Llama3 & DeepSeek v3 (Part 1)

Architecture | Fine-tuning | Inference

Dr. Karthik Mohan, Mar 2 2026

Today's Talk

1. Types of Training

2. Llama3

3. DeepSeekV3

4. Notebook Walkthrough

Types of LLM training

Foundation Model/Pre-Trained Models

Post-Training

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)

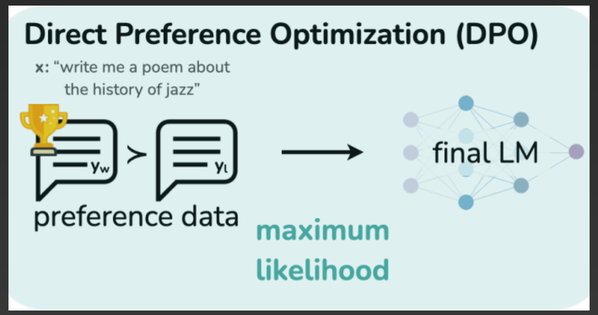
Types of LLM training

Foundation Model/Pre-Trained Models

Supervised Fine-tuning

Instruction Fine-tuning

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



ICE #0

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

1. Single training won’t work
2. Multiple trainings uses different sources of high quality vs medium quality data
3. Single Training has no foundation to build on
4. Multiple training breaks down overall objectives into multiple stages
5. Multiple Training is faster

Pre-Trained vs Instruct Model

Pre-Trained Models are trained 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!”

Pre-Trained Data Mix

Data mix

* Web text
* Code
* Math
* Reasoning-heavy corpus
* Synthetic Data
* Multi-lingual Data

ICE #2

Which of the following statements are true

1. BERT training uses Masked Language Model objective
2. Llama3 Pre-training uses Causal LM objective
3. Llama3 Pre-training uses Next Token Prediction
4. Llama3 Pre-Training uses Masked LM
5. BERT training uses Causal LM objective

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”
Input: “I would love to go to New York and spend time on Times Square”
Output: “New York, Times Square”
Input: “What is the tallest mountain in the world?”
Output: “Mount Everest”

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 #3

What is the loss function for SFT?

1. Cross-entropy
2. Quadratic loss

Llama 3 Instruct Performance

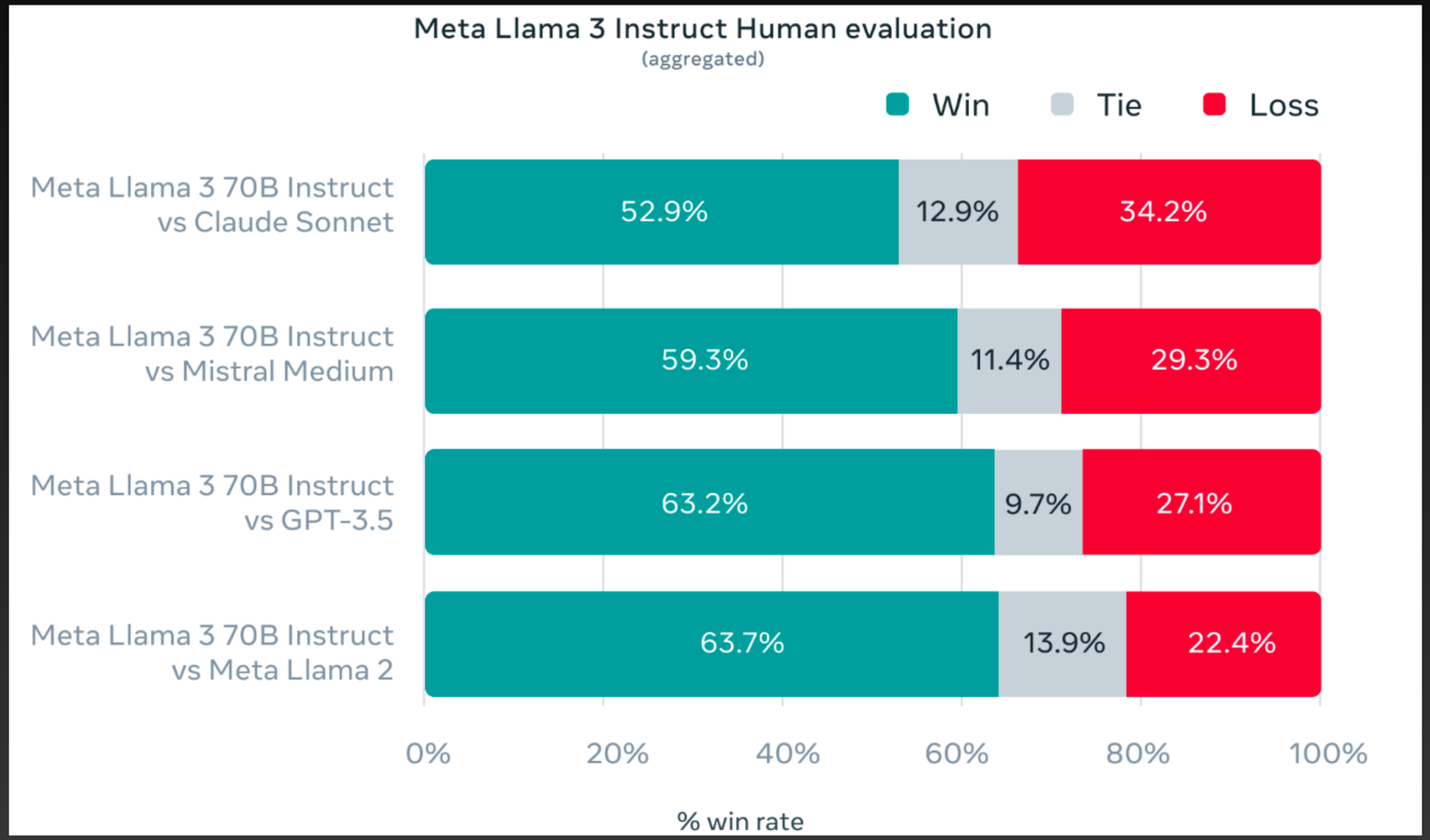
Meta Llama 3 Instruct model performance

	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

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

Llama 3 Instruct Performance



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

ICE #4

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
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

Llama3 Herd

Herd of models including 405B LM, 70B, 8B, 1B versions and also Llama Guard 3 for input/output safety

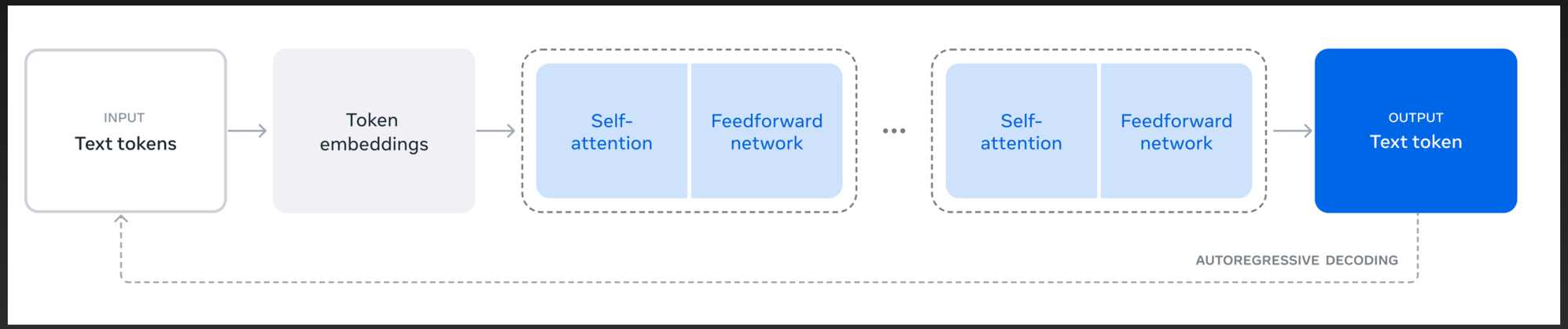
Reference: https://arxiv.org/pdf/2407.21783

Llama 3 Herd of Models

	Finetuned	Multilingual	Long context	Tool use	Release
Llama 3 8B	✗	✗[1]	✗	✗	April 2024
Llama 3 8B Instruct	✓	✗	✗	✗	April 2024
Llama 3 70B	✗	✗[1]	✗	✗	April 2024
Llama 3 70B Instruct	✓	✗	✗	✗	April 2024
Llama 3.1 8B	✗	✓	✓	✗	July 2024
Llama 3.1 8B Instruct	✓	✓	✓	✓	July 2024
Llama 3.1 70B	✗	✓	✓	✗	July 2024
Llama 3.1 70B Instruct	✓	✓	✓	✓	July 2024
Llama 3.1 405B	✗	✓	✓	✗	July 2024
Llama 3.1 405B Instruct	✓	✓	✓	✓	July 2024

Reference: https://arxiv.org/pdf/2407.21783

Llama3 Architecture

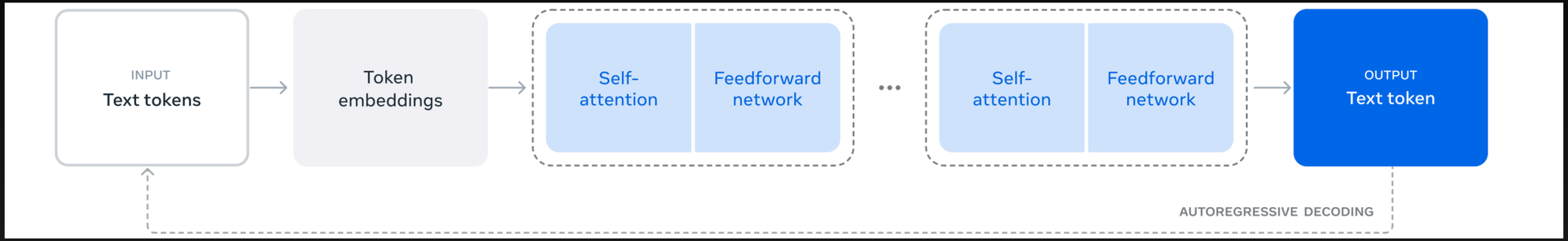


Tokenization

```
def print_tokens_with_ids(txt):
    tokens = tokenizer.tokenize(txt, add_special_tokens=False)
    token_ids = tokenizer.encode(txt, add_special_tokens=False)
    print(list(zip(tokens, token_ids)))

prompt = """<|begin_of_text|><|start_header_id|>system<|end_header_id|>

Based on the information provided, rewrite the sentence by changing its tense fro

She played the piano beautifully for hours and then stopped as it was midnight.<|

"""
print_tokens_with_ids(prompt)

# Token and Token ID
> [('<|begin_of_text|>', 128000), ('<|start_header_id|>', 128006), ('system', 912
```

```
 [128000, 128000, 128006,   9125, 128007,    271,  29815,    389,    279,
           2038,   3984,     11,  18622,    279,  11914,    555,  10223,   1202,
          43787,    505,   3347,    311,   3938,     13, 128009, 128006,    882,
         128007,    271,   8100,   6476,    279,  27374,  32719,    369,   4207,
            323,   1243,  10717,    439,    433,    574,  33433,     13, 128009,
         128006,  78191, 128007,    271]
```

ICE #5



Assume that Llama4 is trained on 40T tokens of data. It has a context window of 256k tokens and has a tokenizer vocab size of 108k tokens and each token has a token embedding size of 4096.
What is the number of classes present in the classifier of the LM head to generate the next token in auto-regressive decoding?

a) 256k
b) 40T
c) 108k
d) 128k
e) 4096

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

* Each GPU has full copy of the model
* Different GPUs work on different mini-batches of data
* Speed ups due to parallel computation of gradients
* Gradients averaged across GPUs

Training Hacks

Data Parallelization

Model Parallelization

Latent Attention

Mixture of Experts

* Tensors are split across GPUs
* Same mini-batch is passed through each GPU
* Useful when the model is quite large to fit on one GPU
* Communication over-heads

Training Hacks

Data Parallelization

Model Parallelization

Latent Attention

Mixture of Experts

Training Hacks

Data Parallelization

Model Parallelization

Latent Attention

Mixture of Experts

ICE #6

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