에서 희소한 MoE 모델로의 전환은 더 적은 리소스로 더 높은 품질의 모델을 배포하는 것이 더 광범위하게 가능해지는 대규모 모델 환경에서 새로운 방향으로의 길을 열 수 있습니다.

기여

SR은 NLG 훈련 실험을 설계하고 추론 시스템을 설계했습니다.
CL은 NLG 훈련 실험(섹션 3)을 이끌었고 섹션 4에 기여했습니다.
ZY는 PR-MoE의 설계 및 실험과 시스템 지원을 주도했습니다(4.1절).
MZ는 MoS(섹션 4.2)와 메모리 효율적인 체크포인트링의 설계 및 실험을 주도했습니다.
RYA와 AAA는 추론 시스템의 개발과 실험을 주도했습니다(섹션 5).
JR은 여러 소프트웨어 기능을 DeepSpeed에 개발, 디버깅 및 통합했습니다.
YH는 전체 연구 프로젝트를 설계, 관리 및 이끌었습니다.

승인

DeepSpeed-MoE 소프트웨어의 개발, 디버깅, 테스트 및 릴리스에 기여한 Microsoft DeepSpeed 팀의 Olatunji Ruwase에게 감사드립니다. 이 작업은 Microsoft Turing 팀의 Brandon Norick, Zhun Liu, Xia Song, Microsoft Z-Code 팀의 Young Jin Kim, Alex Muzio, Hany Hassan Awadalla, 마이크로소프트 SCCL 팀.

참조

- [1] Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley 및 Yuxiong He. DeepSpeed-MoE: 차세대 AI 규모를 강화하기 위해 전문가 추론 및 교육의 혼합을 발전시킵니다. Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu 및 Sivan Sabato, 편집자, 기계 학습에 관한 제39차 국제 회의 절차, 기계 학습 연구 절차의 162권, 18332–18346페이지. PMLR, 2022년 7월 17~23일.
- [2] 엔비디아. DeepSpeed와 Megatron을 사용하여 세계에서 가장 크고 가장 강력한 생성 언어 모델인 Megatron-Turing NLG 530B를 교육합니다. <https://developer.nvidia.com/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/>, 2021.
- [3] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton 및 Jeff Dean. 터무니없이 큰 신경망: 희박하게 게이트된 전문가 레이어의 혼합. arXiv 사전 인쇄 arXiv:1701.06538, 2017.
- [4] Dmitry Lepikhin, Hyukjoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, Zhifeng Chen. Gshard: 조건부 계산 및 자동 샤딩을 사용하여 거대한 모델을 확장합니다. arXiv 프리프린트 arXiv:2006.16668, 2020.
- [5] William Fedus, Barret Zoph, Noam Shazeer. 스위치 변환기: 간단하고 효율적인 희소성으로 1조 개의 매개변수 모델로 확장합니다. arXiv 프리프린트 arXiv:2101.03961, 2021.
- [6] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, Dario Amodei. 신경 언어 모델의 스케일링 법칙. arXiv 프리프린트 arXiv:2001.08361, 2020.
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: 언어 이해를 위한 심층 양방향 변환기의 사전 훈련. NAACL-HLT(1), 2019.
- [8] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov 및 Quoc V Le. Xlnet: 언어 이해를 위한 일반화된 자동 회귀 사전 훈련. 신경 정보 처리 시스템의 발전, 페이지 5753–5763, 2019.
- [9] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer 및 Veselin Stoyanov. Roberta: 강력하게 최적화된 bert 사전 훈련 접근 방식입니다. arXiv 프리프린트 arXiv:1907.11692, 2019.

- [10] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma 및 Radu Soricut. ALBERT: 언어 표현의 자기 지도 학습을 위한 라이트 버트. 학습 표현에 관한 국제 회의에서, 2019.
- [11] Alec Radford, Karthik Narasimhan, Tim Salimans 및 Ilya Sutskever. 언어 개선 생성적 사전 훈련을 통한 이해 OpenAI 블로그, 2018.
- [12] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever 등. 언어 모델은 감독되지 않은 멀티태스킹 학습자입니다. OpenAI 블로그, 1(8):9, 2019.
- [13] 코비 로셋. Turing-nlg: Microsoft의 170억 매개변수 언어 모델입니다. 마이크로소프트 블로그, 1:2, 2020.
- [14] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, Bryan Catanzaro. Megatron-LM: GPU 모델 병렬 처리를 사용하여 수십억 개의 매개변수 언어 모델을 교육합니다. arXiv 프리프린트 arXiv:1909.08053, 2019.
- [15] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu. 통합 텍스트-텍스트 변환기를 사용하여 전이 학습의 한계를 탐색합니다. arXiv 프리프린트 arXiv:1910.10683, 2019.
- [16] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda 및 Raquel Fernández. 람바다 데이터 세트: 광범위한 담화 컨텍스트가 필요한 단어 예측. arXiv 프리프린트 arXiv:1606.06031, 2016.
- [17] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R Bowman. Glue: 자연어 이해를 위한 멀티태스킹 벤치마크 및 분석 플랫폼입니다. arXiv 프리프린트 arXiv:1804.07461, 2018.
- [18] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R Bowman. Superglue: 범용 언어 이해 시스템을 위한 더 끈끈한 벤치마크입니다. arXiv 프리프린트 arXiv:1905.00537, 2019.
- [19] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, James Allen. 상식적인 이야기를 더 깊이 이해하기 위한 말뭉치 및 평가 프레임워크. arXiv 프리프린트 arXiv:1604.01696, 2016.
- [20] 조나단 베란트, 앤드류 추, 로이 프로스티그, 퍼시 랑. 질문-답변 쌍에서 freebase에 대한 시맨틱 구문 분석. 자연어 처리의 경험적 방법에 관한 2013년 회의 절차, 1533–1544, 2013페이지.
- [21] Mandar Joshi, 최은솔, Daniel S Weld, Luke Zettlemoyer. Triviaqa: 독해를 위한 대규모 원거리 감독 챌린지 데이터 세트입니다. arXiv 사전 인쇄 arXiv:1705.03551, 2017.
- [22] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell 등. 언어 모델은 소수 학습자입니다. arXiv 프리프린트 arXiv:2005.14165, 2020.
- [23] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, Yuxiong He. 0: 1조 개의 매개변수 모델 훈련을 위한 메모리 최적화. SC20: 고성능 컴퓨팅, 네트워킹, 스토리지 및 분석을 위한 국제 회의, 1–16페이지. IEEE, 2020.

- [24] 사이드 마수드니아와 레자 에브라힘푸르. 전문가 혼합: 문헌 조사. *인공 지능 검토*, 42(2):275–293, 2014.
- [25] Sepp Hochreiter와 Jürgen Schmidhuber. 장단기 기억. *신경계산*, 9(8):1735–1780, 1997.
- [26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser 및 Illia Polosukhin. 주의가 필요합니다. *신경 정보 처리 시스템의 발전*, 페이지 5998–6008, 2017.
- [27] Junyang Lin, An Yang, Jinze Bai, Chang Zhou, Le Jiang, Xianyan Jia, Ang Wang, Jie Zhang, Yong Li, Wei Lin 등. M6-10t: 효율적인 수조 매개변수 사전 훈련을 위한 공유-연결 해제 패러다임. *arXiv 프리프린트 arXiv:2110.03888*, 2021.
- [28] 김영진, Ammar Ahmad Awan, Alexandre Muzio, Andres Felipe Cruz Salinas, Liyang Lu, Amr Hendy, Samyam Rajbhandari, Yuxiong He, Hany Hassan Awadalla. 멀티태스킹 다국어 모델을 위한 확장 가능하고 효율적인 모에 교육. *arXiv 프리프린트 arXiv:2109.10465*, 2021.
- [29] Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Tuo Zhao, Jianfeng Gao. 확률론적 전문가와 함께 드물게 활성화된 변압기 길들이기. *arXiv 프리프린트 arXiv:2110.04260*, 2021.
- [30] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat 등. Glam: 전문가 혼합을 통한 언어 모델의 효율적인 확장. *arXiv 프리프린트 arXiv:2112.06905*, 2021.
- [31] Mikel Artetxe, Shruti Bhosale, 나만 고알, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giri Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Mona Diab, Zornitsa Kozareva 및 Ves Stoyanov. 전문가가 혼합된 효율적인 대규모 언어 모델링. *arXiv 프리프린트 arXiv:2112.10684*, 2021.
- [32] 김영진, Ammar Ahmad Awan, Alexandre Muzio, Andrés Felipe Cruz Salinas, Liyang Lu, Amr Hendy, Samyam Rajbhandari, Yuxiong He, Hany Hassan Awadalla. 멀티태스킹 다국어 모델을 위한 확장 가능하고 효율적인 모에 교육. *CoRR*, abs/2109.10465, 2021.
- [33] Jiaao He, Jiezhong Qiu, Aohan Zeng, Zhilin Yang, Jidong Zhai 및 Jie Tang. Fastmoe: 빠른 혼합 전문가 교육 시스템입니다. *CoRR*, abs/2103.13262, 2021.
- [34] 마이크로소프트. Tutel: 대규모 dnn 모델 교육을 위한 효율적인 전문가 혼합 구현입니다. <https://www.microsoft.com/en-us/research/blog/tutel-an-efficient-mix-of-experts-implementation-for-large-dnn-model-training/>, 2021.
- [35] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, 최예진 등. Piqa: 자연어의 물리적 상식에 대한 추론. *인공 지능에 관한 AAAI 회의 절차*, 34권, 7432–7439페이지, 2020.
- [36] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, Eduard Hovy. 인종: 시험의 대규모 독해 데이터 세트. *arXiv 프리프린트 arXiv:1704.04683*, 2017.

- [37] 매튜 D 자일러와 롭 퍼거스. 컨볼루션 네트워크 시각화 및 이해 컴퓨터 비전에 관한 유럽 회의, 페이지 818–833. 스프링거, 2014.
- [38] 제이슨 요신스키, 제프 클론, 요슈아 벤지오, 호드 립슨. 기능을 양도할 수 있는 방법 심층 신경망에서? arXiv 사전 인쇄 arXiv:1411.1792, 2014.
- [39] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. 마호니, 커트 Keutzer. Q-BERT: BERT의 헤시안 기반 초저정밀 양자화. 제34차 AAAI 인공 지능 컨퍼런스, AAAI 2020, 인공 지능 컨퍼런스의 제32차 혁신 응용 프로그램, IAAI 2020, 인공 지능 교육 발전에 관한 제10차 AAAI 심포지엄, EAAI 2020, 뉴욕, 뉴욕, 미국, 2월 7-12, 2020, 페이지 8815–8821. AAAI 보도, 2020.
- [40] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, Kurt Keutzer. HAWQ: 혼합 정밀도를 사용한 신경망의 헤시안 인식 양자화. In 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, 서울, 대한민국(남), 2019년 10월 27일 - 11월 2일, 293–302페이지. IEEE, 2019.
- [41] Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W. Mahoney, Kurt Keutzer. HAWQ-V2: 신경망의 헤시안 인식 트레이스 가중 양자화. Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan 및 Hsuan Tien Lin 편집자, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, 12월 6-12 일, 2020, 가상, 2020.
- [42] Rewon Child, Scott Gray, Alec Radford, Ilya Sutskever. 스파스 변환기로 긴 시퀀스 생성. CoRR, abs/1904.10509, 2019.
- [43] François Lagunas, Ella Charlaix, Victor Sanh, Alexander M. Rush. 더 빠른 변환기를 위한 블록 가지치기. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, Scott Wen-tau Yih, 편집자, 2021년 자연 언어 처리의 경험적 방법 회의 절차, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 2021년 11월, 10619–10629페이지. 전산 언어학 협회, 2021.
- [44] Geoffrey E. Hinton, Oriol Vinyals 및 Jeffrey Dean. 신경망에서 지식 추출. CoRR, abs/1503.02531, 2015.
- [45] Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf. BERT의 증류 버전인 Distilbert: 더 작고, 더 빠르고, 더 저렴하고, 더 가볍습니다. CoRR, abs/1910.01108, 2019.
- [46] Siqi Sun, Yu Cheng, Zhe Gan 및 Jingjing Liu. BERT 모델 압축을 위한 환자 지식 증류. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 편집자, 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP IJCNLP 2019, 홍콩, 중국, 11월 3일 -7, 2019, 페이지 4322–4331. 전산 언어학 협회, 2019.
- [47] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang 및 Ming Zhou. Minilm: 사전 훈련된 변환기의 작업에 구애받지 않는 압축을 위한 깊은 자가 주의 증류. Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan 및 Hsuan Tien Lin 편집자, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, 12월 6-12일, 2020, 가상, 2020.

- [48] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, Denny Zhou. Mobilebert: 리소스가 제한된 장치를 위한 소형 작업 독립적 BERT입니다. Dan Jurafsky, Joyce Chai, Natalie Schluter 및 Joel R. Tetreault, 편집자, 전산 언어학 협회 제58차 연례 회의 절차, ACL 2020, 온라인, 2020년 7월 5-10일, 페이지 2158–2170. 전산 언어학 협회, 2020.
- [49] William Fedus, Barret Zoph, Noam Shazeer. 스위치 변환기: 간단하고 효율적인 희소성으로 1조 개의 매개변수 모델로 확장합니다. CoRR, abs/2101.03961, 2021.
- [50] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, Frank Seide. 개선된 대규모 어휘 음성 인식을 위한 KL-divergence 정규화 심층 신경망 적용. 음향, 음성 및 신호 처리에 관한 IEEE 국제 회의, ICASSP 2013, 밴쿠버, BC, 캐나다, 2013년 5월 26-31일, 7893-7897페이지. IEEE, 2013.
- [51] Reza Yazdani Aminabadi, Samyam Rajbhandari, Minjia Zhang, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Jeff Rasley, Shaden Smith, Olatunji Ruwase 및 Yuxiong He. Deepspeed 추론: 전례 없는 규모로 트랜스포머 모델의 효율적인 추론을 가능하게 합니다. <https://arxiv.org/abs/2207.00032>, 2022.
- [52] Zixian Cai, Zhengyang Liu, Saeed Maleki, Madan Musuvathi, Todd Mytkowicz, Jacob Nelson, Olli Saarikivi. SCCL: 최적 집합 알고리즘 합성. CoRR, abs/2008.08708, 2020.
- [53] DeepSpeed 팀, Rangan Majumder 및 Andrey Proskurin. DeepSpeed: 시스템 최적화 및 압축을 통해 대규모 모델 추론 및 교육을 가속화합니다. <https://www.microsoft.com/en-us/research/blog/deepspeed-accelerating-large-scale-model-inference-and-training-via-system-optimizations-and-compression/>, 2021. [온라인].