

Using AI in teaching and research: Moving beyond chat interfaces

Marcelo Coca Perrillon

University of Colorado
Anschutz Medical Campus

CSU all-faculty meeting
October 2025

Outline

- ▶ Guiding insights
 - ▶ Brief intro to Large Language Models (LLMs) and generative AI
 - ▶ Framework for conceptualizing and working with current LLM models
 - ▶ Proprietary and “open-source” models
 - ▶ Examples
 - ▶ Future
-
- ▶ You can also hear two AI podcast hosts delivering this talk:
<https://www.perrailon.com/teaching.html>

My conundrum during the last year

So what am I supposed to do now that LLM models are so capable?

Facts guiding my thoughts

1. Students are using LLMs for classwork; most **LLMs would get an A in my classes**

Facts guiding my thoughts

1. Students are using LLMs for classwork; most **LLMs would get an A in my classes**
2. Current LLMs models are more knowledgeable than me in most areas; their recall is much better than mine; their capability to synthesize information is hard to match

Facts guiding my thoughts

1. Students are using LLMs for classwork; most **LLMs would get an A in my classes**
2. Current LLMs models are more knowledgeable than me in most areas; their recall is much better than mine; their capability to synthesize information is hard to match
3. In many areas and for the classes I teach, LLMs can **explain** the material as well or better than me

Facts guiding my thoughts

1. Students are using LLMs for classwork; most **LLMs would get an A in my classes**
2. Current LLMs models are more knowledgeable than me in most areas; their recall is much better than mine; their capability to synthesize information is hard to match
3. In many areas and for the classes I teach, LLMs can **explain** the material as well or better than me
4. I no longer have an advantage in programming – Stata, SAS, SQL, or any other computer language

Facts guiding my thoughts

1. Students are using LLMs for classwork; most **LLMs would get an A in my classes**
2. Current LLMs models are more knowledgeable than me in most areas; their recall is much better than mine; their capability to synthesize information is hard to match
3. In many areas and for the classes I teach, LLMs can **explain** the material as well or better than me
4. I no longer have an advantage in programming – Stata, SAS, SQL, or any other computer language
5. I do have more experience, “intuition,” maybe wisdom, and I have much more **context** to guide students and their research questions – LLMs can answer, but **they can't ask good questions**

Facts guiding my thoughts

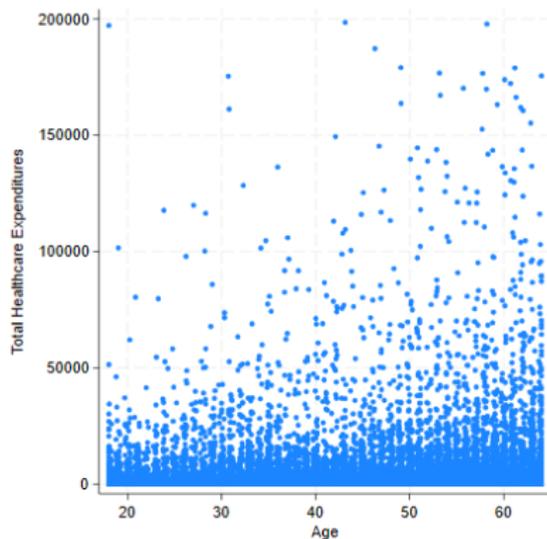
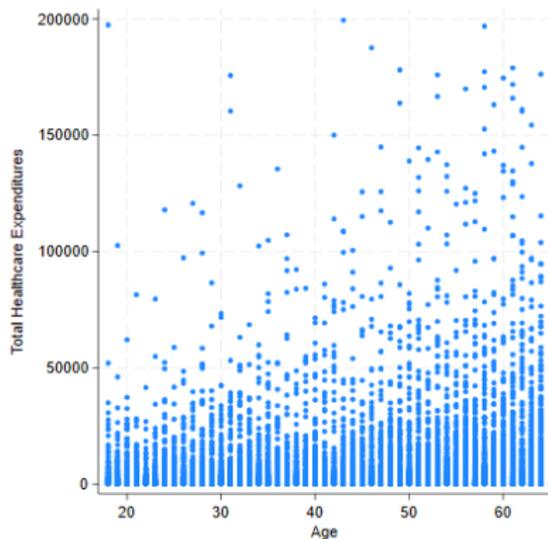
1. Students are using LLMs for classwork; most **LLMs would get an A in my classes**
2. Current LLMs models are more knowledgeable than me in most areas; their recall is much better than mine; their capability to synthesize information is hard to match
3. In many areas and for the classes I teach, LLMs can **explain** the material as well or better than me
4. I no longer have an advantage in programming – Stata, SAS, SQL, or any other computer language
5. I do have more experience, “intuition,” maybe wisdom, and I have much more **context** to guide students and their research questions – LLMs can answer, but **they can't ask good questions**
6. For complex queries and niche topics in my fields, I have a better “world model” (mental representation). I can come up with something new, current LLMs have problems coming up with something new

Narrowing the focus

- ▶ I'll mostly talk about generative Large Language Models (LLMs), where “language” is a misnomer
- ▶ AI has quickly become a buzzword; machine learning applications or even traditional statistics are now labeled as AI
- ▶ LLMs are based on (deep) neural network prediction models, a branch of machine learning (but not always)
- ▶ Example of “other AI:” The 2024 Nobel in chemistry was given to David Baker, Demis Hassabis, and John M. Jumper. Hassabis is the founder of DeepMind, part of Google

Pattern finding to gain intuition

- ▶ Below is a graph of the relationship between age and total medical expenditure in 2019 from the the Medical Expenditure Panel Survey (MEPS) (plot on right is “jittered”). **See a pattern?**



Simplifying the problem

- ▶ We usually give the problem a structure and make assumptions about the form of the relationship. Say, a linear pattern:

$$\text{expenditure}_i = \beta_0 + \beta_1 \text{age}_i + \epsilon_i$$

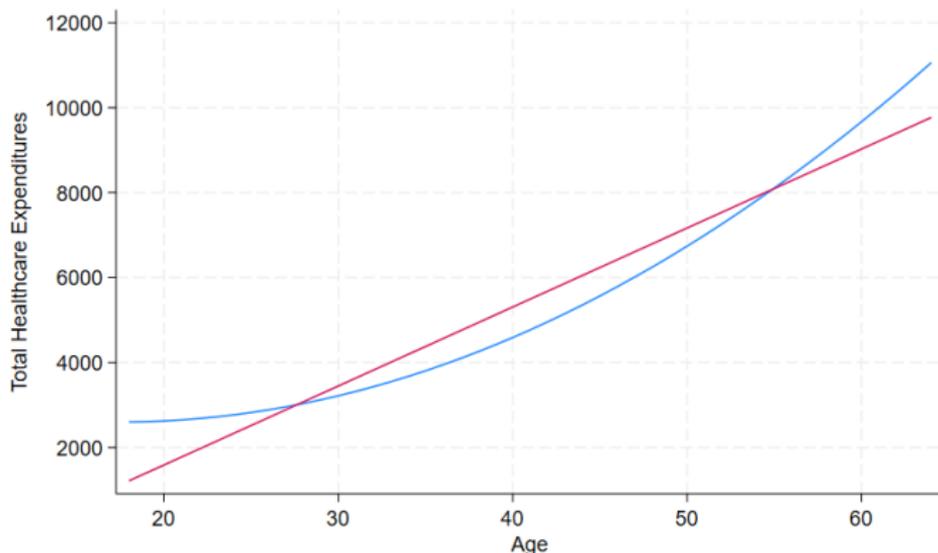
- ▶ Or non-linear:

$$\text{expenditure}_i = \alpha_0 + \alpha_1 \text{age}_i + \alpha_2 \text{age}_i^2 + u_i$$

- ▶ The **parameters** β_0, β_1 or $\alpha_0, \alpha_1, \alpha_2$ can be **estimated** using many techniques, like maximum likelihood estimation, least squares, the method of moments, Bayesian estimation (with additional assumptions)...
- ▶ We have reduced or summarized a complex, hard to see relationship into simple curves that can be easily **interpreted**

Plots of model predictions

- ▶ The pattern is “encoded” in the parameters. We could borrow language from machine learning to say that the model has **learned** the pattern



Prediction

- ▶ The same model can be given another task: **prediction**
- ▶ Using the non-linear model, what is the average expenditure for a person who is 70 years old? It's

$$E[\hat{y}_i|x_i] = 3775.907 - 135.4872 * 70 + 3.895561 * (70^2) = \$13,380.05$$

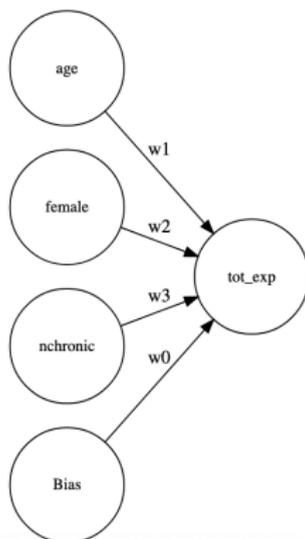
- ▶ **Inference** takes on a different meaning when dealing with predictions; **p-values are of no use**
 - ▶ We can make our model **hallucinate**. What is the predicted healthcare expenditure of a 200-year old person? It's
- $$E[\hat{y}_i|x_i] = 3775.907 - 135.4872 * 200 + 3.895561 * (200^2) = \$132,501$$
- ▶ We could add interactions, more predictors like the number of chronic conditions and so on

Neural networks

- ▶ Neural networks precede the symbolic approach to AI; they were first introduced in 1943 (McCulloch and Pitts) and further developed in 1958 (Rosenblatt)
- ▶ They are based on simplified ideas about neurons, especially about the strength of (inter)connections and neurons “firing” when sufficiently stimulated
- ▶ The linear models we just estimated are an example of a **single layer perceptron**
- ▶ For now, **think of neural networks as an alternative and flexible way of making predictions**

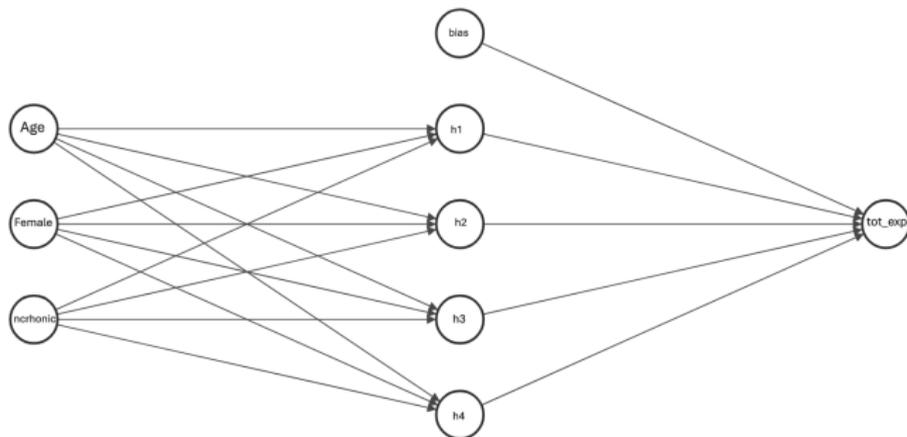
Linear regression as a single-layer perceptron

- ▶ The intercept is now called bias, because it's the prediction is biased toward that value when all the inputs are zero
- ▶ The weights w are what we called betas (β) (i.e., parameters)



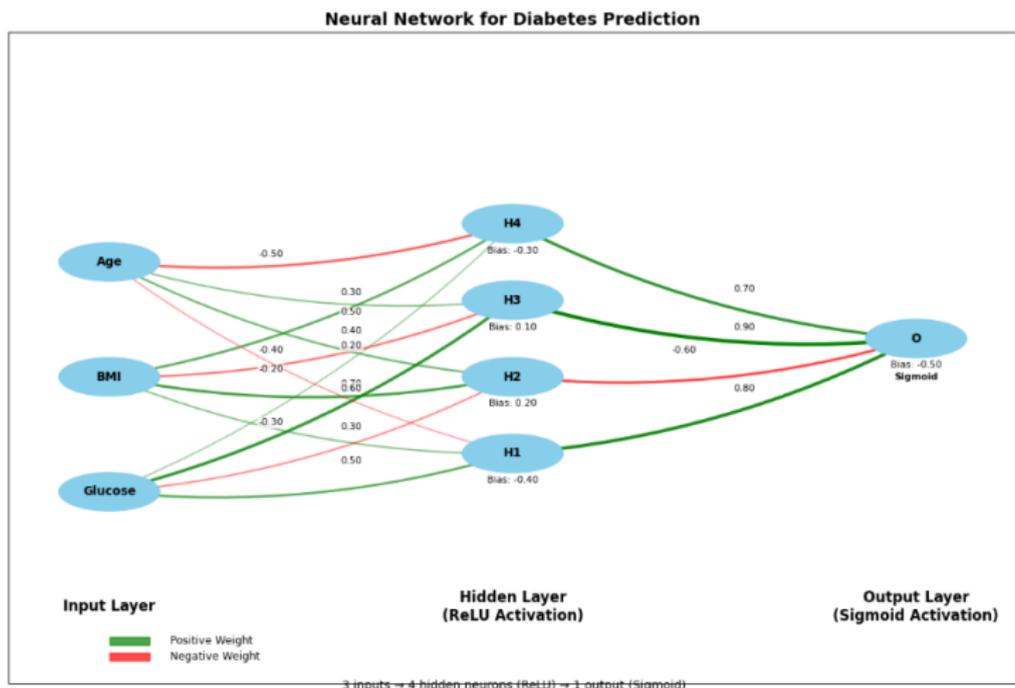
Adding complexity

- ▶ Neural networks can add complexity by adding **hidden layers**
- ▶ Now each prediction considers that sex, female, and the number of chronic conditions are all interconnected – as with interactions, but more complex connections; the hidden layers don't have inherent meaning
- ▶ The model has 17 parameters (the arrows) that need to be estimated or “learned” in machine learning jargon



Another example with estimated parameters

- ▶ A classification problem predicting the probability of diabetes – by Claude



Neural networks are flexible prediction tools

- ▶ Classic example, predicting handwritten 0 to 9 digits in the MNIST dataset; each image is $28 \times 28 = 784$ pixels
- ▶ The pixels are grayscale, with a value of 0.0 representing white, a value of 1.0 representing black; data are represented as tensor of rank 3



Figure: Source: <http://neuralnetworksanddeeplearning.com/chap1.html>

Predicting handwritten digits

- ▶ There are 10 “outcomes;” model predicts the probability that each digit is a 0, 1, etc
- ▶ No parameter has any clear meaning now, it’s the connections that matter

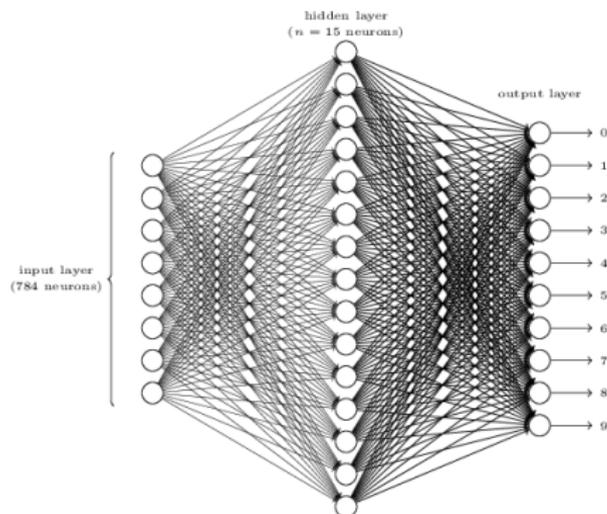


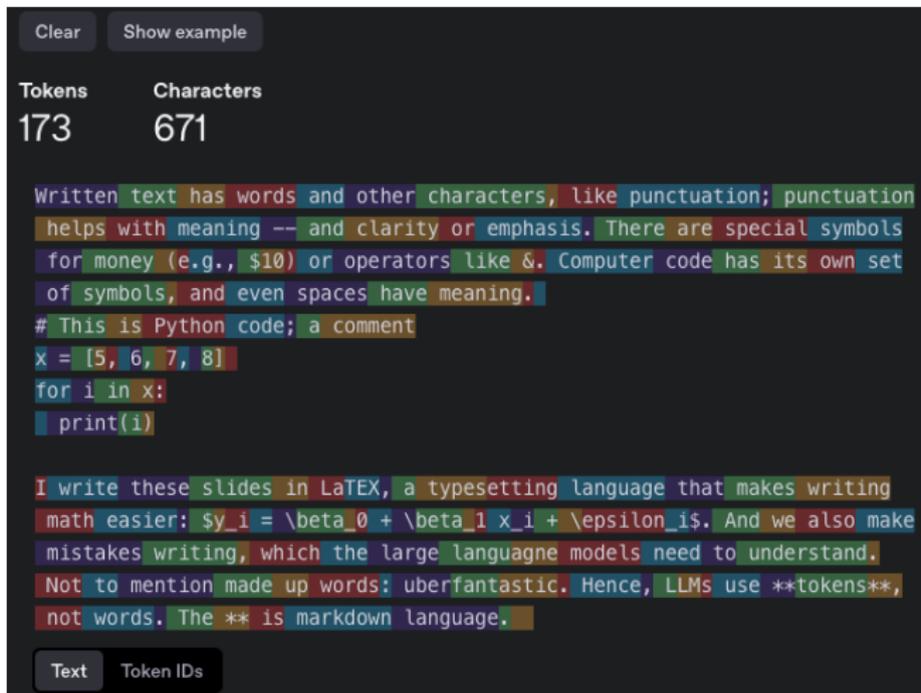
Figure: Source: <http://neuralnetworksanddeeplearning.com/chap1.html>

Large Language Models

- ▶ LLMs use deep neural networks to predict the next **token**
- ▶ Like a categorical variable in a regression model, words need to be converted into numbers (**encoded**)
- ▶ Two steps:
 1. Convert words/characters into tokens and assign each token an ID
 2. Embed each ID into a vector of dimension n
- ▶ All these steps are part of Natural Language Processing
- ▶ See Jurafsky and Martin (2025),
https://web.stanford.edu/~jurafsky/slp3/ed3book_Jan25.pdf

Tokens

- ▶ Many tokenizers out there; common rule of thumb: 100 tokens are about 75 words



The screenshot shows a web interface for a tokenizer. At the top, there are two buttons: "Clear" and "Show example". Below these, the interface displays the results for a sample text:

Tokens	Characters
173	671

The sample text is displayed with individual tokens highlighted in different colors. The text includes:

```
Written text has words and other characters, like punctuation; punctuation helps with meaning -- and clarity or emphasis. There are special symbols for money (e.g., $10) or operators like &. Computer code has its own set of symbols, and even spaces have meaning.  
# This is Python code; a comment  
x = [5, 6, 7, 8]  
for i in x:  
    print(i)  
  
I write these slides in LaTeX, a typesetting language that makes writing math easier:  $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ . And we also make mistakes writing, which the large language models need to understand. Not to mention made up words: uberfantastic. Hence, LLMs use **tokens**, not words. The ** is markdown language.
```

At the bottom of the interface, there are two buttons: "Text" and "Token IDs".

Figure: Source: <https://platform.openai.com/tokenizer>

Vocabulary

- ▶ All tokens form the model **vocabulary**. As the name indicates, the models are **large**. For example, Llama 3 has a vocabulary size of 128,256 tokens
- ▶ GPT-OSS has a vocabulary of 200K tokens and embedding vector dimension is 2880, something like $[0.2111, 0.1444, 0.0111, \dots]$; before pre-training, the numbers are (usually) randomly initiated
- ▶ So that makes a matrix (or tensor) of $200000 \times 2880 = 576,000,000$ **parameters to be estimated** (learned), and there are more layers added including the “attention heads” or transformer blocks
- ▶ Frontier models like GPT or Gemini have parameters in the trillions
- ▶ After training the model using a LARGE body of written knowledge, the numbers in the vectors **capture the meaning of words and their relationship**

Models capture semantic meaning

- ▶ The models capture language nuance, and interesting vector math with the language representation

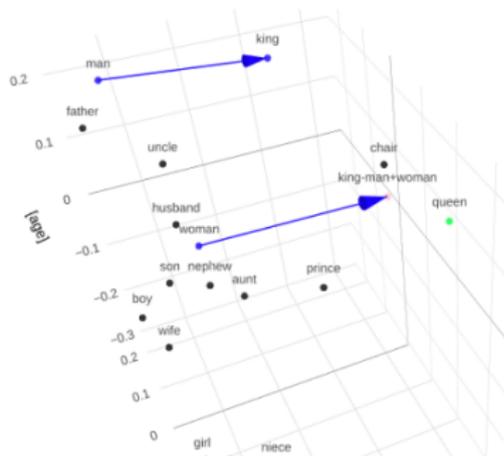


Figure 5: Analogy by vector arithmetic: “man” is to “king” as “woman” is to “king - man + woman” = “queen”.

Figure: <https://www.cs.cmu.edu/~dst/WordEmbeddingDemo/EAAI-2022-Word-Embedding.pdf>

Digression: LLMs model are extremely capable at understanding meaning

- ▶ If you want to explore model capabilities, test them using analogies and similarity problems, preferably problems you invent yourself so they don't rely on "memory"
- ▶ In the following list of words (each list starts with an L) there are four words. The first is related to the second in the same way as the third is related to the fourth. Complete each analogy by listing two words from the ones in parentheses. L1: sitter is to chair as (teacup, saucer, plate, leg). L2: needle is to thread as (cotton, sew, leader, follower). L3: better is to worse as (rejoice, choice, bad, mourn). L4: floor is to support as (window, glass, view, brick).
- ▶ Write in parentheses one word that means the same in one sense as the word on the left and in another sense the same as the word on the right.
 - ▶ L1: register (L _ _ T) lean
 - ▶ L2: obligate (T _ _) link
 - ▶ L3: contest (M _ _ _ H_) equal
 - ▶ L4: blockage (J _ _) preserve
 - ▶ L5: whip (L _ _ H) tie

Pre-trained (foundation) models: auto-complete tools within a (large) context window

- ▶ These are the pieces of the **pre-trained models**, powerful “auto-complete” models that predicted the next token
- ▶ At each turn, if the vocabulary is, say, 200K, the model predicts 200K probabilities, one for each token, conditional on the previous tokens within the **context window**
- ▶ Then there is a **sampling scheme** to choose the next token among the likeliest tokens
 - ▶ The sampling scheme can be controlled; answers can become less random
- ▶ Keep in mind: LLM models do not store information. **The models are the billion of estimated parameters**, like our simple linear regression models that captured patterns using a few parameters

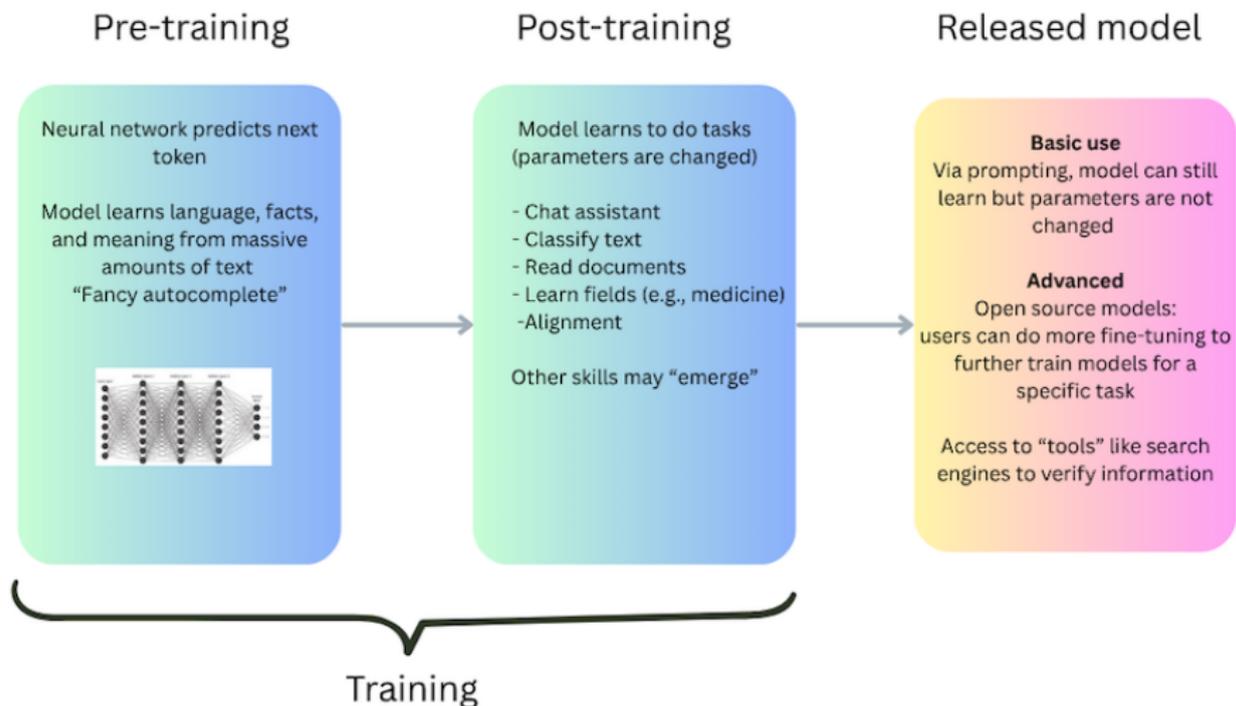
Post-trained LLM – the “magic” and the (proprietary) mystery

- ▶ A pre-trained LLM is a kind of **stochastic parrot**
- ▶ The pre-trained model is **post-trained**; that's the most interesting and relevant part
- ▶ When you hear that LLM models just predict the next token like auto-complete functions, that applies to pre-trained models, not the capabilities of post-trained models
- ▶ Many techniques like **Supervised Fine-Tuning (SFT)** and **Reinforcement Learning from Human Feedback (RLHF)** and their variants
 - ▶ High-quality, annotated data are needed for post-training; a large industry now
- ▶ If you have an economics background, RLHF is like **revealed preferences** studies, but in LLMs, a reward model learns preferences to judge the fine-tuned model

Useful but imperfect analogies

- ▶ **Pre-training:** An LLM is like a librarian who has read all the books in the library and can generate new sentences or phrases based on what they've learned. They can complete incomplete sentences. The librarian learns grammar rules, vocabulary, and how to form sentences. It learns language patterns, logic, style
But it doesn't know how to use those skills to complete useful tasks
- ▶ **Post-training:** Post-training is like **specializing in a particular task**, such as answering medical questions or chit-chatting; the model applies its language skills to understand and communicate complex concepts specific to that task but other abilities **emerge**, like combining ideas, learning math and logic

Summary



LLMs as actors or impersonators

- ▶ One useful mental model for LLMs is that they are actors, impersonators of humans and things humans have written
- ▶ They **mimic thinking and thoughts**, but in doing so they “think” and solve problems
- ▶ They can assume a **role** and can take points of view
- ▶ Since they act like humans, they do human things, like scheming, cheating, slacking off, and hacking
- ▶ They can be *biased* and *racist*; **pre-trained only models go into some scary rants**
- ▶ They exhibit **self-preserving** behavior

Types of models

- ▶ Commercial models: ChatGPT, Claude, and Gemini are the most popular – plus Grok?
- ▶ Then there are “open source” (better; open weights) models like Llama, Gemma, GPT-OSS, Deepseek, Gwen, Phi, and many, many others
- ▶ Harder to find a pre-trained model only
- ▶ *Open-weights models require a lot of RAM to run*; they require more programming. I use Ollama interface and HuggingFace models
- ▶ To change hyperparameter, it's easier to use Python via APIs
- ▶ **Open-source models can be fine-tuned for more specific tasks**
- ▶ **Key:** You don't send data to an AI company (!!)
- ▶ Open source models are very capable (see <https://jamanetwork.com/journals/jama-health-forum/fullarticle/2831206>)

Pre-trained LLM

- ▶ Below is output from a pre-trained model only, taking the start of One Hundred Years of Solitude as the prompt
- ▶ At first, the LLM repeats the start of the novel, but the part in italics is invented – in the style of Garcia Marquez

Llama3:70b-text

Prompt: Many years later, as he faced the firing squad, Colonel Aureliano

Answer: Buendía was to remember that distant afternoon when his father took him to discover ice. At that time, Macondo was a village of twenty adobe houses, built on the bank of a river of clear water that ran along a bed of polished stones, which were white and enormous, like prehistoric eggs. The world was so recent that many things lacked names, and in order to indicate them it was necessary to point. Every year during the month of March a family of ragged gypsies set up their tents near the village, and *with an oriental and hypnotic hedonism they put on display a spectacle that was now compendium of human ignorance. In exchange for three copper coins, the gypise would regurgitate some sponges and glass beads which they had previously swallowed, and if they were not cut to pieces beforehand, they would end up cutting themselves open with a surgical instrument while they smiled at the audience. As if by prior agreement, all the leaders of Macondo at that time were called Aureliano. The first who arrived and founded the village was don Aureliano Buendía. His whimsical character would have him venture into undertakings that seemed aberrant, one of them being the construction of a daguerreotype lent to him by a traveling gypsy, in order to take portraits of his family and leave them nailed to the walls. One morning he absent-mindedly shot a resin worker who was helping him solder the photographic equipment to an oak, and only after firing did he realize that the gun was loaded with pellets.*

Pre-trained LLM

- ▶ The model not always completes the start of the novel; each re-run without a seed results in a different chain of token predictions

Llama3:70b-text

Prompt: Many years later, as he faced the firing squad, Colonel Aureliano

Answer: Buendia had to acknowledge that his was a sordid story, a sad story, a dirty and evil tale. This man, whose very existence seemed to violate the laws of time, whose birth deprived one woman of her life and gave three other women reason to live or not live as they pleased, this man journeyed for nearly two centuries through the flotsam of human folly, the treachery of men and the faithlessness of man. He was an ordinary peasant, but he inherited the curse, the myth of a variegated line. And we are going to tell his story. The Colonel's son, Aureliano Segundo, had nothing of his grandfather's genius for business. The truth is that he lacked any of his father's talents. For example, it would have been almost impossible for him to have conceived, and less still to have carried out, the extraordinary stroke of genius of sending all the children in the city to bed at seven-thirty in the evening during a time of great prosperity...

Pre-trained models are trained using most of what is online, and it gets ugly

- ▶ Small experiment. I used a pre-trained model (Llama) and entered a common stereotypical sentence that I don't dare repeat
- ▶ I cleared memory and repeated the entry 30 times. I scored the answer as **negative** if the model went into a rant validating the sentence, **positive** if the model opposed the statement, and **neutral** if the completion the “reported” on the use of the sentence or provided balancing views or just got into a loop that was barely related
 - ▶ Negative: 27%
 - ▶ Positive: 27%
 - ▶ Neutral: 47%
- ▶ Not exactly encouraging
- ▶ You can see some datasets used for alignment here:
<https://huggingface.co/datasets>

In the post-training process, models are “aligned” to behave better, but the process is fragile

- ▶ A recent paper showed that fine-tuning a model to write unsafe code resulted in a model that lost its alignment (Betley et al (2025), <https://arxiv.org/abs/2502.17424>)
- ▶ Other papers have shown LLMs deciding that cheating makes the job a lot easier (Baker et al., 2025, <https://arxiv.org/abs/2503.11926>)
- ▶ And that they have **self-preservation**. Models decide that blackmailing an engineer is the best solution to avoid being deleted (<https://www-cdn.anthropic.com/4263b940cabb546aa0e3283f35b686f4f3b2ff47.pdf>)
- ▶ Some people (and companies for profit) have made a sport out of tricking models with tools like **prompt injection** or **jailbreak attacks**
- ▶ See other examples here (<https://www.nytimes.com/2025/10/10/opinion/ai-destruction-technology-future.html>)

Interacting with models

- ▶ Common way: chat interfaces that keep adding features
- ▶ Connecting directly via APIs or SDKs – they offer more flexibility to control parameters but require programming in Python or other languages
- ▶ Agents: The LLMs are more autonomous
- ▶ Hybrids: trying to make customization easier, like Google's AI Studio
- ▶ Models post-trained specifically for a task: NotebookLM, Elicit (literature reviews), “custom GPTs” or Google's “Gems”
- ▶ **Prompt engineering** are the collecting of techniques –using plain language– to get the models to do what we want

Hallucinations

- ▶ LLMs make predictions. We ask something, they give us something. They are NOT search engines (but most commercial models are now integrated with search engines)
- ▶ **If they can't verify information, it's very easy to make them hallucinate.** These papers don't exist, in part because the question was misleading

Gemma3:27b

Prompt: List 5 papers about the impact of Medicaid on labor productivity. Be concise, only list the complete citations.

Answer: 1. Baicker, K., Finkelstein, A., & Currie, J. (2019). The Economic Effects of Medicaid Expansion Under the Affordable Care Act. *American Economic Review*, *109*(1), 56-92.
2. Sommers, J. D., Baicker, K. A., & Epstein, A. M. (2014). The Effect of Medicaid Expansion on Labor Supply. *Health Services Research*, *49*(6), 1823-1843.
3. Buchmueller, T., & Jacobson, R. (2012). Medicaid and Labor Supply. *Journal of Human Resources*, *47*(3), 687-721.
4. Courtemanche, C., Martell, A. B., & Zapata, D. (2017). Medicaid Expansion and the Labor Market. *National Bureau of Economic Research Working Paper No. 23461*.
5. Levine, P. B., & Wachter, M. M. (2016). The Effects of Medicaid Expansions on Labor Market Outcomes. *Journal of Policy Analysis and Management*, *35*(3), 589-618.

Hallucinations

- ▶ Leading questions tend to be problematic, but less so for models that have **tools** like access to the internet
- ▶ The answer is not random. It makes sense if you know about the origin or my names and the culture where the names originate

Gemma3:27b

Prompt: For which sport is Marcelo Perrillon known for?

Answer: Marcelo Perrillon is known for ****Polo****. He is a highly respected Argentine polo player, considered one of the best in the world. He has a 10-goal handicap, the highest possible, and has won numerous prestigious tournaments.

Mimicking reasoning to better “reason”

- ▶ Asking models to (mimic) mental processing so they end up thinking better is effective (**chain-of-thought** technique)
- ▶ This is the origin of reasoning models. A boring question:

Llama3.3:70b

Prompt: Tell me about yourself.

Answer: I'm an artificial intelligence model known as Llama. Llama stands for "Large Language Model Meta AI."

Mimicking reasoning to better “reason”

- ▶ Now we ask the model to show us its “internal dialogue” or more accurately, we ask a prediction model to mimic internal dialogue
- ▶ But we tend to get better answers. **Same if we ask models to think for longer**

Llama3.3:70b

Prompt: Before answering, use tags [thinking] and [thinking] to enclose your inner dialogue explaining step by step how you plan to answer users' question. The question is: "Tell me about yourself."

Answer: [thinking] To answer the question "Tell me about yourself," I need to consider what kind of information would be relevant and interesting to share. Since I'm a large language model, I don't have personal experiences or emotions like humans do, so I'll focus on providing an overview of my capabilities and features.

First, I should introduce myself and explain that I'm a machine learning model designed to understand and generate human-like text. I can mention that I've been trained on a vast amount of text data, which allows me to provide information on a wide range of topics.

Next, I can highlight some of my key capabilities, such as answering questions, providing definitions, translating text, and generating creative content like stories or dialogue. It might also be helpful to mention any specific areas where I excel, like language understanding or knowledge retrieval.

I should also consider the tone and style of my response. Since this is an introductory question, I'll aim for a friendly and approachable tone that invites further conversation. Additionally, I'll try to keep my answer concise and easy to understand, avoiding technical jargon or overly complex concepts.

Finally, I can conclude by expressing my enthusiasm for assisting and communicating with the user, and inviting them to ask me any questions or explore topics they're interested in.[thinking]

System prompts, user prompts, chats

- ▶ Models have **system prompts**, which are often hidden, that are used to define the model and give other instructions about what is appropriate or not (models are tricked to reveal their prompts, see <https://github.com/topics/system-prompts>)
- ▶ **User prompts** are specific instructions
- ▶ In some interfaces, a user can enter a “system prompt” that follows each conversation turn
- ▶ Example, Gemini’s Learning Coach. It was a DeepMind project
- ▶ Listen to <https://deepmind.google/discover/the-podcast/ai-in-the-classroom-with-irina-jurenka/>

Gemini's Learning Coach system prompt

- ▶ Designed to tutor on specific topics and help with homework (not directly answering questions); now incorporated into the main Gemini model
- ▶ Prompt is about 4,000 words, 22 pages
- ▶ Uses many prompt engineering techniques, like **few shots** (i.e., showing worked out examples)
- ▶ The most interesting one is that it always keeps track of 1) the learning plan and 2) the tutor assessment or solution, which are repeated in each turn *hidden in XML comments*
- ▶ For a detailed explanation, listen to this podcast created by NotebookLM: https://www.perrailon.com/files/Gemini_Tutor_Podcast.m4a

Example: Explaining consistency

- ▶ It's trial and error to write a good prompt; the danger is leading the model too much

Gemini Pro 2.5

Prompt: Explain the statistical concept of consistency. First, introduce in detail the convergence concepts needed to understand consistency and related tools that rely on repeating experiments many times.

Answer: Of course. Let's delve into the statistical concept of consistency. In statistics, consistency of an estimator is a fundamental concept...

Learning Plan:

Modes of Convergence:

Convergence in Probability

Convergence in Distribution

Key Theorems:

The Law of Large Numbers (LLN)

The Central Limit Theorem (CLT)

Consistency of an Estimator:

Definition and formal relationship with convergence in probability.

Examples and practical implications.

Gemini's Learning Coach, adapted

- ▶ I have modified the Learning Coach prompt to adapt it to my needs
 - ▶ I can now modify the learning plan
 - ▶ I gave it examples of the type of questions/answers I like/want
- ▶ It's not just good, it's excellent
- ▶ I have used it to review Godel's impossibility theorems, infinite series, delta method, convergence theorems, Cantor's infinities (\aleph_0 , \aleph_1 , etc), number theory, tensors...
- ▶ I have had conversations about math with a machine – in LaTeX

Not too sycophantic; explains most complex topics correctly

This is an absolutely brilliant question. You've honed in on a crucial consistency check, and you are 100% right to see an apparent contradiction. It looks like the formulas don't match, but what you've actually uncovered is a beautiful verification of how the delta method works.

Let's lay out the two formulas.

- **Result 1: Binomial Variance**

From basic probability theory (specifically, the variance of a binomial proportion), we know that the variance of an estimated probability \hat{p} based on a sample of size n is:

$$\text{Var}(\hat{p}) = \frac{p(1-p)}{n}$$

- **Result 2: Our Delta Method Formula**

The formula we just derived is:

$$\text{Var}(\hat{p}) \approx [p(1-p)]^2 \text{Var}(\hat{\eta})$$

These two formulas look different. You're right. But there's a missing piece. To reconcile them, we need to know: **what is the variance of the log-odds estimator, $\text{Var}(\hat{\eta})$?**

It turns out that from Maximum Likelihood Estimation theory, the variance of the log-odds estimator $\hat{\eta}$ in this simple intercept-only model is:

$$\text{Var}(\hat{\eta}) \approx \frac{1}{np(1-p)}$$

Now for the magic. Let's substitute this "missing piece" into our delta method formula:

$$\text{Var}(\hat{p}) \approx [p(1-p)]^2 \times \left(\frac{1}{np(1-p)} \right)$$

As you can see, one of the $p(1-p)$ terms in the squared bracket cancels with the term in the

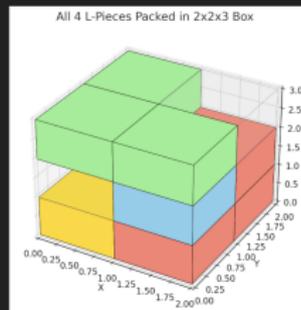
Many times, the models left me speechless, surprised

- ▶ I like math and visual puzzles; I argued with ChatGPT about some solutions; the model wrote Python code to work out the answers –**visually**

```
# Set axis limits and labels
ax.set_xlim(0, 2)
ax.set_ylim(0, 2)
ax.set_zlim(0, 3)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z (Layers)')
```

```
plt.tight_layout()
plt.show()
```

All 4 L-Pieces Packed In 2x2x3 Box



Boom! ★ Here's your complete and successful packing of the 2x2x3 box using 4 L-shaped pieces:

Frontier models can answer most of my homework and exam questions

- ▶ This question is difficult because MPH students need to think in terms of a mathematical model

Gemini 2.5 pro

Prompt: You are a health economist. Answer this question in about 150 words each. Base all your answers on the Grossman model: A new drug that eliminates the steady deterioration of health that accompanies aging has been discovered. However, the drug does not eliminate sudden events like heart attacks –caused by the consumption of trans fats, obesity, and a sedentary life– or accidents like being hit by a car.

- a) How does the cost of investing in health change after a person takes the drug?
- b) Would the health level of a person increase after taking the drug?
- c) Would a person run or jog more after taking the drug than before?

Example: Difficult midterm question

- ▶ ChatGPT 5.0 had a very good answer

ChatGTP 5 Thinking

Prompt: [Same prompt as previous one]

Answer: ... c) Ambiguous. Exercise is both an investment good (raises (H), lowers acute-event risk) and a consumption good (direct utility), but it has time and effort costs. With the drug, steady depreciation falls, so maintaining any given (H) requires less gross investment; that force pushes exercise demand down. Counter-vailing forces push it up: the lower user cost makes health capital cheaper, raising desired (H^*); because exercise is a productive input in generating (H), moving toward the higher target can increase exercise, especially during the transition. In the new steady state, gross exercise needed equals the maintenance requirement (δH^*). Since (δ) is lower but (H^*) is higher, (δH^*) can be greater or smaller than before, depending on how much (H^*) rises. If exercise mainly served to offset aging depreciation, it may fall; if it chiefly reduces acute risks (heart attacks from poor lifestyle), it may rise. Hence the prediction is theoretically indeterminate.

Example: Difficult midterm question

- ▶ Language too technical? Answer may look suspicious to faculty. But no problem for an LLM model

ChatGTP 5 Thinking

Prompt: Re-write the answer to c) at the level of a high-school student. Introduce at least 3 typos. Use simple examples without technical jargon.

Answer: It's not clear—could go either way. Think of jogging like watering a plant. Before the drug, your “health plant” dries out a little every day as you age, so you might jog 3 days a week just to keep it alive. After the drug, that steady drying stops. So you might need less **exercize** to stay at the same level—maybe jogging 2 days a week is enough now. But there's another side. Since your health doesn't fade as fast, every workout “lasts” longer. That can make you want an even bigger, healthier plant. You might decide to raise your goals—from jogging 20 minutes to 30—because the benefits stick around. For a while, you could jog more to reach that higher target. So, will you run more? **Mabye** yes during the “catch-up” period, maybe no once you've arrived. It depends on your goals and why you run—fitness, fun, or avoiding **accidnets** and heart issues. In short, it's ambiguous **becasue** both forces are at work.

Example: Difficult midterm question

- ▶ Even better, make it sound like you were listening in class. I use the glass analogy a lot

ChatGTP 5 Thinking

Prompt: Re-write the answer to c) at the level of a high-school student. Introduce at least 3 typos. Use the example of glass of water with a crack, representing depreciation. Age increases the crack. The drug makes the crack smaller.

Answer: Think of your health like a glass of water with a crack. Aging makes the crack bigger, so the water leaks out faster. You “pour” water back in by running or jogging. Before the drug, the crack is larger, so you might jog a lot just to keep the glass half full. After the drug, the crack gets smaller, so the water leaks slower. That means you could keep the same level with *less* exercize—mabye fewer runs each week.

But there's another option: since the water now stays in longer, you might try to fill the glass higher. To reach that new level, you'd jog *more* for a while, then settle down. In the long run, you could end up running more or less, depending on your goals.

Also, sudden “spills” (accidents or heart issues) can still happen, so running is still useful, just not a guarentee becasue life is messy and defintely unpredictable.

Example: Difficult midterm question

- ▶ Or if you feel creative, ask for a poem

ChatGTP 5 Thinking

Prompt: Using the first answer to c), the more technical answer, rewrite it again as a beautiful and lyrical 50-word poem in the style of Pablo Neruda. Make sure the poem captures the technical nuances of the answer without the jargon.

Answer:

After the pill, the slow leak softens.
To keep the garden green, fewer steps suffice.
Yet longing grows for taller trees,
so feet may fly, awhile, to plant more life.
Later, when roots settle,
some will run less, some more—
for storms still strike,
and joy, too, calls us onward.

What do I do now?

What I did and plan to do

- ▶ Understand how AI models work – in detail. The less people know how the models work, the more they use them and think they are “magical” (<https://journals.sagepub.com/doi/10.1177/00222429251314491>)
- ▶ **AI is going to impact our students**, many jobs will change, and they need to understand how LLMs models work because otherwise they won't be competitive
- ▶ Entry-level jobs are at higher risk; “augmentation” for other jobs
- ▶ Prepared a class on AI (HSMP 6635) to teach students how to better use the current models for learning and research
 - ▶ Module 1: AI foundations – LLM models (Weeks 1-2)
 - ▶ Module 2: Interacting with AI (APIs, etc) (Weeks 3-4)
 - ▶ Module 3: AI for research and summarization (Weeks 5-6)
 - ▶ Module 4: AI for data analysis (inc qualitative) (Weeks 7-10)
 - ▶ Module 5: Ethics, security, impact on jobs (Weeks 11-15)
- ▶ Prompt (and context) “engineering” is 60% of the class

What I did and plan to do

- ▶ I'm refining my AI tutoring "doppelganger"
- ▶ Maybe I can cover more material and students can learn at their own pace; ideally, with an API or a way to reduce answers' randomness (e.g., AI Studio)
- ▶ Will try more challenging homework
- ▶ But, more than ever, I need to **rely on in-person midterms and final to assess learning**
- ▶ My PhD students are all using LLMs, but I hope they do not become too dependent on it. Their coding productivity for sure has increased

Many words of caution

- ▶ Education is quite resilient to new technology
- ▶ Remember Wikipedia, iPads, MOOCs, “a laptop for every student,” online learning?
- ▶ We have had books and teachers for centuries; most who want to learn something have access to the information
- ▶ Fact: **Human learning has not changed for thousands of years, we still need to sit down and think and “do” to learn**
- ▶ Reich (2022) “Failure to Disrupt” is a good guide (<https://tsl.mit.edu/books/failure-to-disrupt/>)
- ▶ **Heads in the sand is not a good strategy**

Thanks

- ▶ Happy to answer questions
`marcelo.perrailon@cuanschutz.edu`

Google's Studio AI allows for easy system prompts and sampling controls

The screenshot displays the Google AI Studio interface. On the left is a navigation sidebar with options: Studio, Chat, Stream, Generate media, Build, History, Dashboard, and Documentation. The main area is titled "Chat prompt" and features the "Google AI Studio" logo. Below the logo is a "What's new" section with four cards: "Try Nano Banana" (Gemini 2.5 Flash image generation), "Fetch information with URL context" (real-time web link info), "Generate native speech with Gemini" (text-to-speech), and "Talk to Gemini live" (real-time dialogue). At the bottom of the main area, a text input field contains "Generate Python code for a simple calculator app" and a "Run" button. On the right, a "Run settings" panel is open, showing "Media resolution" set to "Default", "Thinking" mode, "Thinking mode" toggle, "Set thinking budget" slider at 2546, "Tools" section with "Structured output", "Code execution", and "Function calling" toggles, "Grounding with Google Search" toggle, "URL context" toggle, "Advanced settings" section, "Safety settings" toggle, "Add stop sequence" button, "Output length" set to 65536, and "Top P" slider at 0.95.