Training Llama-2 on Simple, Safe Text

September 13, 2023

1 Introduction

In this report we'll summarize the main results we have obtained on training a Llama-2 model to produce safe and simple text. Please feel free to add the results of any experiments.

2 Dataset Analysis

2.1 Filtering

2.2 EDA

3 Model and Training

4 SFT Inference Results

In this section we will summarize the results of Llama-2 models fine-tuned using supervised fine-tuning.

4.1 HuggingFace Leaderboard

First I'll summarize the results of our fine-tuned models versus the Meta-Llama models on the Huggingface Open LLM leaderboard. The leaderboard is a wrapper for the "Eleuther AI Language Model Evaluation Harness". Specifically, the leaderboard measures the 25-shot performance of LLMs on the arc-challenge dataset, the 10-shot performance on the HellaSwag dataset, the 0-shot performance on TurthfulQA, and the 5-shot performance on MMLU. For arc-challenge and HellaSwag performance is measured using the acc_norm metric of EleutherAI, for TruthfulQA they use the mc2 metric, and for MMLU they average the accuracy of the model across tasks in the MMLU dataset. The nice thing about the leaderboard is one can submit either the full model or submit just the adapter layers and give the base model separately. The problem is, sometimes models disappear from the leaderboard and I'm not sure why. Here Llama-2-7b is the base 7 billion model and Llama-

Model	Average	Arc	HellaSwag	MMLU	TruthfulQA
Llama-2-7b	53.40	53.07	77.74	43.80	38.98
Llama-2-7b-chat	**56.34**	52.90	78.55	**48.32**	**45.57**
Llama-2-7b-eli5	53.92	53.41	77.90	43.56	40.81
Llama-2-7b-wiki	53.72	**54.35**	78.06	45.35	37.11
Llama-2-7b-eli5-wiki	55.46	53.75	**78.76**	46.02	43.31
Llama-2-13b	56.90	58.11	80.97	54.34	34.17
Llama-2-13b-chat	59.93	59.04	81.94	54.64	**44.12**
Llama-2-13b-eli5	**60.61**	**60.41**	**82.58**	**55.86**	43.61
Llama-2-13b-wiki	58.12	59.04	82.33	55.36	35.75
Llama-2-13b-eli5-wiki	59.43	59.98	82.43	55.41	39.90

Table 1: Results for Llama-2 models on Huggingface Open LLM Leaderboard

.

2-7b-chat is the Llama-2 model which has undergone RLHF. The remaining models are fine-tuned versions of the base-model. Specifically, they are trained using SFT and QLORA on the ELI5 SFT dataset, the Simple Wikipedia Instruct dataset, or their combination. The second half of the table is the same, except for the 13B parameter model.

When defining the model there are subtleties about how to merge the LoRA adapter weights with the rest of the model. The subtlety arises because in QLoRA we quantize the base model to 4-bit, but need to dequantize these weights to bfloat16 when performing back-propagation for the LoRA adapter layers. When we merge the adapter layers with the base model we have two natural options: either quantize the model to bfloat16 and then merge or quantize the model to 4-bit, dequantize to bfloat16, and then merge. The second option is arguably more natural since during training we are quantizing and then dequantizing, so we want the model at inference to be as close as possible to the model during training. However, when we quantize and then dequantize we risk losing precision and degrading the model in the process. It is not clear which choice is better and this likely depends on how strong of an effect the LoRA layers have and how sensitive they are to the exact form of weights. On automated benchmarks we have not seen one method give reliably better results than the other.

All that said, to get the above results we quantized and dequantized the 7B model before merging, while for the 13B model we quantized the model to bfloat16 and then merged. For the 7B model we could perform the quantization and dequantization on a 40GB A100. For the 13B model we directly submitted the adapter layers to the HuggingFace leaderboard and used the ungated Llama-2-7b-hf model from NousResearch (the Meta-Llama model is gated and although it can be downloaded, we were not able to use it as a base model on the leaderboard).

In Table **??** we see that of the 7B models, the Meta-Llama-2-7b-chat performs the best

on average with the Llama-2 model trained on ELI5 and Simple Wikipedia performing the second best. One surprising thing is the chat model actually performs worse than the base model on the Arc-Challenge dataset, and here the model trained on just Simple Wikipedia QA pairs performs the best. On the HellaSwag dataset the Llama-2-7B model trained on ELI5 and Simple Wikipedia marginally outperforms the chat model, but the difference is likely too small to be statistically significant. Finally, on MMLU and TruthfulQA the 7B chat model performs significantly better than the other models.

Once we go up to 13B parameters we see that the Llama-2 model trained on the ELI5 SFT dataset performs the best on average with the 13B chat model and 13B model fine-tuned on ELI5 + Simple Wikipedia close behind. It is surprising that the model trained on just ELI5 model performs the best, and this result is driven primarily by its improved performance on the TruthfulQA dataset. On the other datasets it barely improves over the model trained on the combined ELI5 and Simple Wikipedia dataset. We are not sure what the cause of this effect is, somehow training the model more on Reddit data makes the model more honest! This result could also be an artifact of a poor choice of hyperparameters, and perhaps with a different learning rate and/or after averaging over initializations the difference would go away or the model trained on the combined dataset would perform better.

4.2 ROUGE and BERTScore

In this section we will investigate how supervised fine-tuning effects the models ROUGE and BERTScores. ROUGE is a well-known automated benchmark that measures n-gam overlap between generated text and the reference text. Since it just looks at n-gram overlaps, and does not take into account semantic content, ROUGE is effectively measuring to what extent the trained model is adopting the vocabulary of the reference text. One advantage of BERTScore is it uses pre-trained encoder models, such as BERT or RoBERTa, to encode the generated and reference text in some high-dimensional vector space and then measures the cosine-similarity between the two vectors. Of course, this also means that BERTScore is more computationally intensive to compute.

In this section we will only perform inference on a small subset of the validation set, for each validation set we randomly sample 100 question and generate the answers. In addition, for each model we will both merging procedures, either quantizing to BF16 and then merging, or quantizing to 4-bits, dequantizing, and then merging.

4.2.1 Quantizing and Dequantizing

In the tables below we present the ROUGE and BERTScore metrics for the Llama-2 base model, as well as our fine-tuned models, on the (small) validation sets for our three datasets: ELI5, Simple Wikipedia, and combined dataset. We include the original Llama-2 model as a baseline to measure how much fine-tuning changes the Rouge and BERTScores. In

addition, we also include "off-diagonal" elements, where we train a model on one dataset and measure its ROUGE and BERTScores on the validation split of a different dataset. For example, we include cases where we train the model on the ELI5 SFT dataset and then evaluate it on the validation split of the Simple Wikipedia QA dataset. We included these results to serve as additional baselines to see to what extend fine-tuning on *any* QA dataset changes the evaluation metrics. In all cases, the model produces at most 256 new tokens.

Dataset	**Model**	**rouge1**	**rouge2**	**rougeL**	**rougeLsum**
ELI5	Llama-2-7b	**0.3796**	**0.2432**	**0.3000**	**0.3222**
	Llama-2-7b-ELI5	0.3701	0.2140	0.2736	0.2821
	Llama-2-7b-wiki	0.3575	0.2083	0.2660	0.2762
	Llama-2-7b-ELI5-wiki	0.3702	0.2126	0.2733	0.2819
wiki	Llama-2-7b	0.1923	0.0103	0.0937	**0.1392**
	Llama-2-7b-ELI5	**0.2203**	**0.0125**	**0.0964**	0.1271
	Llama-2-7b-wiki	0.1826	0.0073	0.0879	0.1165
	Llama-2-7b-ELI5-wiki	0.1811	0.0076	0.0885	0.1153
full	Llama-2-7b	0.1944	0.0087	0.0905	**0.1400**
	Llama-2-7b-ELI5	**0.2243**	**0.0118**	**0.0971**	0.1312
	Llama-2-7b-wiki	0.1905	0.0079	0.0889	0.1198
	Llama-2-7b-ELI5-wiki	0.1907	0.0080	0.0894	0.1193

Table 2: Rouge Scores

It's difficult to interpret or make sense of these results, somehow the base Llama-2 model often outperforms the fine-tuned models and the model trained on just the ELI5 dataset often performs very well on the Simple Wikipedia validation set! It is probably best to take these numbers with a grain of salt. The BERTScore metrics tend to differ by very small amounts, e.g. the difference in F1 scores between the base model and the model trained on ELI5 for the Simple Wikipedia validation set differ by just 0.001. In addition, we of course do not know on what data the original Llama-2 model was trained and if there is data leakage.

Finally, let's look at the Flesch readability ease (FRE) and Flesch-Kincaid grade metrics. We see that for the most part, the fine-tuned models have a higher readability score and a lower grade level. There are two noticeable exceptions, the model trained on Simple Wikipedia QA pairs and evaluated on the ELI5 validation set has a higher grade level than the original base model. In addition, the model trained on ELI5 and Simple Wikipedia has a slightly higher grade level than the base model when evaluated on the Simple Wikipedia validation set. In the former case, the grade level went up by 0.451 while the readability also went up by 0.472. Given that a text is considered simpler if its grade level is lower

Dataset	Model	Precision	Recall	F1
eli5	Llama-2-7b	**0.8429**	0.8754	0.8583
	Llama-7b-ELI5	0.8372	**0.8799**	0.8575
	Llama-2-7b-wiki	0.8415	0.8796	**0.8598**
	Llama-2-7b-ELI5-wiki	0.8399	0.8798	0.8590
wiki	Llama-2-7b	**0.7879**	0.8067	0.7970
	Llama-7b-ELI5	0.7859	**0.8093**	**0.7971**
	Llama-2-7b-wiki	0.7782	0.8019	0.7898
	Llama-2-7b-ELI5-wiki	0.7783	0.8014	0.7896
full	Llama-2-7b	**0.7940**	0.8092	0.8014
	Llama-7b-ELI5	0.7933	**0.8116**	**0.8022**
	Llama-2-7b-wiki	0.7861	0.8048	0.7952
	Llama-2-7b-ELI5-wiki	0.7841	0.8046	0.7941

Table 3: Precision, Recall, and F1 BERTScores

and its readability is higher, in this case our automatic metrics are inconclusive. In the latter case, the readability also went up by about 1.6518 points while the grade level went up by 0.067. Once again, in this case our metrics for readability are moving in opposite directions. Finally, from the table we also see that the model trained on just ELI5 tends to produce the simplest text.

4.2.2 Simple Merging

4.3 MT(S)-Bench

In this section we will investigate how well our models perform on a new MT-bench like dataset. The idea of MT-bench is to use a LLM, like GPT-4, to judge the output of other, smaller LLMs. There are two types of completion tasks, single-turn and multi-turn. For a single-turn problem, the model is given a prompt and asked to complete it. For a multi-turn problem, the model is prompted with one complete turn of a conversation (i.e. question and answer) and then asked to reply to a new, follow-up question. Given that our models are only trained for single-turn problems, we will see that they do not perform well on multi-turn problems.

Dataset	Model	Flesch Readability Scores	Flesch Kincaid Grade Level
eli5	Llama-2-7b	58.2425	10.075
	Llama-2-7b-eli5	67.6357	8.148
	Llama-2-7b-wiki	58.7145	10.526
	Llama-2-7b-eli5-wiki	65.6862	8.704
wiki	Llama-2-7b-hf	64.1040	8.172
	Llama-2-7b-eli5	72.5682	6.988
	Llama-2-7b-wiki	66.5824	8.055
	Llama-2-7b-eli5-wiki	65.7558	8.239
full	Llama-2-7b	65.0452	8.472
	Llama-2-7b-SFT-eli5	72.5278	7.092
	Llama-2-7b-SFT-wiki	66.8147	8.252
	Llama-2-7b-SFT-eli5-wiki	65.8295	8.417

Table 4: Flesch Readability Metrics