FinLoRA: Finetuning Quantized Financial Large Language Models Using Low-Rank Adaptation on GPUs

Dannong Wang[1], Daniel Kim[1], Bo Jin[1], Xingjian Zhao[1], Tianfan Fu[1], Steve Yang[2], and Xiao-Yang Liu Yanglet[1,3]

[1]Rensselaer Polytechnic Institute, Troy, NY 12180, USA
[2]Stevens Institute of Technology, Hoboken, New Jersey, NJ 07030, USA
[3]Columbia University, New York, NY 10027, USA

Abstract

Finetuned large language models (LLMs) have shown remarkable performance in financial tasks, such as sentiment analysis and information retrieval. Due to privacy concerns, finetuning and deploying financial LLMs (FinLLMs) locally are crucial for institutions and individuals. In this paper, we employ quantized low-rank adaptation (QLoRA) to finetune FinLLMs, which leverage low-rank structure and quantization technique to significantly reduce computational requirements while maintaining model performance. We also employ data and pipeline parallelism to enable local finetuning on commodity GPUs. Experiments on financial datasets validate the efficacy of our approach in yielding notable improvements over the base models.

Introduction

Large language models (LLMs) have demonstrated exceptional capabilities in various applications, such as finance (Liu et al. 2023, 2024b), healthcare (Wang et al. 2024; Chen et al. 2024c,a), scientific discovery (Lu et al. 2022b; Chen et al. 2021, 2024b; Fu et al. 2024), etc. Finetuning using low-rank structures of these models to domain-specific datasets further enhances their performance and improves their applicability to specialized tasks. In the financial domain, finetuned LLMs demonstrate substantial potential for tasks such as sentiment analysis, named entity recognition (NER), and knowledge extraction from financial documents.

FinGPT (Liu et al. 2023, 2024b,a; Tian et al. 2024) applied low-rank adaptation techniques for finetuning quantized LLMs in financial contexts, which displayed noticeable improvement over the base model, while having substantial memory reduction and training speedup. XBRL agent (Han et al. 2024) evaluated the potential of LLM's capabilities in analyzing XBRL reports. The use of Retrieval-Augmented Generation (RAG) and tools-calling techniques on XBRL-related tasks and demonstrated significant improvement in task accuracy.

Due to sensitive data and regulatory constraints, finetuning and inference of LLMs within local environments remain critical requirements for financial institutions. Furthermore, the ability to create personalized and customized LLMs, finetuned for specific tasks, is essential for maximizing the utility of these models in financial applications, such as the FinGPT search agents (Tian et al. 2024).

Building upon prior research (Liu et al. 2024a), we demonstrate that state-of-the-art LLMs can be finetuned for diverse financial tasks locally and cost-effectively using widely accessible GPUs, achieving notable improvements over baseline models. Our main contributions can be summarized as follows:

- We employ Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al. 2023) to alleviate memory requirements and allow more efficient finetuning. Using the low-rank structure reduces the number of trainable parameters required for finetuning, and quantization compresses the model size, further limiting GPU memory consumption.
- We employ distributed data parallelism (DDP) and pipeline parallelism to leverage multiple GPUs effectively. DDP distributes training data across GPUs to accelerate finetuning, while pipeline parallelism partitions the model at the layer level to optimize memory usage during inference. Together, these strategies enable more efficient fintuning and inference for FinLLMs.
- We conduct extensive experiments on diverse financial datasets. Models finetuned with QLoRA exhibit up to a 48% average increase in accuracy compared to baseline models, which validates the effectiveness of low-rank adaptation and quantization in addressing the unique challenges of FinLLMs.

Finetuning LLMs with Quantized Low-rank Adaptation (QLoRA)

Quantized Low-rank Adaptation

Low-rank adaptation (LoRA) (Hu et al. 2021) is a parameter-efficient finetuning method that incorporates a smaller set of trainable weights, such that $\boldsymbol{W} = \boldsymbol{W}_0 + \Delta\boldsymbol{W}$. Let $\boldsymbol{W}_0 \in \mathbb{R}^{n\times n}$ denote the pretrained weights and $\Delta\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$ denote the update weights, where $\boldsymbol{A} \in \mathbb{R}^{r\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times r}$ are trainable parameters. Note that n can be large, e.g., $4,096$ and the rank $r \ll n$, say $r = 4, 8$, or 16. As an example, setting $n = 4,096$, and $r = 8$, then $\boldsymbol{W}_0$ has approximately 16 million parameters, while $\boldsymbol{A}$ and $\boldsymbol{B}$ together have $65,536$ parameters, which is approximately only 0.039% the size of $\boldsymbol{W}_0$.

Table 1: The GPU memory usage during finetuning (blue, batch size and rank of 8) and inference (black, batch size of 1).

	GPU memory (GB)	
Quantization	Llama 3.1-8B	Llama 3.1-70B
16-bit	30.9, 15.0	>300, 131.5
8-bit	11.8, 8.6	258.2, 68.5
4-bit	8.7, 5.6	42.8, 37.8

During the fine-tuning stage, the forward pass can be expressed as:

$$\boldsymbol{y} = \boldsymbol{W}_0\boldsymbol{x} + \Delta\boldsymbol{W}\boldsymbol{x} = \boldsymbol{W}_0\boldsymbol{x} + \boldsymbol{B}\boldsymbol{A}\boldsymbol{x},$$

where $\boldsymbol{W}_0$ denotes the pre-trained weights.

During the inference stage, we do not add $\boldsymbol{A}$ and $\boldsymbol{B}$ back to $\boldsymbol{W}_0$, and we perform

$$\boldsymbol{y} = \boldsymbol{W}_0\boldsymbol{x} + \boldsymbol{B}\boldsymbol{A}\boldsymbol{x}.$$

This is different from (Hu et al. 2021), because we will explore the Mixture of Experts approach that trains multiple LoRA adapters. Therefore, it introduces a small amount of additional costs to the inference process.

Quantized LoRA (QLoRA) (Dettmers et al. 2023) further reduces memory usage by using 8-bit or 4-bit quantization. During finetuning, all weights of the pre-trained model are quantized to 8-bit or 4-bit. Weights will be dynamically dequantized back to 16 bit when performing computation with the input sequence $\boldsymbol{x}$ and the adaptor matrix $\boldsymbol{A}$ and $\boldsymbol{B}$, which remain in 16-bit precision throughout the process.

Table 1 illustrates GPU memory usage with QLoRA during finetuning with batch size and rank of 8 and inference with a batch size of 1. The reductions in GPU memory with quantization displayed practical benefits of resource-efficient finetuning and inference for large-scale models.

High-Performance Optimizations on GPUs

Optimizing Finetuning Process

To accelerate the finetuning process and leverage the computational power of multiple GPUs, we employed Distributed Data Parallel (DDP), which distributes the training data across GPUs. DDP launches one process per GPU, where each process gets its own copy of the model and optimizer. Each process then receives different inputs, and the gradients are computed and synchronized across all GPUs to accelerate training. DDP provides a substantial speedup when multiple GPUs are available (Li et al. 2020).

We also opted to use Brain Floating Point (BF16) during finetuning. BF16 offers the same range of values as FP32 an'd easy conversion to/from FP32. Studies showed that BF16 can achieve similar results as FP32 while having significant speedup and memory savings (Kalamkar et al. 2019).

We used 0/1 Adam optimizer (Lu et al. 2022a), a modified version of the Adam optimizer that linearize each Adam step and allows utilizing 1-bit compression for faster convergence speed, while offering reduced data volume and higher training throughput.

Optimizing Inference Process

Inference on large-scale models such as Llama 3.1 70B demands considerable GPU memory resources, particularly when using higher precision like 8-bit or 16-bit. We employ pipeline parallelism, where the model is partitioned at the layer level and distributed across multiple GPUs; each GPU process computes different micro-batches with different parts of the model concurrently (Liu et al. 2024a).

Experiments

Experimental Setup

The experiments were conducted on a server equipped with four 16-core AMD EPYC 7313 CPUs, 1 TB of RAM, and four NVIDIA RTX A6000 GPUs, each featuring 48 GB of dedicated GPU memory.

We choose **Llama 3.1-8B Instruct** and **Llama 3.1-70B Instruct** (Dubey et al. 2024) models as base models.

Financial Applications

For general financial tasks, our study focuses on three financial language processing tasks: Sentiment Analysis (SA), Named Entity Recognition (NER), and news headline classification.

1. Sentiment Analysis (SA) entails analyzing financial text, such as news articles or tweets, to assign sentiment labels (e.g., positive, negative, or neutral).
2. Named Entity Recognition (NER) is designed to identify and classify critical entities within financial texts, including organizations, locations, and individuals.
3. News headline classification involves categorizing headlines according to predefined criteria or questions, facilitating the automated organization and analysis of financial news.

For eXtensible Business Reporting Language (XBRL) (Saeedi, Richards, and Smith 2007) tasks we focus on tagging and extraction. XBRL is a standardized format designed for the exchange of financial information. Although XBRL documents are based on structured XML (eXtensible Markup Language), their inherent complexity presents challenges that can be addressed using the capabilities of LLMs, thereby facilitating financial reporting and analysis (Han et al. 2024).

Datasets

Sentiment Analysis

- **Financial phrasebank (FPB)** (Malo et al. 2013) contains sentences extracted from financial news and reports. These sentences are annotated with sentiment labels. We manually created the train/test split.
- **Financial question-answering sentiment analysis (FiQA SA)** (Maia et al. 2018) is another sentiment analysis dataset with the same labels as FPB from microblog headlines and financial news.

Table 2: Datasets we used for finetuning and evaluation.

Dataset Name	Type	Train/Test Samples
FPB	Sentiment Analysis	1.2K / 3.6K
FiQA SA	Sentiment Analysis	961 / 150
TFNS	Sentiment Analysis	9.5K / 2.4K
NWGI	Sentiment Analysis	16.2K / 4.1K
Headline	Headline Analysis	82.2K / 20.5K
NER	NER	13.5K / 3.5K
FiNER	XBRL Tagging	900K / 100K
Tags	XBRL Extraction	300 / 150
Values	XBRL Extraction	1K / 150

- **Twitter financial news sentiment (TFNS)** (Rahman 2022) comprises annotated tweets related to financial news labeled with sentiment categories.
- **News with GPT instruction (NWGI)** (Liu et al. 2023) comprises samples with seven labels ranging from "strong negative" to "strong positive". We map the seven labels back to three for simplicity and consistancy with other SA dataset.

Headline classification The Headline dataset (Sinha and Khandait 2020) categorizes headlines into two classes, "yes" and "no", based on predefined questions.

Named entity recognition (NER) The NER dataset (Salinas Alvarado, Verspoor, and Baldwin 2015) annotates one entity per sentence, categorized into one of three classes: "location", "person", and "organization"

XBRL tagging The FiNER dataset (Loukas et al. 2022) includes sentences annotated with 139 types of XBRL Tags. We processed the dataset so each question comprises of the sentence and one highlighted entity and the answer includes the correct tag.

XBRL extraction The XBRL extraction dataset comprises questions and answers derived from XBRL filings from 2019 to 2023 for Dow Jones 30 companies. Each example includes a question, a text segment from an XBRL file containing the answer, and the ground truth generated using an XBRL file extraction library. From this dataset, we selected the following two tasks:

- **XBRL tag extraction**: The extraction of a specific XBRL tag from a large XBRL raw text segment given a natural language description of the tag.
- **XBRL value extraction**: The extraction of a numeric value from a large XBRL raw text segment given a natural language description of the value.

To allow better instruction following for the base model, we use one-shot prompting by providing an example question and answer.

Implementation Details

Finetuning We employed distinct finetuning strategies based on the nature of the tasks:

- **General financial tasks and XBRL tagging:** For sentiment analysis, headline classification, and named entity recognition, and XBRL Tagging, single-task fine-tuning was employed.
- **XBRL Extraction:** For XBRL tag extraction and value extraction, multi-task fine-tuning was adopted.

All fine-tuning experiments utilized the 0/1 Adam optimizer (Lu et al. 2022a) with learning rate of 1e-4, LoRA alpha of 32, LoRA dropout of 0.1. We use both LoRA rank 4 with 4-bit quantization and rank 8 with 8-bit quantization for Llama 3.1 8B. We adjusted the batch size and number of training epochs based on the model size and task:

- **General financial tasks and XBRL tagging:**
  - Llama 3.1 8B: Batch size of 16 with gradient accumulation step of 1; 4 epochs.
  - Llama 3.1 70B: Batch size of 4 with gradient accumulation step of 4; 4 epochs.
- **XBRL extraction:**
  - Llama 3.1 8B: Batch size of 2 with gradient accumulation step of 2; 1 epoch.

Inference We use 8-bit quantized inference for all evaluations to ensure consistency.

Performance Metrics

We evaluate performance using the following metrics:

Accuracy Accuracy is the ratio of the number of correct answers to the total number of queries. An answer is considered correct if the ground truth answer is included in the generated response.

Weighted F1 score For classification tasks, we report the weighted F1 score, calculated as the weighted average of the F1 scores for each class, with weights proportional to the number of instances in each class

Finetuning and Inference Performance

- **Batch size**: The batch size per GPU during finetuning.
- **GPU memory usage**: The sum of the amount of GPU memory used for all GPUs during training.
- **GPU hours**: The product of total training time and number of GPUs used.
- **Adapter size**: The size of the LoRA adapter file.
- **Inference speed**: The number of seconds to process an example.

Results and Analysis

Tables 3 summarize the accuracy and weighted F1 scores under different finetuning configurations. Table 4 and 5 displays resource usage and inference performance for NER and XBRL extraction. The finetuned Llama 3.1 8B demonstrates noticeable improvements in accuracy compared to its base model and even surpasses the results of the Llama 3.1 70B base model.

Table 3: Performance on Classification and XBRL Extraction Tasks: Accuracy (blue) and F1 Score (black).

Model	Classification Datasets						XBRL Extraction		XBRL Tagging
	FPB	FIQA	TFNS	NWGI	NER	Headline	Tags	Values	FiNER
Llama-3.1-8B (base)	68.73% 0.6768	46.55% 0.5571	69.97% 0.6834	46.58% 0.4117	48.89% 0.5686	45.34% 0.5576	79.37% -	55.26% -	2.85% -
Llama-3.1-70B (base)	74.50% 0.7363	47.27% 0.5645	68.42% 0.6864	79.93% 0.7993	46.28% 0.4539	71.68% 0.7294	89.02% -	87.66% -	9.41% -
Llama-3.1-8B-4bits-r4	**86.30%** **0.8600**	73.09% 0.7811	**88.27%** **0.8824**	80.95% 0.8029	96.63% 0.9664	88.03% 0.8864	**95.00%** -	96.05% -	70.45% -
Llama-3.1-8B-8bits-r8	82.84% 0.8302	**80.36%** **0.8177**	84.05% 0.8436	**83.96%** **0.8492**	98.05% 0.9806	84.66% 0.8520	94.37% -	**97.36%** -	**75.21%** -
Llama-3.1-70B-4bits-r4	80.94% 0.8019	60.00% 0.6719	76.01% 0.7219	80.77% 0.8101	**98.88%** **0.9887**	**96.38%** **0.9474**	- -	- -	- -

Table 4: Finetuning and inference performance on one classification task (NER).

Model	Finetuning				Inference
	Batch size	GPU memory (GB)	GPU hours	Adapter size (MB)	Time (s)
Llama-3.1-8B-4bits-r4	16×4	83.6	0.77×4	4.5	0.1
Llama-3.1-8B-8bits-r8	16×4	96.7	0.90×4	9.0	0.1
Llama-3.1-70B-4bits-r4	4×4	184.3	3.50×4	21.3	0.9

Table 5: Finetuning and inference performance on XBRL extraction.

Model	Finetuning				Inference
	Batch size	GPU memory (GB)	GPU hours	Adapter size (MB)	Time (s)
Llama-3.1-8B-4bits-r4	2×4	139.2	0.44×4	4.5	1.9
Llama-3.1-8B-8bits-r8	2×4	152.2	0.48×4	9.0	1.9

Notably, even with lower quantization (4-bit) and rank 4, the finetuned Llama 3.1 8B model achieves comparable performance to its 8-bit, rank 8 counterpart, while requiring less memory. Furthermore, the fine-tuned 70B model demonstrates practical usability with 4-bit quantization, showcasing the feasibility of deploying larger LLMs for complex financial tasks in resource-constrained environments.

While we utilized four GPUs to expedite finetuning, it is important to note that all finetuning are achievable with one GPU with 48GB memory, albeit with longer training times.

Conclusion and Future Work

This study demonstrated the effectiveness of Quantized LoRA (QLoRA) for finetuning large language models (LLMs) for many financial tasks, including sentiment analysis, named entity recognition, news headline analysis, and XBRL filings. We finetuned both Llama 3.1 8B and 70B models on commodity GPUs, achieving up to 48% improvements in accuracy compared to the base models on average across all tasks. Notably, these performance gains can be achieved with only four GPUs and less than 20 hours of training times per task, making local finetuning and deployment of customized models a feasible option for institutions and individuals.

In future work, we plan to explore multi-task finetuning in classification tasks and expand our investigation of XBRL-related tasks. This will enable FinLoRA to perform more complex analysis and reasoning tasks, further increasing their utility in the financial domain.

Acknowledgement

Dannong Wang, Daniel Kim, Bo Jin, Xingjian Zhao, Steve Yang and Xiao-Yang Liu Yanglet acknowledge the support from a NSF IUCRC CRAFT center research grant (CRAFT Grant 22017) for this research. The opinions expressed in this publication do not necessarily represent the views of NSF IUCRC CRAFT. Xiao-Yang Liu Yanglet acknowledges the support from Columbia's SIRS and STAR Program, as well as The Tang Family Fund for Research Innovations in FinTech, Engineering, and Business Operations.

References

Chen, J.; Hu, Y.; Wang, Y.; Lu, Y.; Cao, X.; Lin, M.; Xu, H.; Wu, J.; Xiao, C.; Sun, J.; et al. 2024a. Trialbench: Multi-modal artificial intelligence-ready clinical trial datasets. *arXiv preprint arXiv:2407.00631*.

Chen, L.; Lu, Y.; Wu, C.-T.; Clarke, R.; Yu, G.; Van Eyk, J. E.; Herrington, D. M.; and Wang, Y. 2021. Data-driven detection of subtype-specific differentially expressed genes. *Scientific reports*, 11(1): 332.

Chen, T.; Hao, N.; Lu, Y.; and Van Rechem, C. 2024b. Uncertainty Quantification on Clinical Trial Outcome Prediction. *arXiv preprint arXiv:2401.03482*.

Chen, T.; Hao, N.; Van Rechem, C.; Chen, J.; and Fu, T. 2024c. Uncertainty quantification and interpretability for clinical trial approval prediction. *Health Data Science*, 4: 0126.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. arXiv:2305.14314.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; and et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Fu, Y.; Lu, Y.; Wang, Y.; Zhang, B.; Zhang, Z.; Yu, G.; Liu, C.; Clarke, R.; Herrington, D. M.; and Wang, Y. 2024. DDN3. 0: Determining significant rewiring of biological network structure with differential dependency networks. *Bioinformatics*, btae376.

Han, S.; Kang, H.; Jin, B.; Liu, X.-Y.; and Yang, S. Y. 2024. XBRL Agent: Leveraging Large Language Models for Financial Report Analysis. In *Proceedings of the 5th ACM International Conference on AI in Finance*, ICAIF '24, 856–864. New York, NY, USA: Association for Computing Machinery. ISBN 9798400710810.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.

Kalamkar, D. D.; Mudigere, D.; Mellempudi, N.; Das, D.; Banerjee, K.; Avancha, S.; Vooturi, D. T.; Jammalamadaka, N.; Huang, J.; Yuen, H.; Yang, J.; Park, J.; Heinecke, A.; Georganas, E.; Srinivasan, S. M.; Kundu, A.; Smelyanskiy, M.; Kaul, B.; and Dubey, P. K. 2019. A Study of BFLOAT16 for Deep Learning Training. *ArXiv*, abs/1905.12322.

Li, S.; Zhao, Y.; Varma, R.; Salpekar, O.; Noordhuis, P.; Li, T.; Paszke, A.; Smith, J.; Vaughan, B.; Damania, P.; and Chintala, S. 2020. PyTorch distributed: experiences on accelerating data parallel training. *Proc. VLDB Endow.*, 13(12): 3005–3018.

Liu, X.-Y.; Wang, G.; Yang, H.; and Zha, . D. 2023. Data-centric FinGPT: Democratizing Internet-scale data for financial large language models. In *Workshop on Instruction Tuning and Instruction Following, NeurIPS*.

Liu, X.-Y.; Zhang, J.; Wang, G.; Tong, W.; and Walid, A. 2024a. Efficient Pretraining and Finetuning of Quantized LLMs with Low-Rank Structure . In *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, 300–311. Los Alamitos, CA, USA: IEEE Computer Society.

Liu, X.-Y.; Zhu, R.; Zha, D.; Gao, J.; Zhong, S.; White, M.; and Qiu, M. 2024b. Differentially Private Low-Rank Adaptation of Large Language Model Using Federated Learning. *ACM Transactions on Management Information Systems*.

Loukas, L.; Fergadiotis, M.; Chalkidis, I.; Spyropoulou, E.; Malakasiotis, P.; Androutsopoulos, I.; and Paliouras, G. 2022. FiNER: Financial Numeric Entity Recognition for XBRL Tagging. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics.

Lu, Y.; Li, C.; Zhang, M.; Sa, C. D.; and He, Y. 2022a. Maximizing Communication Efficiency for Large-scale Training via 0/1 Adam. arXiv:2202.06009.

Lu, Y.; Wu, C.-T.; Parker, S. J.; Cheng, Z.; Saylor, G.; Van Eyk, J. E.; Yu, G.; Clarke, R.; Herrington, D. M.; and Wang, Y. 2022b. COT: an efficient and accurate method for detecting marker genes among many subtypes. *Bioinformatics Advances*, 2(1): vbac037.

Maia, M.; Handschuh, S.; Freitas, A.; Davis, B.; McDermott, R.; Zarrouk, M.; and Balahur, A. 2018. WWW'18 Open Challenge: Financial Opinion Mining and Question Answering. 1941–1942.

Malo, P.; Sinha, A.; Takala, P.; Korhonen, P.; and Wallenius, J. 2013. Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts. arXiv:1307.5336.

Rahman, M. A. 2022. Twitter financial news sentiment. http://precog.iiitd.edu.in/people/anupama.

Saeedi, A.; Richards, J.; and Smith, B. 2007. An Introduction to XBRL. In *British Accounting Association's Annual Conference*.

Salinas Alvarado, J. C.; Verspoor, K.; and Baldwin, T. 2015. Domain Adaption of Named Entity Recognition to Support Credit Risk Assessment. In Hachey, B.; and Webster, K., eds., *Proceedings of the Australasian Language Technology Association Workshop 2015*, 84–90. Parramatta, Australia.

Sinha, A.; and Khandait, T. 2020. Impact of News on the Commodity Market: Dataset and Results. arXiv:2009.04202.

Tian, F.; Byadgi, A.; Kim, D. S.; Zha, D.; White, M.; Xiao, K.; and Liu, X.-Y. 2024. Customized FinGPT Search Agents Using Foundation Models. In *Proceedings of the 5th ACM International Conference on AI in Finance*, 469–477.

Wang, Y.; Xu, Y.; Ma, Z.; Xu, H.; Du, B.; Gao, H.; and Wu, J. 2024. TWIN-GPT: Digital Twins for Clinical Trials via Large Language Model. *arXiv preprint arXiv:2404.01273*.

Appendix

Table 6: Finetuning LLMs with QLoRA methods: listing the number of parameters and GPU memory.

Model	Parameters	GPU Memory Batch = 4	GPU Memory Batch = 8	Model size	Percentage
Llama3-8B-16bit (base)	8.03 B	-	-	16.06 GB	-
Llama3-8B-r8-16bit	4.72 M	30.91 GB	30.91 GB	16.08 GB	100.1%
Llama3-8B-r8-8bit	4.72 M	11.41 GB	11.81 GB	8.04 GB	50.1%
Llama3-8B-r8-4bit	4.72 M	8.26 GB	8.65 GB	4.02 GB	25.0%
Llama3-8B-r4-16bit	2.36 M	30.90 GB	30.90 GB	16.07 GB	100.1%
Llama3-8B-r4-8bit	2.36 M	11.40 GB	11.78 GB	8.03 GB	50.0%
Llama3-8B-r4-4bit	2.36 M	8.25 GB	8.61 GB	4.02 GB	25.0%
Llama3-70B-16bit (base)	70.56 B	-	-	151.53 GB	-
Llama3-70B-r8-16bit	22.28 M	-	-	151.57 GB	100.0%
Llama3-70B-r8-8bit	22.28 M	173.57 GB	258.17 GB	75.79 GB	50.0%
Llama3-70B-r8-4bit	22.28 M	42.78 GB	42.78 GB	37.89 GB	25.0%
Llama3-70B-r4-16bit	11.14 M	-	-	151.55 GB	100.0%
Llama3-70B-r4-8bit	11.14 M	173.36 GB	258.11 GB	75.70 GB	50.0%
Llama3-70B-r4-4bit	11.14 M	42.73 GB	42.73 GB	37.89 GB	25.0%