작업별 중요도 기반 프루닝을 활용한 LoRA 어댑터 병합 기법

윤도경 [1], 서지원 [2,3], 조동현 [1*]
[1] 한양대학교, [2] 서울대학교, [3] 서울대학교 자동화연구소

sb0636@hanyang.ac.kr, seojiwon@snu.ac.kr, *doncho@hanyang.ac.kr

TIP: Task Importance-based Pruning for LoRA Adapter Merging

Dogyeong Yun[1], Jiwon Seo[2,3], Donghyeon Cho[1*]
[1]Hanyang University, [2]Seoul National University, [3]Seoul National University ASRI

요 약

본 논문은 다중 작업(multi-task) 환경에서 LoRA(Low-Rank Adaptation) 어댑터를 하나의 어댑터로 효율적으로 병합하기 위한 새로운 기법 TIP(Task Importance-based Pruning)을 제안한다. 기존 병합 방식은 정적인 sparsity 설정이나 단순 선형 결합에 의존함으로써, 레이어 간 중요도 차이를 반영하지 못하고 성능 저하를 유발하는 한계가 있다. 제안 기법은 입력 샘플 기반의 중요도 분석을 통해 레이어별 연산 기여도를 정량화한 뒤, 이에 따라 동적인 sparsity를 적용하고, WANDA[3] 기반 어댑터 pruning을 수행하여 병합을 진행한다. LLaMA 2 7B 모델과 총 7개 벤치마크 태스크에 대한 실험 결과, 제안한 방법은 기존 방식 대비 평균 +17.6% 높은 정확도를 기록하였으며, 특히 CoQA에서는 기존 연구 대비 최대 +43.8%, HellaSwag에서는 +31.2%의 상대적 성능 향상을 달성하였다.

Ⅰ. 서 론

대규모 언어 모델(LLM)은 다양한 자연어 처리 과제에서 뛰어난 성능을 보여주고 있다. 그러나 이러한 모델을 완전 미세조정(full fine-tuning)하는 데에는 막대한 계산 자원과 시간이 요구된다. 이를 해결하기 위한 대안으로, 효율적인 파라미터 튜닝 기법인 LoRA(Low-Rank Adaptation)[1]가 널리 활용되고 있다. 하지만 multi-task 환경에서 LoRA 를 적용할 경우, task 별로 개별 어댑터를 유지·로드해야 하므로, 추론 시 불필요한 오버헤드가 발생하고 시스템 효율성이 저하된다. (Fig. 1)

본 연구에서는 이러한 비효율성을 해소하기 위해, 서로 다른 task 에 특화된 LoRA 어댑터들을 하나의 어댑터로 병합하여 multi-task 를 단일 batch 내에서 효율적으로 처리할 수 있게 하는 방법인 TIP(Task Importance-based Pruning)을 제안한다. 특히, 기존의 단순 선형 결합이나 정적 pruning 기반 병합 방식과 달리, 본 방법은 task 별로 중요한 레이어를 식별한 뒤, 각 레이어에 대해 동적으로 sparsity 를 조정하는 방식을 통해 보다 정밀하고 효과적인 LoRA 병합을 가능하게 한다.

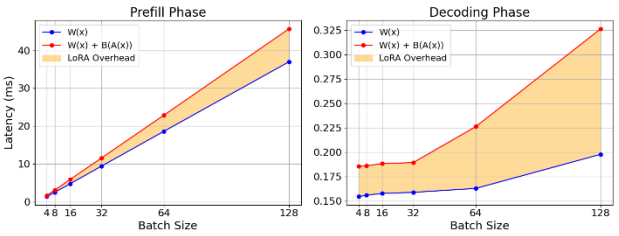

Figure 1: 배치 별 추가적인 LoRA 연산 오버헤드

Ⅱ. 관련 연구

Model Merge

기존 LoRA 어댑터 병합 방식 중 가장 단순한 접근은 선형 결합(linear combination)이다. 그러나 이 방식은 어댑터 간 파라미터 간섭(interference)을 유발할 수 있으며, 특히 task 간 feature 분포가 상이한 경우 병합 후 성능 저하가 발생할 수 있다.

DARE [2]는 병합 시 각 어댑터의 weight 를 무작위로 pruning 하고, 남은 weight 를 sparsity 비율에 따라 rescale 하여 병합함으로써 파라미터 간 간섭을 확률적으로 희석한다. 그러나 무작위 방식은 연산에 중요한 요소까지 제거할 수 있는 위험이 있으며, 모든 레이어에 동일한 sparsity 를 정적으로 적용하기 때문에 작업 간 특성이나 레이어별 중요도를 반영하지 못한다는 한계가 있다.

Ⅲ. 본론

3.1. Methods

본 연구에서는 task 간 레이어 중요도의 차이를 반영하기 위해, 각 task 의 train split 에서 무작위로 추출한 128 개 샘플을 활용하여 각 레이어의 상대적 중요도를 동적으로 측정하고, 이를 기반으로 pruning 비율을 조정하는 방식을 채택한다.

레이어 중요도 측정 과정은 3.1.1 절에서, 이를 활용한 Pruning 및 LoRA 어댑터 병합 과정은 3.1.2 절에서 상세히 설명한다.

3.1.1 레이어 별 중요도 측정

레이어 중요도 측정 과정은 두 단계로 구성된다. 먼저, intra-layer 분석을 통해 각 레이어 l의 가중치 행렬 (W^l)와 입력 샘플(X^l)의 통계적 특성을 바탕으로 평균 활성화 값(A^l)을 산출한다:

$$A_{ij}^l = |W_{ij}^l| \odot \|X_j^l\|_2 \quad (1)$$

여기서 $\|X_j^l\|_2$는 레이어 l에 입력된 샘플들에 대해 j번째 입력 차원의 평균 L2 norm 이다. 이후, A^l의 평균을 취해 전체 레이어의 중요도를 하나의 스칼라 값 I_l로 나타낸다:

$$I_l = mean(A_{ij}^l) \quad (2)$$

레이어 간 상대적 중요도를 비교할 수 있도록 정규화 과정을 수행한다. 이를 위해 $\{I_1, I_2, \dots, I_L\}$의 최소값과 최대

값을 기준으로 각 레이어에 대해 다음과 같은 정규화 스코어($\tilde{I}_l$)를 계산한다:

$$\tilde{I}_l = \left(\frac{I_l - \min(I)}{\max(I) - \min(I)} \right) \quad (3)$$

정규화된 $\tilde{I}_l$ 은 이후 중심 정렬을 통해, 각 레이어의 sparsity 조정치가 평균 0 을 중심으로 분포되도록 한다:

$$\Delta S_l = \tilde{I}_l - mean(\tilde{I}_l) \quad (4)$$

이렇게 조정된 ΔS_l은 전체 pruning 비율 S_{org}와 결합되어, 각 레이어의 최종 sparsity 비율로 사용된다:

$$S_l = S_{org} + \Delta S_l \quad (5)$$

아래의 그림 2 는 본 기법을 적용했을 때, 두 가지 태스크(arc_challenge, hellaswag)에 대해 레이어별로 계산된 최종 sparsity importance 차이를 시각화한 것이다. 이를 통해 태스크별로 중요하게 작동하는 레이어가 어떻게 달라지는지를 확인할 수 있다.

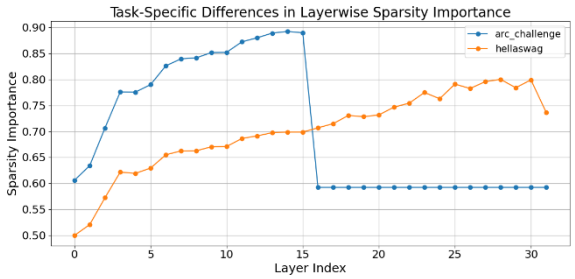

그림 2: layer 별 layer-wise sparsity 비율 ($S_{org} = 0.7$)

3.1.2 TIP 을 활용한 LoRA 어댑터 병합

레이어별 중요도에 따라 정해진 sparsity 비율을 기반으로, 본 연구는 LoRA 어댑터에 WANDA[3] 기법을 적용하여 pruning 을 수행한다. WANDA 는 weight 자체의 크기뿐만 아니라, 해당 weight 에 대응되는 입력값의 정도까지 함께 고려함으로써, 보다 정교한 출력 활성화 기반 pruning 을 가능하게 한다. WANDA 기법은 수식 (1)과 같으며 레이어별로 사전에 결정된 S_l에 따라 중요도가 낮은 weight 부터 순차적으로 제거한다.

이 과정을 모든 LoRA 어댑터에 개별적으로 적용한 뒤, element-wise summation 방식으로 병합함으로써, 여러 task 에 최적화된 통합 어댑터를 구성한다.

아래의 그림은 위 절차를 시각적으로 설명한 것이다. 그림에서 t는 서로 다른 task 를 의미하며, 각각의 $W_{ij}^{l(t)}$와 $\|X_j^{l(t)}\|$는 해당 task 의 LoRA weight 와 입력 샘플을 나타낸다.

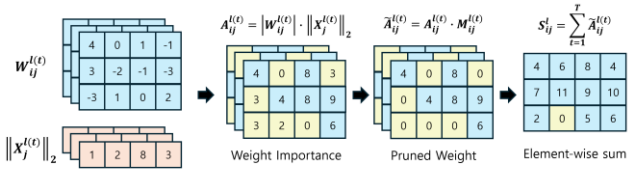

그림 3: Pruning 을 활용한 병합 과정 개요도

3.2. Evaluation

본 실험은 LLaMA 2 7B 모델을 기반으로 하여 총 7 가지 대표적인 벤치마크 태스크(ARC-Challenge, BoolQ, CoQA, HellaSwag, OpenBookQA, PIQA, RTE)를 대상으로 평가하였다. 모든 태스크에 LoRA rank 8 설정을 적용하였으며, 사용된 어댑터는 HuggingFace[4]에서 공개된 사전학습(pre-trained) 어댑터를 그대로 활용하였다. 평가는 각 태스크의 공식적으로 제공된 평가 split(test 또는 validation)을 사용하였으며, 성능 지표는 accuracy 를 기준으로 산출하였다. 모든 실험은 NVIDIA A40 GPU (48GB) 단일 장비에서 수행되었다.

표 1 은 병합하지 않은 단일 LoRA(Single), 단순 선형 병합(Linear), DARE[2], 그리고 본 논문에서 제안한 TIP 병합(TIP)의 정확도를 비교한 결과이다. Linear 와 DARE 는 병합 후 성능 저하가 발생하는 반면, TIP 은 성능을 유지하거나 오히려 기존보다 향상시켜, 일종의 앙상블 효과와 유사한 성능 개선을 달성했음을 보여준다.

특히 CoQA 와 HellaSwag 와 같이 문맥 의존성이 높고 reasoning 이 요구되는 태스크에서 TIP 은 두드러진 성능 향상을 보였다. CoQA 에서는 DARE 대비 약 +43.9%, HellaSwag 에서는 +31.3%의 상대적인 정확도 향상이 관측되었다. 한편, RTE 태스크에서는 단일 LoRA 대비 성능 하락이 있었으나, TIP 은 여전히 Linear 나 DARE 보다 높은 정확도를 기록하여 정보 손실 없이 안정적인 통합 성능을 유지함을 입증하였다.

	Single	Linear	DARE	TIP
ARC	0.4625	0.3370	0.3413	**0.4983**
Boolq	0.7774	0.7275	0.7174	**0.8138**
Coqa	0.6388	0.2260	0.2257	**0.6647**
Hellaswag	0.7600	0.4609	0.4547	**0.7675**
Openbookqa	**0.4420**	0.3840	0.3720	**0.4420**
Piqa	0.7905	0.6627	0.6540	**0.7933**
RTE	**0.6282**	0.5343	0.5379	0.5560

표 1: 태스크 별 정확도

Ⅳ. 결론

본 연구는 multi-task 환경에서 LoRA 어댑터 병합 시 발생하는 연산 비효율성과 파라미터 간섭 문제를 해결하기 위해, 입력 기반 레이어 중요도 분석과 동적 sparsity 조정을 결합한 새로운 병합 기법(TIP)을 제안하였다. 제안된 방법은 기존의 선형 결합 및 정적 sparsity 방식보다 일관되게 높은 정확도를 달성하였으며, 일부 복잡한 태스크에서는 병합 자체가 오히려 성능 향상을 유도하는 효과도 관찰되었다.

ACKNOWLEDGMENT

이 논문은 2025 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 **(No.RS-2020-II201373, 인공지능대학원지원(한양대학교); NO.RS-2021-II211343, 인공지능대학원지원(서울대학교))**

참 고 문 헌

[1] HU, Edward J., et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022, 1.2: 3.

[2] YU, Le, et al. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In: *Forty-first International Conference on Machine Learning*. 2024.

[3] SUN, Mingjie, et al. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

[4] https://huggingface.co/Styxxxx