Supervised-Fine Tuning(SFT) of Llama3-8B (DL-Fall-24 Kaggle Contest)

Sahil Faizal
sf4140
sf4140@nyu.edu

Chenghao Yang
cy2668
cy2668@nyu.edu

Rohit Mohanty
rm6201
rm6201@nyu.edu

Abstract

We are tasked with Supervised-Finetunning (SFT)[1] of Llama3-8B model to predict the correctness of answers to math questions. The goal is to assess whether the provided answer to each question is correct or not. We need to train Llama3-8B to generate is_correct label which is either True or False if the given answer to the the question is correct or not. We are free to use provided solution/explanation.

1 Introduction

The goal of this project is to predict the correctness of answers to mathematical questions using the Llama-3, 8B model. The task involves fine-tuning the language model on a specific dataset. By learning provided explanations and answers, the model is trained to determine whether an answer is True or incorrect False.

This report details the dataset used, model introduction, experimentation setup, hyperparameter tuning, and results achieved during the competition. The fine-tuned model achieved a test set accuracy of **0.86776**, demonstrating its effectiveness in distinguishing between correct and incorrect answers.

2 Dataset Description

The dataset for this task is the Math Question Answer Verification Competition dataset, available on Hugging Face [1]. It consists of questions from various math topics, structured into several columns:

2.1 **question:** The math question posed to the student

2.2 **answer:** The ideal or correct answer to the question

2.3 **solution:** A detailed reasoning or solution that explains the answer, which participants can use to enhance their model's understanding of the answer's correctness

2.4 **is_correct:** The target variable for this competition, indicating whether the answer provided is correct (True) or incorrect (False)

3 Model Description

Llama 3.1 8B is the latest iteration of Meta's smaller language model, offering a balance between performance and efficiency. Released on July 23, 2024, this model features significant improvements over its predecessors [2].

3.1 Expanded context length:

It supports a 128K token context window, a substantial increase from previous versions

3.2 Multilingual capabilities:

The model can handle input and output in eight languages, including English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai

3.3 Improved reasoning:

It demonstrates enhanced reasoning capabilities and stronger performance across various benchmarks

3.4 Efficient resource usage:

With 8 billion parameters, it's suitable for scenarios with limited computational power, making it ideal for edge devices and faster training times.

3.5 Versatility:

Llama 3.1 8B excels at tasks such as text summarization, classification, sentiment analysis, and language translation

3.6 Fine-tuning support:

The model can be customized for specific tasks and domains

Rank (r)	α	Lora Decay	Max Steps	Weight Decay	Learning rate	Batch Size	Loss Value
64	16	0.2	300	0.01	1e-5	16	0.6434
64	32	0.2	500	0.01	1e-5	16	0.6814
128	128	0.0	500	0.01	1e-5	16	0.61
64	128	0.0	2000	0.01	5e-5	16	0.51
64	128	0.0	2000	0.01	5e-5	32	0.43
64*	128	0.0	2500	0.01	5e-5	32	0.41
64	128	0.0	1800	0.01	5e-5	32	0.31

Table 1: Experiments conducted using different hyperparameter configurations. *During the analysis, we noticed that parameters resulting is low loss value did not always perform well on test dataset (**Overfitting**).* row gave the best result on training dataset.*

4 Hyperparameter Description

Fine-tuning Llama models involves adjusting several hyperparameters to optimize performance.

Key hyperparameters include:

4.1 Learning Rate:

Controls the speed at which the model updates its weights.

4.2 Batch Size:

Determines the number of samples processed before updating the model.

4.3 Epochs:

The number of times the entire training dataset is processed.

4.4 Weight Decay:

Regularization parameter to prevent overfitting.

4.5 Max Steps:

Refers to the maximum number of training steps during the optimization process.
$Max\_steps = \frac{Total\_training\_examples}{Batch\_size} * Epochs$

4.6 LoRA (Low-Rank Adaptation):

Method for parameter efficient fine-tuning (PEFT) of large language models. It modifies only a small subset of the model's parameters, which reduces the computational and memory requirements compared to full fine-tuning.

The key LoRA parameters are:

4.6.1 Rank (r):

Determines the size of the low-rank matrices used to adapt the model's weights

4.6.2 Alpha (α):

A scaling factor that amplifies the output of the LoRA module

4.6.3 Dropout (p):

The dropout rate applied to the LoRA modules during training

4.6.4 Target Modules:

Specifies which layers of the original model will be adapted by LoRA

4.6.5 Learning Rate:

Controls how quickly the LoRA parameters are updated during fine-tuning

4.6.6 LoRA Decay:

Regularization parameter in LoRA to prevent overfitting.

5 Hyperparameter Setting:

Various hyperparameter values were set for tuning Llama3-8B model and the result of same are mentioned in **Table 1.**.

6 Prompt Setting:

While setting the prompt, our first approach was to go with the *Chain Of Thought* prompting[3]. A technique were we specify the model to reason on very step of the way to come to a result and also provide it with an example to show the chain of thought to be followed.
The second approach is generic were we just mention the model that it has solve some query and we do not provide any example for reference.

6.1 (Chain of Thought) Example:
""Have the following math question. Understand the question, and reason step by

step. Look at the solution and answer and verify whether it is correct on incorrect as True or False respectively. Here's an example to understand: Question: Solve for x in the equation 3x + 5 = 20.
Solution: Subtracting 5 from both sides, we get 3x = 15. Then, dividing by 3, we find x = 5.
Answer:
5
Output:
True
Now, evaluate the following:""

The following (*Chain of Thought*) prompt led to a model with a very high accuracy on the training data set but on the test data set it performed poorly.

6.2 (Generic) Example:
""As a renowned mathematician, determine if the given answer to the math question is correct. Respond strictly with 'True' or 'False' and provide no additional explanation.""

The following (*Generic*) prompt led to a model with a worse score than the (*Chain of Thought*) prompt on the train dataset but performed way better on the test dataset.

7 Results and conclusion

The final fine-tuned model achieved a test set accuracy of **0.86776** on the test set. In future, a comparative study in terms of performance and efficiency of different LoRA implementations like Adapter-Fusion, Prefix Tuning and QLoRA can be done to arrive at a final conclusion of the model characteristics for this particular task.

8 References:

[1] Wolfe, C. (2023, September 23). Understanding and Using Supervised Fine-Tuning (SFT) for Language Models https://cameronrwolfe.substack.com/p/understanding-and-using-supervised
[2] meta-llama/Llama-3.1-8B · Hugging Face. (2024, July 23). https://huggingface.co/meta-llama/Llama-3.1-8B
[3] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-Thought prompting elicits reasoning in large language models. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2201.11903