**The Era of 1-bit LLMs:**

**All Large Language Models are in 1.58 Bits**

**Shuming Ma[[1]](#footnote-1) Hongyu Wang** **Lingxiao Ma Lei Wang Wenhui Wang Shaohan Huang Li Dong Ruiping Wang Jilong Xue Furu Wei**⋄

<https://aka.ms/GeneralAI>

**Abstract**

최근 연구, 예를 들어 BitNet [23],는 1비트 대형 언어 모델(LLM)의 새로운 시대를 열고 있습니다. 이 연구에서는 BitNet b1.58이라는 1비트 LLM 변형을 소개합니다. 이 모델은 LLM의 모든 매개변수(또는 가중치)가 {-1, 0, 1}의 3진수입니다. 이 모델은 동일한 모델 크기와 학습 토큰을 사용하여 풀 프레시전(FP16 또는 BF16) 트랜스포머 LLM과 동등한 당혹도(perplexity)와 최종 작업 성능을 가지면서도 지연시간, 메모리, 처리량 및 에너지 소비 측면에서 매우 비용 효율적입니다. 더 나아가, 1.58비트 LLM은 고성능이면서 비용 효율적인 새로운 세대의 LLM을 위한 새로운 스케일링 법칙과 학습 레시피를 정의합니다. 또한, 새로운 계산 패러다임을 가능하게 하여 1비트 LLM에 최적화된 특정 하드웨어 설계를 위한 문을 엽니다.

텍스트, 스크린샷, 도표, 번호이(가) 표시된 사진

자동 생성된 설명

그림 1: 1비트 LLM (예: BitNet b1.58)은 모델 성능을 유지하면서 LLM의 추론 비용(지연시간, 처리량, 에너지)을 줄이는 파레토 솔루션을 제공합니다. BitNet b1.58의 새로운 계산 패러다임은 1비트 LLM에 최적화된 새로운 하드웨어 설계를 필요로 합니다.

**1 The Era of 1-bit LLMs**

최근 몇 년 동안, AI 분야는 대형 언어 모델(LLM)의 크기와 능력이 급속히 성장하는 것을 보았습니다. 이러한 모델들은 다양한 자연어 처리 작업에서 놀라운 성능을 보여주었지만, 그 크기가 커지면서 배포에 어려움을 겪고 높은 에너지 소비로 인한 환경 및 경제적 영향에 대한 우려가 제기되었습니다. 이러한 문제를 해결하기 위한 한 가지 접근법은 학습 후 양자화를 사용하여 추론을 위한 저비트 모델을 만드는 것입니다 [23, FAHA23, CCKS23, 24]. 이 기술은 가중치와 활성화의 정밀도를 줄여 LLM의 메모리 및 계산 요구 사항을 크게 줄입니다. 트렌드는 16비트에서 4비트 변형으로 낮추는 방향으로 이동하고 있습니다 [FAHA23, 23]. 그러나 학습 후 양자화는 산업 LLM에서 널리 사용되고 있음에도 불구하고 최적의 방법은 아닙니다.

1비트 모델 아키텍처에 대한 최근 연구, 예를 들어 BitNet [23],는 LLM의 비용을 줄이면서 성능을 유지하는 유망한 방향을 제시합니다. 일반적인 LLM은 16비트 부동 소수점 값(FP16 또는 BF16)으로 되어 있으며, 대부분의 LLM은 행렬 곱셈이 주를 이룹니다. 따라서 주요 계산 비용은 부동 소수점 덧셈 및 곱셈 연산에서 발생합니다. 반면에, BitNet의 행렬 곱셈은 정수 덧셈만 포함하므로 LLM의 에너지 비용을 크게 절약할 수 있습니다. 많은 칩의 계산 성능의 근본적인 한계는 전력에 있기 때문에, 에너지 절약은 더 빠른 계산으로도 이어질 수 있습니다.

계산 외에도, 모델 매개변수를 DRAM에서 칩 내 가속기의 메모리(SRAM)로 전송하는 과정은 추론 중에 비용이 많이 들 수 있습니다. 처리량을 높이기 위해 SRAM을 확대하려는 시도가 있었지만, 이는 DRAM보다 훨씬 높은 비용을 초래합니다. 풀 프레시전 모델과 비교할 때, 1비트 LLM은 용량 및 대역폭 측면에서 훨씬 낮은 메모리 풋프린트를 가지고 있습니다. 이는 DRAM에서 가중치를 로드하는 비용과 시간을 크게 줄여 더 빠르고 효율적인 추론을 가능하게 합니다.

이 연구에서는 BitNet b1.58이라는 중요한 1비트 LLM 변형을 소개합니다. 이 모델의 모든 매개변수는 3진수로 {-1, 0, 1} 값을 가집니다. 원래 1비트 BitNet에 값 0을 추가하여 이진 시스템에서 1.58비트가 되었습니다. BitNet b1.58은 원래 1비트 BitNet의 모든 이점을 유지하면서 거의 곱셈 연산이 필요 없는 새로운 계산 패러다임을 포함하고 있어 높은 최적화가 가능합니다. 또한, 에너지 소비는 원래의 1비트 BitNet과 동일하며 메모리 소비, 처리량, 지연 시간 측면에서 FP16 LLM 기준보다 훨씬 효율적입니다. 게다가, BitNet b1.58은 두 가지 추가적인 장점을 제공합니다. 첫째, 모델 가중치에 0을 포함하여 특징 필터링에 대한 명시적 지원이 가능해져 1비트 LLM의 성능을 크게 향상시킬 수 있습니다. 둘째, 실험 결과 BitNet b1.58은 동일한 구성(예: 모델 크기, 학습 토큰 등)을 사용할 때 3B 크기부터 당혹도(perplexity) 및 최종 작업 성능 측면에서 풀 프레시전(FP16) 기준과 동등한 성능을 발휘할 수 있음을 보여줍니다.

**2 BitNet b1.58**

BitNet b1.58은 nn.Linear를 BitLinear로 대체한 Transformer인 BitNet 아키텍처를 기반으로 합니다. 이 모델은 처음부터 1.58비트 가중치와 8비트 활성화로 학습되었습니다. 원래의 BitNet과 비교하여 몇 가지 수정 사항이 도입되었으며, 아래에 요약합니다.

**양자화 함수(Quantization Function).** 가중치를 -1, 0, 또는 +1로 제한하기 위해 absmean 양자화 함수를 채택했습니다. 이 함수는 먼저 가중치 행렬을 평균 절대값으로 스케일링한 다음, 각 값을 {-1, 0, +1} 중 가장 가까운 정수로 반올림합니다:

텍스트, 폰트, 화이트, 라인이(가) 표시된 사진

자동 생성된 설명

활성화에 대한 양자화 함수는 BitNet에서와 동일한 구현을 따르지만, 비선형 함수 전에 활성화를 [0, ] 범위로 스케일링하지 않습니다. 대신에,

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

표 1: BitNet b1.58과 LLaMA LLM의 당혹도(perplexity) 및 비용.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

표 2: 최종 작업에서 BitNet b1.58과 LLaMA LLM의 제로샷 정확도.

활성화는 제로 포인트 양자화를 없애기 위해 토큰당 [−, ] 범위로 모두 스케일링됩니다. 이는 구현과 시스템 수준 최적화 모두에 대해 더 편리하고 간단하며, 실험에서는 성능에 거의 영향을 미치지 않습니다.

**LLaMA 유사 구성 요소.** LLaMA [23, 23]의 아키텍처는 오픈 소스 LLM의 사실상 기본 백본이 되어 왔습니다. 오픈 소스 커뮤니티를 수용하기 위해, BitNet b1.58의 설계는 LLaMA 유사 구성 요소를 채택했습니다. 구체적으로, RMSNorm [ZS19], SwiGLU [Sha20], 회전 임베딩 [24]을 사용하고 모든 바이어스를 제거했습니다. 이렇게 함으로써 BitNet b1.58은 최소한의 노력으로 Huggingface, vLLM [23], llama.cpp

[[2]](#footnote-2) 등의 인기 있는 오픈 소스 소프트웨어에 통합될 수 있습니다.

**3 Results**

우리는 다양한 크기의 BitNet b1.58과 우리가 재현한 FP16 LLaMA LLM을 비교했습니다. 공정한 비교를 위해 RedPajama 데이터셋 [Com23]에서 모델을 1000억 개의 토큰으로 사전 학습시켰습니다. ARC-Easy [YBS19], ARC-Challenge [YBS19], Hellaswag [19], Winogrande [SBBC20], PIQA [19], OpenbookQA [MCKS18], BoolQ [19]를 포함한 다양한 언어 작업에서 제로샷 성능을 평가했습니다. 또한, WikiText2 [MXBS16]와 C4 [19] 데이터셋에서 검증 당혹도(perplexity)를 보고했습니다.

LLaMA LLM과 BitNet b1.58의 실행 시간 GPU 메모리와 지연 시간을 비교했습니다. 결과는 GPU 장치에서 LLM 추론 지연 시간에 최적화된 FasterTransformer[[3]](#footnote-3) 코드베이스를 사용하여 측정되었습니다. Ladder [23]의 2비트 커널도 BitNet b1.58에 통합되었습니다. 우리는 추론의 주요 비용이 되는 출력 토큰당 시간을 보고했습니다.

표 1은 BitNet b1.58과 LLaMA LLM의 당혹도와 비용을 요약한 것입니다. BitNet b1.58은 3B 모델 크기에서 당혹도 측면에서 풀 프레시전 LLaMA LLM과 동등한 수준에 도달하기 시작하며, 2.71배 더 빠르고 3.55배 적은 GPU 메모리를 사용합니다. 특히, 3.9B 모델 크기의 BitNet b1.58은 2.4배 더 빠르고 3.32배 적은 메모리를 소비하면서도 LLaMA LLM 3B보다 성능이 훨씬 뛰어납니다.

텍스트, 라인, 도표, 그래프이(가) 표시된 사진

자동 생성된 설명

그림 2: 모델 크기에 따른 BitNet b1.58의 디코딩 지연 시간(왼쪽)과 메모리 소비(오른쪽).

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

표 3: BitNet b1.58 70B와 LLaMA LLM 70B의 처리량 비교.

표 2는 최종 작업에서 제로샷 정확도의 자세한 결과를 보고합니다. 우리는 lm-evaluation-harness[[4]](#footnote-4)의 파이프라인을 따라 평가를 수행했습니다. 결과는 BitNet b1.58과 LLaMA LLM 간의 성능 격차가 모델 크기가 증가함에 따라 좁아진다는 것을 보여줍니다. 더 중요한 것은, BitNet b1.58이 3B 크기부터 풀 프레시전 기준의 성능을 맞출 수 있다는 것입니다. 당혹도 관찰과 유사하게, 최종 작업 결과는 BitNet b1.58 3.9B가 더 낮은 메모리와 지연 시간 비용으로 LLaMA LLM 3B를 능가한다는 것을 보여줍니다. 이는 BitNet b1.58이 최첨단 LLM 모델들에 비해 파레토 개선을 제공함을 시사합니다.

**메모리와 지연 시간** 우리는 모델 크기를 7B, 13B, 70B로 더욱 확장하고 비용을 평가했습니다. 그림 2는 지연 시간과 메모리의 경향을 보여주며, 모델 크기가 커짐에 따라 속도가 증가함을 나타냅니다. 특히, BitNet b1.58 70B는 LLaMA LLM 기준보다 4.1배 빠릅니다. 이는 nn.Linear의 시간 비용이 모델 크기에 따라 증가하기 때문입니다. 메모리 소비도 유사한 경향을 따르며, 임베딩은 풀 프레시전 상태로 유지되고 큰 모델일수록 메모리 비율이 작아집니다. 지연 시간과 메모리는 모두 2비트 커널로 측정되었기 때문에 비용을 더욱 줄이기 위한 최적화 여지가 여전히 남아 있습니다.

**에너지** 우리는 BitNet b1.58과 LLaMA LLM의 산술 연산 에너지 소비를 추정했습니다. 주로 행렬 곱셈 계산에 집중했는데, 이는 LLM의 비용에 가장 큰 기여를 하기 때문입니다. 그림 3은 에너지 비용의 구성을 보여줍니다. BitNet b1.58의 대부분은 INT8 덧셈 계산으로 구성되어 있는 반면, LLaMA LLM은 FP16 덧셈과 FP16 곱셈으로 구성되어 있습니다. [Hor14, ZZL22]의 에너지 모델에 따르면, BitNet b1.58은 7nm 칩에서 행렬 곱셈의 산술 연산 에너지 소비를 71.4배 절감합니다. 우리는 또한 512 토큰 모델의 종단 간 에너지 비용을 보고했습니다. 결과는 모델 크기가 커짐에 따라 BitNet b1.58이 FP16 LLaMA LLM 기준에 비해 에너지 소비 측면에서 점점 더 효율적임을 보여줍니다. 이는 nn.Linear의 비율이 모델 크기에 따라 증가하는 반면, 다른 구성 요소에서 발생하는 비용은 큰 모델에서 더 작기 때문입니다.

**처리량** 우리는 두 개의 80GB A100 카드에서 파이프라인 병렬 처리 [19]를 사용하여 BitNet b1.58과 LLaMA LLM의 70B 매개변수를 비교했습니다. 이를 통해 LLaMA LLM 70B가 장치에서 실행될 수 있었습니다. 배치 크기를 GPU 메모리 한계에 도달할 때까지 증가시켰으며, 시퀀스 길이는 512였습니다. 표 3은 BitNet b1.58 70B가 LLaMA LLM보다 최대 11배 더 많은 배치 크기를 지원할 수 있어, 처리량이 8.9배 더 높음을 보여줍니다.

텍스트, 도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

그림 3: 7nm 공정 노드에서 BitNet b1.58과 LLaMA LLM의 에너지 소비 비교. 왼쪽은 산술 연산 에너지의 구성 요소이고, 오른쪽은 다양한 모델 크기에서의 종단 간 에너지 비용입니다.

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

표 4: 2T 토큰으로 학습된 BitNet b1.58과 StableLM-3B의 비교.

BitNet b1.58은 모델 성능과 추론 비용 측면에서 새로운 스케일링 법칙을 가능하게 합니다. 참고로, 그림 2와 3의 결과를 기반으로 1.58비트와 16비트의 다양한 모델 크기 간 다음과 같은 동등성을 가질 수 있습니다.

* 13B BitNet b1.58은 지연 시간, 메모리 사용량, 에너지 소비 측면에서 3B FP16 LLM보다 더 효율적입니다.
* 30B BitNet b1.58은 지연 시간, 메모리 사용량, 에너지 소비 측면에서 7B FP16 LLM보다 더 효율적입니다.
* 70B BitNet b1.58은 지연 시간, 메모리 사용량, 에너지 소비 측면에서 13B FP16 LLM보다 더 효율적입니다.

**2T 토큰으로 학습** 학습 토큰 수는 LLM에 있어 중요한 요소입니다. BitNet b1.58의 토큰 측면에서의 확장성을 테스트하기 위해, 최첨단 오픈 소스 3B 모델인 StableLM-3B [TBMR]의 데이터 레시피를 따라 2T 토큰으로 BitNet b1.58 모델을 학습시켰습니다. 두 모델 모두 Winogrande [SBBC20], PIQA [19], SciQ [WLG17], LAMBADA [16], ARC-easy [YBS19]로 구성된 벤치마크에서 평가되었습니다. 표 4에 제로샷 정확도를 보고했습니다. 정확도와 정규화된 정확도로 측정된 작업의 경우 두 값을 평균으로 산출했습니다. 2T 토큰에서의 StableLM 3b 결과는 기술 보고서에서 직접 가져왔습니다. 우리의 연구 결과, BitNet b1.58이 모든 최종 작업에서 우수한 성능을 발휘하여 1.58비트 LLM도 강력한 일반화 능력을 가지고 있음을 나타냅니다.

**4 Discussion and Future Work**

**1-bit Mixture-of-Experts (MoE) LLMs**

전문가 혼합(Mixture-of-Experts, MoE)은 LLM에 있어 비용 효율적인 접근법으로 입증되었습니다. 이는 계산 FLOP를 크게 줄이지만, 높은 메모리 소비와 칩 간 통신 오버헤드로 인해 배포와 적용에 한계가 있습니다. 이러한 문제는 1.58비트 LLM으로 해결할 수 있습니다. 첫째, 메모리 풋프린트가 줄어들어 MoE 모델을 배포하는 데 필요한 장치 수가 감소합니다. 또한, 네트워크 간 활성화 전송의 오버헤드를 크게 줄입니다. 궁극적으로, 전체 모델을 단일 칩에 배치할 수 있다면 오버헤드가 전혀 발생하지 않을 것입니다.

**Native Support of Long Sequence in LLMs**

LLM의 시대에는 긴 시퀀스를 처리하는 능력이 중요한 요구사항이 되었습니다. 긴 시퀀스 추론에서의 주요 과제 중 하나는 KV 캐시로 인해 발생하는 메모리 소비입니다. BitNet b1.58은 활성화를 16비트에서 8비트로 줄여 동일한 자원으로 컨텍스트 길이를 두 배로 늘릴 수 있기 때문에 긴 시퀀스에 대한 네이티브 지원을 향한 중요한 단계입니다. 이는 1.58비트 LLM의 경우 4비트 또는 그 이하로 무손실 압축할 수 있으며, 이는 향후 연구 과제로 남겨둡니다.

**LLMs on Edge and Mobile**

1.58비트 LLM의 사용은 엣지 및 모바일 장치에서 언어 모델의 성능을 크게 향상시킬 잠재력을 가지고 있습니다. 이러한 장치들은 종종 메모리와 계산 능력이 제한되어 있어 LLM의 성능과 규모를 제한할 수 있습니다. 그러나 1.58비트 LLM의 메모리와 에너지 소비 감소는 이러한 장치에 배포할 수 있게 하여 이전에는 불가능했던 다양한 응용 프로그램을 가능하게 합니다. 이는 엣지 및 모바일 장치의 능력을 크게 향상시키고 LLM의 새로운 흥미로운 응용 프로그램을 가능하게 합니다. 게다가, 1.58비트 LLM은 엣지 및 모바일 장치에서 주로 사용되는 CPU 장치에 더 친화적입니다. 이는 BitNet b1.58이 이러한 장치에서 효율적으로 실행될 수 있음을 의미하며, 성능과 기능을 더욱 향상시킵니다.

**New Hardware for 1-bit LLMs**

Groq[[5]](#footnote-5)와 같은 최근 연구는 LLM을 위한 특정 하드웨어(예: LPU)를 구축하는 데 있어 유망한 결과와 큰 잠재력을 보여주었습니다. 한 단계 더 나아가, BitNet [23]에서 가능해진 새로운 계산 패러다임을 고려하여 1비트 LLM에 최적화된 새로운 하드웨어와 시스템 설계를 촉구합니다.

**References**

[19] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. CoRR, abs/1911.11641, 2019.

[CCKS23] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees. CoRR, abs/2307.13304, 2023.

[19] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. CoRR, abs/1905.10044, 2019.

[Com23] Together Computer. Redpajama: an open dataset for training large language models, 2023.

[FAHA23] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: accurate quantization for generative pre-trained transformers. In The Eleventh International Conference on Learning Representations, 2023.

[19] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Xu Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, Yonghui Wu, and Zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In Advances in Neural Information Processing Systems, pages 103–112, 2019.

[Hor14] Mark Horowitz. 1.1 computing’s energy problem (and what we can do about it). In 2014 IEEE International Conference on Solid-State Circuits Conference, ISSCC 2014, Digest of Technical Papers, San Francisco, CA, USA, February 9-13, 2014, pages 10–14, 2014.

[23] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles, 2023.

[23] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. AWQ: activation-aware weight quantization for LLM compression and acceleration. CoRR, abs/2306.00978, 2023.

[MCKS18] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. CoRR, abs/1809.02789, 2018.

[MXBS16] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

[16] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers. The Association for Computer Linguistics, 2016.

[19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. CoRR, abs/1910.10683, 2019.

[24] Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. Neurocomputing, 568:127063, 2024.

[SBBC20] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: an adversarial winograd schema challenge at scale. In The Thirty-Fourth AAAI Conference on Artificial Intelligence, pages 8732–8740, 2020.

[Sha20] Noam Shazeer. GLU variants improve transformer. CoRR, abs/2002.05202, 2020.

[TBMR] Jonathan Tow, Marco Bellagente, Dakota Mahan, and Carlos Riquelme. Stablelm 3b 4e1t.

[24] Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better LLM quantization with hadamard incoherence and lattice codebooks. CoRR, abs/2402.04396, 2024.

[23] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: open and efficient foundation language models. CoRR, abs/2302.13971, 2023.

[23] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, and et al. Llama 2: open foundation and fine-tuned chat models. CoRR, abs/2307.09288, 2023.

[WLG17] Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin, editors, Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017, pages 94–106. Association for Computational Linguistics, 2017.

[23] Lei Wang, Lingxiao Ma, Shijie Cao, Ningxin Zheng, Quanlu Zhang, Jilong Xue, Ziming Miao, Ting Cao, , and Yuqing Yang. Ladder: Efficient tensor compilation on customized data format. In OSDI, 2023.

[23] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language models. CoRR, abs/2310.11453, 2023.

[23] Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: accurate and efficient post-training quantization for large language models. In International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, 2023.

[YBS19] Vikas Yadav, Steven Bethard, and Mihai Surdeanu. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, EMNLP-IJCNLP, 2019.

[19] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: can a machine really finish your sentence? In Proceedings of the 57th Conference of the Association for Computational Linguistics, pages 4791–4800, 2019.

[ZS19] Biao Zhang and Rico Sennrich. Root mean square layer normalization. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, Advances in Neural Information Processing Systems, pages 12360–12371, 2019.

[ZZL22] Yichi Zhang, Zhiru Zhang, and Lukasz Lew. PokeBNN: A binary pursuit of lightweight accuracy. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12465–12475. IEEE, 2022.

1. 동등 기여. ⋄ 교신 저자. S. Ma, L. Ma, L. Wang, W. Wang, S. Huang, L. Dong, J. Xue, F. Wei는 Microsoft Research 소속입니다. H. Wang과 R. Wang은 중국과학원 대학(University of Chinese Academy of Sciences) 소속입니다. [↑](#footnote-ref-1)
2. <https://github.com/ggerganov/llama.cpp> [↑](#footnote-ref-2)
3. <https://github.com/NVIDIA/FasterTransformer> [↑](#footnote-ref-3)
4. <https://github.com/EleutherAI/lm-evaluation-harness> [↑](#footnote-ref-4)
5. <https://groq.com/> [↑](#footnote-ref-5)