Effects of Combined Knowledge Distillation and Quantization on Model Performance

Anonymous COLING 2025 submission

Abstract

Quantization and knowledge distillation are two well-studied optimization techniques to decrease the resource needs of a model, but these techniques have only ever been used separately. This work combines the two by distilling a full-sized model into a quantized version of itself. While the performance of the fine-tuned quantized model can't quite compare to that of its full-sized teacher model on GSM8K, our findings show that this hybrid process can yield significant improvements over the pre-trained quantized model, suggesting that quantized models can be fine-tuned to better mimic full-sized model outputs in selected areas of expertise via knowledge distillation. Code from this project is largely adapted from https://anonymous.4open.science/r/knowledge-distillation-8C37/.

1 Introduction

Quantization and knowledge distillation have been proven to be effective ways to reduce model size without overly compromising performance (Cho and Hariharan, 2019) (Gholami et al., 2021). A study at the University of California, Berkeley observed a 16 times reduction in memory footprint and latency in select quantization cases (Gholami et al., 2021). Similarly, knowledge-distilled models have demonstrated performance comparable to their teacher models while consuming much less resources than their respective teacher models (Cho and Hariharan, 2019). We intend to explore the intersection of these two methods to obtain the benefits of both by distilling a larger, full-sized Llama 3.1-8B-Instruct teacher model into a 4-bit quantized instance of the same model. The ensuing performance is then measured on the GSM8K benchmark. Results show that this hybrid pipeline yields a model that is the size of the quantized model with superior results than the pretrained model.

2 Related Works

Quantization is a method of reducing a model's size by decreasing the number of bits of precision used to represent each floating point parameter, thereby reducing resource consumption (Gholami et al., 2021). For example, a model whose full-size form uses 32-bit precision downsized to its 16-bit and 8-bit counterparts by removing the last 16 and 24 precision bits, respectively (Trusov et al., 2024). This necessarily comes at the cost of performance, but as study by Trusov et al. (2024) demonstrates that the gap may be negligible. Researchers tested 4.6 bit quantization and discovered that the quality is close to the mean of the 4-bit and 8-bit neural networks while being 1.5-1.6 times faster than the 8-bit neural network.

Unlike quantization, *knowledge distillation*, is the process of training a smaller "student" model on the outputs of a larger "teacher" model to train the student model to mimic the performance of its teacher(Gou et al., 2021). This was formally introduced in a 2015 study at Google (Jaiswal et al., 2023). Knowledge distillation can be broken down into three separate parts: the knowledge the student should learn, the algorithm for the teacher model to teach the student model, and lastly the teacher-student architecture(Gou et al., 2021). In our experiment, the knowledge section will be the GSM8K benchmark. The student model will be trained on the output *logits* of its teacher. Finally, unique to our experiment, the student model will not be an entirely different model from the teacher, but rather a quantized instance of the teacher.

3 Methodology

In this study, we evaluated the performance of the quantized 4-bit Llama 3.1 8B model before and after knowledge distillation. Both models were fine-tuned and benchmarked against the GSM8k dataset, which involves mathematical problem-

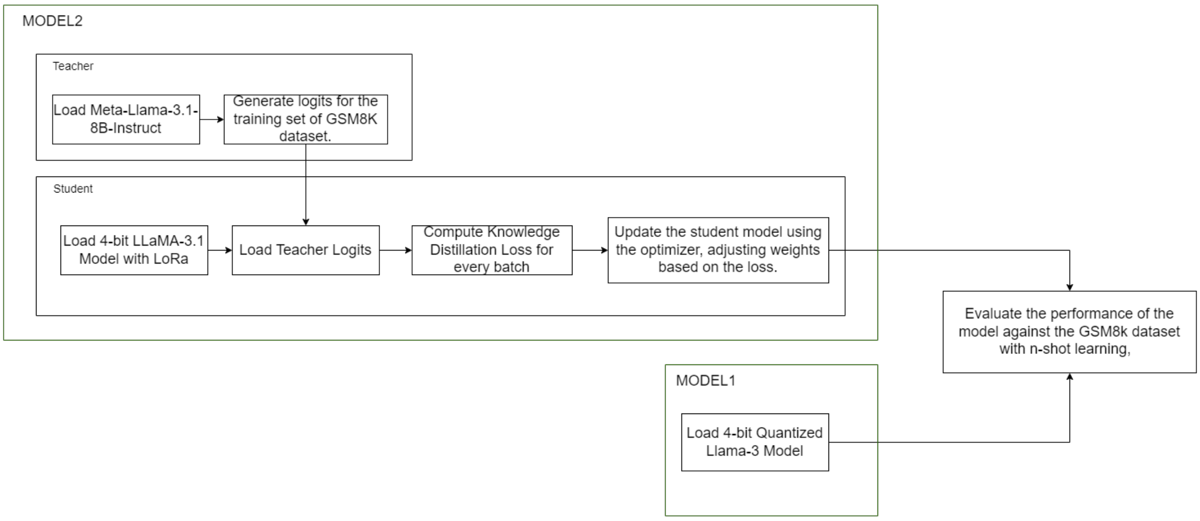
MODEL2
Teacher
Load Meta-Llama-3.1-8B-Instruct
Generate logits for the training set of GSM8K dataset.
Student
Load 4-bit LLaMA-3.1 Model with LoRa
Load Teacher Logits
Compute Knowledge Distillation Loss for every batch
Update the student model using the optimizer, adjusting weights based on the loss.
Evaluate the performance of the model against the GSM8k dataset with n-shot learning,
MODEL1
Load 4-bit Quantized Llama-3 Model

Figure 1: Implementation Workflow

solving tasks. Below, we describe the methodology used to configure and train the models as well as the evaluation process employed for benchmarking their performance.

3.1 Evaluation Metric

The GSM8k dataset was chosen as the benchmark for all models in this expeirment as the performance gap between the quantized and full-sized versions the model in question was significant enough such that knowledge distillation would not be trivial.

The 8-shot prompts to the models involve 8 question-and-answer pairs from the training set of GSM8K dataset with an additional standard prompt to format the answer in the last line. The answers are then compared for accuracies.

3.2 Model choice

The first model was based on Meta-LLaMA-3 (specifically the 8B parameter model) and loaded in a quantized 4-bit format. This configuration was designed for efficient memory usage while maintaining model performance. The model was loaded using the AutoModelForCausalLM API from Hugging Face, with a configuration that applied NF4 quantization, double quantization, and bfloat16 for computation.

The text generation pipeline was used to evaluate the model's baseline response to mathematical queries from the GSM8k dataset.

3.3 Knowledge distillation pipeline

Knowledge distillation is comprised of two steps: (1) transfer set creation and (2) student fine-tuning.

3.3.1 Transfer set creation

During this initial step, a full-sized 16-bit instance of Llama 3.1 8B Instruct downloaded from Hugging Face (meta-llama/Meta-Llama-3.1-8B-Instruct) was run on the first 1000 questions in the GSM8K benchmark set. The logits associated with the response are recorded along with its corresponding benchmark question, forming a key-value dictionary known as a *transfer set*, the to-be training set for the student model.

3.3.2 Student fine-tuning

Once the transfer set has been completed, the student instance, a 4-bit quantized instance of Llama 3.1 8B instruct, is fine-tuned on the logits within that set. LoRa (Low-Rank Adaptation) was used to fine-tune specific layers of the student model, making the process more memory-efficient while retaining knowledge from the teacher model. The following LoRa hyperparameter configuration was applied to the student model:

Parameter	**Value**
LoRa rank	4
LoRa α	8
Dropout	0.2

The loss function used in training combined two components:

- a hard loss (cross-entropy between the student's predictions and true labels) and

- a soft loss (KL-divergence between the teacher and student logits).

The hyperparameters used to train the model are as follows:

Parameter	Value
Number of epochs	2
Batch Size	8
Learning Rate	5e-8
Momentum	0.9

4 Experimental Results

The fine-tuned model achieved a 50% accuracy on the first 100 questions of the GSM8K test set when 0-shot prompted as compared to a mere 40% accuracy for the baseline pre-grained model, a significant difference. The difference between the 8-shot prompt accuracies was less marked, as the fine-tuned model achieved a 76.9% accuracy, and 0.3% increase over the baseline model. The results suggest that knowledge distillation on a quantized model is much more effective when the downstream task involves fewer-shot prompting, implying that the model would actually be more useful in cases where the model isn't artificially prompted to increase performance on a very specific benchmark. This study actually partially demonstrates just how powerful even few-shot prompting can be in increasing model performance.

5 Limitations and Future Work

This study was only conducted on a single data set (GSM8K) with a very limited scope (math questions). Further research is required in various other areas of expertise to verify whether the phenomena in this paper can be seen in other areas, giving us an idea of whether our results are endemic to the hybrid knowledge distillation-quantization method or to the field of expertise itself. Furthermore, due to resource limitations, we used default parameters for all models and for training. A more extensive study should conduct hyperparameter tuning to assess whether more extensive training could produce student model performance approaching that of its teacher.

6 Conclusion

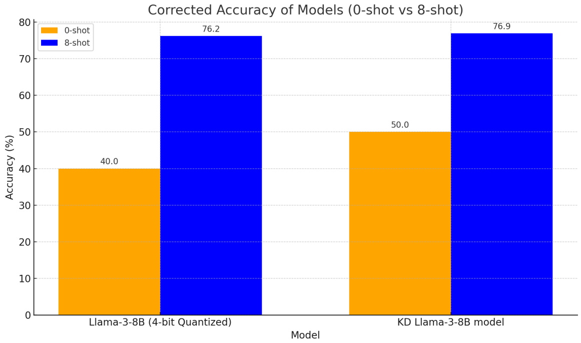

Figure 2: Evaluation of the models on GSM8K dataset

In this paper, we proposed a new way to improve AI models by combining together techniques already being used by researchers. Using quantized Llama-3.18b and by using logit-based quantization, we were able to improve it's performance on the testing dataset GSM8K from 76.2% accuracy to 76.9% accuracy.

References

Jang Hyun Cho and Bharath Hariharan. 2019. On the efficacy of knowledge distillation. *arXiv:1910.01348*.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A survey of quantization methods for efficient neural network inference. *arXiv:2103.13630*.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819.

Ajay Jaiswal, Zhe Gan, Xianzhi Du, Bowen Zhang, Zhangyang Wang, and Yinfei Yang. 2023. Compressing llms: The truth is rarely pure and never simple. *arxiv:2310.01382*.

Anton Trusov, Elena Limonova, Dmitry Nikolaev, and Vladimir V. Arlazarov. 2024. 4.6-bit quantization for fast and accurate neural network inference on cpus. *Mathematics*, 12.