Harnessing the Power of AI and LLMs in Mathematics

Current Trends and Future Directions

Marly Gotti

October 15, 2024

Simple Words Seminar

Objective

- Understand the concept of Large Language Models (LLMs) and the key terms related to their implementation.
- Learn about the history of LLMs.
- Become familiar with tools related to LLMs that can assist in mathematical research.

Overview

LLaMA by Meta

FALCON 180B

Claude 3

ChatGPT

Rubik's AI

Copilot

Gemini Google

IBM Granite

MISTRAL AI_

Outline

Training an LLM

What are LLMs?

Evolution of LLMs

LLMs in Mathematics

Training an LLM

Task: Predict the next word in a fragment.

Training an LLM

The Road Not Taken by Robert Frost

Two roads diverged in a yellow wood
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;

Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same,
(...)

Training an LLM (Cont.)

1. Text selection: *The Road Not Taken* by Robert Frost.
2. Tokenization: word-level tokenization
   $\{$*two*, *roads*, *diverged*, *in*, ...$\}$.
3. Training Objective: next word prediction.
   Two roads diverged in a **yellow** _____

   > Two roads diverged in a **yellow** wood,
   > And sorry I could not travel both
   > And be one traveler, long I stood
   > And looked down one as far as I could
   > To where it bent in the undergrowth;
   > (...)

   Two roads diverged in a **yellow** wood.

Training an LLM (Cont.)

3. Training objective: next word prediction.
   One traveler looked down **as** _____

   Two roads diverged in a yellow wood,
   And sorry I could not travel both
   And be one traveler, long I stood
   And looked down one **as** far **as** I could
   To where it bent in the undergrowth;

   Then took the other, **as** just **as** fair,
   And having perhaps the better claim,
   Because it was grassy and wanted wear;
   Though as for that the passing there
   Had worn them really about the same,
   (...)

Training an LLM (Cont.)

3. Training objective: next word prediction.
   Context is important!

   Two roads diverged in a yellow wood,
   And sorry I could not travel both
   And be one traveler, long I stood
   And looked down one as far as I could
   To where it bent in the undergrowth;

   Then took the other, as just as fair,
   And having perhaps the better claim,
   Because it was grassy and wanted wear;
   Though as for that the passing there
   Had worn them really about the same,
   (...)

What are LLMs?

What are LLMs?

- LLMs (Large Language Models) are AI models designed to understand, generate, and manipulate human language.
- They are built on neural network architectures and trained on large-scale text data.
- LLMs use self-supervised learning, where they learn patterns in language by predicting the next word in a sentence.
- They are capable of performing a wide range of tasks such as
  - Text generation (e.g., writing essays, articles)
  - Question answering
  - Translation
  - Summarization
- LLMs are pre-trained on massive datasets and can be fine-tuned for specific applications.

Evolution of LLMs

Evolution of LLMs

The Beginning of AI and NLP (1950s–1970s)

- **Early NLP Programs**: ELIZA (1964) - A rule-based conversational agent.
- **N-grams**: Used for predicting the next word based on the previous “n” words.

Statistical NLP (1980s–1990s)

- **Probabilistic Models**:
  - Hidden Markov Models (HMMs)
  - Maximum Entropy Models
- **Statistical Machine Translation**: IBM’s models improved translation accuracy using large datasets.

Evolution of LLMs (Cont.)

Neural Networks in NLP (1990s–2000s)

- Early attempts with **Feedforward Neural Networks** in NLP.
- **Word Embeddings**:
  - **Word2Vec** (2013): Representing words in continuous vector space.

The Shift to Deep Learning (2010s)

- The rise of **deep learning** with neural networks.
- **RNNs and LSTMs** for sequence modeling, capturing dependencies across long text sequences.

Evolution of LLMs (Cont.)

The Transformer Architecture (2017)

- **Attention Is All You Need (2017)**: Vaswani et al. introduce the Transformer.
- **Self-Attention Mechanism**:
  - Enables models to focus on different parts of a text sequence in parallel.
  - Impact: Improved scalability and performance in NLP tasks.

Pre-trained Language Models (2018–2020)

- **BERT (2018)**: Bidirectional, pre-trained using both left and right context.
- **GPT (2018)**:
  - Introduces the **pre-train, fine-tune** paradigm.
  - GPT-2 (2019) with 1.5 billion parameters.
- **T5 (2019)**: Unifies NLP tasks as text-to-text generation.

Evolution of LLMs (Cont.)

The Rise of LLMs (2020s)

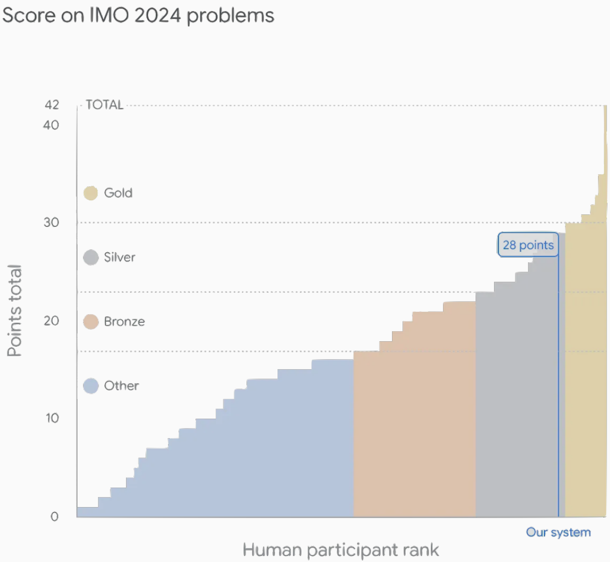
- **GPT-3 (2020)**:
  - 175 billion parameters.
  - Strong performance in text generation, translation, and coding.
- **GPT-4 (2023)**:
  - Multimodal capabilities (text and image understanding).
  - OpenAI's **DALL·E** and **CLIP**.

Scaling Laws and Ethical Considerations

- **Scaling Laws**: Performance improves with larger models, but so do the computational costs.
- **Ethical Concerns**:
  - Bias, misinformation, and energy consumption.
- **Ongoing research**: Reducing the environmental footprint and increasing model efficiency.

LLMs in Mathematics

AI achieves top-score silver-medal at IMO



AI achieves top-score silver-medal at IMO

- **AlphaProof:** a system that trains itself to prove mathematical statements in the formal language Lean.
- **AlphaGeometry 2:** a hybrid system in which the language model was trained from scratch on an order of magnitude more of synthetic data than its predecessor.

What is Lean?

- **Lean** is an open-source **interactive theorem prover** and **functional programming language** developed for formalizing mathematical proofs and verifying software correctness.
- It is used to write definitions, theorems, and proofs in a formal language that can be verified for logical correctness.
- Lean is designed for use in **formal reasoning** across mathematics, logic, and computer science.

Key Features of Lean

- **Formal Proofs**: Lean is used to create formal proofs that are verifiable and free from logical errors.
- **Interactive Theorem Proving**: Users can incrementally build and verify proofs with real-time feedback.
- **Mathlib**: Lean includes an extensive library of formalized mathematics, called **Mathlib**, with thousands of theorems and definitions.
- **Functional Programming**: Lean supports functional programming to define and manipulate mathematical objects.
- **Automation**: Lean includes tactics and automation tools to assist in proof construction.

LLMs Tools

- GPT-4 (o1-preview) by OpenAI
- GitHub Copilot with Lean Extension
- OpenAI Codex
- Wolfram Alpha
- Gemini
- Nova by Rubik's AI

Live Example

Definition: An additive submonoid of the nonnegative rationals is called a Puiseux monoid.

Definition: Let M be a Puiseux monoid. An element $a \in M \setminus \{0\}$ is an atom (or irreducible) if, for all $u, v \in M$ the fact that $a = u + v$ implies that either $u = 0$ or $v = 0$. Let $\mathcal{A}(M)$ denote the set of all atoms of M, and call M atomic if $M = \langle \mathcal{A}(M) \rangle$.

Example: Let $M := \mathbb{N}_0 \setminus \{1\} := \{0, 2, 3, 4, \dots\}$ is a Puiseux monoid.

- 5 is not an atom because $5 = 2 + 3$.
- The atoms are 2 and 3.
- Each element of M can be written as a finite sum of copies of 2 and 3, so M is atomic and we write $M = \langle 2, 3 \rangle$.

Live Example (Cont.)

Prompts:

- Give me an example of a Puiseux monoid.
- Give me an example of an atomic Puiseux monoid.
- Give me an example of a Puisuex monoid that is not atomic.

Thank You!