The Building Blocks of Large Language Models (LLMs)

Marcelo Coca Perraillon

University of Colorado
Anschutz Medical Campus

CU - Data Week
February 2026

Outline

- From linear regression to neural networks
- Hypothesis testing versus prediction
- Pre-training
- Post-training
- LLM models as impersonators
- Alignment problems

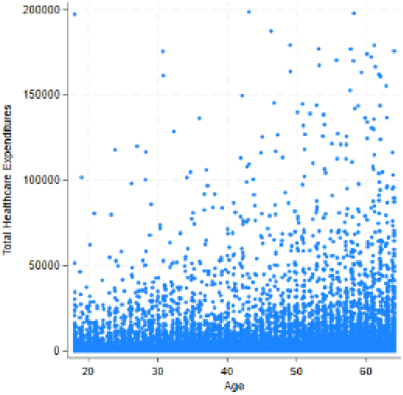
- You can hear NotebookLM deliver this lecture here:
  `https://perraillon.com/files/LLM_intro_Perraillon_Podcast.m4a`

Focus

- Large Language Models (LLMs) used in generative AI
- 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** models, a branch of cognitive science and later machine learning; machine learning was developed in computer science but now part of statistics... It gets confusing
- 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). **See a pattern?**



Simplifying the problem

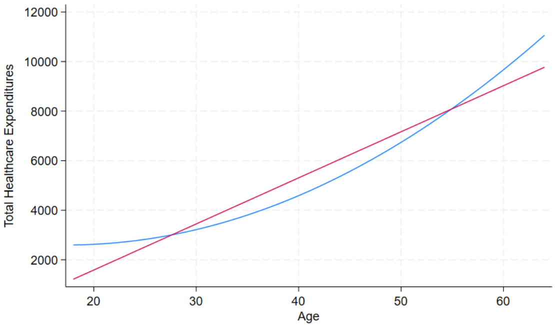
- We usually give the problem a structure and make assumptions about the form of the relationship, like:

  $\text{expenditure}_i = \beta_0 + \beta_1 \text{age}_i + \epsilon_i$
- 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 α_1, α_1, α_2 can be **estimated** using many techniques, like maximum likelihood estimation or least squares
- **Key**: We "reduced" or summarized a complex, hard to see relationship into simple curves that can be easily **interpreted**
- (Non-parametric models relax the structure we just imposed)

Plots of model predictions

- We could say that the pattern is now "**encoded**" in the parameters
- We could borrow language from machine learning to say that the models have "**learned**" the pattern; they have been "**trained**"



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**
- Inference is now the process of **making predictions using other data not used for training**
- We can make our model **hallucinate**. What is the predicted healthcare expenditure of a 200-year old person? It's

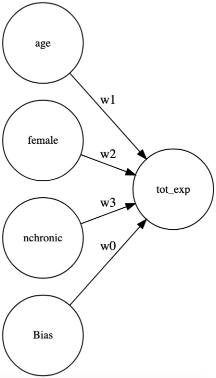
  $E[\hat{y}_i|x_i] = 3775.907 - 135.4872 * 200 + 3.895561 * (200^2) = \$132,501$
- We could add **interactions**, more predictors like the number of chronic conditions and so on to improve predictions

Neural networks

- Neural networks preceded the symbolic approach to AI; they were first introduced in 1943 (McCulloch and Pitts) and further developed in 1958 (Rosenblatt)
- In 1943, the word "computer" meant a person who performed calculations; NNs were pen and paper things
- Neural networks are based on simplified ideas about neurons, especially about the strength of (inter)connections and neurons "firing" when sufficiently stimulated
- For now, **think of neural networks as an alternative and flexible way of making predictions**
- Flexible in the sense that NNs do not impose a rigid structure on the relationship between covariates and outcome

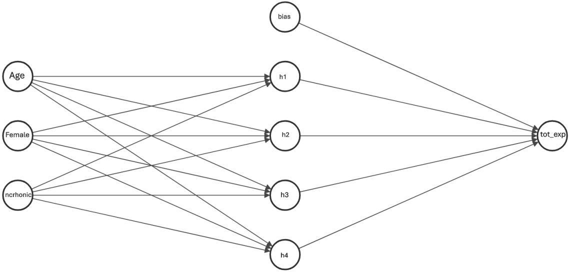
Linear regression as a single-layer perceptron

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



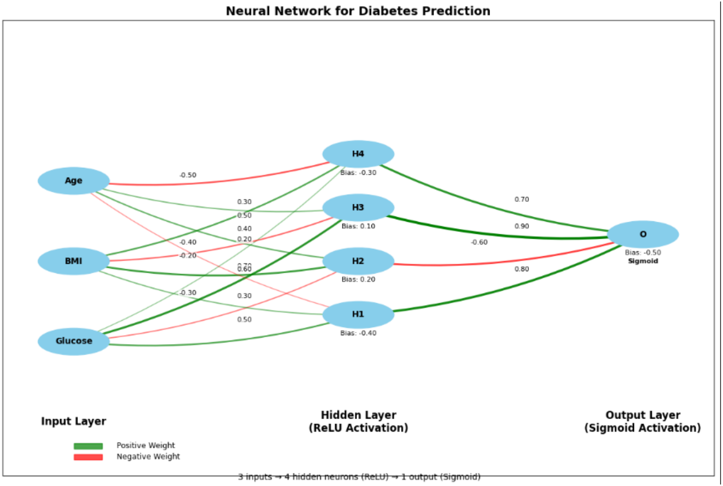
Adding complexity

- 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
- The NN is used to make predictions, **but no parameter has a clear meaning**



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

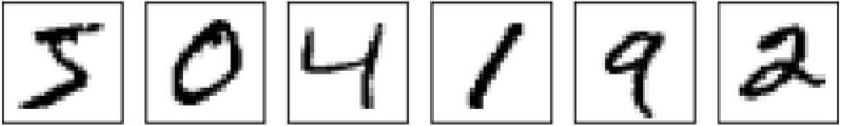


Figure: Source: http://neuralnetworksanddeeplearning.com/chap1.html

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

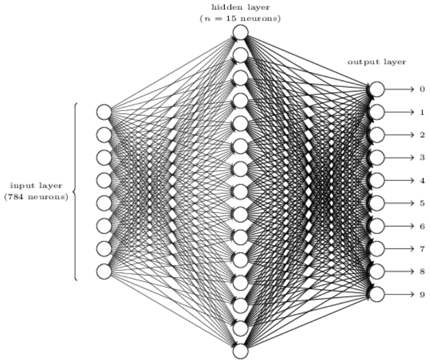


Figure: Source: http://neuralnetworksanddeeplearning.com/chap1.html

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 the Natural Language Processing field
- 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

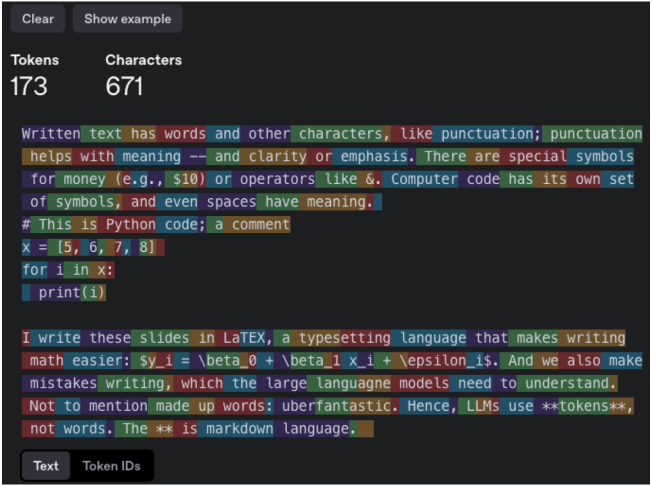


Figure: Source: `https://platform.openai.com/tokenizer`

Vocabulary

- All tokens form the model **vocabulary**
- Models have a vocabulary of 120K to 200K tokens
- So just to represent the vocabulary (so called embeddings), it takes about $200000 \times 2880 = 576,000,000$ **parameters to be estimated**
- 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**

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

- 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**

Models capture semantic meaning

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

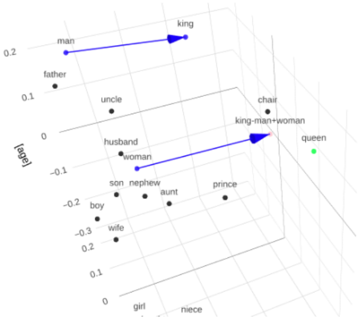


Figure 5: Analogy by vector arithmetic: "man" is to "king" as "woman" is to "king − man + woman" = "queen".

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

Scaling laws

- Scaling laws mostly explain 1) why models are better now and 2) the race to increase the size of the models

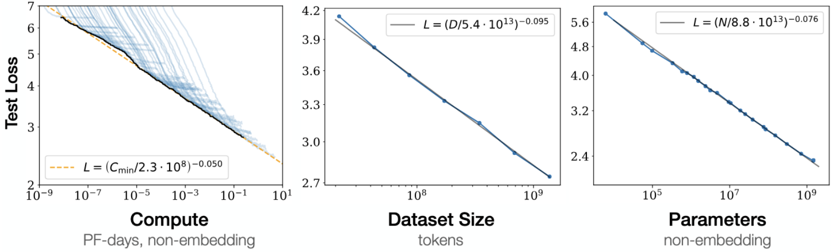


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

Figure: Kaplan et al., (2020), https://arxiv.org/pdf/2001.08361

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`

Post-trained LLM – the “magic” and the mystery

- A pre-trained LLM is a kind of **stochastic parrot**
- The pre-trained model is **post-trained**
- 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
- The LLM model learns to do useful things in the post-training stage
- Post-training changes the value of the learned parameters

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; they just auto-complete

- **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

LLMs as actors or impersonators

- One useful mental model for LLMs is that they are actors, impersonators of humans and things humans have written or said or filmed
- 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

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 at 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`)

Putting all the pieces together

- LLM models are prediction tools
- The way questions and tasks are framed (**context**) impacts predictions so "**prompt engineering**" is still important
- Whether mimicking intelligence and more scaling will result in truly autonomous tools is an open question – the hype is getting ahead of reality
- **Human-in-the-loop-approach**: you always need to check the output is correct, and that means that AI often does **not** save any time
- Companies are solving basic problems by adding tools, like checking LLM's math or giving the models access to databases or the web
- This is a good guide on choosing models: `https://www.oneusefulthing.org/p/an-opinionated-guide-to-using-ai`

Suggestions

- Evaluate models for things you know well. It's the best way to assess capabilities
- Be careful with leading questions
- Use "anti-sycophant" prompts. The post-training makes some models too agreeable, which is not great for objectivity
- You can't trust model outputs, you need to verify the information
- Learn prompt engineering tricks but **they will be less important over time**. See for example: `https://cloud.google.com/discover/what-is-prompt-engineering`
- LLM models change quickly, what is a problem today may not be a problem tomorrow

Thanks

- Happy to answer questions
  `marcelo.perraillon@cuanschutz.edu`
- I may offer a semester-long class on AI: HSMP 6635
- More material on my website: `https://perraillon.com/teaching.html`