IJLAI Transactions on Science and Engineering (ISSN: 3006-3418) 2024, vol. 2, no. 4, pp: 65–71

Large Language Model Fine-tuning Method Based on Adaptive Quantization

Lijuan Feng[1], Jiaxiang Wang[1], Jiangjiang Li[1], and Yachao Zhang[1]

School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology
Zhengzhou 450064, China
857003841@qq.com
Corresponding author:Jiangjiang Li and Yachao Zhang
Received Dec. 21, 2024; Revised and Accepted Dec. 24, 2024

Abstract. In recent years, large language models (LLMs) have excelled in comprehensive AI tasks such as language text generation, mathematics, abstraction, and code, and people have seen the embryonic form of general artificial intelligence. However, the fine tuning of the model also needs to consume a lot of computer memory, and the computing resources are extremely high, which is difficult to meet the general consumer grade graphics card. Therefore, an adaptive quantized low-rank (ADAQ-LoRA) fine-tuning algorithm is proposed to solve the problem of video memory consumption during fine-tuning of large language models. The solution is to use both quantification and pruning methods to dramatically reduce video memory usage without losing accuracy. ADAQ-LoRA is applied to ChatGLM2-6B model and its effectiveness is verified in different fine-tuning datasets and downstream scenarios. Compared with the existing large language model fine-tuning methods, ADAQ-LoRA shows better performance and lower memory usage.

Keywords: Large language model, AI, ADAQ-LoRA.

1. Introduction

In recent years, large language models (LLMs) have made remarkable achievements in many natural language processing tasks." Although these models have many parameters and perform well, performance on specific tasks is not always ideal, so the model parameters need to be fine-tuned to meet the specific needs of downstream tasks and improve performance. However, as the size of the model increases, so do the resources required for fine tuning, making fine tuning more complex [1-5].

As a result, researchers have developed a variety of novel fine-tuning methods designed to fine-tune large language models more efficiently and with less resource consumption. At present, the main fine-tuning methods can be divided into two kinds. The first is full fine-tuning. The approach starts with a language model pre-trained with a large amount of text, and then continues training on a small amount of text for a specific task. In this process, the weights of the pre-trained model are updated to better fit the specific task. Although full tuning can achieve good results, it requires a lot of computing resources and takes a long time to train. The second is efficient parameter fine-tuning [6-9]. This method solves the problem of large resources required by traditional fine-tuning techniques by freezing some parameters and training only a small number of parameters. These parameters can be a subset of the model's existing parameters, or they can be a newly added set of parameters. These methods have different characteristics in terms of parameter efficiency, memory efficiency, training speed, model quality and extra inference cost, but can achieve nearly full fine-tuning effect on some tasks.

From the development of pre-training model to the present, many excellent and efficient fine-tuning models have emerged, mainly including Adapter-Tuning [10], LoRA [11] and other methods.

The Adapter-Tuning method is a kind of efficient tuning method for large language models. The idea is to insert a low-rank feed-forward neural network (FFN) module (12-15) in serial between the Transformer layers. The body of the pre-trained model is frozen during fine-tuning, and knowledge of specific downstream tasks is learned by the Adapter module, as shown in Figure 1.

As shown in Figure 1, the Adapter structure consists of a layer of reduced FFN, a layer of nonlinear transformation, and a layer of increased FFN. The dimensions after dimensionality reduction are compressed very small, so the amount of "plug-in parameters" introduced is also very small. During the model fine-tuning process, only fine-tuning the Adapter structure can greatly reduce the video memory consumption. While the Adapter Tuning method reduces the cost of fine tuning, it increases the number of parameters by inserting an Adapter layer into the model. Although the number of new parameters is small, it may reduce the efficiency of model inference in practical application.

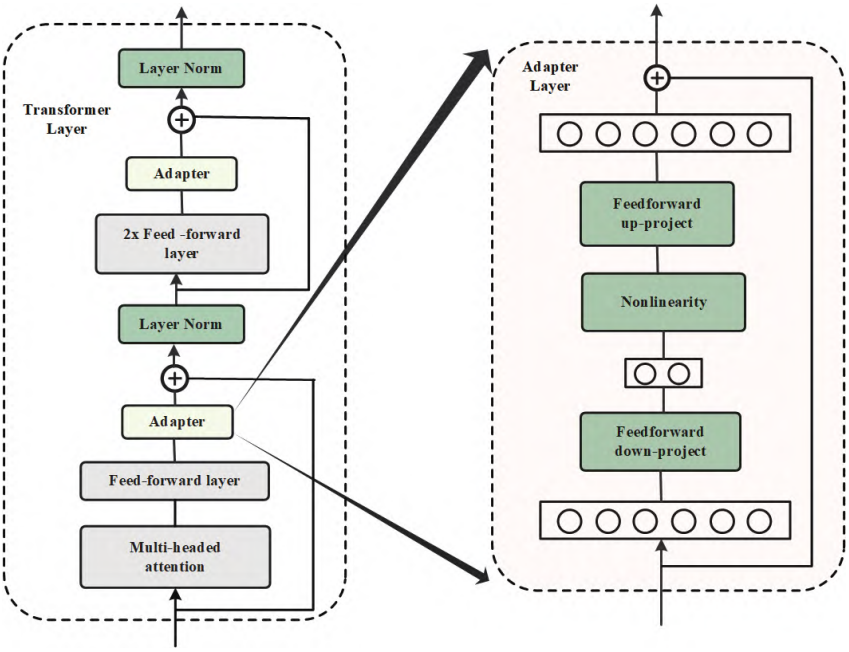

Fig. 1. OAdapter-Tuning model

In recent years, a more mainstream approach is low-rank adaptive fine-tuning (LoRA). LoRA is able to achieve close to full fine-tuning with a small amount of memory. It is to decompose the increment of the pre-training weight, decompose into two low-rank matrix multiplication, then its forward propagation process will become the following process:

$$Y = W_0 x + \Delta W x = W_0 x + \frac{a}{r} BAx. \tag{1}$$

Where W_0, $\Delta W \in R^{n \times d}$, $A \in R^{r \times d}$, $B \in R^{d \times r}$, and $r \ll n, d$, a is a constant scale hyperparameter. One matrix controls the ascending rank and one controls the descending rank. In general, A is initialized with random Gaussian, while B is initialized with zero so that $\Delta W = 0$ at the beginning. Especially in LoRA, the level r is a hyperparameter that should be adjusted for each task.

The method in this paper is based on the LoRA fine-tuning method to add quantization technology, and score according to the importance of the matrix. Less important matrices are hidden, and more computing resources are allocated to important matrices. This strategy greatly reduces the memory usage of model fine-tuning without reducing the performance of model fine-tuning.

2. Model Description

2.1. Transformers

Transformers model has efficient parallel computing capabilities and strong presentation capabilities, can adapt to long sequence data, and has excellent performance in many fields such as natural language processing and computer vision [16-20]. It captures global information through self-attention mechanism, improves modeling ability, and has good universality and expansibility.

Due to the excellent effect of Transformers, most models in recent years have been Transformers based models, Transfoemers-based models are usually stacked by 1 block, where each block consists of two sub-modules to form the multi-head attention and full connection layer. For a given input, the attention of multiple heads to the h head of the input is calculated as:

$$MHA(X) = Concat(head_1, \cdots, head_h) W_0. \tag{2}$$

$$head_i = Softmax(XW_{q_i}(XW_{k_i})^T / \sqrt{d_h}) XW_{v_i}. \tag{3}$$

In the formula, $W_0 \in R^{n\times d}$ is the projection of the output, W_{q_i}, W_{k_i} and W_{qv_i} are the query, key and value projections of the $i-th$ header. d_h is usually equal to d/h. The other submodule in each block is the fully connected layer FFN, which contains two linear transformations.

$$FFN(X) = ReLU(XW_{f_1} + b_1)W_{f_2} + b_2. \tag{4}$$

Where $W_{f_1} \in R^{d\times d_m}$, $W_{f_2} \in R^{d_m \times d}$.

Finally, the two submodules are connected by residuals, and then the result is output through the normalization layer.

As shown in Figure 2, the model structure diagram of ChatGLM2-6B [8] is composed of Decoder structure in Transformers as the basic structure.

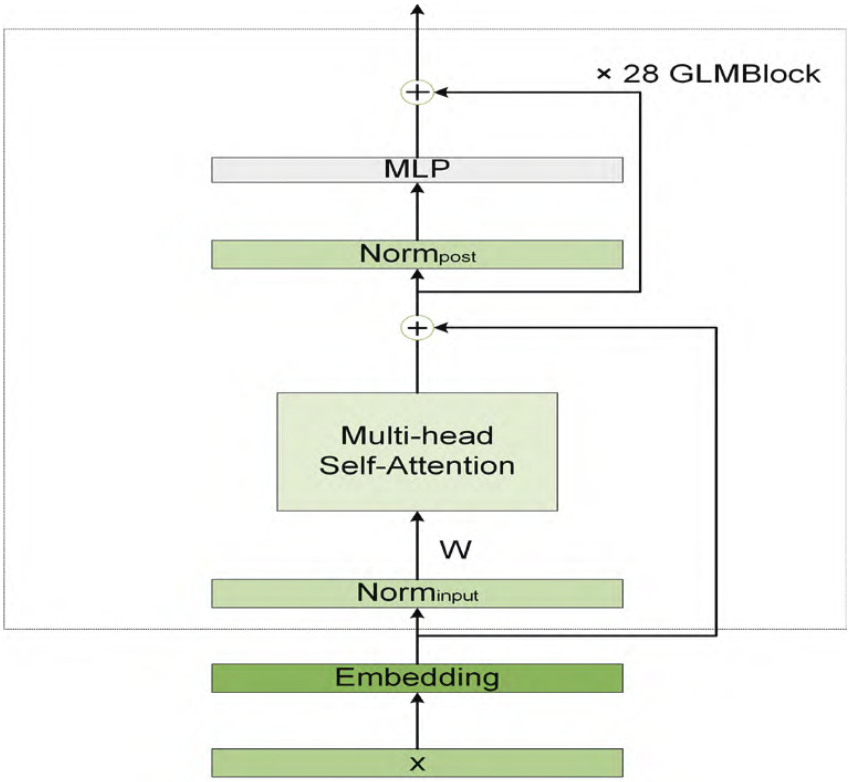

Fig. 2. ChatGLM2-6B model

Where, there are 28 GLMblocks in ChatGLM2-6B model, and there are normalization layer, self-attention layer, post-normalization layer and multi-layer perceptron layer in GLMBlock. The final output is calculated through these layers.

2.2. Adaptive Based on Singular Value Decomposition

In order to adjust the level of the incremental matrix to control its budget, the incremental matrix is decomposed by SVD [21-23].

$$Wx = W_0x + \Delta Wx = W_0x + P\Lambda Qx. \tag{5}$$

In the formula, $P \in R^{d\times r}$ and $Q \in R^{dr\times k}$ represent the singular vectors around $\Lambda \in R^{r\times r}$, and the diagonal matrix contains the singular value $\lambda_{i1\leq i\leq r}$, $r \ll d, k$. $G_i = (P_{*i}, \lambda_i, Q_{i*})$ represents the triplet that holds the $i-th$ singular value and vector. Λ is initialized to zero, while P and Q are initialized with random Gaussian to ensure that $\Delta = 0$ at the start of training. To force the orthogonality of P and Q, i.e. $P^TP = Q^TQ = I$, the following regularizer is used.

$$R(P,Q) = |P^TP - I|_F^2 + |QQ^T - I|^{2_F}. \tag{6}$$

This article notes that structured pruning can be used to control rankings in LoRA applications. However, this approach has the following disadvantages. First, when the measurement finds that both parameters AB are not important, all of their elements must be cut off, that is, all of the elements of A are discarded. But doing so makes it almost impossible to reactivate the pruned two matrices because their values are zeroed and they are not trained. In contrast, this method simply masks the singular values according to the above formula, leaving the singular vector unchanged. This preserves the possibility of recovery of triples that might have been dropped by error. Second, A and B in LoRA are not orthogonal, which means that A and B are dependent. Compared with the truncated minimum singular value method, discarding A and B may result in greater changes in the original matrix. Therefore, if the truncated least singular value method is not used, the incremental matrix usually changes dramatically after each step of ranking assignment, which can cause training instability and even impair the generalization ability for downstream tasks.

2.3. Importance Ranking

In the above formula, LoRA is decomposed by SVD and applied to each weight matrix. In order to reduce the training cost, the computational resources are allocated according to the important fractions of the weight matrix, and the singular values are pruned. For visualization, the resulting incremental matrix is indexed by k, i.e. $W_k = P_k \Lambda_k Q_k$, where $k = 1, 2, \cdots, n$, where n is the number of incremental matrices that need to be fine-tuned. The $i-th$ triplet in ΔW_k is represented by $(\Re_{\hbar,i} = P_{k,i}, E_{k,i}, Q_{k,i})$ and its importance score is $S_{k,i}$.

Further parameters are represented as $P = {P_k}_{k=1}^{n}$, $E = {\Lambda_k}_{k=1}^{n}$, $Q = {Q_k}_{k=1}^{n}$, the cost of training is defined as $C(P, E, Q)$. According to the formula of regularization, the goal of training for $L(P, E, Q) = C(P, E, Q) + \gamma \sum_{k=1}^{n} R(P_k, Q_k)$, including $\gamma > 0$ is the regularization coefficient. When running to step t, the random gradient is used to update the parameter $(P_k^t, \Lambda_k^t, Q_k^t)$, $k = 1, 2, \cdots, n$. In particular, for Λ_k^t, there are:

$$\tilde{\Lambda_k^t} = \Lambda_k^t - \eta \Delta_{\Lambda_k} L(P^t, E^t, Q^t). \tag{7}$$

3. Experiment and Result Analysis

In order to evaluate the natural language generation capability and video memory consumption of fine-tuning methods on large language models, the ADAQ-LoRA method is compared with mainstream fine-tuning methods on two publicly available large language model fine-tuning datasets to evaluate their advantages and disadvantages.

To evaluate natural language generation ability, two datasets are selected for experiments: School_math and Alpaca_Chinese. In the dataset Alpaca_Chinese, feedback and comparative data from GPT-4 are collected to achieve a comprehensive evaluation and reward of model training. The dataset contains 52000 instruction follow data, which are generated by ChatGPT and translated into Chinese. The dataset School-math contains about 250000 Chinese math problem data generated by the BELLE project and includes the problem learning process. This data consists of three parts: prompt, input, and output. By adding prompt instructions before input, the model can better understand the problem and improve the quality of the model output.

In order to conduct quantitative analysis on the generative ability of the model after fine tuning, this paper chooses to use Rouge-1, Rouge-2 and Rouge-L in ROUGE (recall-oriented understudy for gisting evaluation). These indicators are mainly concerned with the degree of overlap between the generated text and the reference text.

Rouge-1: the similarity between the generated summary and the reference summary is measured mainly based on the coincidence degree of unigram.

Rouge-2: it evaluates the similarity of the generated text to the reference text based on the coincidence of the binary phrase (bigram). A binary phrase is a combination of two words that occur consecutively in a text.

Rouge-L: Based on the longest common subsequence (LCS) calculation [24,25], considering word order and not requiring continuous matching, suitable for evaluating sentence level structural similarity.

BLEU (bilingual evaluation understudy) is used to evaluate machine translation and text generation quality, primarily based on Precision. BLEU measures translation quality by calculating the overlap of n-byte phrases between the generated text and the reference text. The larger the BLEU value, the better the performance.

In the data partitioning step, the data is divided into training set and test set with 0.8 as the demarcation point. 80% of the data is extracted for model training, and the remaining 20% is used as a test set.

During the fine-tuning process, 5 epochs experiments are performed using the pagination AdamW optimizer with a maximum gradient norm of 0.1 and a batch size of 1. The constant learning rate plan is selected and the learning rate is set to 3×10^{-4}. All experiments are conducted on Tesla A100 Gpus to ensure adequate computing power and efficiency.

After fine-tuning, as shown in Tables 1,2, ChatGLM2-6B's ROUGE indicator on the Alpaca_Chinese dataset improved by nearly 5% compared to the original model. Compared to LoRA and QLoRA methods, the fine-tuned model has an 80% reduction in memory usage compared to LoRA and a 4% improvement in ROUGE metrics. Compared to the QLoRA method, the fine-tuned model also has some improvement in the ROUGE metric with 5% less memory.

Table 1. Comparison results on Alpaca_Chinese

Method	Rouge-1	Rouge-2	Rouge-L	BLEU	Avg	Memory/MB
GLM2	0.2962	0.1288	0.1612	0.1337	0.1799	51875
GLM2+LoRA	0.3036	0.1484	0.1950	0.1572	0.2010	40132
GLM2+QLoRA	0.3368	0.1657	0.2117	0.1828	0.2242	9231
GLM2+ADAQ-LoRA	0.3411	0.1688	0.2122	0.1833	0.2263	8446

Table 2. Comparison results on School_math

Method	Rouge-1	Rouge-2	Rouge-L	BLEU	Avg	Memory/MB
GLM2	0.5204	0.2849	0.3408	0.3213	0.3668	48885
GLM2+LoRA	0.5542	0.3113	0.3595	0.3403	0.3913	59551
GLM2+QLoRA	0.5490	0.3245	0.3716	0.3508	0.3989	12626
GLM2+ADAQ-LoRA	0.5553	0.3294	0.3789	0. 3592	0.4057	9072

On School_math dataset, the ADAQ-LoRA method significantly improves the language generation ability of the original model, and the ROUGE index is increased by 3%. Compared with LoRA method, ADAQ-LoRA reduces the memory by 75% and improves the language generation ability. Compared to the QLoRA method, ADAQLoRA reduces memory by 30% and improves ROUGE performance metrics.

These results show that with fine-tuning, ChatGLM2-6B has a significant improvement in natural language generation. At the same time, the model has achieved remarkable results in memory usage and performance index, which proves the effectiveness and high efficiency of ADAQ-LoRA method.

4. Conclusion

In this work, the fine tuning of large language models is discussed. It is found that the LoRA fine-tuning method consumes a lot of video memory and neglects the importance difference between the weight matrices. Therefore, this paper proposes ADAQ-LoRA, which integrates quantitative awareness into LoRA fine-tuning to effectively reduce the use of video memory. Allocate computing resources according to the importance of pre-trained weights, and trim unimportant weights to further reduce memory consumption. Extensive experimental results show that on the base model ChatGLM26B, ADAQ-LoRA significantly reduces the use of video memory compared to other fine-tuning methods, and improves the model's ability to generate natural language on the dataset. In future work, we plan to further investigate effective fine-tuning methods in low resource Settings and explore their applications in image generation and multimodal models.

5. Conflict of Interest

The authors declare that there are no conflict of interests, we do not have any possible conflicts of interest.

Acknowledgments. This work was supported by the "Research on language adaptive model based on deep learning in future-oriented teaching scenarios" Project number: 25B413010.

References

1. Thirunavukarasu A J, Ting D S J, Elangovan K, et al. Large language models in medicine[J]. Nature medicine, 2023, 29(8): 1930-1940.

2. Naveed H, Khan A U, Qiu S, et al. A comprehensive overview of large language models[J]. arxiv preprint arxiv:2307.06435, 2023.
3. Yin S, Li H, Laghari A A, et al. FLSN-MVO: Edge Computing and Privacy Protection Based on Federated Learning Siamese Network With Multi-Verse Optimization Algorithm for Industry 5.0[J]. IEEE Open Journal of the Communications Society, 2024.
4. Chang Y, Wang X, Wang J, et al. A survey on evaluation of large language models[J]. ACM Transactions on Intelligent Systems and Technology, 2024, 15(3): 1-45.
5. Yin S, Ivanović M. Recent Advances in AI Methods for Image Processing: Theory, Algorithms, and Applications. Computer Science and Information Systems, Vol. 21, No. 4, v-viii. (2024), https://doi.org/10.2298/CSIS240400vY.
6. Chen Y, Qian S, Tang H, et al. Longlora: Efficient fine-tuning of long-context large language models[J]. arxiv preprint arxiv:2309.12307, 2023.
7. Yin S, Laghari A A. Multi-branch Collaboration Based Person Re-identification[J]. Journal of Science and Engineering, 2024, 1(1): 19-24.
8. Ibrar M, Sun Y. SEIR Model Based Epidemic Transmission Risk Deep Prediction[J]. Journal of Science and Engineering, 2024, 1(1): 25-31.
9. Teng L, Wang J. A Novel Gesture Recognition Network based on LSTM[J]. Journal of Science and Engineering, 2024, 1(1): 1-6.
10. Chen Y, Fu Q, Fan G, et al. Hadamard adapter: An extreme parameter-efficient adapter tuning method for pre-trained language models[C]//Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023: 276-285.
11. Devalal S, Karthikeyan A. LoRa technology-an overview[C]//2018 second international conference on electronics, communication and aerospace technology (ICECA). IEEE, 2018: 284-290.
12. Lin D, Zou R. Applications, Risk, Challenges, and Future Prospects of ChatGPT in Electronic Records Management[J]. Journal of Artificial Intelligence Research, 2024, 1(1): 1-18.
13. Wang Z, Wang Y. Digital Library Book Recommendation System Based on Tag Mining[J]. Journal of Artificial Intelligence Research, 2024, 1(1): 19-32.
14. Zhang M, Qin C, Qiang F. Leveraging Artificial Intelligence to Assess Physicians Willingness to Share Electronic Medical Records in a Hierarchical Diagnostic Ecosystem[J]. Journal of Artificial Intelligence Research, 2024, 1(1): 55-72.
15. Liu W. Channel Reorganization for Few-Shot Segmentation[J]. Journal of Artificial Intelligence Research, 2024, 1(1): 73-82.
16. Khan S, Naseer M, Hayat M, et al. Transformers in vision: A survey[J]. ACM computing surveys (CSUR), 2022, 54(10s): 1-41.
17. Wolf T, Debut L, Sanh V, et al. Transformers: State-of-the-art natural language processing[C]//Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. 2020: 38-45.
18. S. Yin, H. Li, Y. Sun, M. Ibrar, and L. Teng. Data Visualization Analysis Based on Explainable Artificial Intelligence: A Survey[J]. IJLAI Transactions on Science and Engineering, vol. 2, no. 2, pp. 13-20, 2024.
19. Yin S, Li H, Laghari A A, et al. An anomaly detection model based on deep auto-encoder and capsule graph convolution via sparrow search algorithm in 6G internet-of-everything[J]. IEEE Internet of Things Journal, vol. 11, no. 18, pp. 29402-29411, 2024. DOI: 10.1109/JIOT.2024.3353337.
20. Jiang, Y., Yin, S. Heterogenous-view Occluded Expression Data Recognition Based on Cycle-Consistent Adversarial Network and K-SVD Dictionary Learning Under Intelligent Cooperative Robot Environment[J]. Computer Science and Information Systems, vol. 20, no. 4, pp. 1869-1883, 2023.
21. Markkandan S, Logeshwaran R, Venkateswaran N. Analysis of precoder decomposition algorithms for MIMO system design[J]. IETE Journal of Research, 2023, 69(6): 3398-3405.
22. Hou B, Wang D, Xia T, et al. Difference mode decomposition for adaptive signal decomposition[J]. Mechanical Systems and Signal Processing, 2023, 191: 110203.
23. Teng L, Qiao Y, Shafiq M, et al. FLPK-BiSeNet: Federated learning based on priori knowledge and bilateral segmentation network for image edge extraction[J]. IEEE Transactions on Network and Service Management, 2023, 20(2): 1529-1542.
24. Jisi A and Shoulin Yin. A New Feature Fusion Network for Student Behavior Recognition in Education [J]. Journal of Applied Science and Engineering. vol. 24, no. 2, pp.133-140, 2021.
25. Yu J, Zhao L. A novel deep CNN method based on aesthetic rule for user preferential images recommendation[J]. Journal of Applied Science and Engineering, 2021, 24(1): 49-55.

Biography

Lijuan Feng is with the School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology. Research direction is computer application and AI.
Jiaxiang Wang is with the School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology. Research direction is computer application and AI.

Jiangjiang Li is with the School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology. Research direction is computer application and AI.
Yachao Zhang is with the School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology. Research direction is computer application and AI.