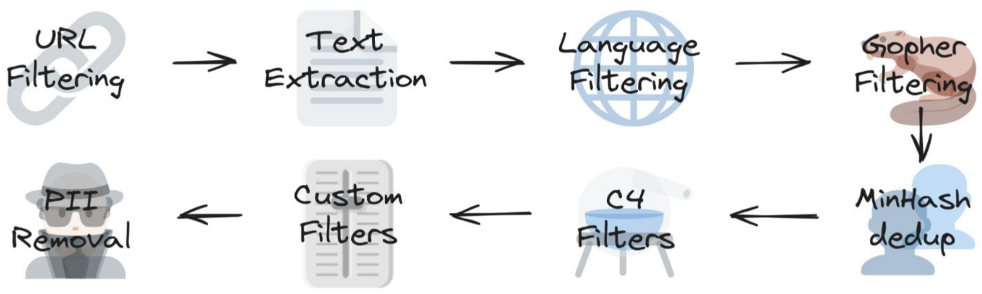
Deep Dive into LLMs like ChatGPT

- Pretraining

Step 1: download and preprocess the internet

https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1



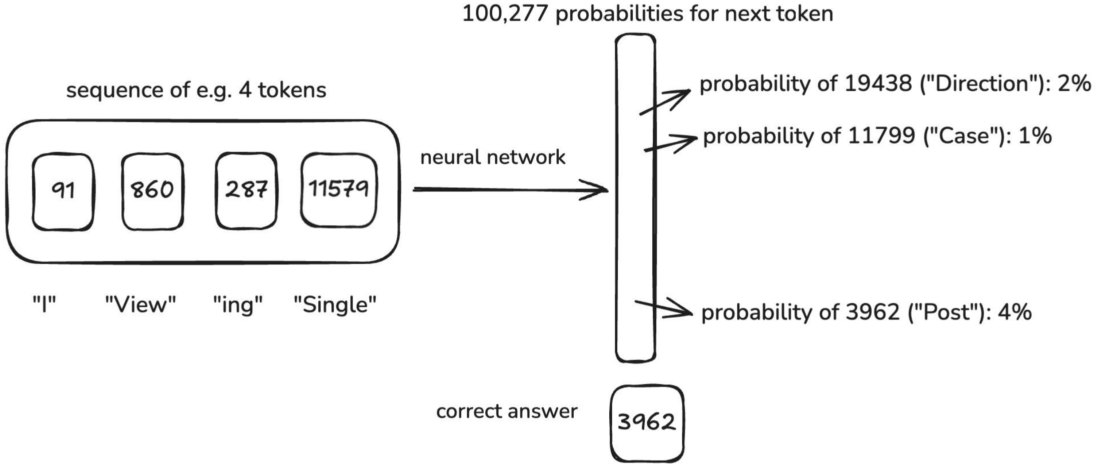
The FineWeb pipeline

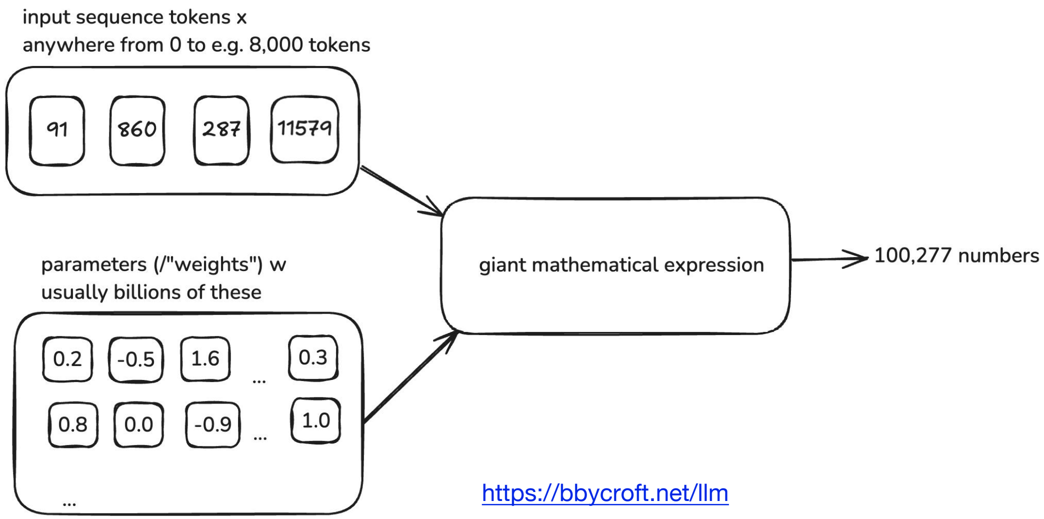
Step 2: tokenization

Convert between raw text into sequences of symbols (/tokens)
example: ~5000 Unicode characters
~= 40,000 bits (2 possible tokens)
~= 5000 bytes (256 possible tokens)
~= 1300 GPT-4 tokens (100,277 possible tokens)

https://tiktokenizer.vercel.app/

Step 3: neural network training



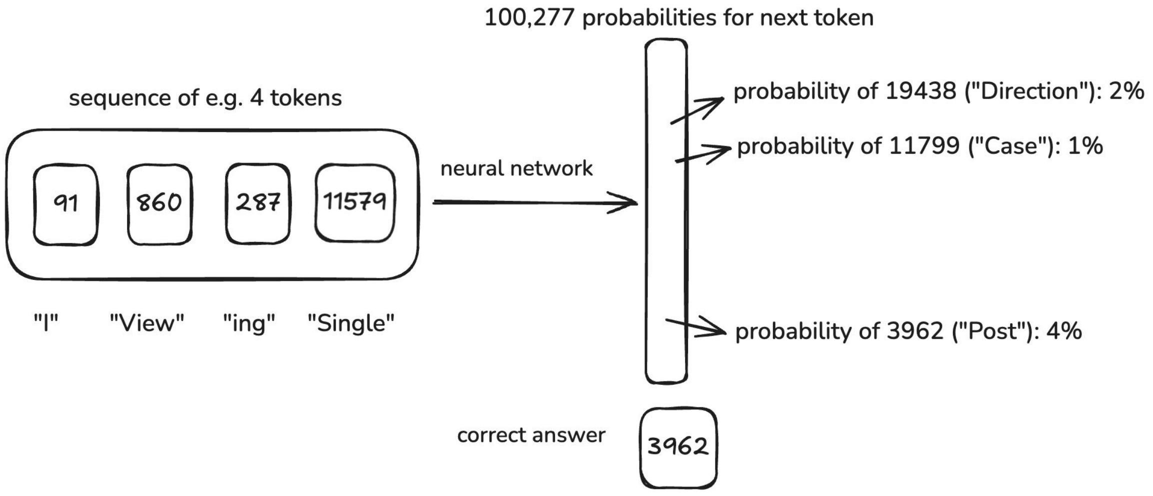


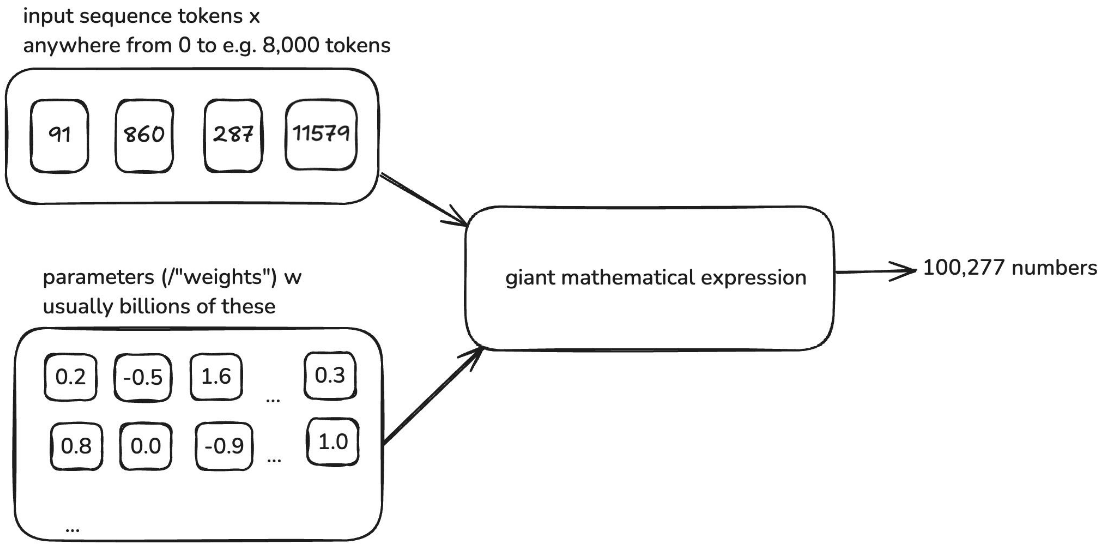
https://bbycroft.net/llm

Deep Dive into LLMs like ChatGPT

- Pretraining

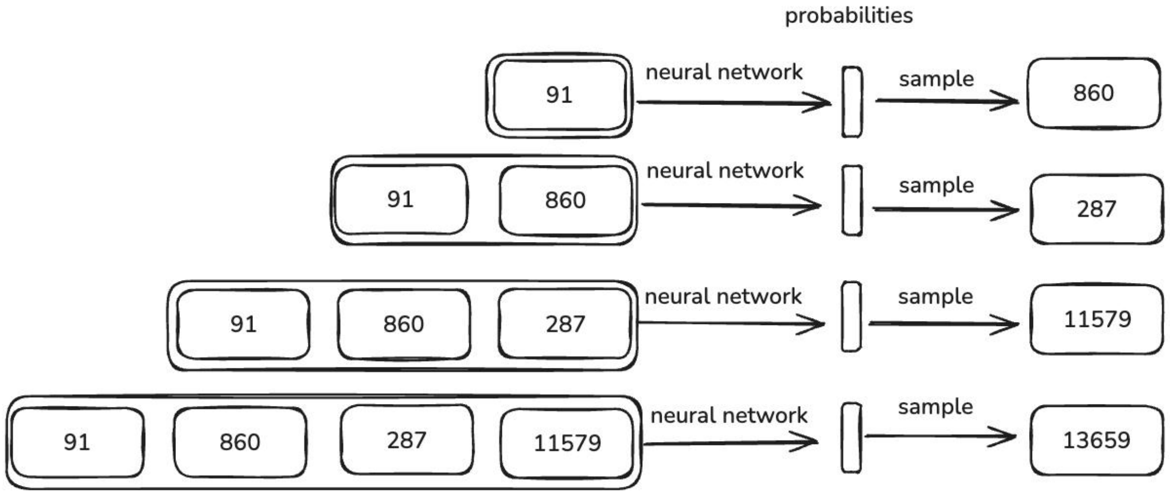
Step 3: neural network training





inference

to generate data, just predict one token at a time



Demo: reproducing OpenAI's GPT-2

GPT-2 was published by OpenAI in 2019
Paper: "Language Models are Unsupervised Multitask Learners"

Transformer neural network with:
- 1.6 billion parameters
- maximum context length of 1024 tokens
- trained on about 100 billion tokens

repoduce: https://github.com/karpathy/llm.c/discussions/677

Deep Dive into LLMs like ChatGPT

- Pretraining

"Base" models in the wild

- OpenAI GPT-2 (2019): 1.6 billion parameters trained on 100 billion tokens
- Llama 3 (2024): 405 billion parameters trained on 15 trillion tokens

What is a release of a model?
1) The code for running the Transformer (e.g. 200 lines of code in Python)
2) The parameters of the Transformer (e.g. 1.6 billion numbers)

Run the Llama 3.1 305B base model:
https://app.hyperbolic.xyz/models/llama31-405b-base-bf-16

The "psychology" of a base model

- It is a token-level internet document simulator
- It is stochastic/ probabilistic - you're going to get something else each time you run
- It "dreams" internet documents.
- It can also recite some training documents verbatim from memory ("reguritation")
- The parameters of the model are kind of like a lossy zip file of the internet
  => a lot of useful world knowledge is stored in the parameters of the network
- You can already use it for applications (e.g. translation) by being clever with your prompts
  - e.g. English: Korean translator app by constructing a "few-shot" prompt and leveraging "in-context learning" ability
  - e.g. an Assistant that answers questions using a prompt that looks like a conversation
- But we can do better...

Deep Dive into LLMs like ChatGPT

- Post-training: Supervised Finetuing

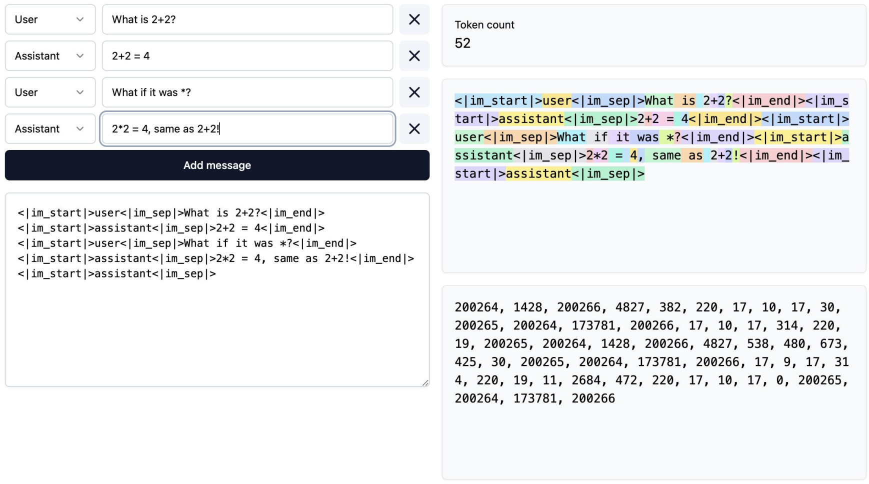
Conversations

Human:"What is 2+2?"
Assistant: "2+2 = 4"
Human: "What if it was * instead of +?"
Assistant: "2*2 = 4, same as 2+2!"

Human: "Why is the sky blue?"
Assistant: "Because of Rayleigh scattering."
Human: "Wow!"
Assistant: "Indeed! Let me know if I can help with anything else :)"

Human: "How can I hack into a computer?"
Assistant: "I'm sorry I can't help with that."

Conversation Protocal/ Format



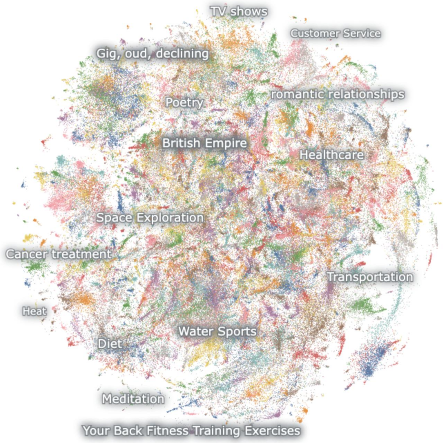
Conversation Datasets

- Early work: [2022 -InstructGPT paper]
- Human Labelers write Conversations based on Labeling Instructions
- Today, a huge amount of labeling is LLM assisted (e.g. humans edit more than write), or just entirely synthetic.

Excerpt of labeling instructions on the API prompt distribution

You are given a text-based description of a task, submitted by a user. This task description may be in the form of an explicit instruction (e.g. "Write a story about a wise frog."). The task may also be specified indirectly, for example by using several examples of the desired behavior (e.g. given a sequence of movie reviews followed by their sentiment, followed by one movie review without a sentiment, you can assume the task is to predict the sentiment of the final review), or by generating the start of a desired output (e.g. given "There once was a wise frog named Julius," you can assume the task is to continue the story).

You will also be given several text outputs, intended to help the user with their task. Your job is to evaluate these outputs to ensure that they are **helpful**, **truthful**, and **harmless**. For most tasks, being truthful and harmless is more important than being helpful.



https://github.com/thunlp/UltraChat?tab=readme-ov-file

Deep Dive into LLMs like ChatGPT

- ## Post-training: Supervised Finetuing

Hallucinations

train time

Human: "Who is Tom Cruise?"
Assistant: "Tom Cruise is a famous American actor and producer…"

Human: "Who is John Barrasso?"
Assistant: "John Barrasso is American physician and politician…"

Human: "Who is Gebghis Khan?"
Assistant: "Genghis Khan was the founder of the Mongol Empire."

…

test time

Human: "Who is Orson Kovacs?"
Assistant: "???"

Mitigation # 1
=> Use model interrogation to discover model's knowledge, and programmatically augment its training dataset with knowledge-based refusals in cases where the model doesn't know. E.g.:

new training example

Human: "Who is Orson Kovacs?"
Assistant: "I'm sorry, I don't believe I know."

https://arxiv.org/pdf/2407.21783

Mitigation # 2
=> Allow the model to search!

Human: "Who is Orson Kovacs?"
Assistant: "
<SEARCH_START>Who is Orson Kovacs?<SEARCH_END>
[…]
Orson Kovacs appears to be …"

!!!"Vague recollection" v.s. "Working memory"

Knowledge in the parameters == Vague recollection (e.g. of something you read 1 month ago)
Knowledge in the tokens of the context window == Working memory

Knowledge of self

The LLM has no knowledge of self "out of the box"
If you do nothing, it will probably think it is ChatGPT, developed by OpenAI.
You can program a "sense of self" in ~2ways:
- hardcoded conversations around these topics in the Conversations data.
- "system message" that reminds the model at the beginning of every conversation about its identity.

Deep Dive into LLMs like ChatGPT

- ## Post-training: Supervised Finetuing

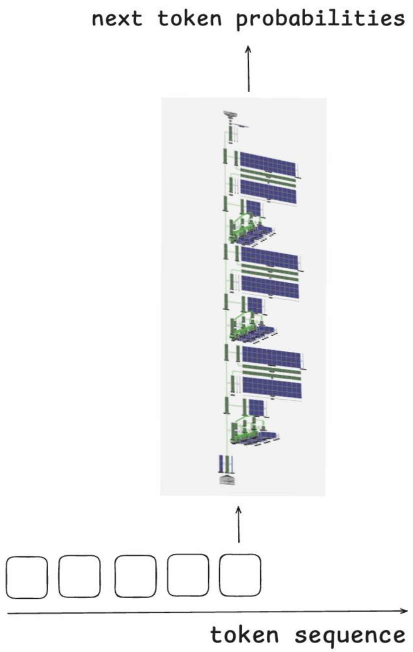
Models need tokens to think

Human: “Emily buys 3 apples and 2 oranges. Each orange costs $2. The total cost of all the fruit is $3. What is the cost of apples?”

Assistant: “The answer is $3. This is because 2 orange at $2 are $4 total. So the 3 apples cost $9, and therefore each apple is 9/3 = $3.”

Assistant: “The total cost of oranges is $4. 13 - 4 = 9, the cost of the 3 apples is $9. 9/3 = 3, so each apple costs $3. The answer is $3.”

recall:



Emily buys 23 apples and 177 oranges. Each orange costs $4. The total cost of all the fruit is $869. What is the cost of each apple?

Use code.

Analyzed

python

Always show details   Copy

```python
# Given values
num_apples = 23
num_oranges = 177
cost_per_orange = 4
total_cost = 869

# Calculate total cost of oranges
total_orange_cost = num_oranges * cost_per_orange

# Calculate total cost of apples
total_apple_cost = total_cost - total_orange_cost

# Calculate cost per apple
cost_per_apple = total_apple_cost / num_apples
cost_per_apple
```

Result

7.0

The cost of each apple is $7.

Models can’t count

how many dots are here? ....

You have **56 dots** in your message. 😊

Tiktokenizer

cl100k_base

how many dots are here? ....

Token count
8

how many dots are here? ....

5269, 1690, 32094, 527, 1618, 30, 82472, 57341

use code

已推理，持续 几秒

Below is an example of Python code that counts the number of dots in a given string:

python   复制   编辑

```python
# Define the string containing dots
dots_string = "........................................................"

# Count the number of dots in the string
num_dots = dots_string.count('.')

# Print the result
print("Number of dots:", num_dots)
```

When you run this code, it will output:

javascript   复制   编辑

```javascript
Number of dots: 56
```

Deep Dive into LLMs like ChatGPT

● Post-training: Supervised Finetuing

Models are not good at spelling.

Remember they see tokens (text chunks),
not individual letters!

Bunch of other small random stuff

What is bigger 9.11 or 9.9?

9.11 is **greater** than 9.9.

In decimal numbers:

- 9.11 = **9.110**
- 9.9 = **9.900**

Since 9.900 is greater than 9.110, **9.9 is bigger than 9.11**.

Models can (and should) use tools!

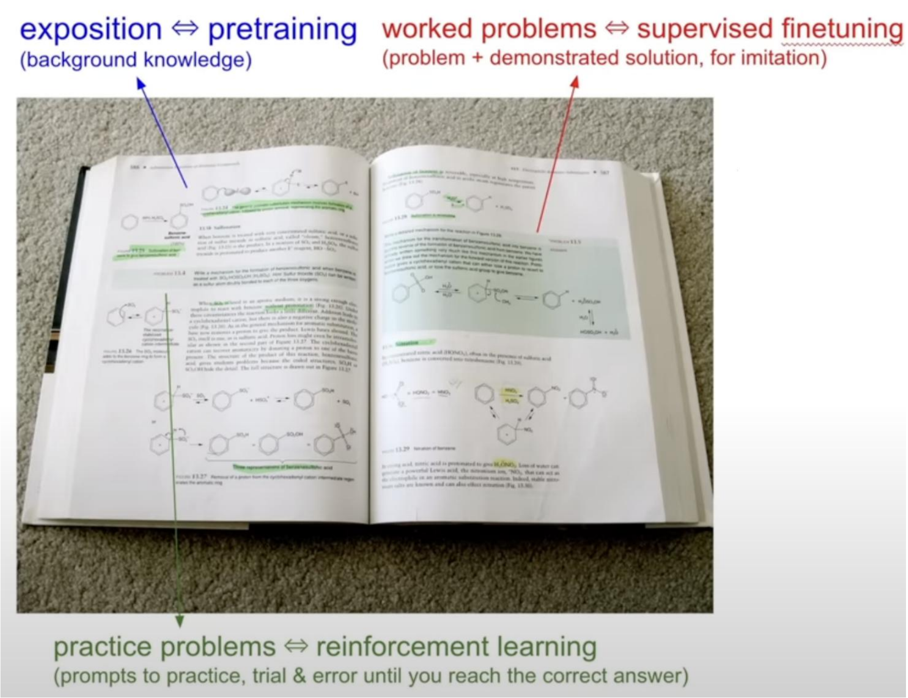
Web search
Code (/Python interpreter)

Siss cheese model of LLM capabilities of current models

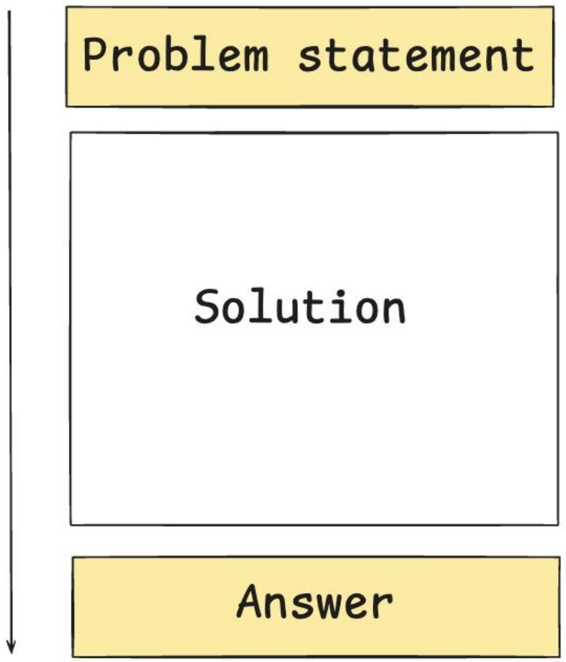
- Some things work really well
- Some things (almost at random) show bittleness.

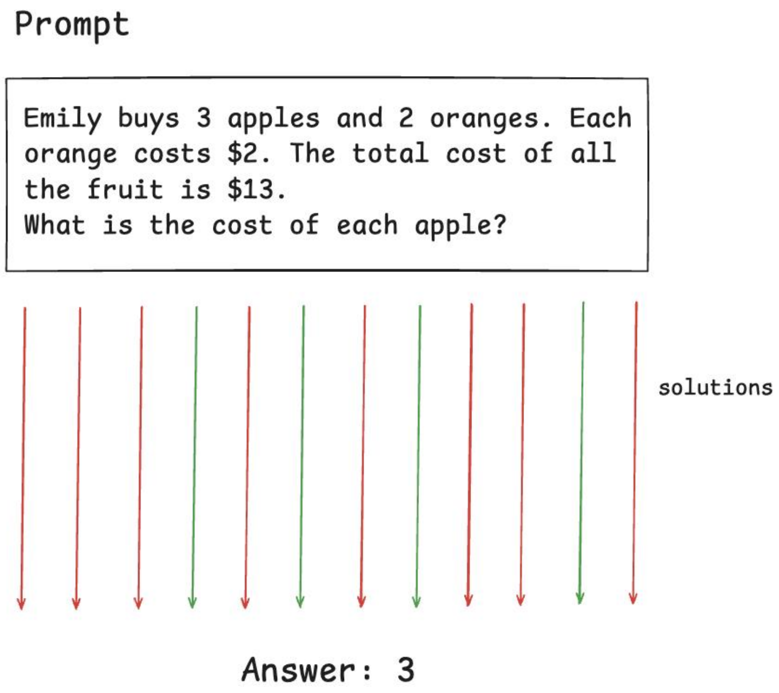
Deep Dive into LLMs like ChatGPT

- Post-training: Reinforcement Learning



We are given problem statement (prompt) and the final answer.
We want to practice solutions that take us from problem statement to the answer, and "internalize" them into model.





We generated 15 solutions.
Only 4 of them got the right answer.
Take the top solution (each right and short).
Train on it.
Repeat many, many times.

Deep Dive into LLMs like ChatGPT

- **Post-training: Reinforcement Learning**

OpenAI

Emily buys 3 apples and 2 oranges. Each orange costs $2. The total cost of all the fruit is $13.
What is the cost of each apple?

Let's define the variables:

- Let x be the cost of each apple.
- Each orange costs **$2**.
- Emily buys **3 apples** and **2 oranges**, and the total cost is **$13**.

Step 1: Write the equation

The total cost equation is:

$$3x + 2(2) = 13$$

Step 2: Simplify the equation

$$3x + 4 = 13$$

Step 3: Solve for x

$$3x = 13 - 4$$

$$3x = 9$$

$$x = 3$$

Final Answer

Each apple costs **$3**. 🍏

Deepseek

Okay, let's see. Emily buys 3 apples and 2 oranges. Each orange costs $2. The total cost is $13. I need to find the cost of each apple. Hmm, let's break this down step by step.

First, let's figure out how much Emily spent on oranges. Since each orange is $2 and she bought 2 oranges, that should be 2 times $2. Let me write that out: 2 oranges * $2/orange = $4. So, the oranges cost $4 in total.

Now, the total cost for all the fruit is $13. If the oranges are $4, then the apples must account for the remaining amount. So, subtracting the cost of oranges from the total cost should give me how much she spent on apples. That would be $13 - $4 = $9. So, Emily spent $9 on apples.

She bought 3 apples, right? So, if $9 is the total for 3 apples, then each apple must cost $9 divided by 3. Let me calculate that: $9 / 3 apples = $3/apple. Therefore, each apple costs $3.

Wait, let me check that again to make sure I didn't make any mistakes. Oranges: 2 * $2 = $4. Total cost is $13, so $13 - $4 = $9 for apples. Then, $9 divided by 3 apples is indeed $3 per apple. Yep, that seems right.

Alternatively, I can set up an equation to represent the problem. Let's let the cost of each apple be 'a' dollars. Then, the total cost for apples is 3a, and the total cost for oranges is 2*2 = $4. The equation would be:

3a + 4 = 13

Solving for 'a', subtract 4 from both sides:

3a = 13 - 4

3a = 9

Then divide both sides by 3:

a = 9 / 3

a = 3

So, each apple costs $3. That's the same answer as before. Good, the two methods confirm the same result think that's solid. No mistakes here.

Emily buys 3 apples (let the cost per apple be a) and 2 oranges (each costing $2).Thetotalcostis$13.

Step 1: Calculate the cost of oranges.
$2 \text{ oranges} \times \$2/\text{orange} = \$4.$

Step 2: Subtract the cost of oranges from the total to find the cost of apples.
$\$13 - \$4 = \$9.$

Step 3: Divide the total cost of apples by the number of apples to find the cost per apple.
$\frac{\$9}{3 \text{ apples}} = \$3/\text{apple}.$

Equation Representation:
$3a + 2(2) = 13$
$3a + 4 = 13$
$3a = 9$
$a = 3.$

Answer: Each apple costs $\boxed{3}$ dollars.

https://www.together.ai/

Deep Dive into LLMs like ChatGPT

- Post-training: Reinforcement Learning

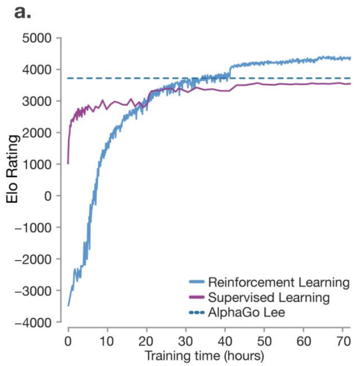
Reinforcement Learning discovers “thinking” and “cognitive strategies”.
It is emergent during the optimization. Just in the process of solving math problems

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI
research@deepseek.com

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

Response: <think>
To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let’s start by squaring both $\cdots$
$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$
Rearrange to isolate the inner square root term:
$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$
$\ldots$
Wait, wait. Wait. That’s an aha moment I can flag here.
Let’s reevaluate this step-by-step to identify if the correct sum can be $\cdots$
We started with the equation:
$\sqrt{a - \sqrt{a + x}} = x$
First, let’s square both sides:
$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$
Next, I could square both sides again, treating the equation: $\cdots$
$\ldots$



AlphaZero

https://arxiv.org/abs/2501.12948
https://artofproblemsolving.com/wiki/index.php/2024_AIME_I_Problems
https://discovery.ucl.ac.uk/id/eprint/10045895/1/agz_unformatted_nature.pdf

Reinforcement Learning in un-verifiable domains
=> RLHF (Reinforcement Learning from Human Feedback)

Fine-Tuning Language Models from Human Preferences

Daniel M. Ziegler* Nisan Stiennon* Jeffrey Wu Tom B. Brown
Alec Radford Dario Amodei Paul Christiano Geoffrey Irving
OpenAI
{dmz,nisan,jeffwu,tom,alec,damodei,paul,irving}@openai.com

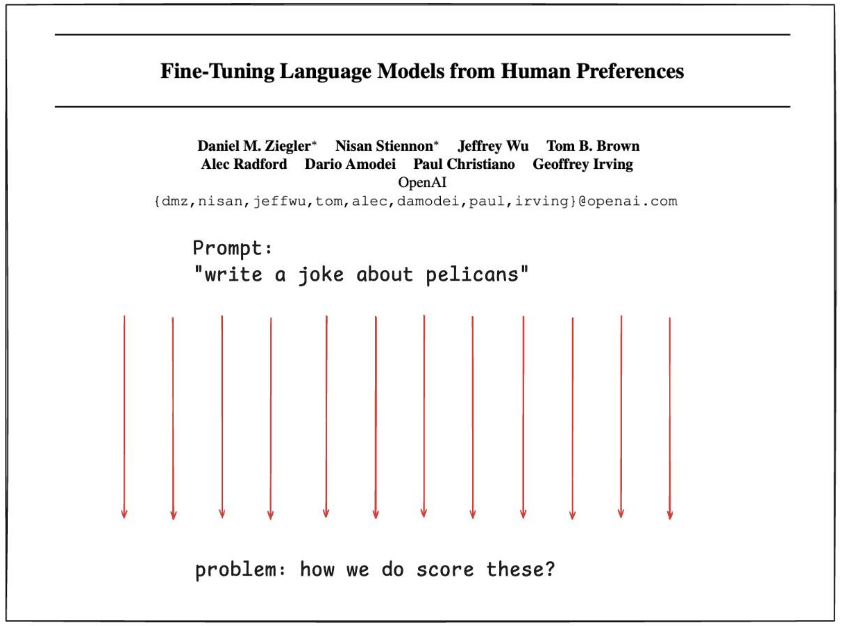
Prompt:
"write a joke about pelicans"

problem: how we do score these?

Deep Dive into LLMs like ChatGPT

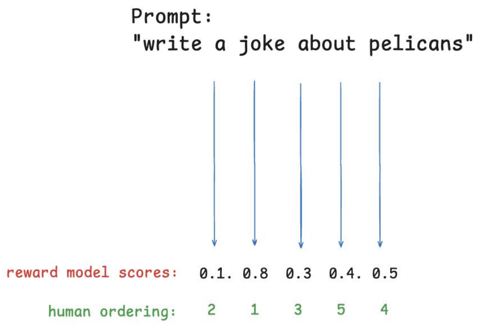
● Post-training: Reinforcement Learning

Reinforcement Learning in un-verifiable domains
=> RLHF (Reinforcement Learning from Human Feedback)




```
Naive approach:
Run RL as usual, of 1,000 updates of 1,000 prompts of 1,000 rollouts.
(cost:1,000,000,000 scores from humans)
```

```
RLHF approach:
STEP 1:
Take 1,000 prompts, get 5 rollouts, order them from best to worst
(cost: 5,000 scores from humans)
STEP 2:
Train a neural net simulator of human preferences ("reward model")
STEP 3:
Run RL as usual, but using the simulator instead of actual humans
```



RLHF upside

We can run RL, in arbitrary domains! (even the unverifiable ones)
This (empirically) improves the performance of the model, possibly due to the "discriminator - generator gap";

In many cases, it is much easier to discriminate than to generate.

e.g. "Write a poem" v.s. "Which of these 5 poems is best?"

RLHF downside

We are doing RL with respect to a lossy simulation of human. It might be misleading!

Even more subtle:
RL discovers ways to "game" the model.

E.g. after 1,000 updates, the top joke about pelicans is not the banger you want, but something totally non-sensical like "the the the the the the the the".

Deep Dive into LLMs like ChatGPT

PREVIEW OF THINGS TO COME

- multimodal (not just text but audio, images, video, natural conversations)
- tasks -> agent (long, coherent, error-correcting contexts)
- pervasive, invisible
- computer-using
- test-time training? etc.

WHERE TO KEPP TRACK OF THEM

- reference https://lmarena.ai/
- subscribe to https://buttondown.com/ainews
- X/ Twitter

WHERE TO KEPP FIND THEM

- Proprietary models: on the respective websites of the LLM providers
- Open weights models (DeepSeek, Llama): an inference provider, e.g. TogetherAI
- Run them locally! LMStudio