**上海大学 2023 ～ 2024 学年 春 季学期**

**学科前沿与实践**  **研究生课程成绩评价表**

课 程 名称： 学科前沿与实践 课程编号： 4XS1080001

报 告 题目： 基于计算加速技术的大语言模型微调实践

学 生 姓名： 郎志远 学 号： 23724590

课程报告评语：

课程报告成绩：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 文献总结（30分） | | 算法实验（40分） | | 报告综合评价（30分） | | | 总分  （100分） |
| 相关性 | 深度 | 工作量 | 效果 | 规范性 | 质量 | 创新 |
| 10分 | 20分 | 30分 | 10分 | 10分 | 10分 | 10分 | 100 |
|  |  |  |  |  |  |  |  |

任课教师:

评阅日期: 年 月 日

基于计算加速技术的大语言模型微调实践

郎志远23724590

**摘要：**本文专注于大型语言模型在训练与推理阶段所遭遇的计算资源与内存需求问题，探索了计算加速技术在此领域的应用。在部署了两张A100的服务器上使用Yi-6B-Chat模型上进行微调实践，实验结果表明，KV Cache 和 Flash Attention 技术能够有效降低模型的存储和计算需求，提高训练和推理效率。特别是在使用 Flash Attention时，训练时间平均减少了约26%，显示出显著的性能提升。

**关键字:** 大语言模型 计算加速 模型微调

**Fine-tuning practice of large language models based on computational acceleration technology**

Zhiyuan Lang 23724590

**Abstract:** This paper focuses on the computing resources and memory requirements encountered by large language models in the training and inference stages, and explores the application of computing acceleration technology in this field. The experimental results show that KV Cache and Flash Attention technologies can effectively reduce the storage and computing requirements of the model and improve the training and inference efficiency. Especially when using Flash Attention, the training time was reduced by about 26% on average, showing a significant performance improvement.

**Key words:** Large Language Models ; Computational Acceleration; Model fine-tuning

**1 引言**

在当今人工智能的快速发展浪潮中，大语言模型已经成为推动自然语言处理领域革新的核心动力。这些模型，以其庞大的参数规模和深度学习能力，能够在多种语言任务上展现出令人瞩目的表现，从而在机器翻译、文本生成、情感分析等多个领域实现了前所未有的突破[1] [2][3]。然而，随着模型规模的不断扩大，它们对计算资源的需求也呈现出指数级增长，这不仅对硬件设施提出了更高的要求，也对研究人员提出了新的挑战。特别是在训练和部署这些模型时，所需的计算力、内存容量以及数据传输速度成为了制约其进一步发展的关键因素[4]。

为了应对这些挑战，计算加速技术应运而生，它们通过一系列创新的方法来优化模型的训练和推理过程。本文旨在深入探讨计算加速技术在大语言模型中的应用，通过在在本地部署大语言模型并使用KV Cache、Flash Attention等计算加速方法进行测试，评估它们在实际应用中的性能表现，并探讨它们在提升大语言模型训练效率和推理速度方面的潜力。同时，也将讨论在实际应用中遇到的挑战，并对未来的研究方向提出展望。

**2 文献总结**

在大模型计算加速领域，许多研究人员和团队通过创新的方法和理论贡献，推动了大语言模型（大语言模型）训练和推理效率的提升。Yann LeCun等人提出了LoRA（Low-Rank Adaptation）[5]，这是一种通过在Transformer模型中引入低秩矩阵来实现参数高效更新的方法，LoRA显著减少了模型更新时所需的参数数量，同时保持了模型性能。R Child等人提出了了Sparse Transformer[6]，它是一种通过稀疏化注意力机制来减少计算需求的方法。Sparse Transformer在保持模型性能的同时，显著降低了计算和内存消耗。Samyam Rajbhandari等人共同开发了ZeRO（Zero Redundancy Optimizer）[7]，这是一种内存优化技术，通过消除数据并行训练中的冗余内存消耗，显著提高了训练大型模型的效率。ZeRO使得在有限的硬件资源下训练万亿参数模型成为可能。张等人提出了FlashAttention[8]，这是一种通过优化内存访问模式来加速Transformer模型推理的技术，FlashAttention显著提高了模型的推理速度，降低了延迟。Nathan Srebro等人提出了一种名为ALiBi（Adaptive Linear Blockwise Inference）[9]的方法，该技术通过动态调整模型的计算粒度，实现了在保持模型性能的同时，显著提高了推理效率。这些研究人员和团队的工作不仅推动了计算加速技术的发展，也为大语言模型在实际应用中的部署和使用提供了可能。他们的研究成果在学术界和工业界都产生了深远的影响，为未来的AI研究和应用奠定了坚实的基础。

**3 算法实现描述**

**3.1 KVCache**

KV Cache[10]是一种在大型语言模型推理过程中常用的优化技术，特别是在基于Transformer架构的模型中。它的核心思想是利用自注意力机制中的重复计算来提高推理效率。在自注意力机制中，模型在处理序列数据时，会多次计算相同的键（Key）和值（Value）对，这些计算在标准的自注意力过程中是重复进行的。

如图1所示，针对计算重复，KV Cache使用了缓存机制：在模型的自注意力层中，KV Cache会将已经计算过的键值对（K,V）存储在一个缓存中。这些键值对代表了模型在处理输入序列时的上下文信息。当模型在后续的推理步骤中再次需要相同的键值对时，KV Cache可以直接从缓存中检索这些值，而无需重新计算。这样可以显著减少计算量，因为自注意力机制中的计算是模型推理过程中最耗时的部分之一。

KV Cache通过牺牲一定的内存空间来换取推理时间的减少。在资源有限的环境中，这种权衡可能是必要的，因为它可以显著提高模型的响应速度，尤其是在需要快速处理大量请求的场景中。KV Cache适用于那些具有自注意力机制的模型，如GPT、BERT[11] [12]等。这些模型在处理长序列时，KV Cache可以显著提高效率。尽管KV Cache在提高推理速度方面表现出色，但它也有一些局限性。例如，如果输入序列的长度变化很大，或者模型需要处理动态长度的输入，KV Cache可能需要更复杂的管理策略来确保缓存的有效性。此外，缓存的维护和更新也需要额外的计算资源[13]。

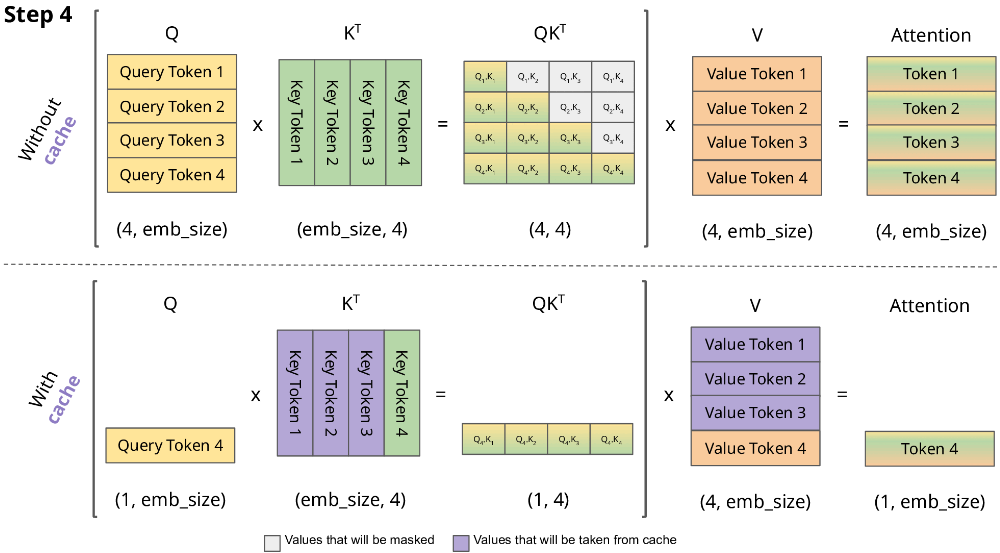


图 1 KVCache

**3.2 FlashAttention**

在大语言模型的自注意力计算中，Q、K、V矩阵的矩阵乘法操作是核心步骤，但由于其规模巨大，直接进行全矩阵乘法会导致缓存不友好的问题[14]。为了解决这一挑战，研究者们提出了多种优化策略，其中FlashAttention是一种特别有效的方法。FlashAttention通过将传统的自注意力计算过程分解为更小的计算块，以适应GPU的缓存结构。这种方法的核心在于，它能够在不牺牲性能的情况下，减少对高带宽内存（HBM）的读写次数，从而提高整体的计算效率。FlashAttention 的主要思想如下：

IO感知：如图2所示现代GPU的计算速度远超过内存访问速度，IO成为了性能瓶颈。FlashAttention关注GPU高带宽内存（HBM）和GPU片上SRAM之间的数据IO操作。它通过减少在HBM上的操作来提高效率。

分块：FlashAttention通过将输入矩阵（Q, K, V）分割成小块，然后逐块进行处理，这样可以避免在GPU的慢速HBM中存储和读取整个注意力矩阵，从而减少IO操作。

重计算：在反向传播过程中，FlashAttention 不存储中间的注意力矩阵，而是通过存储softmax归一化因子来重计算注意力矩阵，这在内存效率和计算速度上都有所提升。

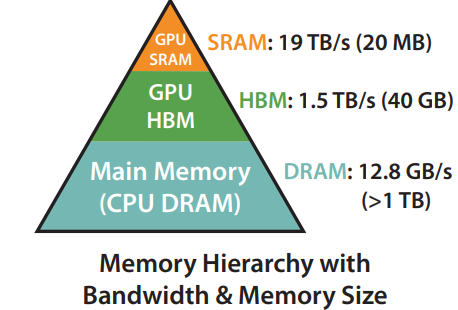


图 2 GPU的分级缓存

FlashAttention分为前向传播和反向传播两部分。如图3所示，在前向传播过程中，输入矩阵Q,K,V被分割成小块并加载到GPU的快速片上SRAM中。对于每个小块，计算Q和 K的点积，然后应用softmax函数。使用softmax的归一化因子来调整输出，得到最终的注意力输出。在反向传播反向传播过程中使用前向传播中保存的softmax归一化因子和dropout[15]掩码来重计算注意力矩阵。根据重计算的注意力矩阵，计算输入矩阵的梯度。

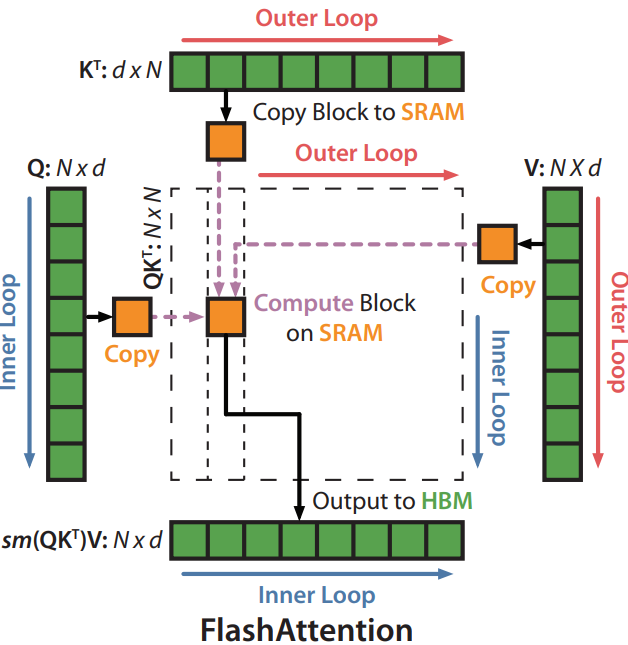


图 3 FlashAttention

**4 实验描述**

**4.1实验设计**

**4.1.1 实验目标**

本实验旨在评估KV Cache、Flash Attention等计算加速技术在提升大语言模型）训练效率和推理速度方面的实际效果。通过比较使用相关技术前后的训练时间判断来评测这些技术在减少内存占用、降低训练时间以及提高模型并行性方面的贡献。

**4.1.2 实验环境**

硬件配置：实验在配备两块NVIDIA A100 GPU的高性能计算集群上进行。每个GPU拥有40GB显存。

软件环境：操作系统为Red Hat Enterprise Linux3.10.0-693使用PyTorch 2.0.1作为深度学习框架，DeepSpeed 0.12.6[16]作为模型训练的分布式训练框架。

**4.1.3 实验方法**

KV Cache：考虑到大模型的训练资源以及服务器节点的不稳定性，在Transformer模型的解码阶段，使用KV Cache来存储和重用已经计算过的K和V矩阵以加快训练速度，尽量避免训练过程中由服务器不稳定性引起的重新训练。KV Cache的设置关键如图4所示



图 4 KVCache设置

FlashAttention：DeepSpeed框架提供了FlashAttention支持，如图5所示只需要在使用时进行设置即可选择是否使用FlashAttention。

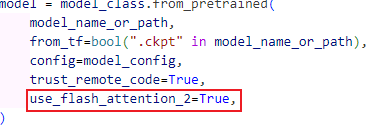


图 5 FlashAttention设置

**4.2实验过程**

**4.2.1 数据及模型准备**

数据集：本文在微调过程中使用的数据集来自Chinese Open Instruction Generalist（COIG）相比其他的中文指令数据集，COIG数据集在领域适应性、多样性、数据质量等方面具有一定的优势。目前COIG数据集主要包括：通用翻译指令数据集、考试指令数据集、价值对其数据集、反事实校正数据集、代码指令数据集。

模型选择：为了保证所有实验环节能够正常执行，本文选择了Yi-6B-Chat模型，它具有接近60亿个参数。

**4.2.2**实施步骤

首先，初始化模型参数，使用KV Cache，分别设置模型使用FlashAttention和不使用FlashAttention然后通过数据并行在多个GPU上进行微调。在微调过程中，记录了训练时间、内存使用情况以及模型的损失值。微调完成后，通过分别向初始模型与微调后的模型进行提问，比较微调后是否起到作用。

**4.3结果分析**

通过对比实验，发现在都使用KV Cache的情况下如果不使用FlashAttention一次微调所需要的时间平均在38分钟，如果使用FlashAttention，一次微调所需要的时间平均在28分钟，训练速度提升了约26%。

表1 对Yi-6B-Chat微调所用时间(单位：分钟)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 实验1 | 实验2 | 实验3 | 实验4 |
| 使用FlashAttention | 28 | 30 | 28 | 27 |
| 未使用FlashAttention | 38 | 38 | 40 | 37 |

在实验中每一个epoch中计算时每块A100所需要的显存在26GB左右，这已经超出了消费级显卡的最大显存容量，可以看出针对大语言模型的微调仍需要继续优化。

微调前后，针对模型提问效果如图6、7所示，可以看出基础模型虽能回答出关于节日信息的部分内容，但相比微调后的模型，他缺少的具体的节日时间等信息，并且输出格式，由此可见微调后的模型相比基础模型有所改善。

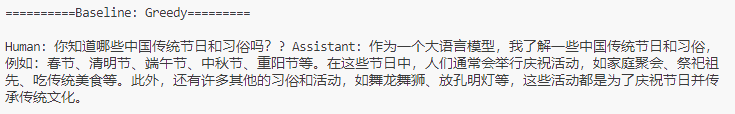


图 6 基础模型回答结果

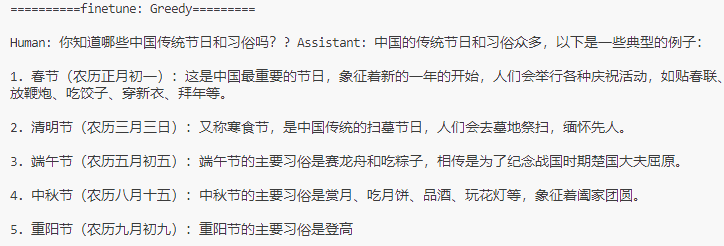


图 7 微调后模型输出结果

本实验表明，KV Cache、Flash Attention等计算加速技术能够有效提升大语言模型的训练效率。这些技术不仅提高了模型的运行速度，还降低了对硬件资源的需求。尽管在实施过程中遇到了一些挑战，但通过不断的和调整，我们成功地解决了这些问题。未来的工作将集中在进一步优化这些技术，以及探索新的加速策略，以实现更高效的AI模型训练和部署。

**5 结论**

经过实验研究和数据分析，可以看到KV Cache和FlashAttention，能够在不牺牲模型性能的前提下，显著降低模型的存储和计算需求。这一发现对于资源受限的环境尤为重要，使得该模型能够更广泛地部署在各种硬件平台上。其次，推理优化技术，展现了在提高模型推理效率方面的潜力。这些技术通过减少重复计算和动态调整计算资源，使得模型在处理复杂任务时更加灵活和高效。特别是在GPU等专用硬件上的应用，在推理速度的提升非常显著，这对于需要快速响应的应用场景具有重要意义。这一策略对于处理大规模数据集和高并发请求的NLP应用尤为关键，为构建可扩展的智能系统提供了有效途径。

**6 课程总结**

通过这门课程的学习，我对计算机科学这个日新月异领域的最新进展有了更加深入的了解。这不仅仅是一段知识的累积之旅，更是一次自我挑战与成长的历程。通过实战经验的积累，我在面对未来学术深海的未知挑战时，能够更加自信地潜行探索，也在我职业生涯的征途上，铺设了一条稳固而宽广的道路。总而言之，这门课程不仅提升了我的专业技能，也为我未来的学术探索和职业发展奠定了坚实的基础。我期待将这些新获得的知识和技能应用到未来的工作和研究中，以实现个人和学术上的更大成就。

参考文献：

[1] Vaswani A , Shazeer N , Parmar N ,et al.Attention Is All You Need[J].arXiv, 2017.DOI:10.48550/arXiv.1706.03762.

[2] Dong L , Yang N , Wang W ,et al.Unified Language Model Pre-training for Natural Language Understanding and Generation[J]. 2019.DOI:10.48550/arXiv.1905.03197.

[3] Devlin J , Chang M W , Lee K ,et al.BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.

[4] 柯沛,雷文强,黄民烈.以ChatGPT为代表的大型语言模型研究进展[J].中国科学基金, 2023, 37(5):714-723.

[5] Hu E J , Shen Y , Wallis P ,et al.LoRA: Low-Rank Adaptation of Large Language Models[J]. 2021.DOI:10.48550/arXiv.2106.09685.

[6] Child R , Gray S , Radford A ,et al.Generating Long Sequences with Sparse Transformers[J]. 2019.DOI:10.48550/arXiv.1904.10509.

[7] Rajbhandari S , Rasley J , Ruwase O ,et al.ZeRO: Memory Optimization Towards Training A Trillion Parameter Models[J]. 2019.DOI:10.48550/arXiv.1910.02054.

[8] Yu Z , Pengyuan Z , Yonghong Y .Long short-term memory with attention and multitask learning for distant speech recognition[J].Qinghua Daxue Xuebao/Journal of Tsinghua University, 2018, 58(3):249-253.DOI:10.16511/j.cnki.qhdxxb.2018.25.016.

[9] Press O , Smith N A , Lewis M .Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation[J]. 2021.DOI:10.48550/arXiv.2108.12409.

[10] Waddington D , Colmenares J , Kuang J ,et al.KV-Cache: A Scalable High-Performance Web-Object Cache for Manycore[C]//Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing.ACM, 2013.DOI:10.1109/UCC.2013.34.

[11] Brown T B , Mann B , Ryder N ,et al.Language Models are Few-Shot Learners[J]. 2020.DOI:10.48550/arXiv.2005.14165.

[12] Devlin J , Chang M W , Lee K ,et al.BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.DOI:10.48550/arXiv.1810.04805.

[13] Choi H , Bajic I V .Near-Lossless Deep Feature Compression for Collaborative Intelligence[J].IEEE, 2018.DOI:10.1109/MMSP.2018.8547134.

[14] Cobbe K , Kosaraju V , Bavarian M ,et al.Training Verifiers to Solve Math Word Problems[J]. 2021.DOI:10.48550/arXiv.2110.14168.

[15] Srivastava N , Hinton G , Krizhevsky A ,et al.Dropout: A Simple Way to Prevent Neural Networks from Overfitting[J].Journal of Machine Learning Research, 2014, 15(1):1929-1958.DOI:10.5555/2627435.2670313.

[16] 李天健,林达华.一种针对超大模型的分布式推理部署系统.CN202211373207.4[2024-01-25].