

PRESENTATION

How LLMs Actually Work

Team notes from Andrej Karpathy's one-hour talk, Intro to Large Language Models.



Made with MagicSlides

OVERVIEW

Just Two Files

Karpathy opens with Llama 2 70B: an LLM is **a parameters file and a small run file**. The run file can be about 500 lines of C, and the whole thing runs on a laptop with no internet.

Explore the **4 core pillars** that define this topic — each backed by research and real-world insight.

01 Parameters

KEY

70 billion weights at 2 bytes each, a 140 GB file.

02 Run code

CORE

A short program that runs the network on those weights.

03 One task

FOCUS

Predict the next word in a sequence, again and again.

04 Lossy zip

IMPACT

Training compresses internet text into a gestalt, not an exact copy.

What it took to train Llama 2 70B

Numbers Karpathy gives for the expensive first stage.

[↑ Key Metrics](#)

~10 TB

INTERNET TEXT

A crawl of web documents as raw input.

▲ Quantity over quality

6,000

GPUS

A dedicated cluster for one training run.

▲ Specialized hardware

12 days

TRAINING TIME

Continuous training to produce the base model.

▲ Done maybe once a year

~\$2M

COMPUTE COST

Rough cost for this model size, by 2023 standards.

▲ Frontier runs cost far more

Pre-training gives a **document generator**. Fine-tuning turns it into an assistant.



Key Insight

Made with MagicSlides

From base model to assistant

4-Step Framework

01

Pre-train

Learn next-word prediction on a huge internet crawl.

- Tens to hundreds of TB of text
- Low quality, high quantity
- Expensive GPU cluster
- Run rarely

02

Fine-tune

Swap the data for ideal Q and A conversations.

- About 100,000 conversations
- Written by people from labeling instructions
- Quality over quantity
- Cheap, can run weekly

03

Compare

An optional third stage using comparison labels.

- Labelers pick the better answer
- Easier than writing from scratch
- Used in RLHF
- Improves the assistant further

04

Iterate

Collect misbehaviors and fix them in the data.

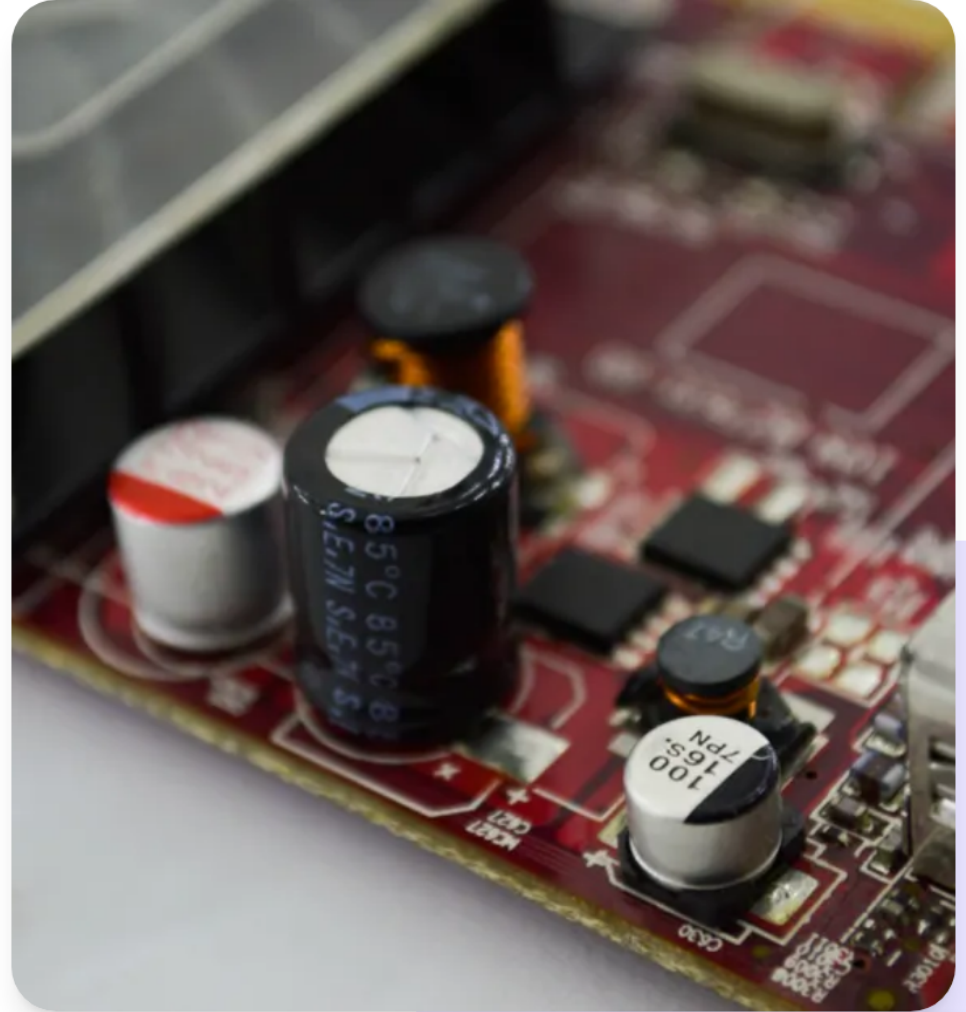
- Deploy and monitor
- Find wrong answers
- Have people write the right response
- Fine-tune again



The LLM as an operating system

Karpathy argues it is more correct to see an LLM as the kernel process of an emerging operating system than as a chatbot. It coordinates memory and tools, like a browser, a calculator or a Python interpreter, to solve problems.

He also points to where research is heading: a System 2 mode that thinks longer for better answers, and AlphaGo-style self-improvement in narrow domains with a clear reward.



SECURITY

New attack surfaces

The last part of the talk covers attacks that only exist because of LLMs. These matter for any team shipping LLM features.

- Evidence-backed insights
- 4 proven advantages
- Ready to implement

1

Jailbreaks

Roleplay or encoded prompts that get the model past its refusals.

NEW

2

Prompt injection

Hidden instructions in images or web pages hijack the model.

CORE

3

Data poisoning

A trigger phrase planted in training data corrupts outputs.

PRO

4

Cat and mouse

Defenses exist for each, and new attacks keep appearing.

KEY



Made with MagicSlides