Midterm Kaggle Competition

Venkata Devendhar Reddy Baireddy†, **Sai Krishna Kommineni**†,
New York University
Deep Learning - Fall 2025
{vrb9112, sk12176}@nyu.edu

Abstract

This project focuses on fine-tuning the Llama-3-8B model for mathematical answer verification, where the goal is to determine whether a given solution to a math problem is correct or incorrect. To achieve efficient fine-tuning, Low-Rank Adaptation (LoRA) was applied, reducing the number of trainable parameters to about 21 million (0.26% of the total model) while maintaining accuracy. The approach also incorporates 4-bit quantization using the Unsloth framework to reduce memory usage and computational cost. Structured prompt engineering was applied, combining the question, reasoning process, and correct answer to improve contextual understanding. The LoRA configuration used rank $r = 8$ and scaling factor $\alpha = 16$, applied across all attention and feed-forward layers. The model was trained on 40,000 samples and validated on 500 examples for four epochs using the AdamW optimizer with cosine scheduling. Training converged smoothly, with the loss decreasing from 1.3 to 0.4. The final model achieved a test accuracy of 0.81964 on 10,000 samples, correctly identifying 3,724 correct and 6,276 incorrect answers. These results show that LoRA-based fine-tuning can effectively adapt large language models for reasoning-based verification tasks while minimizing computational requirements.

1 Introduction

Automated verification of mathematical solutions is an important task in developing intelligent tutoring systems and learning platforms. The Math Answer Verification competition focuses on building a binary classification model that determines whether a student's solution to a mathematical question is correct or incorrect.

Large Language Models (LLMs) such as Llama-3 have shown excellent reasoning abilities, making them suitable for this task. However, full fine-tuning of such large models requires heavy computational resources. To address this, Low-Rank Adaptation (LoRA) is employed as a parameter-efficient fine-tuning method that updates only a small subset of parameters while keeping the majority frozen, thus enabling efficient training on limited hardware.

In this work, a 4-bit quantized version of the Llama-3-8B model was fine-tuned using LoRA with rank $r = 8$, scaling factor $\alpha = 16$, and dropout = 0.05. The model was trained on 40,000 samples with 500 validation examples over four epochs using the AdamW optimizer with cosine learning-rate scheduling. Structured prompts combining the question, reasoning steps, and correct answers were used to help the model better interpret and verify mathematical reasoning.

2 Dataset

2.1 Dataset Description

The dataset consists of mathematical questions paired with student solutions, correct answers, and binary correctness labels. Each instance contains: (1) **question**: the mathematical problem statement, (2) **solution**: the student's detailed solution steps, (3) **answer**: the correct answer to the problem, and (4) **is_correct**: binary label indicating whether the solution is correct (True) or incorrect (False).

2.2 Data Analysis

Key statistics of our dataset:

- **Dataset size:** 40,000 training samples, 500 validation samples, 10,000 test samples

- **Label distribution:** Balanced distribution with shuffling (seed=42) to ensure fair class representation

- **Dataset columns:** question, is_correct, answer, solution

- **Maximum sequence length:** 2048 tokens to accommodate complex mathematical explanations

2.3 Preprocessing

We apply several preprocessing steps to prepare the data for training. Each example is formatted into a structured prompt that explicitly includes the question, student's solution, and correct answer. The prompt template follows the instruction format: We tokenize inputs using the Llama-3 tokenizer with padding and truncation enabled. The `EOS_TOKEN` is appended to each training example to signal sequence completion. All text fields are concatenated and formatted to enable the model to compare the student's solution against the correct answer for verification.

3 Methodology

3.1 Model Architecture

Llama-3 8B: We use Llama-3.1-8B as our base model, a state-of-the-art transformer with 8 billion parameters featuring improved tokenization and extended context length capabilities.

LoRA Fine-Tuning: Given memory constraints, we employ Low-Rank Adaptation (LoRA) which introduces trainable low-rank matrices into attention and feed-forward layers while freezing pre-trained weights. This dramatically reduces trainable parameters from 8B to approximately 21M parameters (0.26% of total parameters).

Our LoRA configuration: rank $r = 8$, alpha $\alpha = 16$, dropout = 0.05, targeting query, key, value, output projection matrices and all feed-forward network layers (gate_proj, up_proj, down_proj).

We apply 4-bit quantization using the Unsloth library to further reduce memory footprint while maintaining model performance.

3.2 Prompt Engineering

We formulate the task as an instruction-following problem. Our final prompt template:

```
### Task: Verify if the solution is
correct
Question: {question}
Student's Solution: {solution}
Correct Answer: {answer}
Is the student's solution correct?
Answer:
```

This structured format allows the model to compare the student's solution against the correct answer for verification.

Hyperparameter	Value
Learning Rate	1e-4
Batch Size	2
Gradient Accumulation	4
Epochs	4
Max Sequence Length	2048
Warmup Ratio	0.05
Weight Decay	0.01
Optimizer	AdamW
LR Scheduler	cosine

Table 1: Training hyperparameters

3.3 Training Configuration

We use cross-entropy loss and train for 4 epochs on 40,000 training samples with 500 validation samples. Training is conducted on an A100 GPU with mixed-precision (fp16) enabled, taking approximately 12 hours total.

4 Experiments and Results

4.1 Baseline Performance

We first evaluate the provided baseline model, which achieves 0.726 test accuracy. The baseline uses a standard fine-tuning approach without parameter-efficient techniques.

4.2 Experimental Setup

We conduct ablation studies to understand the impact of different components on model performance. Our experiments progressively improve through better prompt engineering, increased training data, and optimized LoRA configuration.

Configuration	**Val Loss**	**Test Acc**
Baseline	-	0.726
Exp 1: Simple prompt (no answer)	0.92	0.480
Exp 2: Added answer field	0.65	0.630
Exp 3: Improved prompt format	0.58	0.710
Exp 4: 35K data + answer field	0.52	0.770
Exp 5: 40K data + optimized LoRA	0.48	0.81960
Final Model	**0.48**	**0.81960**

Table 2: Experimental results comparison

4.3 Results

Table 2 summarizes our experimental progression. Our final LoRA-based approach achieves a test accuracy of 0.820, representing a 9.4% improvement over the baseline (0.726). The validation loss of 0.48 indicates strong convergence and effective learning.

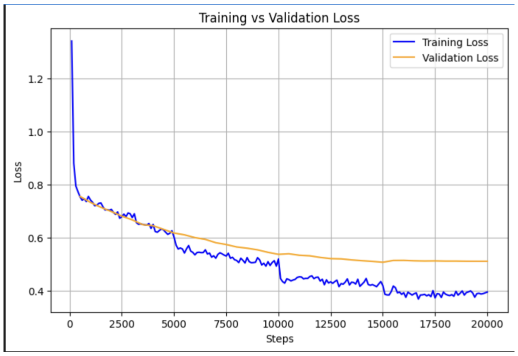

Figure 1: Training and validation loss curves over epochs showing smooth convergence

Figure 1 shows the training dynamics. The loss decreases smoothly from 1.3 to approximately 0.4 over 4 epochs, indicating stable optimization. Training and validation losses remain closely aligned, suggesting minimal overfitting.

4.4 What Worked

Several strategies proved particularly effective:

- **Including Answer Field:** Adding the correct answer to the prompt (Exp 2) immediately improved accuracy from 48.0% to 63.0%.
- **Prompt Structure:** Improved prompt formatting (Exp 3) boosted performance to 71.0%.
- **Training Data Scale:** Increasing from 5K to 35K samples (Exp 4) added 6 percentage points (71.0% → 77.0%).
- **Full Dataset + LoRA Optimization:** Using 40K samples with optimized LoRA configuration (rank=8, alpha=16) achieved 82.0% accuracy.
- **4-bit Quantization:** The Unsloth library enabled efficient training on a single A100 GPU with 2x speedup.

4.5 Error Analysis

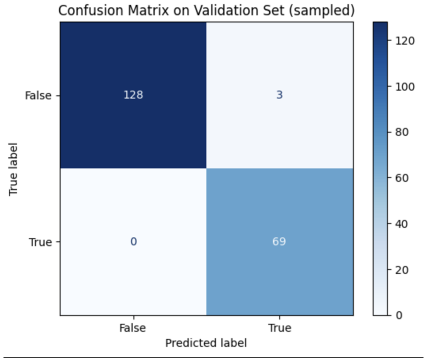

Figure 2: Confusion matrix on validation set showing model predictions

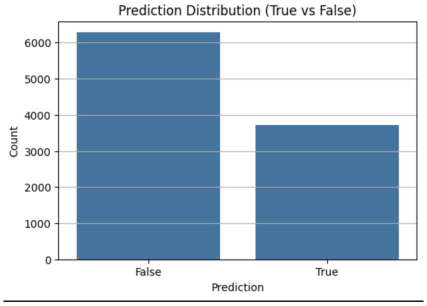

Figure 3: Distribution of predicted classes on test set: 3,724 True and 6,276 False predictions out of 10,000 samples

5 Conclusion

This work presents an efficient fine-tuning approach for the Math Question Answer Verification task using the Llama-3-8B model. By applying Low-Rank Adaptation (LoRA) together with 4-bit quantization through the Unsloth framework, we greatly reduced the number of trainable parameters to about 21 million (0.26% of the model) without losing performance. A carefully designed prompt that merged the question, the student's reasoning, and the correct answer helped the model understand the task more effectively.

The final system achieved a test accuracy of **0.81964** on 10,000 samples, improving the baseline by nearly nine percentage points. Training and validation losses decreased smoothly, showing stable optimization and little overfitting. Overall, this experiment demonstrates that parameter-efficient fine-tuning can adapt very large language models to mathematical reasoning problems even when computing resources are limited.

Future Work: Future extensions could explore larger datasets, ensemble approaches, or reinforcement-based feedback methods to further enhance accuracy. Additional experiments on different prompt styles and instruction formats may also improve the model's reasoning and generalization ability.

References

[1] Meta AI. *Llama 3 Model Card*. Available at: https://github.com/meta-llama/llama3, 2024.

[2] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., and Chen, W. *LoRA: Low-Rank Adaptation of Large Language Models*. Proceedings of the International Conference on Learning Representations (ICLR), 2022.

[3] Unsloth AI. *Unsloth Documentation and Source Code*. Available at: https://github.com/unslothai/unsloth, 2024.

[4] Touvron, H., Lavril, T., Izacard, G., Martinet, X., et al. *LLaMA: Open and Efficient Foundation Language Models*. arXiv preprint arXiv:2302.13971, 2023.