Lecture 4: train a LLM

谢丹
清华大学数学系

2025/08

Two-Stage LLM Training Pipeline

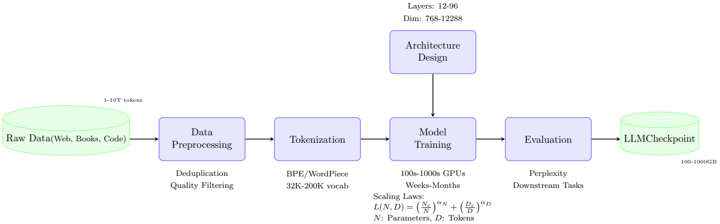
1. Pre-Training Phase

- **Objective**: Learn general language understanding and generation
- **Data**: Massive text corpora (1-10 trillion tokens)
- **Method**: Unsupervised learning via next-token prediction
- **Outcome**:
  - Base model with world knowledge
  - Text generation capability
  - Fundamental language understanding

2. Post-Training Phase

- **Supervised Fine-Tuning (SFT)**:
  - Uses labeled instruction datasets
  - Aligns model with human preferences
  - Enables task-specific behaviors
- **Reinforcement Learning (RLHF)**:
  - Further refines model outputs
  - Uses human/AI feedback signals
  - Optimizes for helpfulness/safety
- **Reinforcement Learning to gain reasoning capabilities**:

LLM Pre-Training Pipeline



Step 1: Data Collection

Key Data Sources

- Web pages (Common Crawl, Wikipedia)
- Books (Project Gutenberg, proprietary collections)
- Scientific papers (arXiv, PubMed)
- Code repositories (GitHub)
- Dialogue data (for conversational ability)

Data Volume

- Typically 1-10 trillion tokens
- GPT-3: 300B tokens
- LLaMA 2: 2T tokens; LLaMA 3: 15T tokens;
- Deepseek V3: 14.8T tokens
- Qwen 3: 36T tokens

Step 2: Data Preprocessing

Cleaning Steps

- Deduplication
- Quality filtering
- Toxicity removal
- Language identification
- PII redaction

Data quality is very important.

Tokenization

- Subword algorithms:
  - BPE (GPT series)
  - WordPiece (BERT)
  - Unigram (SentencePiece)
- Vocabulary size: 32K-200K

Step 3: Model Architecture

Transformer Specifications

- Decoder-only architecture (GPT-style)
- Key hyperparameters:
  - Layers: 12-96
  - Hidden dim: 768-12288
  - Attention heads: 12-128
  - Context window: 2K-32K tokens

Example Configurations

Model	Layers	Dim	Params
GPT-3	96	12288	175B
LLaMA 2	80	8192	70B
PaLM	118	18432	540B

	8B	70B	405B
Layers	32	80	126
Model Dimension	4,096	8192	16,384
FFN Dimension	14,336	28,672	53,248
Attention Heads	32	64	128
Key/Value Heads	8	8	8
Peak Learning Rate	3×10^{-4}	1.5×10^{-4}	8×10^{-5}
Activation Function	SwiGLU		
Vocabulary Size	128,000		
Positional Embeddings	RoPE ($\theta = 500,000$)		

Figure: Architecture for LLaMa3

Step 4: Training Objectives

Primary Objective

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \log P(x_t | x_{<t}; \theta)$$

Autoregressive language modeling (next token prediction)

Common Variants

- Causal masking (left-to-right)
- Fill-in-the-middle (for code models)
- Mixed objective (sometimes with span corruption)

Step 5: Optimization

Key Techniques

- AdamW optimizer
- Learning rate warmup
- Cosine decay schedule
- Gradient clipping
- Mixed precision training

Challenges

- Batch size: 1M-10M tokens
- Hardware: 100s-1000s GPUs/TPUs
- Training time: Weeks-months
- Memory optimization:
  - ZeRO
  - Pipeline parallelism
  - Tensor parallelism

Step 6: Evaluation & Scaling

Training Monitoring

- Loss curves
- Perplexity
- Downstream task performance
- Zero-shot capabilities

Scaling Laws

$$L(N, D) = \left(\frac{N_c}{N}\right)^{\alpha_N} + \left(\frac{D_c}{D}\right)^{\alpha_D}$$

- N: Model parameters
- D: Training tokens
- α_N, α_D: Scaling exponents

Final Pre-Training Output

Model Checkpoint	100-1000GB
Vocabulary	1-10MB
Training Logs	Comprehensive

Zero-Shot Learning in Large Language Models

Definition

- **Zero-shot**: Model performs tasks *without* task-specific training examples
- Leverages only:
  - Pretrained knowledge
  - Task description in prompt
  - General reasoning ability

Mechanism

$$P(\text{output}|\text{task description}, \text{input})$$

- No gradient updates or fine-tuning
- Pure inference-time adaptation
- Relies on model's pretrained representations

Examples & Applications

Text Classification

Prompt:

> *"Classify this tweet sentiment:*
> *'I love this product!'*
> *Options: [positive, negative, neutral]"*

Output: "positive"

Key Advantages

- No need for labeled data
- Instant task adaptation
- Broad task generalization

Question Answering

Prompt:

> *"Q: What's the capital of France?*
> *A:"*

Output: "Paris"

Technical Foundations

What Enables Zero-Shot?

- **Scale**: Massive pretraining data coverage
- **Architecture**: Transformer's attention mechanism
- **Objectives**: Causal/MLM pretraining

Limitations

- Performance $\ll$ fine-tuned models
- Sensitive to prompt phrasing
- May generate plausible but wrong answers

Post-Training: The Alignment Phase

Three Key Stages

1. **Supervised Fine-Tuning (SFT)**
   - Trains on curated (input, output) pairs
   - Adapts base model to follow instructions
   - Requires 10K-100K high-quality examples
2. **Reinforcement Learning from Human Feedback (RLHF)**
   - Learns from preference rankings (good vs bad outputs)
   - Uses reward model trained on human judgments
   - Optimizes with PPO algorithm
3. **Constitutional AI**
   - Applies self-critique against principles
   - Reduces harmful outputs without human labels
   - Uses chain-of-thought feedback

Technical Details

RLHF Mathematics

- Reward modeling:

$$\mathcal{L}_{\text{RM}} = -\mathbb{E}[\log \sigma(r_\theta(y_w) - r_\theta(y_l))]$$

- PPO optimization:

$$\mathcal{L}_{\text{PPO}} = \mathbb{E}[\min(\rho_t \hat{A}_t, \mathsf{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\hat{A}_t)]$$

Emergent Methods

- **DPO**: Direct preference optimization
- **RAFT**: Reward-ranked fine-tuning
- **Self-Play**: Model-as-its-own-judge

Why Post-Train?

- Base LLMs lack:
  - Safety guardrails
  - Instruction following
  - Consistent formatting

Every step involves tuning the parameters of the model!

Reinforcement learning plays an important role.

What is Reinforcement Learning?

- Learning by interaction with an environment
- Goal: Maximize cumulative reward
- No supervised labels—only rewards/penalties

Key Components

- Agent
- Environment
- State (s)
- Action (a)
- Reward (r)

Chain-of-thought

Q: A bookstore had 80 books. They sold 25 on Monday and 30 on Tuesday. How many remain?

A: Let's think step-by-step: 1. Start with 80 books 2. Sold 25 on Monday: 80 - 25 = 55 books left 3. Sold 30 on Tuesday: 55 - 30 = 25 books left Final answer: 25 books remain.

Chain-of-Thought Prompting

- **Goal**: Elicit step-by-step reasoning from LLMs.
- **Methods**:
  - Zero-Shot CoT: Add "Let' s think step by step".
  - Few-Shot CoT: Provide reasoning examples.
- **Impact**: +20% accuracy on math benchmarks (GSM8K).

Example

Q: A bat and a ball cost \$1.10. The bat costs \$1 more than the ball. How much is the ball? **A**: Let the ball cost \$x. The bat costs \$x + 1. Total: $x + (x + 1) = 1.10 \rightarrow 2x = 0.10 \rightarrow x = 0.05$. The ball costs \$0.05.

DeepSeek: Key Innovations

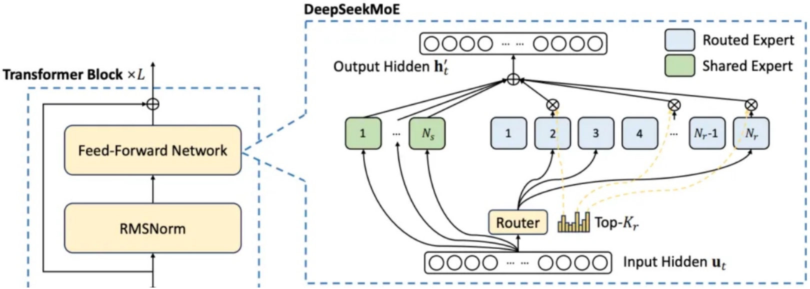
Cutting-Edge Contributions

1. **Multi-Latent Attention (MLA)**: An enhanced attention mechanism improving model efficiency and scalability.
2. **Mixture of Experts (MoE)**: Optimizes training and inference by dynamically routing tasks to specialized subnetworks.
3. **GRPO (Group Relative Policy Optimization)**: A novel RL framework for stable and efficient alignment.

Training Breakthroughs

Pioneering architectures and methodologies for high-performance LLM training.

Modern MOE architecture



Other topics

1. The knowledge of LLM is base on the data it is trained. One can add knowledge by following method
    - Supervised fine tuning or Reinforcement learning using your own data. (Parameter-efficient fine-tuning method)
    - Web search
    - RAG using your own data and transforms it into a data base
2. Tool use and AI agent