Llama3 & DeepSeek v3

Architecture | Fine-tuning | Inference

Dr. Karthik Mohan, Feb 26 2025

Today's Talk

1. Types of Training

2. Llama3

3. DeepSeekV3

4. Notebook Walkthrough

Types of LLM training

① Pre-Training

Foundation Model/Pre-Trained Models

② Post-Training

Post-Training

Types of LLM training

① Pre-training

Foundation Model/Pre-Trained Models

→ Next lecture

② Post-training

Supervised Fine-tuning

Instruction Fine-tuning

Reinforcement Learning with Human Feedback (RLHF)/Direct Preference Optimization (DPO)

Types of LLM training

Foundation Model/Pre-Trained Models

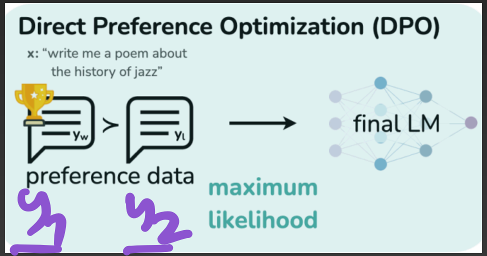
Supervised Fine-tuning

Instruction Fine-tuning

Reinforcement Learning with Human Feedback (RLHF)/Direct Preference Optimization (DPO)

Popularity of GPT models GPT-3

Post-Training


Direct Preference Optimization (DPO)
x: "write me a poem about the history of jazz"
y_w
>
y_l
preference data
maximum likelihood
final LM
y_1
y_2

ICE #0

Why not just do a single training instead of multiple trainings for LLMs?

1. Single training won’t work
2. Multiple trainings also different sources of high quality vs medium quality data
3. Single Training has no foundation to build on
4. Multiple Training is faster

Pre-Trained vs Instruct Model

Pre-Trained Models are trained as a Masked Language Model and for Next Token Prediction or Multiple Token Prediction

Instruct Fine-tuning is fine-tuning a pre-trained model to follow instructions

Pre-Trained vs Instruct Model

Pre-Trained Models are trained as a Masked Language Model and for Next Token Prediction or Multiple Token Prediction

Instruct Fine-tuning is fine-tuning a pre-trained model to follow instructions on a wide variety of tasks

Pre-Trained vs Instruct Fine-tuned Model

Pre-Trained

Input: “The red fox ___”
Output: “chased”
Input: “The red fox chased the ___”
Output: “blue”
Input: “The red fox chased the blue ___”
Output: “bird”

Instruct Fine-tuned

Input: “You are to complete the following sentence. Sentence: ‘The red fox ’ ”
Output: “The red fox chased the blue bird. And the bird flew away in the nick of time!”

Instruct Fine-tuning vs Supervised Fine-Tuning

Instruct Fine-tuning is fine-tuning a pre-trained model to follow instructions on a wide variety of tasks

Supervised Fine-tuning is fine-tuning the pre-trained LLM on specific tasks: Sentiment analysis, text summarization, question answering, intent detection, etc

Instruct Fine-tuning vs Supervised Fine-Tuning

Instruct Fine-tuning is fine-tuning a pre-trained model to follow instructions on a wide variety of tasks

Supervised Fine-tuning is fine-tuning the pre-trained LLM on specific tasks: Sentiment analysis, text summarization, question answering, intent detection, etc

SFT vs Instruct Fine-tuned Model

Supervised Fine-tuning

Input: “I am not feeling good today”
Output: “Unhappy”

Sentiment Analysis

Input: “I would love to go to New York and spend time on Times Square”
Output: “New York, Times Square”

Named Entity Recognition

Input: “What is the tallest mountain in the world?”
Output: “Mount Everest”

Q&A

Instruct Fine-tuning

Input: “You are to complete the following sentence. Sentence: ‘The red fox ’ ”
Output: “The red fox chased the blue bird. And the bird flew away in the nick of time!”

Instruct Fine-tuning vs Supervised Fine-Tuning

SFT trains a pre-trained LLM to do well on specific NLP tasks.

Instruction Fine-tuning looks at the ability of an LLM to follow instructions on variety of tasks. There is no clear indication of task + makes the LLM behave more like an AI agent

ICE #1

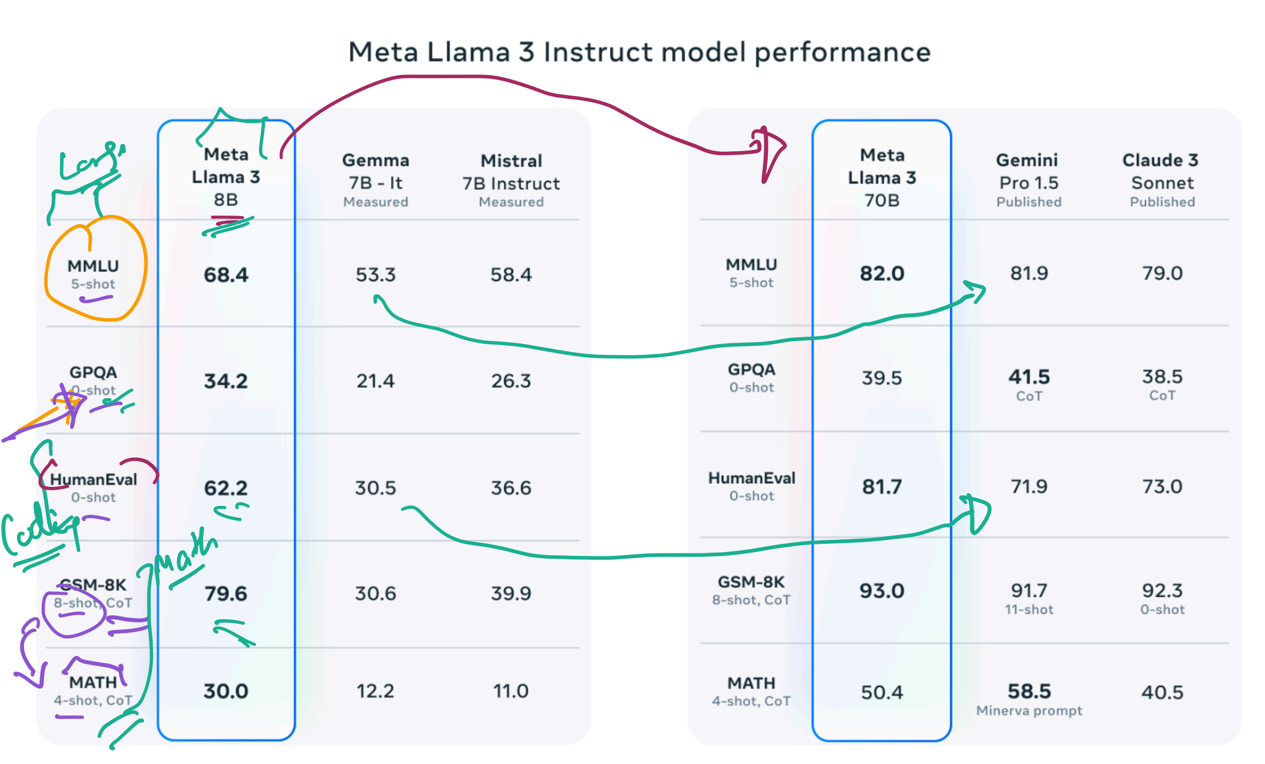
What is the loss function for SFT?

1. Cross-entropy
2. Quadratic loss

What is the loss function for Instruct models?

1. Cross-entropy
2. Quadratic loss

Llama 3 Instruct Performance



Meta Llama 3 Instruct model performance

	Meta Llama 3 8B	Gemma 7B - It Measured	Mistral 7B Instruct Measured
MMLU 5-shot	**68.4**	53.3	58.4
GPQA 0-shot	**34.2**	21.4	26.3
HumanEval 0-shot	**62.2**	30.5	36.6
GSM-8K 8-shot, CoT	**79.6**	30.6	39.9
MATH 4-shot, CoT	**30.0**	12.2	11.0

	Meta Llama 3 70B	Gemini Pro 1.5 Published	Claude 3 Sonnet Published
MMLU 5-shot	**82.0**	81.9	79.0
GPQA 0-shot	39.5	**41.5** CoT	38.5 CoT
HumanEval 0-shot	**81.7**	71.9	73.0
GSM-8K 8-shot, CoT	**93.0**	91.7 11-shot	92.3 0-shot
MATH 4-shot, CoT	50.4	**58.5** Minerva prompt	40.5

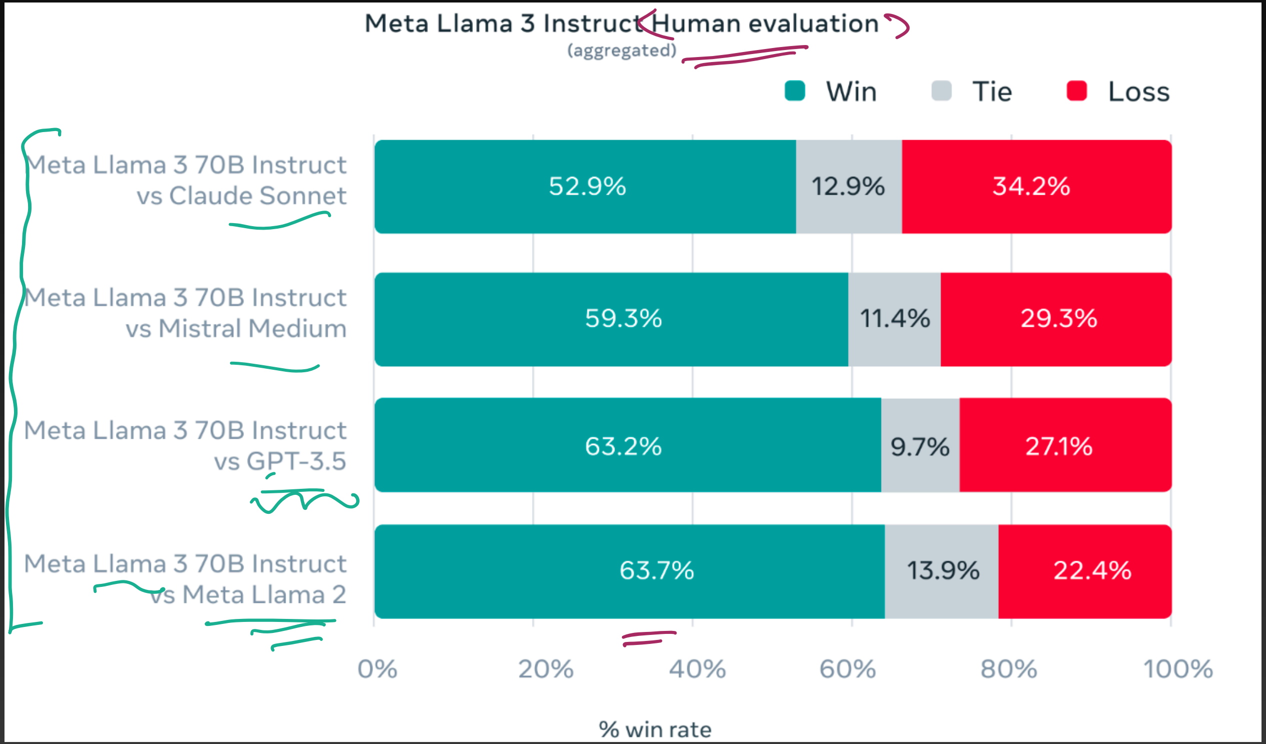
Large

Coding

Math

Reference: https://ai.meta.com/blog/meta-llama-3/

Llama 3 Instruct Performance



Reference: https://ai.meta.com/blog/meta-llama-3/

ICE #2

If you had to fine-tune a LLM model that can detect an emotion from a review - which would you pick?

1. Fine-tune an instruct model ✓
2. Fine-tune the pre-trained model → More useful for a company/org build their custom foundation + fine-tuned model.
3. Fine-tune the SFT model ✓

Llama2 vs Llama3

7 times larger pre-train data set. **15 Trillion Tokens** of data ~ 150 million books

High-quality filters to filter out bad data in training - Use Llama2

Better “data mix” - Trivia, STEM, coding, historical knowledge

Larger model means better performance (8B vs 70B)

But more data = better performance (also avoids over-fitting). Log-linear improvement from 200B to 15T tokens

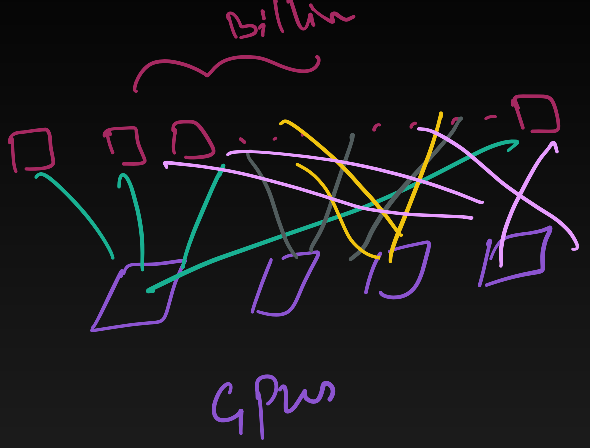
Training Hacks

Data Parallelization

Model Parallelization

Latent Attention

Mixture of Experts



Training Hacks

Data Parallelization

Model Parallelization

Latent Attention

Mixture of Experts

Parallelizing model parameters

H100 GPU

80GB Ram

70B parameter model

→ 280GB

#GPUs needed ≥ 4

Training Hacks

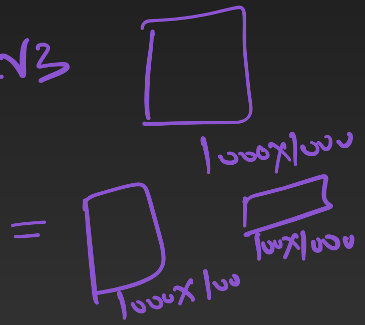
Data Parallelization

Model Parallelization

Latent Attention

Mixture of Experts

→ Deepseek V3



Training Hacks

Data Parallelization

Model Parallelization

Latent Attention

Mixture of Experts

⤳ used in Deepseek v3

ICE #3

In the model parallelism regime - Assume a new **DeepFetch** model got released on the market with 1 trillion parameters. Assume that for pre-training, you are using H100 GPUs with 80 GB ram. How many GPUs would you need to have to hold the model in memory?

1. **25**
2. **50**
3. **75**
4. **100**

1 Trillion paramt

$\simeq$ 4TB

Need $\geq \frac{4000}{80}$

$= 50$ GPUs!