Sanjeev Arora

Parrots No More: How AI Models Learn, Reason, and Self-Improve

On Thursday, May 22, Sanjeev Arora joined Markus' Academy for a conversation on Parrots No More: How AI Models Learn, Reason, and Self-Improve. Arora is the Charles C. Fitzmorris Professor of Computer Science and the Director of Princeton Language and Intelligence at Princeton University.

A few highlights from the discussion.[1]

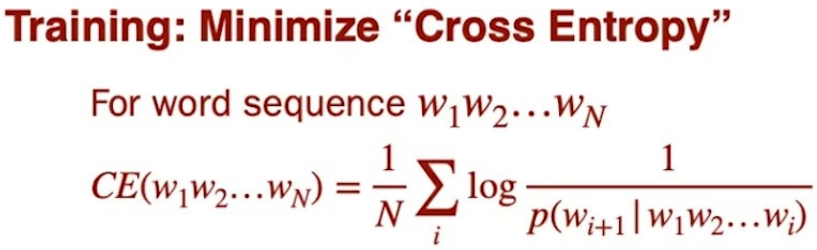
- **A summary in four bullets**
  - LLMs don't just memorize their training data—they learn to combine abstract skills in new ways, showing signs of metacognition ("thinking about thinking")
  - Synthetic data—generated by another LLM—improves the quality of a model's training data. Human data need not be the gold standard!
  - Post-training (fine-tuning on curated Q&A or reinforcement learning) teaches models how to access and apply their internal knowledge more effectively
  - Building on these ideas, the talk covered six current AI techniques: (1) extrapolation of scaling laws for training, (2) multimodal models, (3) models helping improve future models, (4) distillation, (5) self-improvement loops, and (6) AI agents.

- [0:00] **Markus' introduction**
  - One can try to connect AI/machine learning methods and the econometrics/statistics framework economists are more familiar with
  - In both fields there is an overfitting problem. AI/LLMs train on samples and use techniques like shrinkage (smaller models, fewer layers), just as econometrics relies on in- and out-of-sample validation and shrinkage to limit the number of regressors or polynomial terms (e.g., HP filters)
  - Using synthetic data in AI can be seen as analogous to bootstrapping or simulated method of moments
  - Small models in AI often prepare data for large models—similar to overidentified moment matching in economics, where small models generate moments and large models match them
  - Metacognition in AI connects to basis function discovery—learning functions of the state variables that simplify complex decision problems

- [4:15] **Next-word prediction is more powerful than it seems**
  - Many initially dismissed LLMs as more like autocomplete than as intelligence. Not too far from word processors.

[1] Summary produced by Pablo Balsinde (PhD student, Stockholm School of Economics)

  - The skepticism of LLMs was exemplified by Bender et al. (2021), which dubbed them “Stochastic Parrots,” implying that LLMs basically parrotted back what they saw in the training data.
  - Another example of such skepticism is Chiang (2024) who argues that AI will never be able to make art. This article seems to suffer from popular misconception that LLMs only have two ways to write: (1) averaging past writing, or (2) mimicking specific authors. As we will see, this is not true.
  - The basic notion of next-word prediction (NWP) using probabilistic models dates back to Claude Shannon in the 1950s. The modern era began in 2013 and took off with the invention of the transformer architecture in 2017 (Vaswani et al., 2017)
  - The training objective of these prediction models is to minimize an information-theoretic measure of cross-entropy: the average of the log of the reciprocals of the predicted probabilities for the next word in a sequence



Training: Minimize “Cross Entropy”

For word sequence $w_1 w_2 \ldots w_N$

$$CE(w_1 w_2 \ldots w_N) = \frac{1}{N} \sum_i \log \frac{1}{p(w_{i+1} \mid w_1 w_2 \ldots w_i)}$$

  - Chaining next-word predictions yields probabilistic text generation; randomness is the source of diversity but also sometimes hallucinations
  - Models are evaluated on held-out text, with better models predicting rare or novel sequences more accurately
  - Researchers turned to using NWP to train language models because they needed a scalable way to train massive deep neural networks. Today’s deep nets of ~1 trillion parameters need ~20 trillion words of training data
  - Researchers have tried other approaches to language modeling, but NWP seems best at improving model capabilities as we grow the model size. NWP has shortcomings but they can be patched by other stages in training

- [18:02] **Skills and implications for originality**
  - The skills that models pick up have no formal definition, but can be thought of as programming procedures. Clearly, even if the term “metaphor” did not exist, the model would figure out the concept of a metaphor (and how to create new ones) after seeing examples of metaphors in text. Complex reasoning in this sense is procedural.
  - Yu et al. (2023) developed “Skillmix evaluation” to understand whether models were doing more than copying from training data. They generated queries that asked models to exhibit random combinations of linguistic skills (from Wikipedia, for example “metaphor” or “ad hominem attack”)
  - The number of combinations of k skills scales exponentially in k. A simple calculation shows that for large k, such as 5 or 6, the number of such combinations is too large and so most of them could not have occurred in the training data. The skillmix evaluation showed that models in 2023 were indeed capable of generating text correctly combining up to 6 skills in random

    combinations, and so we must conclude that most of these pieces of text were not present in the training set.
    - To understand how an LLM acquires the ability to understand text it did not see during training, it is helpful to consider pre-training. Using automated means (e.g., tiny LLMs) training data is partitioned into categories like code, webpages, literature, news, Wikipedia
    - Training then happens in multiple phases, with each using a different mix from data categories. This helps because there may be positive or negative synergies across data types—e.g., programming data can improve general reasoning and STEM capabilities
    - A neural network can be thought of as a population of small "experts" in individual concepts whose interactions allow the model to gain understanding
    - Prediction models are trained using gradient-based (calculus) methods. The gradient can be seen as message-passing across these "experts," adjusting the strength of their connections to favor those that contribute to predicting the actual next word
    - The key point is that when exposed to new text (i.e., not seen during training) the model extracts its semantics using this internal web of "expert interactions"—not from explicit memorization of the training data. The model doesn't have enough parameters to memorize the massive training set.
    - The post-training process turns a model from a next-word predictor into a chat agent. One can think of pre-training as primary school: creating an inner web of knowledge and skills but which is error prone
    - Post-training is like high school or college. It uses specialized (and expensive) data to refine the model's ability to access its internal knowledge while adding conversational and reasoning skills. It is done in several steps:
    - **(1) Fine-tuning on Q&A data:** Thanks to the model's exposure to the vast training set, its internal knowledge store is also vast. Thus one must train a model with question-answer pairs of high quality —training on answers that are short and shallow makes the model lazy and unhelpful, since it is in effect being taught to give lower quality answers than what it is capable of.   As a result, this fine-tuning stage requires high quality data, usually with humans in the loop.
    - **(2) Reinforcement learning:** the idea is to improve Q&A capability using human feedback; the model generates multiple answers to a query which are then rated by qualified humans. It is also possible to replace human graders by LLM "graders" trained using human answers.
    - **(3) Safety training:** also uses reinforcement learning, but with a focus on ethics and safety (e.g., learning to spot inappropriate/malicious questions)

- **[41:27] Metacognition: are LLMs aware of how they are solving tasks?**
    - In humans, metacognition is the ability to reason about one's own thought processes
    - Metacognition or "thinking about thinking" appears to arise automatically in LLMs (Didolkar et al., 2024). The slides had some examples of their ability to automatically construct a taxonomy of different cases and skills such as "synonym substitution technique" or "tone moderation adjustment".

  - We can leverage LLMs' metacognition to create synthetic data that improves training. Kaur et al. (2024) developed a procedure to create high-quality Q&A pairs to train a chat model. They asked GPT-4o to list instruction-following skills 4,000 times. For each iteration, a pair of these skills was randomly selected, and GPT-4o was asked to generate a user query and chatbot answer that demonstrated both skills This yielded 4000 Q&A pairs, which were used to train a chat model based on Lamma 3. Despite using far fewer training examples—orders of magnitude less than typical—the model achieved performance comparable to frontier models
  - The key was the quality of the synthetic data created: it was designed to engage the model's metacognitive capacities—training it to dig deeper into its inner web of knowledge
  - Note that here synthetic data was meant to supplement data not in terms of quantity, but in terms of quality. This is an example of how LLM-generated text can be superior to human data for certain types of training.

- [46:07] **Current AI techniques**
  - "The biggest lesson from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin" ("Bitter lesson" Sutton, 2019)
  - Human intuition can be misleading. In particular, methods that use humans in the training loops tend not to scale well. Data and compute increase at an exponential rate, so any method that leverages them will usually be most effective.
  - Scaling laws —where (in log scale) model size linearly improves quality— are an empirical finding. Training recipes that guide things like dataset composition and the learning rate (the hyperparameter that controls how much the model's parameters are adjusted in each training step) are being extrapolated from scaling experiments
  - The latest models are multimodal (language + vision + audio + motion). Vision encoders convert images, audio, and motion into semantic tokens, which the LLM learns to process similarly to words
  - This approach is promising in robotics; a longstanding issue in robotics was brittleness—robots trained in one environment would fail in others. Giving robots a language core helps them reason about how to adapt to changed environments.
  - Models are helping improve future models in many ways: (1) Small models help clean up data for training larger models (simple example: fixing LaTeX or html rendering issues;), (2) current models are used to rewrite human-written text to improve the training data of the next model, and (3) current models generate specialized training data for the next model (as discussed before)
  - The main reason for why the price of AI has dropped so much is "distillation," where a large, capable model generates massive amounts of training data for a smaller model. The net effect is that you get a small model whose performance is close to that of the large model. For example, GPT-4o-mini was probably trained from GPT-4o, yet performs better than what traditional scaling laws would predict given its size.

  - Self-improvement loops is a key idea I want you to pay attention to, because this concept not yet very well-known in the general public. These originated in game-playing models like AlphaGo. Reasoning models like o1 and DeepSeek R1 were trained in this way, as well as math theorem-provers like AlphaProof. You start with a large question bank (possibly human-written), with questions varying in difficulty, and answers being “auto-gradeable” (usually using a small LLM). The model gets multiple attempts to solve each question. The correct answers are used to train the model further —this wouldn’t make any sense in the “parrot” view of LLM but actually helps reinforce good “reasoning patterns,” and results in improved performance on questions and topics not seen in training. In effect, the model is improving its own performance without relying on human solutions.

- [1:03:34] **Peeking 5-7 years ahead**
  - Expect to see weird phenomena. Burns et al. (2023) showed that a strong “student” AI model can outperform a “weak” AI teacher. The weak AI tries to answer all questions but may get only 60% right (say). It shows its answers to half of the questions to the stronger AI. The strong AI ends up learning the task more effectively (say to 70% accuracy) just by training on the weak AI’s outputs.
  - This result illustrates that once a model reaches a certain capacity it gains some level of understanding—even from noisy or flawed data. The strong AI learned the task better from observing the weak AI’s mistakes. Recent work of Arora and coauthors shows that this “weak to strong” phenomenon also happens with simpler models.
  - There is increasing talk about superAGI, and discussions are beginning on how to define superalignment (“how should we teach a superAGI to behave?”)
  - Researchers have noticed an increase in the hallucination rate “reasoning” models that trained with self-improvement loops. It is unclear if this is inherent or not.
  - The language used for training plays a role, but models often generalize capabilities across languages. This generalization across languages is the dominant way in which models acquire capabilities in under-resourced languages

Timestamps:

[0:00] **Markus’ introduction**
[4:15] **Next-word prediction is more powerful than it seems**
[18:02] **Skills and implications for originality**
[41:27] **Metacognition: are LLMs aware of how they are solving tasks?**
[46:07] **Current AI techniques**
[1:03:34] **Peeking 5-7 years ahead**