DA Adapter : Domain Adaptation using Low Rank Adapter

Anonymous ACL submission

Abstract

A recent trend in natural language processing involves fine-tuning a pre-trained Large Language Model (LLM) for a specific task or domain. However, domain adaptive learning of LLMs can be computationally expensive, and requires retraining the LLM using a lot of computing resources to add new domains. To address these issues, this paper proposes a Domain Adaptation Adapter (DA Adapter) model that utilizes Low-Rank Adapter. It reduces the computing budget consumed in domain adaptation by freezing the LLM applied and improves performance. In addition, the low-rank adapter is designed to be easily replaced, making adaptation to multiple domains easy.

1 Introduction

LLMs that use the Transformer's decoder(Vaswani et al., 2017) have demonstrated strong performance in most domains, even surpassing human accuracy(OpenAI, 2023). However, these models still face a challenge; their performance drops significantly in certain domains compared to others To address this issue, a proposed solution is to fine-tune LLMs using domain-specific datasets. The resulting model is referred to as a Domain-Specific LLM. One issue with Domain-Specific LLMs is their high computing requirements. Due to the nature of LLMs, they demand significant computing power, resulting in increased budgetary needs(Touvron et al., 2023).

To enhance computing power, a technique called *Adapter* has been stuided(Poth et al., 2023). The Adapter approach adds a straightforward adapter model to the applied LLM. LLaMA-Adapter(Zhang et al., 2023) is a LLM adapter model which enables parameter-efficient fine-tuning and allows multi-modality inputs to LLM. Another LLM adapter model is LoRA that uses low-rank adapters to achieve more Efficient Fine-Tuning(Hu et al., 2021). LoRA performs fine-tuning by freezing the weights of LLM and attaching Low-Rank adapters to all linear latyers. The authers of (Hu et al., 2021) say this apporach can achieve similar or identical performance to fine-tuned LLM without adapters. The inference process uses the merged LLM, which is obtained by adding the low-rank adapter weight to the LLM weight, resulting in no slowdown in performance. There is another LLM adapter model, called UDApter, which performs domain adaptation by modifying the location of straightforwar adapters(Malik et al., 2023)

(Hu et al., 2021) proposed a method for replacing multiple adapters. In LoRA-Merged LLM Weight, another Low Rank Adapter is merged into the original LLM Weight obtained by unmerging the LoRA Weight. This approach has the advantage of not introducing additional bottlenecks in the inference process, but it has the disadvantage of causing a delay in the adapter replacement process.

This paper proposes a Domain Adaptation Adapter, called DA Adapter, which utilizes a Low Rank Adapter for domain adaptation. Unlike the approach proposed in (Hu et al., 2021), the DA Adapter applies a low-rank adapter to a decoder layer in parallel like a skip connection in ResNet. It maintains the same structure for the inference process. Low-rank adapter has fewer parameters compared to the LLM, resulting in minimal speed degradation even when used without merging it into the LLM. During the fine-tuning process, Low Rank Adapters are applied to all Decoder Layers to facilitate the replacement of Low Rank Adapter, even if it is used without merging it. The experimental results in Section 3 demonstrate that this structure leads to higher Domain Adaptation performance.

2 The Proposed Method: DA Adapter

To comprehend the proposed DA Adapter method, this section describes the structure of LoRA first, and then presents the DA Adapter method.

2.1 Low Rank Adapter

Figure 1 displays the structure of applying Low Rank Adapters during training, as proposed in the LoRA paper(Hu et al., 2021). For the purpose of explanation, we refer to the adapter applied to each layer, as shown in Figure 1, as the 'Low Rank Adapter', which reflects its structural characteristics. The overall structure of the LoRA paper is referred to as the 'LoRA structure' or simply 'LoRA'.

2.2 DA(Domain Adaptation) Adapter

Low Rank Adapters are applied to the Q, K, V, O Linear Layers in all Decoder Layers, including the Embedding Layer and Head Layer. During the fine-tuning process, we freeze the remaining layers except Low Rank Adapters and only update the parameter weights of Low Rank Adapters. For inference, merged Linear Layers are used by adding the weights of a Linear Layer to the weights of the relevent Low Rank Adapter.

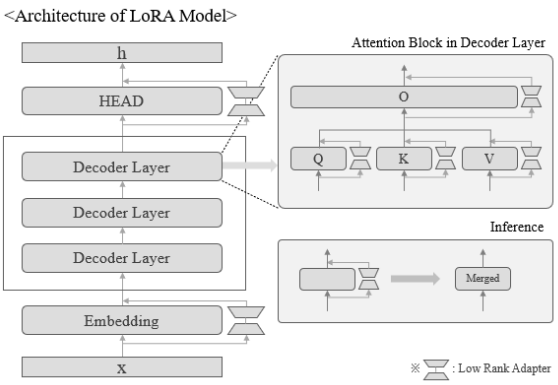

Figure 1: LoRA Training Structure (Inference is performed by merging the weights of low-rank adapters)

Figure 2 displays the proposed DA Adapter structure in this paper. For convenience, we will refer to the structure as the 'DA Adapter structure' or 'DA Adapter'. The DA Adapter uses the same Low Rank Adapter as LoRA, but applies it to Decoder Layers instead of the Linear Layer. Unlike the LoRA structure, which requires access to the internals of the Decoder Layer, the DA Adapter structure can add Low Rank Adapters without modifying the Decoder Layer.

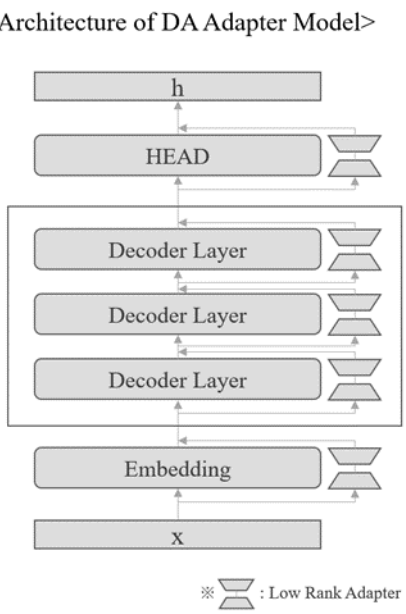

Figure 2: shows the proposed structure for the DA Adapter (Low Rank Adapters are available for both fine-tuning and inference)

Equation 1 and Equation 2 calculate the formulas of computing in the LoRA and DA Adapter's Attention Block, respectively. Equation 1 is computed inside the Decoder Layer's Attention Block, which may affect the performance of Domain Adaptation. On the other hand, DA Adapter operates low rank adapters outside of the Decoder Layer's Attention Block, as shown in Equation 2. LoRA can deliver value regardless of the Attention Block, resulting in improved Domain inference performance.

$$h = x[softmax\{\frac{(Q + B_Q A_Q)(K + B_K A_K)^T}{\sqrt{d_k}}\}(V + B_V A_V)] \quad (1)$$

$$h = xsoftmax(\frac{QK^T}{\sqrt{d_k}})V + BA \quad (2)$$

3 Evaluation

3.1 Experiment Environment

Table 1 displays the computing environment utilized in our experiments. As taking the computing budget into account, we configured the computing resources to run on a personal computer.

Table 2 presents the specifications of the LLMs utilized in our experiments. To assess the learning performance, we employed the DA Adapter structure outlined in this paper, which utilizes the 13b model with 4-bit and 8-bit quantization using BitsandBytes(Dettmers et al., 2022a) (Dettmers et al., 2022b)to fit LLMs into the available VRAM.

CPU	Intel® Core™ i9-13900k
GPU	RTX 4090 24GB
RAM	DDR5 128GB
OS	Ubuntu 22.04.2 LTS

Table 1: Computing Environments used in the Experiment

The comparison models used are based on Vicuna(Chiang et al., 2023) and include LLM Only and LLM with LoRA. The dataset used for training is the one published by Medalpaca(Han et al., 2023). We aime to observe the difference in performance due to the change in structure when training with the same dataset. Each model is trained for 10 epochs with a learning rate of 2e-5 and hyperparameters of lora r = 64 and lora alpha = 64. The experiments use the USMLE (United States Medical Licensing Examination) dataset(Jin et al., 2020), which Medalpaca uses as a benchmark.

3.2 Experimental Results

Figure 3 shows accuracy measurements on the USMLE exam for models with 4-bit quantization. The results are averaged over 10 experiments. Model A, which is not trained on the Medalpaca dataset and only used Vicuna. Model A exhibits lower accuracy than Model C. Model B, which was fine-tuned on the Medalpaca dataset by adding LoRA to Model A, exhibits higher accuracy. However, Model B, which is fine-tuned with the Medalpaca Dataset by adding LoRA to Model A, exhibits worse performance than Model A without fine-tuning. However, the DA Adapter proposed in this paper demonstrates higher accuracy than both A and B.

3.3 THROUGHPUT

Figure 4 displays the throughput of the inference process for model A and model B witch are trained with 8-bit and 4-bit using the LoRA structure, and model C and model D witch are trained with 8-bit and 4-bit using the DA Adapter. The results are averaged over 30 experiments. Due to the quantization process, the inference speed of the 8-bit LLM is faster than the 4-bit. While models A and B experience a speed loss of 2$\tilde{1}$0%, models C and D proposed in this paper only have a maximum speed loss of 2%.

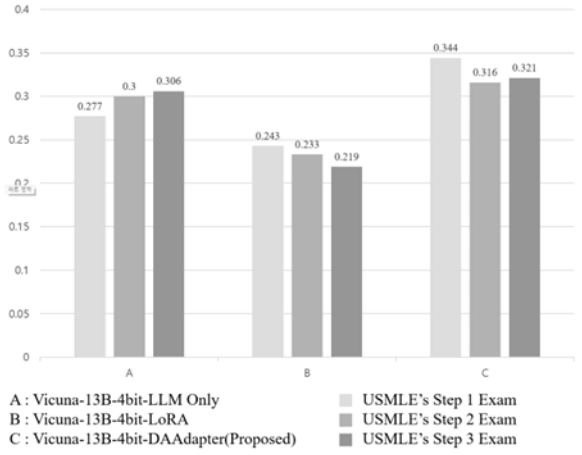

Figure 3: Comparison of Accuracies of 3 Models for USMLE Medical Examinations.

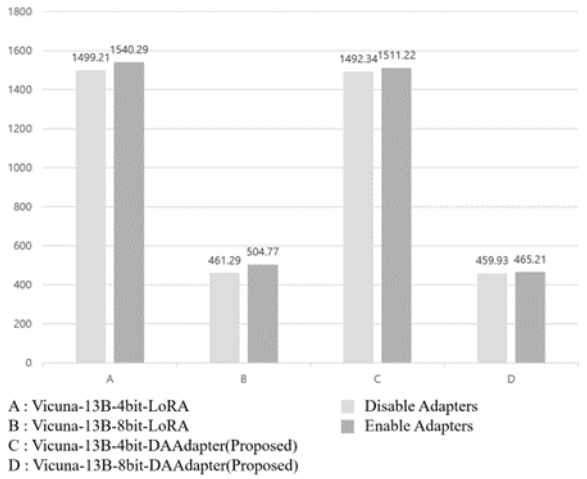

Figure 4: throughput graph for the inference process of four models, measured in milliseconds.

3.4 Latency for Loading an Adapter

Figure 5 displays the latency of loading the Adapter from storage to LLM. Similar to Figure 4, it is an average of 30 measurements. As described in Section 3.3, A and B represent experimental results for LoRA 4-bit and 8-bit models, while C and D represent experimental results for DA Adapter 4-bit and 8-bit models. The proposed structures, C and D, are approximately 16 times faster than A and B. This is because our approach is more efficient than the one described in Section 2. This is because accessing fewer layers, as explained in Section 2, eliminates the need to access the inside of the decoder layer.

4 Conclusion

This paper proposes DA Adapter, which applies low rank adapters to decoder layers only, unlike LoRA which applies low rank adapters to the entire

Model	size	Quantize	Require VRAM
Vicuna	13b	8bit	13.54GiB
Vicuna	13b	4bit	7.86GiB
Vicuna-LoRA	13b	8bit	13.75GiB
Vicuna-LoRA	13b	4bit	8.07GiB
Vicuna-DA Adapter (Proposed)	13b	8bit	13.59GiB
Vicuna-DA Adapter (Proposed)	13b	4bit	8.04GiB

Table 2: Specifications of LLMs used in the experiment

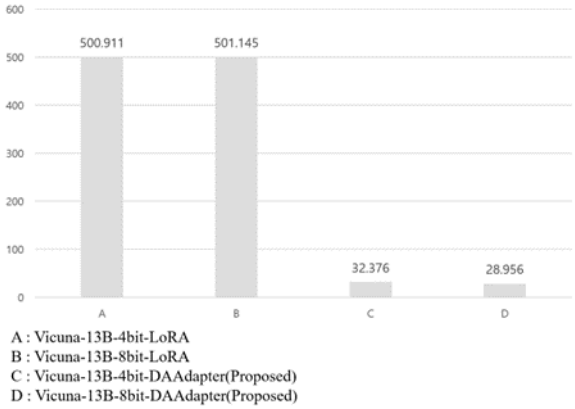

Figure 5: latency graph of the adapter loading process for the four models in milliseconds.

linear layers. DA Adapter can be loaded more than 10 times faster than LoRA, with only a 2% decrease in speedup, as verified through experiments. When using multiple domain-specific LLMs with a single LLM, loading a fine-tuned DA adapter into the domain can provide fast and accurate results.

5 Acknowledgements

This research was partially supported by Basic Science Research Programthrough the National Research Foundation of Korea (NRF) funded by theMinistry of Education (No. 2020R1A6A1A03046811).

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2021R1A2C2094943)

This paper was supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (P0020536, HRD Program for Industrial Innovation)

References

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022a. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022b. 8-bit optimizers via block-wise quantization. *9th International Conference on Learning Representations, ICLR*.

Tianyu Han, Lisa C. Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K. Bressem. 2023. Medalpaca – an open-source collection of medical conversational ai models and training data.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams.

Bhavitvya Malik, Abhinav Ramesh Kashyap, Min-Yen Kan, and Soujanya Poria. 2023. Udapter – efficient domain adaptation using adapters.

OpenAI. 2023. Gpt-4 technical report.

Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. 2023. Adapters: A unified library for parameter-efficient and modular transfer learning.

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. 2023. Llama: Open and efficient foundation language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all

you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. 2023. Llama-adapter: Efficient fine-tuning of language models with zero-init attention.