ATALINK

Series: Trí tuệ nhân tạo (Artificial Intelligence)

Phần 1/4: Từ thai nghén đến tạo sinh

Feb 28, 2026

Agenda

1. Các giai đoạn phát triển chính của AI
2. Các bước ngoặt trong phát triển AI
3. AlexNet: Phát súng khởi đầu của Deep Learning
4. VGG và ResNet: Cải tiến sau sự ra đời của AlexNet
5. Transformer: Động cơ phản lực của AI
6. LLM – Mô hình Ngôn ngữ lớn
7. LMM – Mô hình Đa phương thức lớn
8. AI Agent – AI Platform

1 Các giai đoạn phát triển chính của AI

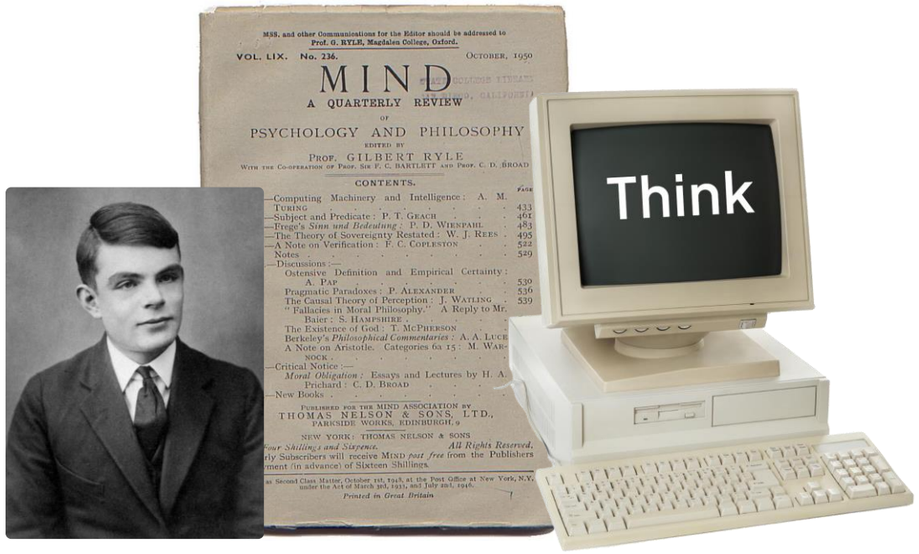
Tóm tắt các giai đoạn chính

STT	Giai đoạn	Đặc điểm chính
1	**1950s**	Hình thành khái niệm và Phép thử Turing
2	**1956 – 1974**	Hội nghị Dartmouth - Khởi đầu chính thức & Thời kỳ hoàng kim #1
3	**1970s – 1980s**	Các "Mùa đông AI" do thiếu kinh phí và công nghệ
4	**1990s – 2010s**	Machine Learning và chiến thắng cờ vua
5	**2012 – Nay**	Deep Learning, Big Data và Trí tuệ nhân tạo tạo sinh (Generative AI)

Lịch sử AI không chỉ là câu chuyện về công nghệ, mà là nỗ lực của con người trong việc giải mã chính trí thông minh của mình.

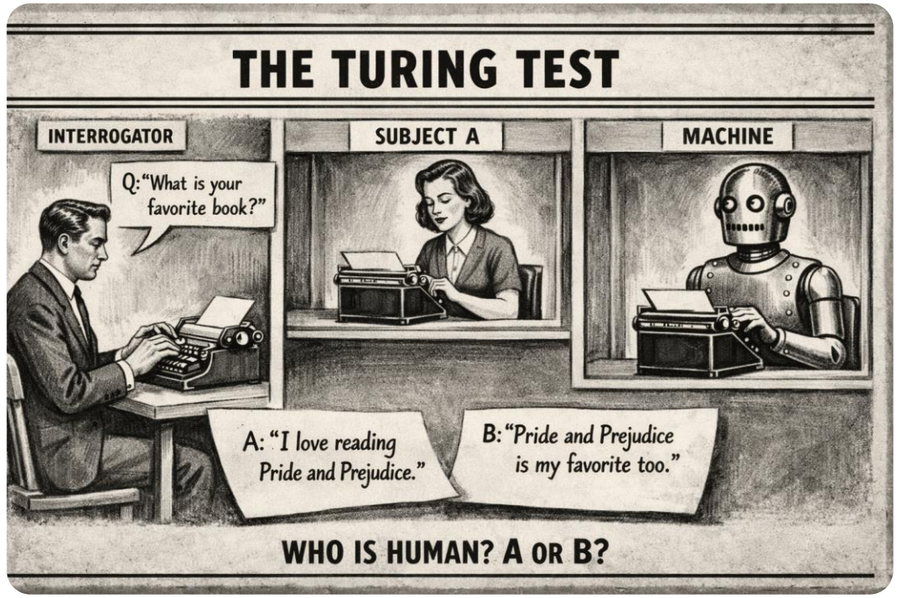
1. Thời kỳ “Thai nghén” (Trước năm 1950)

Ý tưởng về những cỗ máy biết suy nghĩ đã tồn tại từ lâu trong triết học và văn học. Tuy nhiên, bước ngoặt khoa học thực sự đến từ Alan Turing.



1950: Turing xuất bản bài báo “Computing Machinery and Intelligence”

Đưa ra câu hỏi: “**Máy móc có thể suy nghĩ không?**”



Phép thử Turing (Turing Test)

Ông đề xuất Phép thử Turing (Turing Test) để đánh giá trí thông minh của máy tính.

2. Sự ra đời và Thời kỳ Hoàng kim đầu tiên (1956 – 1974)

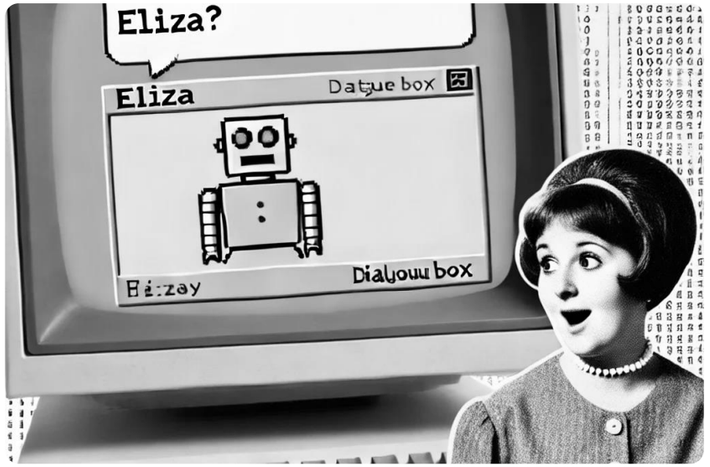
Cụm từ “Trí tuệ nhân tạo” chính thức được khai sinh.

1956: Hội nghị Dartmouth

Hội nghị Dartmouth diễn ra, quy tụ những bộ óc vĩ đại như John McCarthy, Marvin Minsky. Đây được coi là cột mốc khai sinh ra ngành AI.

Sự lạc quan

Các nhà khoa học lạc quan cho rằng máy móc sẽ có khả năng làm mọi việc của con người chỉ trong vòng 20 năm.



Các chương trình tiên phong

Các chương trình như ELIZA (chatbot đầu tiên) hay máy giải toán hình học đã tạo nên cơn sốt.

3. Các “Mùa đông AI” do thiếu kinh phí và công nghệ

“Mùa đông AI” lần thứ nhất (1974 – 1980)

Sự lạc quan quá mức dẫn đến thất vọng khi công nghệ thời đó không đáp ứng được kỳ vọng.

Hạn chế

- Sức mạnh xử lý của máy tính quá yếu
- Bộ nhớ đắt đỏ
- Các thuật toán chưa đủ sâu

Hậu quả

Chính phủ và các tổ chức cắt giảm ngân sách nghiên cứu, khiến ngành AI rơi vào trạng thái đóng băng.

Hệ chuyên gia và Mùa đông thứ hai (1980 – 1993)

Hệ chuyên gia (Expert Systems):

AI hồi sinh thông qua các chương trình mô phỏng kỹ năng ra quyết định của các chuyên gia con người (như chẩn đoán y tế).

Tuy nhiên, các hệ thống này quá tốn kém để duy trì và khó cập nhật, dẫn đến **“Mùa đông AI” lần thứ hai** vào cuối những năm 80.

4. Sự trỗi dậy của Machine Learning (1993 – 2011)

AI bắt đầu chuyển hướng từ việc “dạy máy quy tắc” sang việc **"để máy tự học từ dữ liệu"**.

1997: Siêu máy tính Deep Blue của IBM

Đánh bại nhà vô địch cờ vua thế giới Garry Kasparov.
Đây là một cú hích tâm lý cực lớn.

Sự xuất hiện của Big Data

Internet phát triển tạo ra nguồn dữ liệu khổng lồ,
là “thức ăn” hoàn hảo cho các thuật toán học máy.

5. Kỷ nguyên Deep Learning và AI tạo sinh (2012 – Nay)

Đây là giai đoạn bùng nổ mạnh mẽ nhất nhờ vào sự phát triển của mạng thần kinh nhân tạo (Neural Networks) và chip đồ họa (GPU) mạnh mẽ.

2012: AlexNet

AlexNet giành chiến thắng trong cuộc thi nhận diện hình ảnh, mở ra kỷ nguyên **Deep Learning**.

2016: AlphaGo

AlphaGo của Google DeepMind đánh bại Lee Sedol trong trò chơi cờ vây, một cột mốc cho thấy AI có thể xử lý các bài toán có độ phức tạp cực cao.

2017: Transformer

Kiến trúc **Transformer** ra đời, thay thế các mô hình xử lý tuần tự trước đó, cho phép AI hiểu ngữ cảnh toàn cục và huấn luyện trên dữ liệu quy mô lớn — nền tảng của LLMs.

2022 – Nay: Mô hình ngôn ngữ lớn (LLM)

Sự ra đời của các mô hình ngôn ngữ lớn (LLM) như **ChatGPT**, **Claude** và các công cụ tạo ảnh như **Midjourney**, giúp AI trở nên phổ biến và gần gũi với mọi cá nhân.

2

Các bước ngoặt trong phát triển AI

Tóm lược sự chuyển dịch

Giai đoạn	Cơ chế chính	Trọng tâm
Sơ khai	Logic & Quy tắc	Giải quyết bài toán đố
Trung hạn	Thống kê & Machine Learning	Nhận diện mẫu (Pattern)
Hiện tại	Deep Learning & Transformer	Sáng tạo nội dung (Generative)

Sự thật thú vị:

Bạn có biết rằng Geoffrey Hinton (người đặt nền móng cho Deep Learning ở bước 3) đã được trao giải **Nobel Vật lý** năm 2024 vì những đóng góp này không?

Điều đó cho thấy AI đã trở thành một phần không thể thiếu của khoa học cơ bản.

Các bước ngoặt chính trong phát triển AI

Để hiểu rõ tại sao AI lại bùng nổ như hiện nay, chúng ta cần nhìn vào những cú hích (milestones) thay đổi hoàn toàn cuộc chơi. Nếu lịch sử AI là một con đường, thì đây là những "khúc cua" quan trọng nhất:

Phép thử Turing (1950) – Đặt ra "Định nghĩa"

Trước khi có máy tính hiện đại, Alan Turing đã đặt nền móng triết học. Thay vì hỏi "Máy tính có linh hồn không?", ông hỏi: "Máy tính có thể mô phỏng hành vi con người đến mức không thể phân biệt được không?".

Ý nghĩa: Chuyển trọng tâm từ lý thuyết viển vông sang việc kiểm chứng bằng thực nghiệm.

Hội thảo Dartmouth (1956) – Khai sinh cái tên

Đây là lúc thuật ngữ "Artificial Intelligence" chính thức ra đời.

Bước ngoặt: Những bộ óc vĩ đại như John McCarthy và Marvin Minsky tin rằng mọi khía cạnh của trí thông minh đều có thể được mô tả chính xác đến mức máy móc có thể mô phỏng được. Nó biến AI từ sở thích cá nhân thành một ngành khoa học thực thụ.

Sự trỗi dậy của Mạng Neural và Lan truyền ngược (Backpropagation – 1986)

Trong một thời gian dài, AI dựa trên các quy tắc "Nếu-Thì" cứng nhắc (Logic).

Bước ngoặt: Geoffrey Hinton và các cộng sự phổ biến thuật toán Backpropagation.

Ý nghĩa: Giúp các mạng thần kinh nhân tạo "học" được từ lỗi sai của chính mình. Đây chính là tiền thân của Deep Learning ngày nay.

Các bước ngoặt chính trong phát triển AI

Deep Blue đánh bại Garry Kasparov (1997) – Biểu tượng sức mạnh

Lần đầu tiên một nhà vô địch cờ vua thế giới bị đánh bại bởi máy tính trong một trận đấu chính thức.

Ý nghĩa: Chứng minh rằng với sức mạnh tính toán đủ lớn (Brute force), máy tính có thể vượt qua trí tuệ con người trong các bài toán logic phức tạp.

Cuộc cách mạng Deep Learning & ImageNet (2012)

Đây là "vụ nổ Big Bang"của AI hiện đại. Tại cuộc thi nhận diện hình ảnh ImageNet, mô hình **AlexNet** (sử dụng GPU và mạng thần kinh sâu) đã nghiền nát mọi đối thủ.

Bước ngoặt: Chứng minh rằng **Dữ liệu lớn (Big Data) + Chip đồ họa (GPU) + Deep Learning** là công thức chiến thắng.

AlphaGo và "Nước đi thứ 37" (2016)

Cờ vây phức tạp hơn cờ vua gấp tỷ tỷ lần, đòi hỏi "trực giác". Khi AlphaGo đánh bại Lee Sedol, nó không chỉ tính toán mà còn đưa ra những nước đi sáng tạo mà con người chưa từng nghĩ tới.

Ý nghĩa: AI bắt đầu thể hiện những khả năng giống như "sáng tạo" và "chiến lược" thay vì chỉ tính toán thuần túy.

Kiến trúc Transformer và ChatGPT (2017 – 2022)

Năm 2017, Google công bố bài báo "Attention is All You Need", giới thiệu kiến trúc **Transformer**.

Bước ngoặt: Transformer cho phép AI hiểu ngữ cảnh cực tốt và xử lý dữ liệu song song (nhanh hơn rất nhiều).

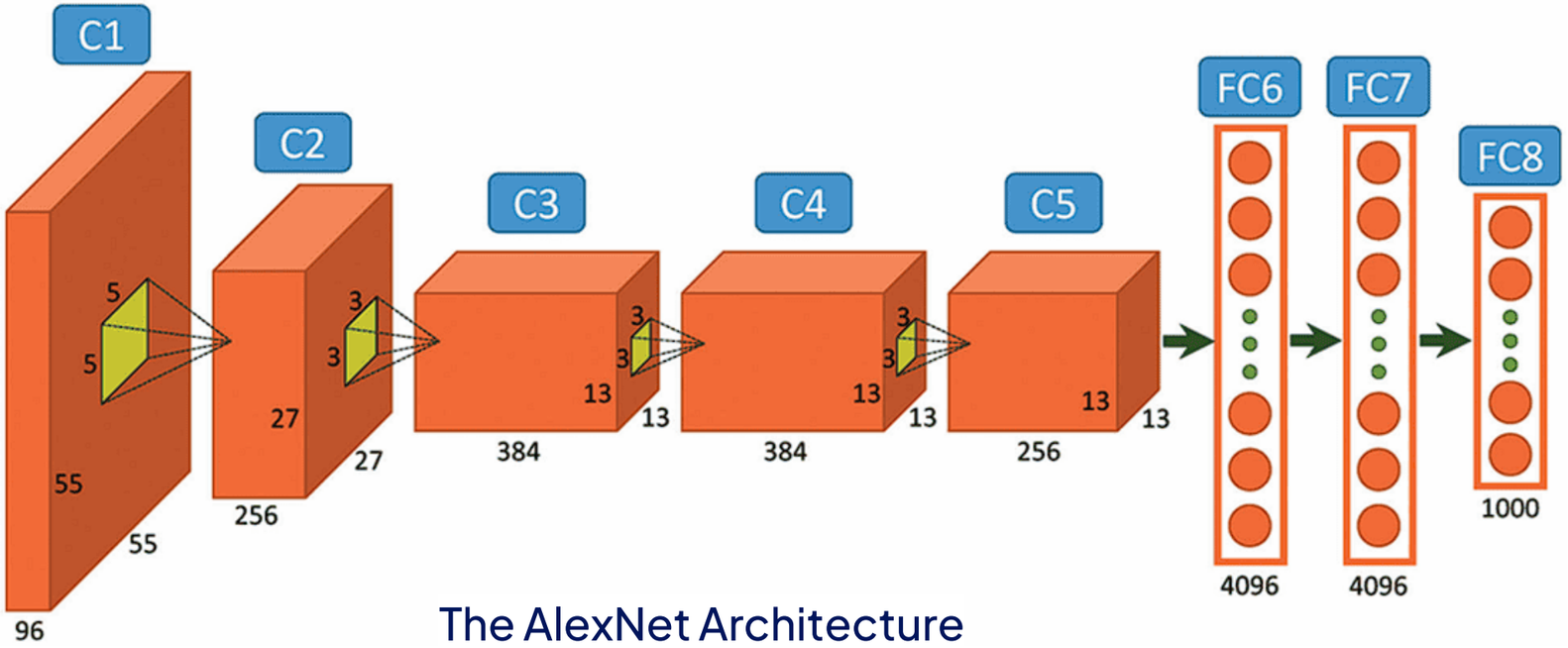
Hệ quả: Dẫn đến sự ra đời của GPT (OpenAI), Claude (Anthropic) và Gemini (Google). AI giờ đây có thể trò chuyện, lập trình và sáng tác như một cộng sự thực thụ.

3 AlexNet: Phát súng khởi đầu của Deep Learning

AlexNet: Phát súng khởi đầu của Deep Learning

Nếu bạn coi lịch sử AI là một cuộc đua, thì **AlexNet** chính là phát súng khởi đầu cho kỷ nguyên bùng nổ của Deep Learning (học sâu) như chúng ta thấy ngày nay.

Được giới thiệu vào năm 2012 bởi **Alex Krizhevsky, Ilya Sutskever** (người sau này đồng sáng lập OpenAI) và **Geoffrey Hinton**, AlexNet đã làm rung chuyển cộng đồng khoa học máy tính tại cuộc thi nhận diện hình ảnh ImageNet (ILSVRC 2012).



The AlexNet Architecture

1. Bối cảnh: “Cú sốc” tại ImageNet 2012

Trước năm 2012, các phương pháp nhận diện hình ảnh truyền thống dựa trên các thuật toán thủ công (hand-crafted features) rất phức tạp nhưng hiệu quả thấp.

15.3%

Tỷ lệ lỗi của AlexNet

26.2%

Tỷ lệ lỗi của các phương pháp khác

Ý nghĩa: Khoảng cách 10% này lớn đến mức nó chứng minh rằng **Mạng thần kinh nhân tạo (Neural Networks)** đã thực sự vượt trội.

2. Tại sao AlexNet lại đặc biệt?

AlexNet không chỉ là một mạng thần kinh "sâu" hơn bình thường, nó giới thiệu những kỹ thuật mà hiện nay vẫn là tiêu chuẩn vàng:

Sử dụng GPU (Đơn vị xử lý đồ họa)

AlexNet là một trong những mạng đầu tiên tận dụng sức mạnh tính toán của card đồ họa (NVIDIA GTX 580) để huấn luyện. Thay vì mất hàng tháng trên CPU, nó chỉ mất vài ngày.

Hàm kích hoạt ReLU (f(x)=max(0,x))

Thay vì dùng hàm Sigmoid hay Tanh truyền thống, ReLU giúp tốc độ hội tụ (học) nhanh hơn gấp 6 lần.

Kỹ thuật Dropout

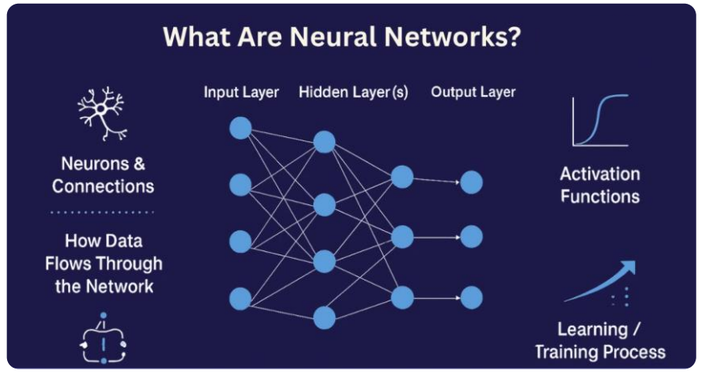
Để tránh việc máy "học vẹt" (overfitting), AlexNet ngẫu nhiên "tắt" một số nơ-ron trong quá trình huấn luyện, buộc mạng phải học những đặc trưng tổng quát hơn.

Kiến trúc 8 lớp

Gồm 5 lớp tích chập (**Convolutional Layers**) và 3 lớp kết nối đầy đủ **(Fully Connected Layers)**.

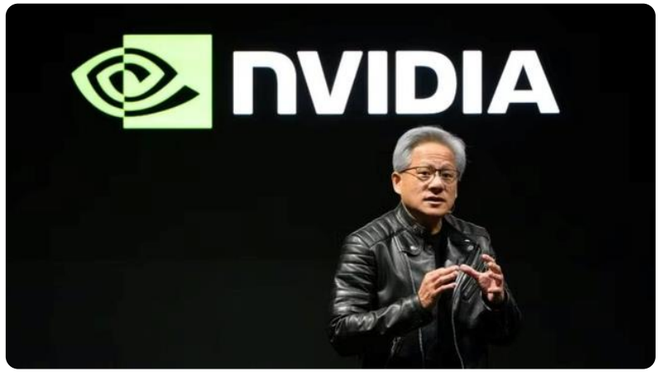
3. Tác động lâu dài

AlexNet giống như một “chìa khóa” mở ra cánh cửa cho tất cả những gì chúng ta có bây giờ:



Hồi sinh Neural Networks

Sau nhiều thập kỷ bị ghẻ lạnh (Mùa đông AI), các nhà khoa học đổ xô quay lại nghiên cứu mạng thần kinh.



Sự trỗi dậy của NVIDIA

Cơn sốt AI bắt đầu khiến nhu cầu về GPU tăng vọt, biến NVIDIA từ một công ty làm card màn hình chơi game thành “gã khổng lồ” AI.

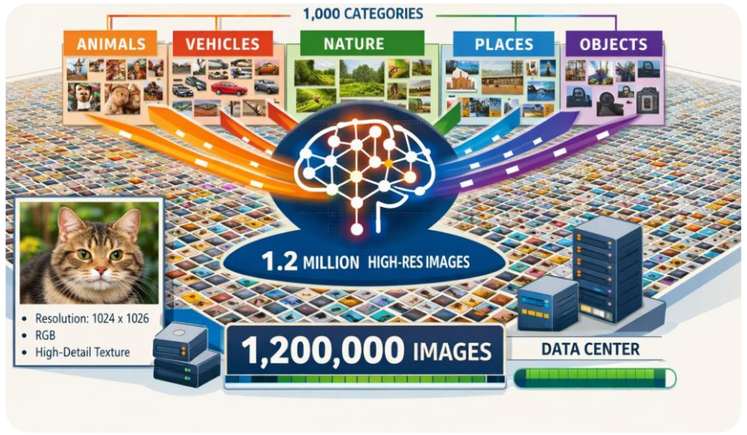
Tiền thân của “mọi thứ”

Từ FaceID trên iPhone, xe tự lái Tesla đến khả năng nhận diện hình ảnh của Gemini hay ChatGPT đều có “DNA” từ những nguyên lý mà AlexNet đã khẳng định.

Tóm tắt thông số

Tác giả:

Ilya Sutskever,
Geoffrey Hinton,
Alex Krizhevsky,



Dữ liệu huấn luyện:

1.2 triệu hình ảnh
độ phân giải cao
thuộc 1,000 lớp khác nhau

Số lượng tham số:

Khoảng 60 triệu
tham số

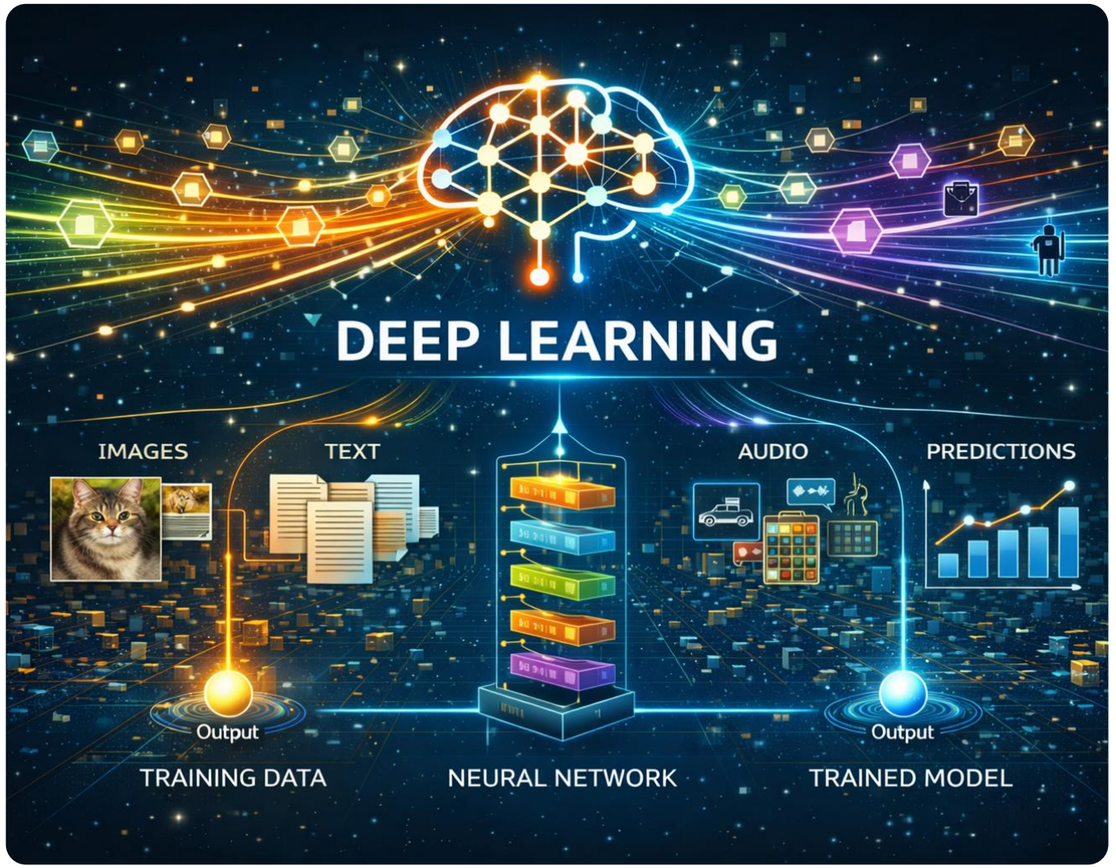
4

VGG và ResNet: Cải tiến sau sự ra đời của AlexNet

VGG và ResNet: Cải tiến sau sự ra đời của AlexNet

Sau thành công rực rỡ của AlexNet vào năm 2012, cộng đồng AI nhận ra một chân lý: **“Càng sâu càng tốt” (Deeper is better)**. Tuy nhiên, việc làm cho mạng thần kinh sâu hơn không hề đơn giản vì các rào cản vật lý và toán học.

VGG và **ResNet** chính là hai bước nhảy vọt kế tiếp để giải quyết bài toán “độ sâu” này.



Bảng so sánh tóm tắt

Đặc điểm	AlexNet (2012)	VGG16 (2014)	ResNet152 (2015)
Số lớp	**8**	**16**	**152**
Kích thước bộ lọc	Đa dạng (11 x 11, 5 x 5,3 x 3)	Đồng nhất (3 x 3)	Đồng nhất (3 x 3) + Skip Connection
Triết lý chính	Sử dụng GPU & ReLU	Tăng độ sâu bằng filter nhỏ	Sử dụng "đường tắt" để học sâu cực đại
Tỷ lệ lỗi (Top 5)	**~15.3%**	**~7.3%**	**~3.57%** (Vượt ngưỡng của con người)

Tóm lại

- **AlexNet** chứng minh Deep Learning có thể hoạt động tốt
- **VGG** chứng minh cấu trúc mạng đồng nhất và sâu sẽ hiệu quả hơn
- **ResNet** chứng minh chúng ta có thể xây dựng những mạng cực sâu mà không làm hỏng dữ liệu, mở ra kỷ nguyên cho các AI nhận diện khuôn mặt và xe tự lái hiện nay

1. VGG (2014): Sự đồng nhất và Sức mạnh của sự đơn giản

Được phát triển bởi nhóm Visual Geometry Group (Đại học Oxford), VGG (thường là VGG16 hoặc VGG19) đã thay đổi cách thiết kế cấu trúc mạng.

Cải tiến so với AlexNet

Thay vì dùng các bộ lọc (filter) kích thước lớn và lộn xộn (như 11×11 hay 5×5 ở AlexNet), VGG chỉ sử dụng các bộ lọc **siêu nhỏ 3×3** xếp chồng lên nhau.

Tại sao lại hiệu quả?

Việc chồng 3 lớp 3×3 có vùng nhìn tương đương với 1 lớp 7×7 nhưng lại có ít tham số hơn và có nhiều hàm kích hoạt phi tuyến tính hơn, giúp mạng học được các đặc trưng phức tạp hơn.

Độ sâu

Tăng từ 8 lớp (AlexNet) lên **16 – 19 lớp**.

Hạn chế

VGG rất "nặng". Do có nhiều lớp kết nối đầy đủ (Fully Connected layers) ở cuối, nó có tới hơn 138 triệu tham số, khiến việc tính toán rất tốn bộ nhớ.

2. ResNet (2015): Kẻ phá vỡ giới hạn “Độ sâu”

Khi các nhà khoa học cố gắng làm mạng sâu hơn VGG (ví dụ 30, 50 lớp), họ gặp phải hiện tượng **Tiêu biến đạo hàm (Vanishing Gradient)**: Càng đi sâu, thông tin để học càng bị “loãng” dần và biến mất, khiến mạng không thể hội tụ.

Microsoft Research đã giải quyết điều này bằng **ResNet (Residual Network)** với khái niệm **Kết nối tắt (Skip Connections / Residual Learning)**.

Cải tiến đột phá

Thay vì bắt mỗi lớp phải học toàn bộ thông tin mới, ResNet cho phép thông tin "nhảy cóc" qua một vài lớp. Nếu một lớp không học được gì hữu ích, nó chỉ cần giữ nguyên thông tin từ lớp trước đó

Công thức

Thay vì học ánh xạ H(x), mạng sẽ học phần dư $F(x) = H(x) - x$

Độ sâu không tưởng

ResNet đã phá vỡ mọi kỷ lục với **152 lớp** (gấp gần 10 lần VGG) nhưng lại có ít tham số hơn VGG vì không dùng nhiều lớp Fully Connected cồng kềnh

5

Transformer: Động cơ phản lực của AI

Transformer:
Động cơ phản lực của AI

Nếu AlexNet là "phát súng khởi đầu" thì **Transformer** chính là "động cơ phản lực" đưa AI từ việc nhận diện hình ảnh đơn giản đến khả năng tư duy ngôn ngữ như con người (ChatGPT, Claude, Gemini).

Được Google giới thiệu vào năm 2017 qua bài báo nổi tiếng "**Attention is All You Need**", Transformer đã thay thế hoàn toàn các kiến trúc cũ (như RNN hay LSTM) để trở thành tiêu chuẩn vàng cho mọi mô hình AI hiện đại.

Các tác giả của bài báo
"**Attention is All You Need**"

Ashish Vaswani
Co-founder & CEO
at Essential AI

Noam Shazeer
Technical Lead, Gemini
at Google

Niki Parmar
Member of Technical Staff
at Anthropic

Illia Polosukhin
Co-founder at NEAR Protocol
CEO at NEAR Foundation

Aidan N. Gomez
Co-founder & CEO
at Cohere

Jakob Uszkoreit
Co-founder & CEO
at Inceptive

Łukasz Kaiser
Member of Technical Staff
at OpenAI

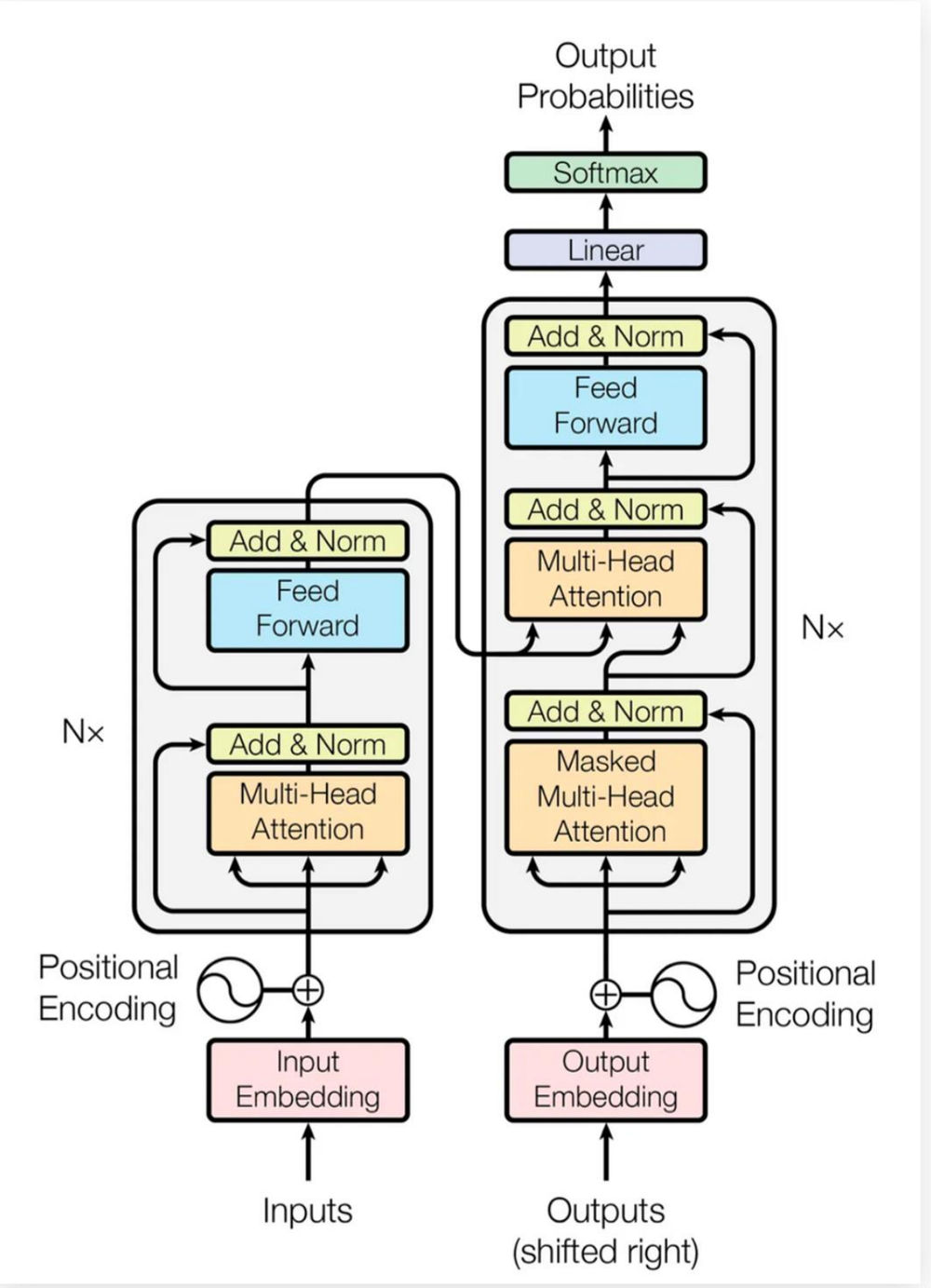
Llion Jones
Co-founder & CTO
at Sakana AI

Transformer:
Động cơ phản lực của AI

Attention Is All You Need

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Query (Q), **Key (K)**, and **Value (V)** are the core components of the attention mechanism in Transformer models, **simulating a search process to determine how much focus to place on other words**. Q is the current query, K is the label for matching, and V holds the content.



1. Tại sao các kiến trúc cũ thất bại?

Trước Transformer, AI xử lý ngôn ngữ theo kiểu tuần tự (giống như cách bạn đọc từng từ một từ trái sang phải).

Vấn đề

Với những câu dài, AI sẽ "quên" những từ ở đầu câu khi đọc đến cuối câu (Lỗi mất trí nhớ ngắn hạn).

Tốc độ

Vì phải xử lý tuần tự từng từ, việc huấn luyện AI diễn ra rất chậm và không tận dụng được sức mạnh của GPU.

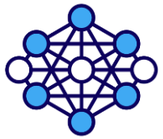
2. Bí mật của Transformer: Cơ chế Self-Attention

Transformer không đọc tuần tự. Nó "nhìn" toàn bộ câu cùng một lúc. Cơ chế Self-Attention cho phép mỗi từ trong câu "chú ý" đến tất cả các từ khác để hiểu ngữ cảnh.

Ví dụ: Xét câu: "**Con báo** đang đuổi theo con mồi vì **nó** đang đói."
Khi xử lý từ "nó", cơ chế Attention sẽ giúp AI hiểu rằng "nó" đang kết nối mạnh nhất với từ "con báo" chứ không phải "con mồi". Điều này giúp AI hiểu được ý nghĩa sâu xa và các mối quan hệ logic trong văn bản.

3. Cấu trúc của Transformer

Kiến trúc này gồm hai phần chính (nhưng tùy dòng máy mà có thể lược bỏ):



Encoder (Bộ mã hóa)

Đọc và hiểu văn bản đầu vào, chuyển nó thành các vector số học chứa đựng ý nghĩa (Context).

Decoder (Bộ giải mã)

Dựa trên ý nghĩa đó để dự đoán và tạo ra văn bản đầu ra (từng từ một).

4. Tại sao Transformer thay đổi thế giới?

Sự ra đời của Transformer tạo ra 3 bước ngoặt lớn:

Xử lý song song (Parallelization)

Vì không phải đợi xử lý từ trước xong mới đến từ sau, Transformer có thể được huấn luyện trên hàng nghìn GPU cùng lúc. Đây là lý do chúng ta có thể dạy AI trên toàn bộ dữ liệu của Internet.

Hiểu ngữ cảnh cực dài

Nó có thể ghi nhớ mối liên hệ giữa các đoạn văn cách xa nhau hàng nghìn từ, giúp viết luận văn hay lập trình mã nguồn dài mà không bị "râu ông nọ cắm cằm bà kia".

Đa nhiệm (Multimodal)

Kiến trúc này linh hoạt đến mức không chỉ dùng cho văn bản mà còn áp dụng cho hình ảnh (**Vision Transformer – ViT**), âm thanh và cả video.

“Cây phả hệ” của các mô hình dựa trên Transformer

Dòng mô hình	Đại diện	Thế mạnh
Chỉ Encoder	BERT (Google)	Hiểu văn bản, phân loại cảm xúc, tìm kiếm
Chỉ Decoder	GPT (OpenAI)	Sáng tạo nội dung, viết lách, trò chuyện (Generative)
Encoder – Decoder	T5, BART	Dịch thuật, tóm tắt văn bản

Một cách ví von:

Nếu các AI “cũ” giống như một người đọc sách bằng cách dùng ngón tay chỉ từng chữ, thì Transformer giống như một người có khả năng nhìn một lúc cả trang sách và nhanh chóng hiểu các ý tứ liên quan đến nhau.

6 LLM – Mô hình Ngôn ngữ lớn

LLM – Mô hình Ngôn ngữ lớn

LLM (Large Language Model – Mô hình Ngôn ngữ lớn) chính là "đỉnh cao" hiện tại của dòng chảy lịch sử AI mà chúng ta vừa điểm qua. Nếu Transformer là động cơ, thì LLM chính là chiếc siêu xe được lắp động cơ đó và nạp đầy "nhiên liệu" là toàn bộ tri thức của nhân loại trên Internet.

Để được gọi là một LLM, mô hình đó phải hội tụ đủ 3 yếu tố: **Large (Lớn)** về số lượng tham số, **Large** về dữ liệu huấn luyện, và khả năng **Ngôn ngữ** vượt trội.

Ba cột trụ tạo nên một LLM

A. Quy mô tham số (Parameters)

Tham số giống như các "nút thắt" thần kinh trong não bộ AI.

- Các mô hình cũ có vài triệu tham số
- LLM hiện đại (như GPT-4 hay Gemini) có từ hàng trăm tỷ đến hàng nghìn tỷ tham số

Hiệu ứng mới nổi (Emergent Abilities): Khi đạt đến một ngưỡng quy mô nhất định, LLM đột ngột xuất hiện những khả năng mà người tạo ra nó cũng không ngờ tới, như giải toán đố, hiểu chuyện cười hay lập trình.

B. Dữ liệu khổng lồ (Big Data)

LLM được huấn luyện trên hàng petabyte dữ liệu văn bản bao gồm:

- Sách, bài báo khoa học
- Mã nguồn (Github)
- Wikipedia và các diễn đàn (Reddit)

Nó không "học thuộc lòng" mà học **quy luật xác suất** của ngôn ngữ.

C. Kiến trúc Transformer

Như đã nói ở phần trước, nhờ cơ chế **Attention**, LLM có thể hiểu được mối quan hệ giữa các từ ngữ trong một văn cảnh cực dài.

Quy trình “luyện” một LLM

Để một mô hình ngôn ngữ lớn từ chỗ chỉ là các con số vô hồn trở thành một trợ lý thông minh như ChatGPT, LLM phải trải qua 2 giai đoạn chính:

Pre-training (Tiền huấn luyện)

AI đọc hàng tỷ câu văn và tự chơi trò “điền vào chỗ trống”. **Ví dụ**: “Thủ đô của Việt Nam là [?]?”. Giai đoạn này giúp AI có kiến thức cực rộng nhưng chưa biết cách trò chuyện.

Fine-tuning & RLHF (Tinh chỉnh)

Con người sẽ chấm điểm các câu trả lời của AI. Nếu AI trả lời sai hoặc thô lỗ, nó bị điểm thấp; nếu trả lời hay và hữu ích, nó được điểm cao.

RLHF (Reinforcement Learning from Human Feedback):
Giúp AI trở nên an toàn, lịch sự và biết tuân thủ mệnh lệnh của người dùng.

Khả năng đặc biệt của LLM: “In-context Learning”

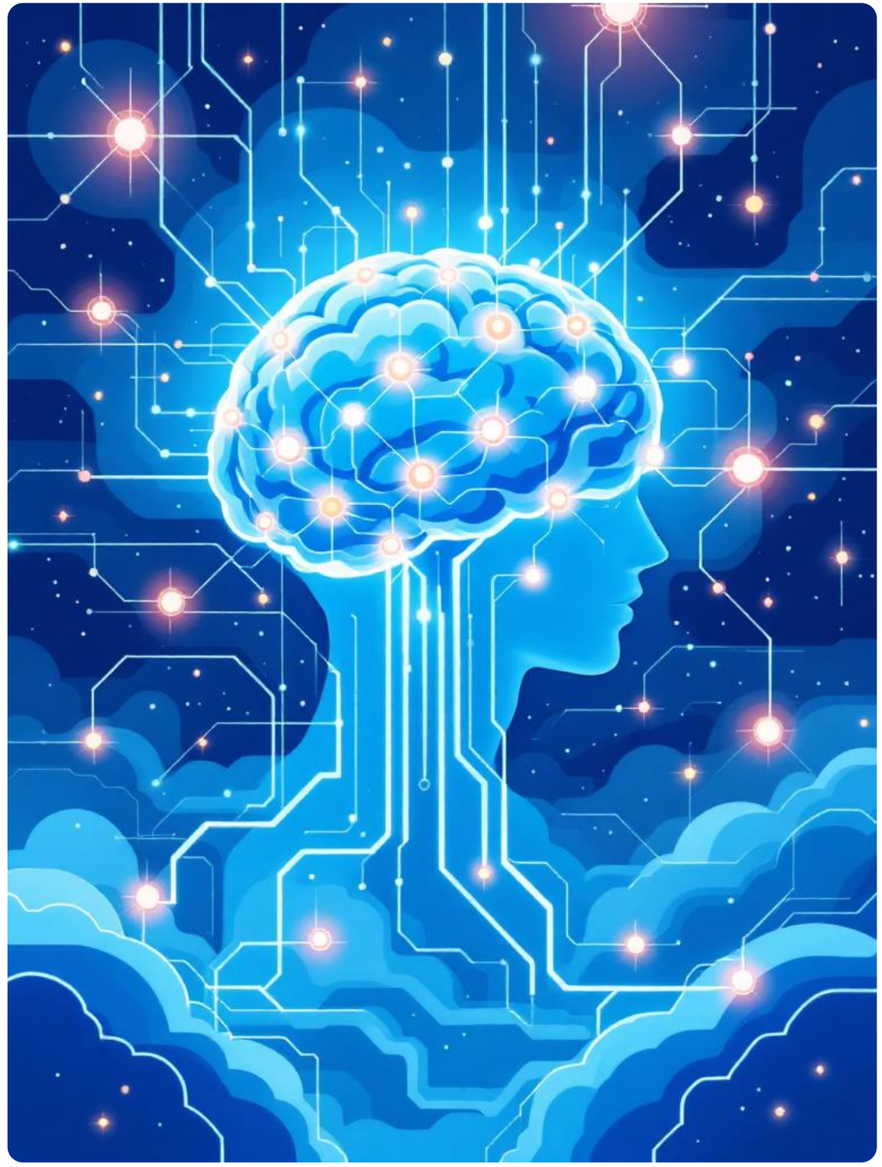
Đây là điểm khiến LLM khác biệt hoàn toàn với các phần mềm truyền thống:

Zero-shot

Bạn yêu cầu AI làm một việc nó chưa từng được dạy trực tiếp (ví dụ: viết bài thơ về một loại linh kiện máy tính mới), nó vẫn làm được dựa trên kiến thức tổng quát.

Few-shot (Prompt Engineering)

Bạn đưa cho AI một vài ví dụ mẫu, nó sẽ bắt chước phong cách hoặc logic đó ngay lập tức mà không cần phải lập trình lại mã nguồn.



Các dòng LLM phổ biến hiện nay

Nhóm	Đại diện tiêu biểu	Đặc điểm
Đóng (Proprietary)	GPT-4 (OpenAI), Gemini (Google), Claude 3 (Anthropic)	Hiệu suất cao nhất, được tối ưu hóa cực tốt nhưng không công bố mã nguồn.
Mở (Open Weights)	Llama 3 (Meta), Mistral, Qwen (Alibaba), DeepSeek	Cộng đồng có thể tải về, tự cài đặt trên máy chủ riêng và tùy chỉnh.

Thách thức lớn nhất: "Ảo giác" (Hallucination)

Vì LLM hoạt động dựa trên xác suất (từ nào đi sau từ nào là hợp lý nhất), đôi khi nó viết những câu nghe rất thuyết phục nhưng **hoàn toàn sai sự thật**.

Ví dụ: Nó có thể tự tin khẳng định "Năm 1995, Việt Nam đã phóng tàu vũ trụ lên Sao Hỏa" chỉ vì cấu trúc câu đó nghe rất "thuận tai".

Bạn có biết? Hiện nay xu hướng đang chuyển dịch sang **Multimodal LLM (LMM)** – nơi AI không chỉ hiểu văn bản mà còn "nhìn" được ảnh và "nghe" được âm thanh trong cùng một bộ não.

Top 3 “Siêu mẫu” toàn năng (Proprietary)

Đây là những mô hình trả phí, có hiệu suất cao nhất và được tối ưu hóa tốt nhất cho người dùng phổ thông qua web / mobile app.

GPT-5.2 (OpenAI)

Vẫn giữ ngôi vương về khả năng đa nhiệm và độ phổ biến. Nó được đánh giá cao nhất ở khả năng “Agentic” (tự động thực hiện các tác vụ phức tạp thay vì chỉ trả lời câu hỏi) và sự ổn định trong công việc chuyên môn.

Gemini 3 Pro (Google)

Thế mạnh tuyệt đối nằm ở cửa sổ ngữ cảnh cực lớn và khả năng tích hợp sâu vào hệ sinh thái Google (Docs, Gmail, Drive). Đây là lựa chọn số 1 nếu bạn cần phân tích hàng chục file PDF hoặc video dài cùng lúc.

Claude 4.5 Sonnet (Anthropic)

Được cộng đồng lập trình viên và giới trí thức yêu thích nhất nhờ phong cách trả lời “giống người”, ít giáo điều và khả năng lập luận (reasoning) cực kỳ sắc bén, ít khi bị ảo giác hơn các đối thủ.

Các “Chuyên gia” theo lĩnh vực

Nếu bạn có một nhu cầu cụ thể, hãy chọn những cái tên sau:

Lĩnh vực	Mô hình tốt nhất	Lý do
Lập trình (Coding)	**Claude 4.5 Sonnet & GPT-5.2**	Khả năng viết mã sạch, giải thích logic và gỡ lỗi (debug) vượt trội
Suy luận chuyên sâu	**OpenAI o3 & DeepSeek R1**	Sử dụng cơ chế “Chain-of-Thought” (suy nghĩ từng bước) để giải các bài toán logic, toán học hóc búa
Xử lý dữ liệu dài	**Llama 4 Scout & Gemini 2.5/3**	Hỗ trợ ngữ cảnh từ 1 triệu đến 10 triệu token (tương đương hàng nghìn trang sách)
Giá rẻ / Tốc độ cao	**Gemini 3 Flash**	Cực nhanh, phản hồi gần như tức thì, phù hợp cho các chatbot CSKH

Top mô hình Nguồn mở (Open-Weights)

Năm 2026 chứng kiến sự trỗi dậy mạnh mẽ của các mô hình mà bạn có thể tự tải về và chạy trên máy chủ riêng để đảm bảo bảo mật:

Llama 4 (Meta)

Tiêu chuẩn vàng của thế giới nguồn mở, mạnh tương đương với các mô hình đóng của năm ngoái.

DeepSeek V3.2 (China)

Gây sốt toàn cầu vì hiệu suất cực cao trong khi chi phí vận hành cực thấp, đặc biệt mạnh về toán và code.

Qwen 3 (Alibaba)

Mô hình đa ngôn ngữ tốt nhất, đặc biệt hiệu quả nếu bạn làm việc nhiều với tiếng Việt hoặc các ngôn ngữ châu Á.

Nên chọn cái nào?

Dùng hàng ngày, làm văn phòng

ChatGPT (GPT-5.2)

Nghiên cứu tài liệu đồ sộ, dùng hệ Google

Gemini 3

Lập trình viên, viết lách sáng tạo

Claude 4.5

Doanh nghiệp cần bảo mật dữ liệu

Llama 4 hoặc DeepSeek

Cách LLM tư duy

Thực tế, việc dùng từ “tư duy” đối với LLM (Large Language Models) là một cách nói nhân hóa. Về mặt kỹ thuật, LLM không “suy nghĩ” như con người (dựa trên trải nghiệm, ý thức và cảm xúc). Thay vào đó, nó **“tư duy bằng xác suất”** thông qua một quá trình cực kỳ phức tạp.

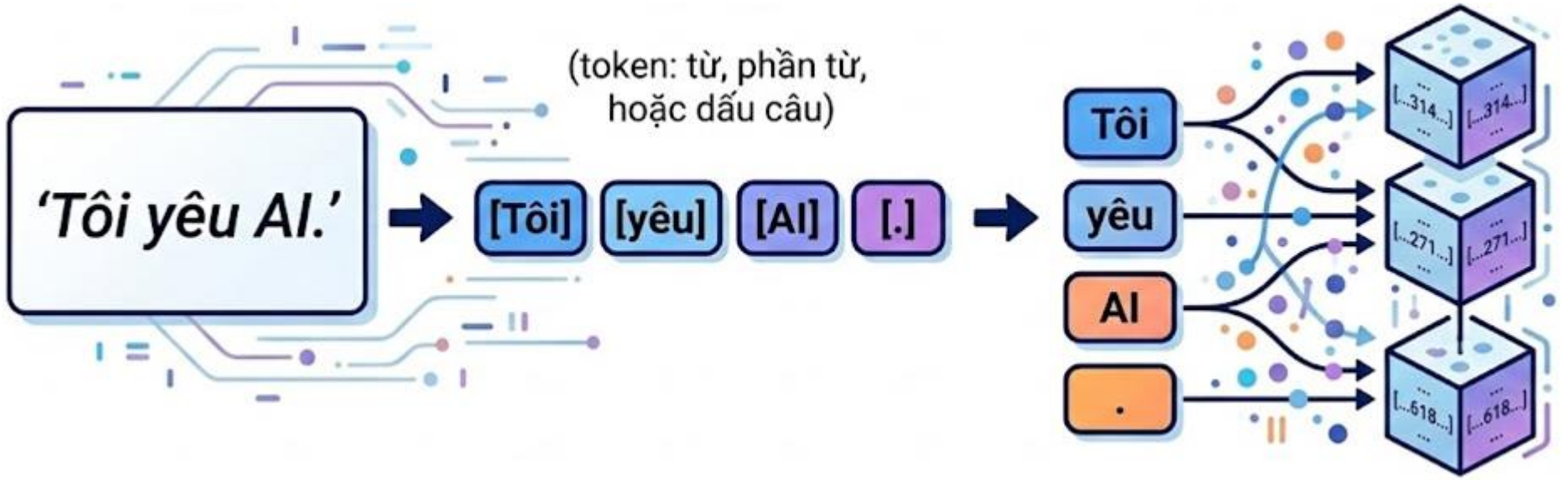
Dưới đây là cách mà một bộ não AI vận hành để đưa ra câu trả lời cho bạn.

1. Biến từ ngữ thành con số (Tokenization & Embedding)
2. “Nhìn” và hiểu ngữ cảnh (Self-Attention)
3. Dự đoán từ tiếp theo (Next Token Prediction)
4. Các mức độ “tư duy” mới (Reasoning)

1. Biến từ ngữ thành con số (Tokenization & Embedding)

AI không đọc chữ cái. Bước đầu tiên, nó chia văn bản của bạn thành các mảnh nhỏ gọi là Tokens (có thể là một từ, một phần của từ hoặc dấu câu).

Sau đó, mỗi token được chuyển thành một chuỗi số (Vector) trong không gian hàng nghìn chiều.



Nguyên lý: Các từ có ý nghĩa gần nhau (như "vua" và "hoàng hậu") sẽ nằm gần nhau trong không gian này.

2. “Nhìn” và hiểu ngữ cảnh (Self-Attention)

Đây là phần quan trọng nhất trong “tư duy” của AI. Với cơ chế **Self-Attention**, AI không nhìn từ một cách độc lập. Nó xem xét mối quan hệ giữa từ đó với **tất cả các từ còn lại** trong câu.

Ví dụ 1

“Pha nước cam thì nhớ cho thêm ít **đường** cho bớt **chua** nhé”

→ Đường ăn (sugar)

Ví dụ 2

“Giờ cao điểm đi **đường** này hay bị **kẹt xe** lắm”

→ Đường giao thông (road)

Nhờ các từ xung quanh (“chua” hay “kẹt xe”),
AI sẽ gán trọng số khác nhau cho từ “đường”
để hiểu đó là đường ăn (sugar) hay là đường giao thông (road).

3. Dự đoán từ tiếp theo (Next Token Prediction)

Về bản chất, LLM là một cái máy dự đoán cực kỳ thông minh. Khi bạn đặt câu hỏi, nó không tra cứu một thư mục có sẵn. Nó tự hỏi: **"Dựa trên toàn bộ những gì tôi đã học, từ nào có khả năng xuất hiện tiếp theo cao nhất sau chuỗi ký tự này?"**

Nó thực hiện việc này lặp đi lặp lại:
Dự đoán từ thứ 1 → lấy từ thứ 1 làm đầu vào để dự đoán từ thứ 2 → cứ thế cho đến khi hoàn thành câu.

4. Các mức độ "tư duy" mới (Reasoning)

Trong các dòng mô hình mới nhất (như **OpenAI o1** hay **DeepSeek R1**), AI đã có một bước tiến mới gọi là **Chain-of-Thought (Chuỗi tư duy)**:

Tự đối thoại nội bộ

Trước khi trả lời, AI được dạy để tạo ra một "bản nháp" trong đầu. Nó tự đặt câu hỏi, tự kiểm tra lỗi logic và điều chỉnh hướng đi.

Phân rã vấn đề

Thay vì đưa ra đáp án ngay lập tức (dễ sai), nó chia bài toán khó thành các bước nhỏ 1→2→3.

So sánh tư duy Người vs. LLM

Đặc điểm	Tư duy Con người	"Tư duy" LLM
Cơ chế	Xung thần kinh, sinh hóa, cảm xúc	Phép nhân ma trận và xác suất thống kê
Dữ liệu	Trải nghiệm thực tế qua 5 giác quan	Văn bản và hình ảnh trên Internet
Sự hiểu biết	Hiểu bản chất sự vật (Semantic)	Hiểu mối liên hệ giữa các ký hiệu (Syntactic)
Sáng tạo	Đột phá từ cái chưa có	Kết hợp các mẫu (patterns) đã có theo cách mới

Tại sao đôi khi nó trông rất “thông minh”?

Đó là nhờ quy mô. Với hàng nghìn tỷ tham số, AI có khả năng nhận diện các quy luật logic cực kỳ tinh vi. Khi bạn thấy nó giải toán hay lập luận, thực chất nó đang áp dụng các “mẫu logic” mà nó đã thấy hàng triệu lần trong dữ liệu huấn luyện.

Một cách ví von: Tư duy của LLM giống như một người cực kỳ uyên bác, đã đọc hết mọi cuốn sách trên đời nhưng lại bị nhốt trong một căn phòng tối. Người đó không biết thế giới bên ngoài trông ra sao, nhưng vì đã đọc quá nhiều nên có thể dự đoán cực chính xác câu tiếp theo trong một cuộc trò chuyện là gì.

Tại sao LLM trở nên "thông minh" đột biến?

Sự "thông minh" đột biến của LLM không phải là một sự tình cờ, mà là kết quả của việc đạt đến một ngưỡng tới hạn về quy mô, được các nhà khoa học gọi là **"Sự trỗi dậy của các khả năng"** (Emergent Abilities).

Dưới đây là 4 lý do chính giải thích cho sự bùng nổ trí tuệ này.

1. Định luật quy mô (Scaling Laws)
2. Sự chuyển dịch từ "Học vẹt" sang "Học quy luật"
3. Kỹ thuật RLHF: "Cú hích" từ con người
4. In-context learning

1. Định luật quy mô (Scaling Laws)

Khi "Lượng" đổi dẫn đến "Chất" đổi

Đây là yếu tố quan trọng nhất. Các nghiên cứu từ OpenAI và Google chỉ ra rằng khi bạn tăng đồng thời ba yếu tố: **Số lượng tham số** (não bộ lớn hơn), **Dữ liệu huấn luyện** (đọc nhiều hơn) và **Sức mạnh tính toán** (học nhanh hơn), mô hình sẽ đột ngột xuất hiện những khả năng mà nó không được dạy trực tiếp.

Ví dụ: Một mô hình nhỏ có thể chỉ biết nối từ. Nhưng khi tăng quy mô lên 100 lần, nó đột nhiên biết giải toán, hiểu sự mỉa mai, hoặc thậm chí là lập trình, dù mục tiêu ban đầu chỉ là "dự đoán từ tiếp theo".

2. Sự chuyển dịch từ "Học vẹt" sang "Học quy luật"

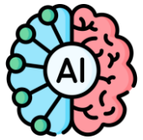
Khi dữ liệu huấn luyện bao phủ gần như toàn bộ tri thức nhân loại (Internet, sách, mã nguồn), AI không còn đủ bộ nhớ để "học thuộc lòng" từng câu văn. Thay vào đó, nó buộc phải tìm cách **nén thông tin**.

- Để nén thông tin hiệu quả nhất, nó phải tự tìm ra các quy luật logic, ngữ pháp và mối quan hệ nhân quả ẩn sau ngôn ngữ
- Chính việc hiểu được các **quy luật** này giúp nó có khả năng xử lý những câu hỏi mới mà nó chưa từng thấy trong quá trình học

3. Kỹ thuật RLHF: “Cú hích” từ con người

Trước khi có ChatGPT, các mô hình ngôn ngữ lớn đã rất giỏi nhưng thường trả lời lộn xộn hoặc vô nghĩa. Bước ngoặt thực sự đến từ **RLHF (Reinforcement Learning from Human Feedback – Học tăng cường từ phản hồi của con người)**.

Các chuyên gia con người sẽ chấm điểm các câu trả lời của AI, dạy nó biết thế nào là một câu trả lời “có ích”, “logic” và “an toàn”.



Quá trình này giúp “mài giũa” đống kiến thức khổng lồ nhưng hỗn loạn của AI thành một trí tuệ có định hướng và biết tuân thủ mệnh lệnh.

4. Khả năng “Trong văn cảnh” (In-context Learning)

Kiến trúc Transformer cho phép AI duy trì một sự tập trung (Attention) cực lớn vào toàn bộ đoạn hội thoại. Điều này tạo ra cảm giác AI rất thông minh vì nó có thể:

- Nhớ những gì bạn nói ở trên
- Học theo ví dụ bạn vừa đưa ra (Few-shot prompting)
- Suy luận dựa trên những thông tin mới mà nó chưa từng được học trong quá trình huấn luyện

Tóm lại

Sự thông minh đột biến này giống như quá trình một đứa trẻ học nói. Ban đầu nó chỉ lặp lại các âm thanh (mô hình nhỏ), nhưng đến một ngày, khi vốn từ và trải nghiệm đủ lớn, các kết nối thần kinh tự động khớp lại với nhau, giúp đứa trẻ bắt đầu hiểu logic và biết tư duy độc lập.

Hiện tượng này khiến ngay cả các nhà khoa học tạo ra chúng cũng cảm thấy bất ngờ vì họ không trực tiếp lập trình ra những khả năng đó.

Hạn Chế Của LLM

Dù LLM có vẻ rất “thông minh”, chúng vẫn chỉ là các mô hình toán học dựa trên xác suất, không phải một trí tuệ sinh học hoàn chỉnh. Vì vậy, chúng tồn tại những điểm yếu cố hữu mà người dùng cần hiểu rõ để tránh bị phụ thuộc quá mức.

1. Ảo giác (Hallucination)
2. Thiếu hiểu biết thực sự về thế giới (Lack of Grounding)
3. Cửa sổ ngữ cảnh hạn chế (Context Window)
4. Định kiến và độc hại (Bias & Toxicity)
5. Khả năng suy luận logic và toán học vẫn còn “mong manh”
6. Chi phí năng lượng và môi trường
7. Quyền riêng tư và Bản quyền

1. Ảo giác (Hallucination)

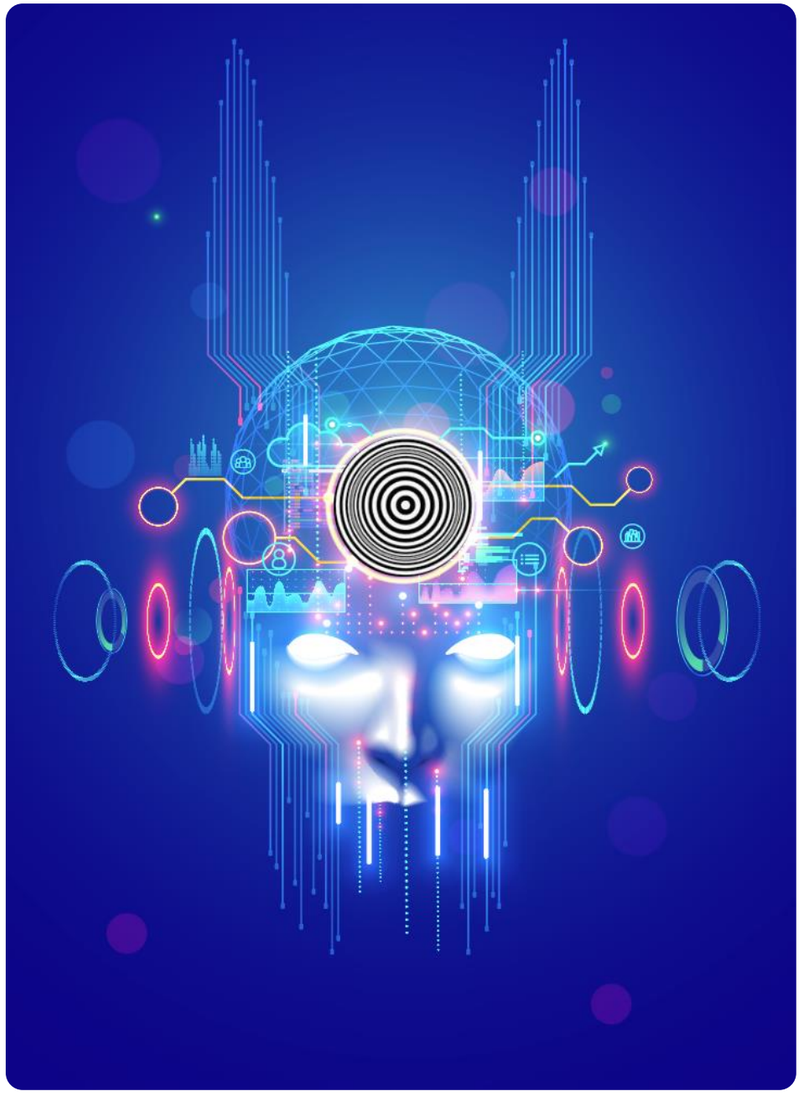
Đây là điểm yếu “chí mạng” và khó khắc phục nhất. AI có thể đưa ra những thông tin sai sự thật nhưng trình bày một cách cực kỳ tự tin và logic.

Nguyên nhân

Vì AI chỉ dự đoán từ tiếp theo dựa trên xác suất, nó sẽ chọn từ “nghe có vẻ hợp lý” thay vì từ “đúng sự thật”.

Hệ quả

Nó có thể bịa ra một sự kiện lịch sử, một điều luật không tồn tại hoặc trích dẫn một nguồn tài liệu giả.



2. Thiếu hiểu biết thực sự về thế giới (Lack of Grounding)

LLM không có trải nghiệm thực tế. Nó hiểu từ “quả táo” qua hàng triệu dòng mô tả, nhưng nó không biết vị ngọt, mùi thơm hay cảm giác cầm quả táo trên tay là gì.

AI hoạt động dựa trên **ký hiệu (syntax)** chứ không phải **ý nghĩa thực thể (semantics)**.

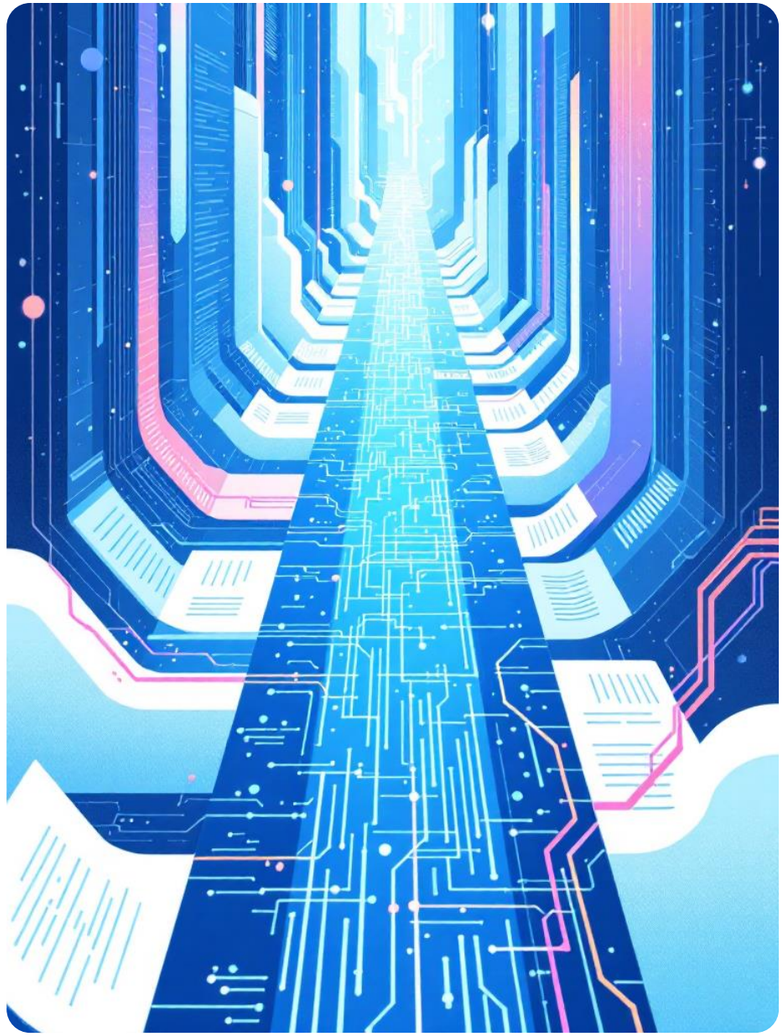
Nếu bạn đưa ra một tình huống đòi hỏi logic vật lý cơ bản mà chưa từng được viết trong sách vở, AI rất dễ đưa ra câu trả lời ngớ ngẩn.

3. Cửa sổ ngữ cảnh hạn chế (Context Window)

Mặc dù các mô hình như Gemini đã mở rộng cửa sổ ngữ cảnh lên hàng triệu từ, nhưng AI vẫn có xu hướng “quên” hoặc “loãng” thông tin ở giữa các tài liệu quá dài.

Hiện tượng “Lost in the Middle”

AI thường nhớ tốt phần đầu và phần cuối của một yêu cầu dài, nhưng lại bỏ sót những chi tiết quan trọng nằm ở giữa.



4. Định kiến và độc hại (Bias & Toxicity)

AI được huấn luyện trên dữ liệu từ Internet – nơi chứa đầy những định kiến về giới tính, sắc tộc, tôn giáo và văn hóa.

Mặc dù các nhà phát triển đã dùng kỹ thuật RLHF để lọc bỏ, nhưng các định kiến ngầm vẫn có thể xuất hiện trong cách AI ưu tiên đưa ra câu trả lời, dẫn đến những kết quả không công bằng hoặc thiếu khách quan.

5. Khả năng suy luận logic và toán học vẫn còn “mong manh”

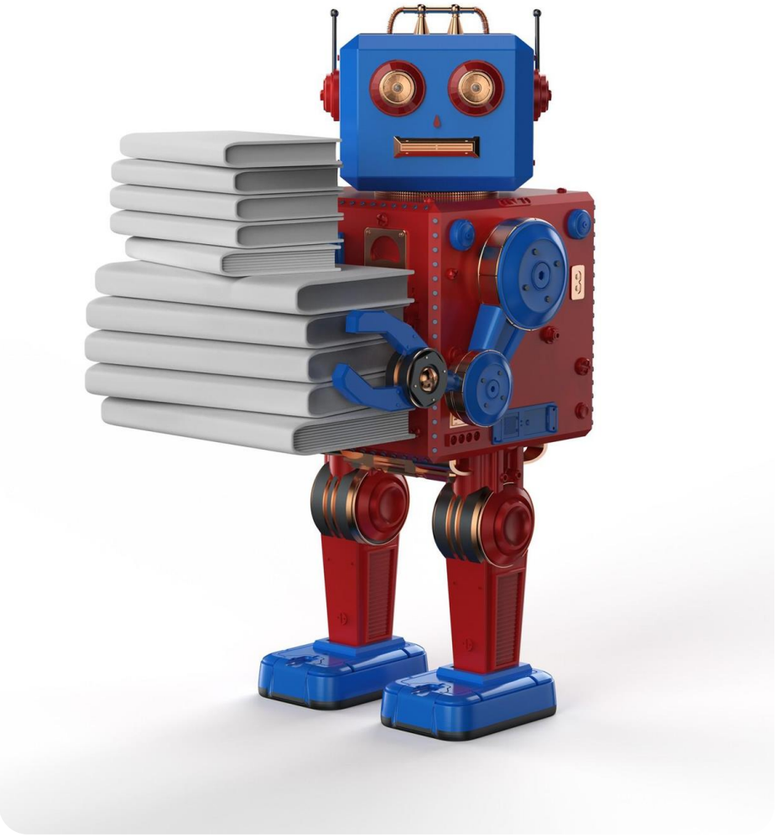
Dù đã cải tiến với các mô hình như **OpenAI o1**,
nhưng đa số LLM thông thường vẫn gặp khó khăn với:

Các bài toán đa bước

Chỉ cần sai một bước nhỏ ở giữa, toàn bộ kết quả sau đó sẽ đổ vỡ.

Logic đảo ngược

Ví dụ, AI có thể biết “Mẹ của A là B”, nhưng khi hỏi “B là gì của A”, đôi khi nó lại lúng túng (hiện tượng **Reversal Curse**).



6. Chi phí năng lượng và môi trường

Việc huấn luyện và vận hành các LLM khổng lồ tiêu tốn một lượng điện năng và nước (để làm mát máy chủ) cực kỳ lớn.

Mỗi câu hỏi bạn đặt ra cho AI tiêu tốn năng lượng gấp nhiều lần so với một lượt tìm kiếm trên Google truyền thống.

7. Quyền riêng tư và Bản quyền

Đây là vấn đề pháp lý lớn nhất:

Dữ liệu huấn luyện

Nhiều tác giả kiện các công ty AI vì sử dụng tác phẩm của họ mà không xin phép.

Rò rỉ dữ liệu

Nếu bạn nhập thông tin nhạy cảm của công ty vào AI, thông tin đó có thể bị sử dụng để huấn luyện cho các phiên bản sau, dẫn đến nguy cơ lộ bí mật kinh doanh.

Tóm tắt một số hạn chế của LLM

Hạn chế	Mô tả ngắn gọn	Cách khắc phục cho người dùng
Ảo giác	Nói dối một cách tự tin	Luôn kiểm chứng lại nguồn (Fact-check)
Cập nhật	Kiến thức bị giới hạn tại thời điểm huấn luyện	Sử dụng các tính năng có kết nối Internet (Search)
Logic	Dễ sai ở các bài toán nhiều bước	Yêu cầu AI “suy nghĩ từng bước”
Bảo mật	Có thể lưu trữ dữ liệu người dùng	Không nhập mật khẩu, dữ liệu cá nhân & doanh nghiệp nhạy cảm

7

LMM – Mô hình Đa phương thức lớn

LMM (Large Multimodal Model – Mô hình Đa phương thức Lớn)

LMM (Large Multimodal Model – Mô hình Đa phương thức Lớn) là bước tiến hóa tiếp theo của LLM. Nếu LLM chỉ là một "bộ não" giỏi về chữ nghĩa, thì LMM là một bộ não có đầy đủ các giác quan: mắt (nhìn hình ảnh / video) và tai (nghe âm thanh).

Nói cách khác, LMM không chỉ đọc văn bản mà còn **hiểu và xử lý đồng thời nhiều loại dữ liệu khác nhau** trong cùng một kiến trúc thống nhất.

1. Sự khác biệt giữa LLM và LMM

LLM (Chỉ văn bản)

Bạn gửi một tấm ảnh, nó không thấy gì cả. Bạn phải mô tả tấm ảnh đó bằng lời thì nó mới hiểu.

LMM (Đa phương thức)

Bạn gửi một tấm ảnh chụp tủ lạnh, nó “nhìn” thấy trứng, sữa, rau và gợi ý cho bạn công thức nấu ăn ngay lập tức.

2. Cách LMM “nhìn” và “nghe”

LMM thường sử dụng một kiến trúc kết hợp để kết nối các loại dữ liệu khác nhau vào chung một không gian hiểu biết:

01

Bộ mã hóa thị giác (Vision Encoder)

Thường sử dụng kiến trúc **ViT (Vision Transformer)** để chia hình ảnh thành các ô vuông nhỏ (giống như token trong văn bản) và trích xuất đặc trưng.

02

Bộ kết nối (Connector / Adapter)

Một lớp trung gian giúp “dịch” các đặc trưng hình ảnh/âm thanh sang ngôn ngữ mà bộ não ngôn ngữ (LLM) có thể hiểu được.

03

Bộ não ngôn ngữ (LLM Core)

Xử lý tất cả thông tin tổng hợp để đưa ra câu trả lời cuối cùng.

3. Những khả năng đột phá của LMM

Hiểu hình ảnh phức tạp

Giải các bài toán hình học từ ảnh chụp, giải thích các meme (ảnh chế), hoặc đọc biểu đồ tài chính phức tạp.

Phân tích Video theo thời gian thực

Bạn có thể gửi một video bóng đá và hỏi “Tại sao trọng tài lại thổi phạt?”, AI sẽ phân tích các khung hình để trả lời.

Thiết kế và Sáng tạo

Bạn vẽ một bản phác thảo thô sơ lên giấy, LMM có thể hiểu ý tưởng đó và viết code HTML / CSS để tạo ra trang web hoàn chỉnh.

Hỗ trợ người khiếm thị

AI có thể đóng vai trò là “đôi mắt”, mô tả chi tiết những gì đang diễn ra trước camera điện thoại cho người dùng.

4. Các dòng LMM hàng đầu hiện nay (2026)

Mô hình	Điểm mạnh đa phương thức
GPT-4o / GPT-5	Tốc độ phản hồi cực nhanh, có thể hội thoại bằng giọng nói với cảm xúc tự nhiên (Omni)
Gemini 2.0 / 3.0	Khả năng xử lý video cực dài (lên đến vài tiếng) và hiểu ngữ cảnh không gian rất tốt
Claude 3.5 / 4 Vision	Khả năng đọc hiểu tài liệu, bảng biểu và sơ đồ kỹ thuật với độ chính xác cực cao
LLaVA / Llama 4-V	Dòng mã nguồn mở cho phép cộng đồng tự phát triển các ứng dụng thị giác máy tính riêng

5. Thách thức của LMM

Dù mạnh mẽ, LMM vẫn gặp những khó khăn riêng:

Chi phí tính toán

Xử lý hình ảnh và video tốn kém tài nguyên gấp nhiều lần so với văn bản.

Ảo giác thị giác

AI có thể "nhìn gà hóa cuốc", đếm sai số ngón tay trên bàn tay hoặc bỏ lỡ các chi tiết nhỏ trong một bức ảnh dày đặc thông tin.

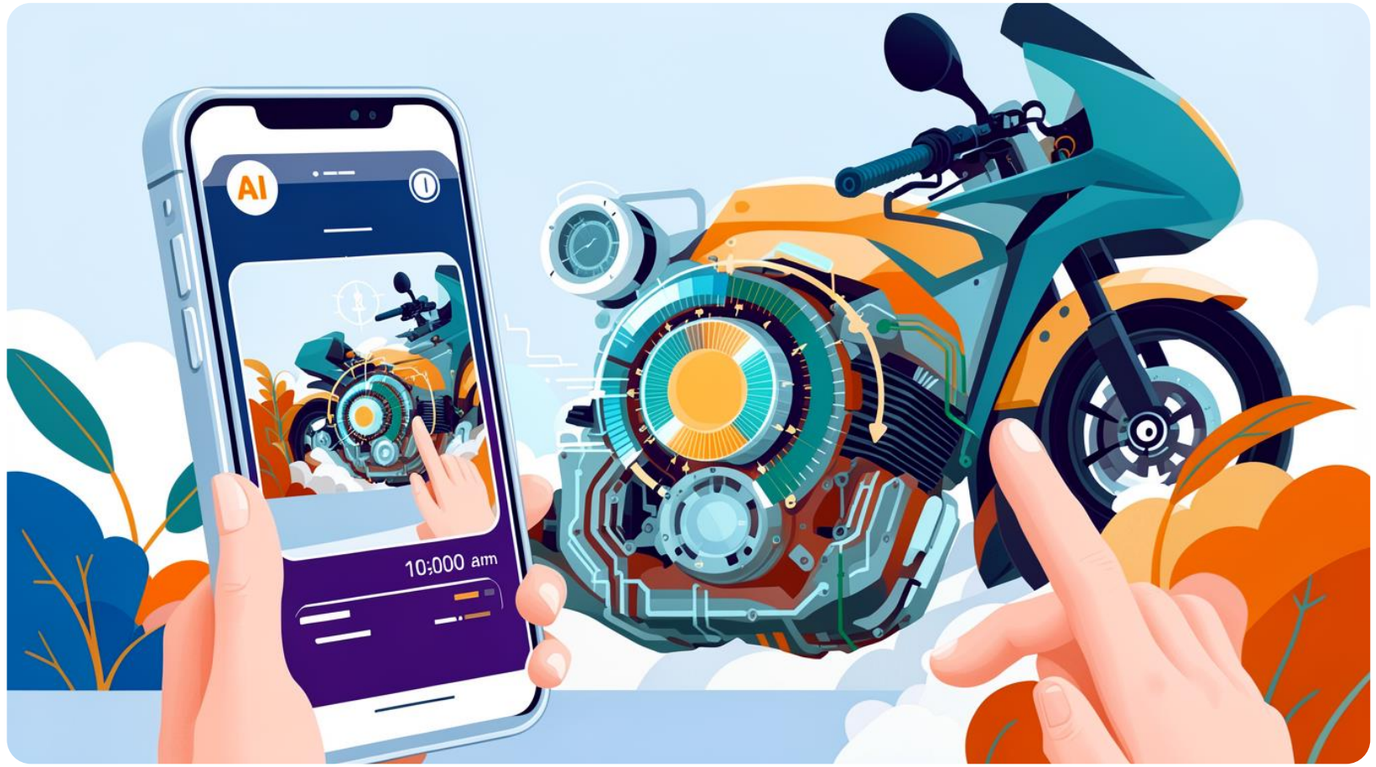
Logic không gian

Đôi khi AI vẫn nhầm lẫn về vị trí trái / phải hoặc khoảng cách giữa các vật thể trong ảnh.

Ví dụ thực tế

Với LMM, bạn có thể quay video động cơ xe máy đang kêu lọc cọc và hỏi: “Máy bị làm sao vậy?”.

AI sẽ nghe tiếng động, nhìn các bộ phận rung lắc và chỉ ra bugi hoặc dây curoa có vấn đề. Đó là điều mà LLM thuần túy không bao giờ làm được.

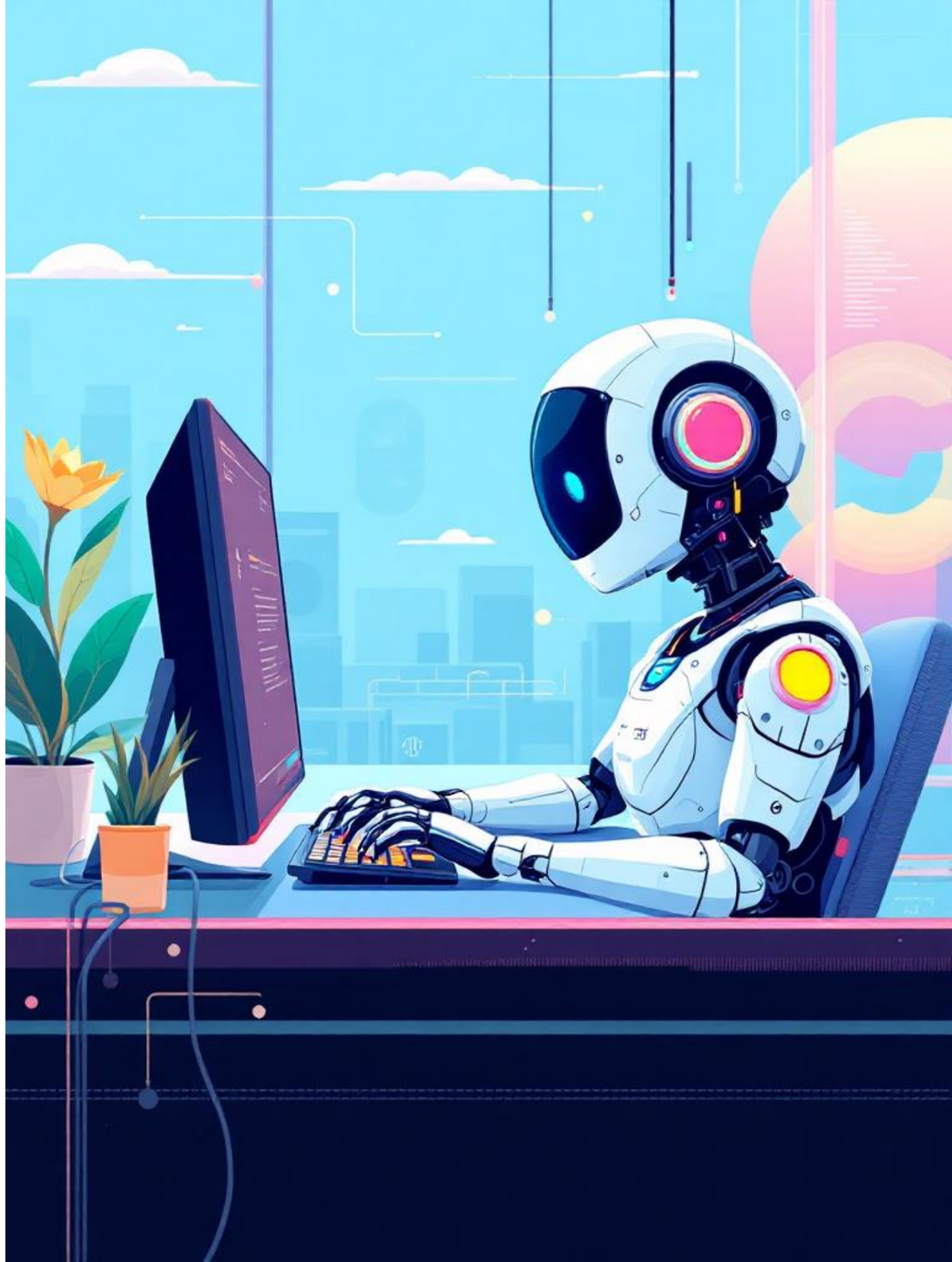


8 AI Agent – AI Platform

Từ “Người Chỉ Biết Nói” đến “Người Làm Được Việc”

Nếu LLM là một “người biết tuốt” (chỉ biết nói), thì **AI Agent** (Tác nhân AI) là một “người làm được việc”.

Đây là bước chuyển mình lớn nhất của công nghệ AI trong năm 2025-2026: Chuyển từ việc chỉ biết trả lời câu hỏi sang việc tự lên kế hoạch và thực thi nhiệm vụ để đạt được mục tiêu cuối cùng mà không cần con người cầm tay chỉ việc từng bước.



AI Agent là gì?

Một AI Agent là một hệ thống sử dụng LLM làm "bộ não", nhưng được trang bị thêm các công cụ và khả năng ra quyết định để tương tác với thế giới bên ngoài.

Công thức tạo nên AI Agent:

AI Agent =

- **LLM (Suy luận)**
- **+ Planning (Lập kế hoạch)**
- **+ Memory (Ghi nhớ)**
- **+ Tool Use (Sử dụng công cụ)**

4 Thành phần cốt lõi của một AI Agent

Brain (Bộ não)

Thường là các LLM mạnh như GPT-4o, Claude 3.5 hoặc Gemini 1.5. Nó đóng vai trò suy luận, hiểu ý định của người dùng.

Planning (Lập kế hoạch)

Agent biết chia nhỏ một yêu cầu phức tạp thành các bước nhỏ hơn.

Ví dụ: Để "lên kế hoạch chuyến đi", nó biết phải: 1. Tìm vé máy bay → 2. Đặt khách sạn → 3. Lên lịch trình tham quan.

Memory (Trí nhớ)

Trí nhớ ngắn hạn: Lưu ngữ cảnh cuộc hội thoại hiện tại.

Trí nhớ dài hạn: Lưu thói quen, sở thích của người dùng qua nhiều ngày (thường dùng Vector Database).

Action / Tools (Hành động)

Đây là điểm khác biệt nhất. Agent có thể sử dụng:

- Trình duyệt web để tìm thông tin
- Email để gửi báo cáo
- Code để giải toán hoặc vẽ đồ thị
- API để kết nối với các phần mềm khác (Lark, Slack, Zapier)

Cách AI Agent hoạt động

Vòng lặp Suy nghĩ

AI Agent thường hoạt động theo mô hình **ReAct (Reason + Act)**:

1

Suy nghĩ (Thought)

Tôi cần làm gì tiếp theo?

2

Hành động (Act)

Sử dụng công cụ (ví dụ: Tìm kiếm Google).

3

Quan sát (Observation)

Kết quả tìm kiếm có ích không?

4

Lặp lại

Nếu chưa đủ, nó lại suy nghĩ và hành động tiếp cho đến khi hoàn thành.

Các loại AI Agent phổ biến hiện nay

Loại AI Agent	Chức năng chính	Ví dụ thực tế
Research Agent	Tìm kiếm, tổng hợp tài liệu chuyên sâu từ Internet.	Perplexity, GPT Researcher.
Coding Agent	Tự viết code, chạy thử, tìm lỗi và sửa lỗi.	Devin (AI Software Engineer), Cursor.
Personal Assistant	Tự đặt lịch hẹn, quản lý email, đặt đồ ăn.	Các Agent tích hợp trong Siri mới hoặc Gemini.
Multi-Agent Systems	Nhiều AI Agent nói chuyện với nhau để giải quyết việc lớn.	Một AI Agent làm “Sếp”, một AI Agent “Viết code”, một AI Agent “Kiểm thử”

Tại sao AI Agent lại quan trọng?

Trước đây

Bạn phải copy-paste dữ liệu giữa các tab:
Copy từ PDF → Paste vào dịch → Copy vào Email.

Với AI Agent

Bạn chỉ cần ra lệnh: “**Hãy đọc file PDF này, tóm tắt ý chính và gửi email cho đối tác của tôi bằng tiếng Pháp.**”
Agent sẽ tự mở file, tự dịch và tự vào Gmail để gửi.

Thách thức hiện tại

Vòng lặp vô tận

AI Agent có thể bị “kẹt” trong một vòng lặp logic nếu không tìm thấy lời giải.

Độ tin cậy

Liệu bạn có dám để AI Agent tự ý quẹt thẻ tín dụng của mình để mua vé máy bay?

Bảo mật

AI Agent có quyền truy cập vào các tài khoản cá nhân, nếu bị tấn công sẽ rất nguy hiểm.

Các Coding Agent hàng đầu hiện nay

Trong năm 2026, các **Coding Agent** đã tiến hóa từ những công cụ hỗ trợ gõ code (autocomplete) thành những "kỹ sư phần mềm AI" thực thụ, có khả năng tự chẩn đoán lỗi, viết test case và quản lý toàn bộ vòng đời của một dự án (Repository).

Dưới đây là những cái tên hàng đầu đang thống trị thị trường:

Các Coding Agent hàng đầu

1 Devin (by Cognition) – Kỹ sư AI "độc lập" đầu tiên

Devin được coi là cột mốc thay đổi cuộc chơi khi lần đầu tiên xuất hiện với tư cách một Agent có thể tự hoàn thành dự án từ đầu đến cuối.

Khả năng:

Tự lập kế hoạch, tự mở terminal để cài đặt môi trường, tự viết mã, tự sửa lỗi (debug) dựa trên log và tự deploy sản phẩm.

Điểm mạnh:

Hoạt động như một nhân viên thực sự. Bạn có thể giao cho Devin một issue trên GitHub, nó sẽ tự đọc mã nguồn hiện có và đưa ra giải pháp (Pull Request).

2 Cursor – Trình soạn thảo mã nguồn (IDE) phổ biến nhất

Cursor không hẳn là một Agent độc lập mà là một IDE (dựa trên VS Code) tích hợp sâu AI Agent vào bên trong.

Khả năng:

Tính năng Composer cho phép bạn yêu cầu thay đổi mã nguồn trên nhiều tệp tin cùng lúc. Nó hiểu toàn bộ cấu trúc thư mục của bạn thay vì chỉ một tệp đang mở.

Điểm mạnh:

Trải nghiệm người dùng cực mượt. Nó biết chính xác vị trí cần sửa và cho phép bạn "Accept" hoặc "Reject" từng dòng code chỉ bằng một cú click.

Các Coding Agent hàng đầu

3 GitHub Copilot Workspace – Hệ sinh thái của Microsoft

Tận dụng lợi thế sở hữu kho mã nguồn lớn nhất thế giới, GitHub đã ra mắt Workspace để biến mọi ý tưởng trong phần "Issue" thành code.

Chuyển đổi từ ngôn ngữ tự nhiên thành một kế hoạch thực thi, sau đó tạo ra một không gian làm việc đám mây để Agent tự động viết code.

Tích hợp sâu nhất với quy trình CI / CD và các công cụ quản lý dự án của GitHub.

4 OpenDevin & SWE-agent – Lựa chọn Nguồn mở (Open Source)

Nếu bạn lo ngại về quyền riêng tư hoặc muốn tùy chỉnh Agent theo ý mình, đây là những dự án hàng đầu:

OpenDevin (nay là OpenManus): Một nỗ lực của cộng đồng để tái hiện lại khả năng của Devin nhưng hoàn toàn miễn phí và minh bạch.

SWE-agent: Được phát triển bởi các nhà nghiên cứu tại Đại học Princeton, tập trung vào việc biến LLM thành kỹ sư phần mềm chuyên giải quyết các lỗi thực tế trong kho mã nguồn.

So sánh nhanh các Coding Agent

Công cụ	Mô hình bộ não	Cách hoạt động chính	Đối tượng phù hợp
Devin	GPT-4o / Proprietary	Chạy trong môi trường riêng biệt (Sandboxed)	Thuê làm việc như "Cộng tác viên"
Cursor	Claude 3.5 Sonnet / GPT-4o	Tích hợp trực tiếp vào trình soạn thảo	Lập trình viên cá nhân muốn tăng tốc
GitHub Workspace	GPT-5 (tương đương)	Dựa trên Issue và Pull Request	Các team dùng hệ sinh thái GitHub
SWE-agent	Llama 3 / GPT-4	Xử lý các lỗi kỹ thuật (Software Engineering)	Nhà nghiên cứu và Startup muốn tùy chỉnh

Tại sao Coding Agent lại mạnh hơn Chatbot truyền thống?

Khác với việc bạn hỏi ChatGPT "Viết cho tôi đoạn code này", các Coding Agent sử dụng một vòng lặp "**Think – Act – Observe**":

Đọc hiểu

Quét toàn bộ mã nguồn của dự án để hiểu ngữ cảnh (Context).

Lên kế hoạch

Quyết định cần sửa những file nào, thêm thư viện nào.

Thực thi & Kiểm tra

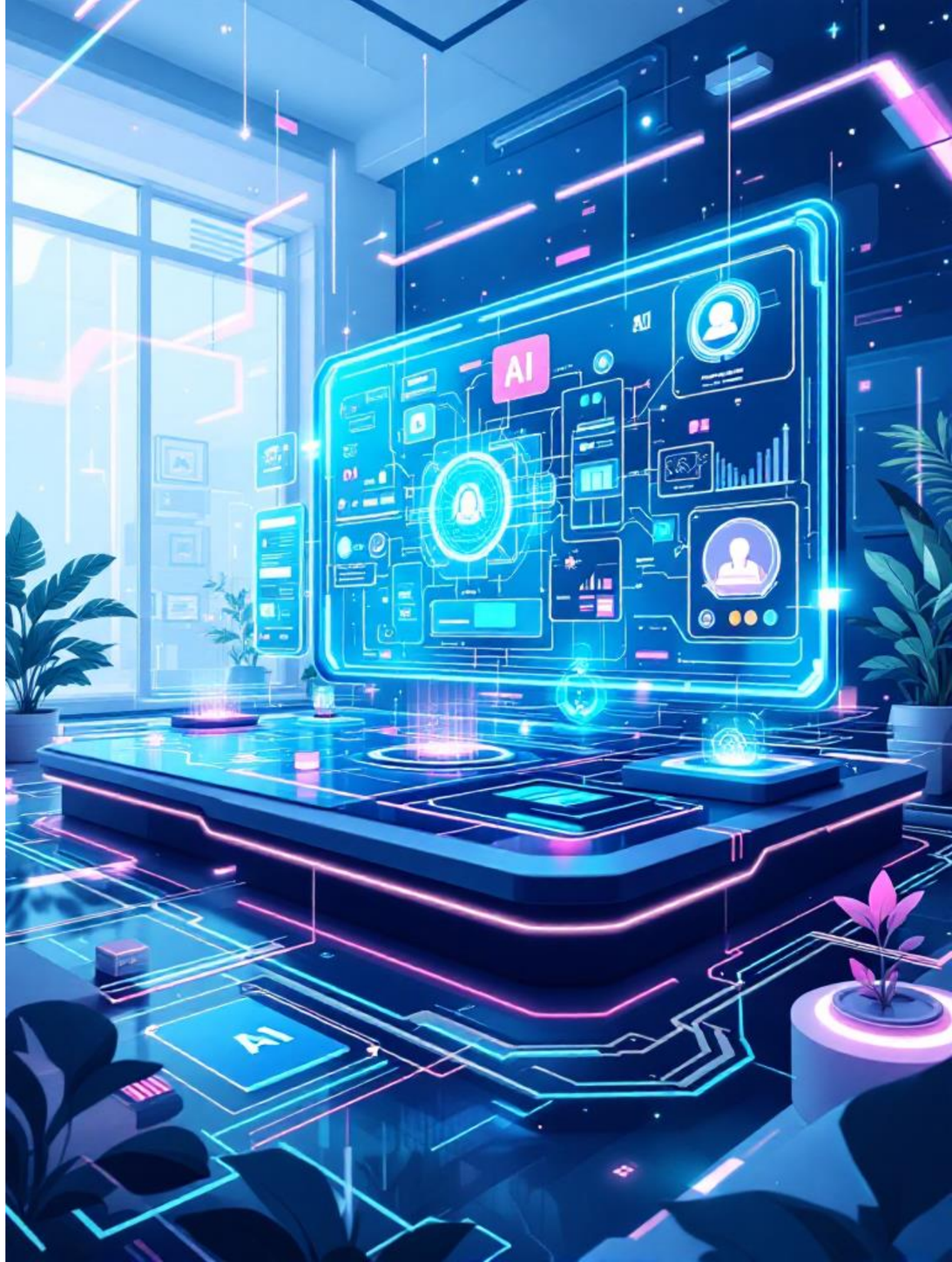
Viết code xong sẽ tự chạy lệnh test. Nếu lỗi, nó sẽ đọc thông báo lỗi và tự sửa lại cho đến khi vượt qua bài kiểm tra.

Lời khuyên: Nếu bạn mới bắt đầu, **Cursor** là công cụ dễ tiếp cận và mang lại hiệu quả tức thì nhất. Nếu bạn muốn giao phó những mảng việc lớn hơn (như bảo trì cả một hệ thống), hãy thử nghiệm với **Devin** hoặc **GitHub Workspace**.

Nền tảng AI 2026: Từ Chat sang Hệ điều hành Trí tuệ

Trong năm 2026, các nền tảng AI (AI Platforms) đã chuyển dịch từ việc chỉ là những trang web để chat sang những **hệ điều hành trí tuệ** toàn diện. Chúng không chỉ cung cấp mô hình ngôn ngữ (LLM) mà còn tích hợp cả công cụ tạo ảnh, video, phân tích dữ liệu và tự động hóa quy trình.

Dưới đây là các nền tảng mạnh nhất hiện nay được phân chia theo mục đích sử dụng.



1. Top các Nền tảng “Hệ sinh thái” Toàn năng

Đây là những nơi bạn có thể tìm thấy mọi công cụ AI trong cùng một giao diện.

OpenAI Platform (ChatGPT Plus / Team / Enterprise)

Thế mạnh: Đi đầu với mô hình **GPT-5** và khả năng suy luận vượt trội từ dòng **o3**.

Tính năng mới: Khả năng **“Advanced Voice & Vision”** cho phép hội thoại trực tiếp bằng giọng nói và camera theo thời gian thực không có độ trễ. Các **GPTs** giờ đây hoạt động như những Agent thực thụ, có thể tự động thực hiện các tác vụ liên ứng dụng.

Google AI Studio & Gemini

Thế mạnh: Sở hữu cửa sổ ngữ cảnh (Context Window) lớn nhất thị trường (lên đến **10 triệu tokens**).

Tính năng mới: Tích hợp cực sâu vào hệ sinh thái Google (Workspace). Bạn có thể yêu cầu AI đọc toàn bộ lịch sử email và tài liệu trong Drive của 5 năm qua để lập một báo cáo tổng hợp chỉ trong vài giây.

Microsoft Azure AI Studio

Thế mạnh: Dành cho doanh nghiệp và các nhà phát triển chuyên nghiệp.

Tính năng mới: Cung cấp quyền truy cập vào tất cả mô hình mạnh nhất (OpenAI, Meta, Mistral) kèm theo các công cụ kiểm soát an toàn dữ liệu và bản quyền cấp độ doanh nghiệp.

2. Top các Nền tảng “Nhà máy AI”
(Open Source & Infrastructure)

Nơi các nhà phát triển xây dựng và triển khai các mô hình tùy chỉnh.

Hugging Face

Được mệnh danh là “GitHub của giới AI”.

Thế mạnh: Lưu trữ hàng triệu mô hình mã nguồn mở, bộ dữ liệu và các ứng dụng demo (Spaces). Đây là nơi bạn tìm thấy các bản cập nhật mới nhất của **Llama 4**, **Mistral** hay **Stable Diffusion 4**.

Poe

Thế mạnh: Nền tảng tổng hợp. Chỉ với 1 gói đăng ký, bạn có thể sử dụng tất cả các mô hình hàng đầu (GPT-4o, Claude 3.5, Gemini 1.5, Llama 3) trên cùng một giao diện.

Tính năng mới: Cho phép người dùng tạo ra các Bot đa phương thức vô cùng nhanh chóng mà không cần code.

3. Top các Nền tảng "Chuyên dụng" (Specialized Platforms)

Dành cho các lĩnh vực đòi hỏi chất lượng cao nhất.

Lĩnh vực	Nền tảng hàng đầu	Đặc điểm nổi bật
Lập trình	**Cursor / Replit Agent**	Tự động viết, sửa và deploy ứng dụng hoàn chỉnh từ ý tưởng.
Nghiên cứu	**Perplexity / Elicit**	AI thay thế Google Search, trích dẫn nguồn cực kỳ chính xác và uy tín.
Sáng tạo Video	**Sora / Luma Dream Machine**	Tạo video từ văn bản với độ chân thực không thể phân biệt được với phim thực tế.
Thiết kế Ảnh	**Midjourney v7 / Flux.1**	Đỉnh cao về nghệ thuật thị giác và khả năng hiểu ngôn ngữ điều khiển phức tạp.

4. Xu hướng mới: Nền tảng AI "Tự trị" (Agentic Platforms)

Đây là biên giới mới nhất của năm 2026. Các nền tảng này không đợi bạn ra lệnh từng bước mà tự vận hành:

Coze (by ByteDance)

Cho phép người dùng bình thường xây dựng các AI Agent phức tạp có thể kết nối với mạng xã hội, công cụ văn phòng và API bên ngoài.

CrewAI / LangChain

Các framework mạnh nhất để các công ty xây dựng một "đội ngũ AI" (ví dụ: một AI Agent viết nội dung, một AI Agent kiểm tra lỗi, một AI Agent đăng bài lên mạng xã hội).

Nên chọn nền tảng nào? Hãy bắt đầu với ...

Cá nhân, học tập

Chọn **ChatGPT** hoặc **Claude** (để có sự logic và văn phong hay nhất).

Phân tích tài liệu siêu dài

Chọn **Gemini 2.5/3 Pro**.

Lập trình viên

Chắc chắn là **Cursor**.

Doanh nghiệp cần bảo mật

Chọn **Azure AI** hoặc tự host **Llama 4** qua **Hugging Face**.

Về tác giả bài tổng hợp:
Mr. Eric Lại Đức Nhuận

Kinh nghiệm:

- 25 năm kinh nghiệm tư vấn & phát triển các giải pháp công nghệ cho các DN lớn trong & ngoài nước, chuyên sâu về các lĩnh vực:
  - Giải pháp quản trị chuỗi cung ứng, sàn giao dịch TMĐT B2B, Logistics
  - Giải pháp quản lý mua sắm trang thiết bị cho hệ thống bệnh viện tại Mỹ
  - Giải pháp tự động mua bán chứng khoán cho thị trường Mỹ và Châu Âu
  - Giải pháp công nghệ cho các ngân hàng quốc tế như ANZ, Shinhan, Standard Chartered
- Xây dựng và phát triển phần mềm theo tiêu chuẩn CMMI, ISO 9001:2015, và ISO 27001:2022
- Đã trực tiếp đào tạo PMBOK cho 600+ nhà quản lý dự án CNTT tại VN
- Diễn giả tại các hội thảo, hội nghị về phát triển phần mềm, giải pháp quản trị chuỗi cung ứng, TMĐT B2B và tối ưu hóa hiệu quả kinh doanh

Kỹ sư CNTT (Honor)
ĐHBK TP. HCM (2000)

Thạc sĩ CNTT,
ĐHBK TP. HCM (2004)

Chứng chỉ Quản lý dự án PMP,
Viện Quản lý Dự án Hoa Kỳ (2010)

2018 — ATALINK

Nhà sáng lập ATALINK - Giải pháp quản trị chuỗi cung ứng tích hợp Sàn giao dịch TMĐT B2B tiên phong tại thị trường Việt Nam

2009 — Meperia

Đồng sáng lập MEPERIA - Giải pháp hàng đầu để các bệnh viện quản lý mua sắm thiết bị y tế tại Mỹ

2003 — LARION

Nhà sáng lập, TGĐ LARION - Công ty gia công phần mềm CNTT hàng đầu Việt Nam, với quy mô 200+ nhân sự

2001 — Arrive

Trưởng phòng phần mềm (hệ thống nhúng - chip viễn thông thế hệ mới). Công ty này hiện là một bộ phận của tập đoàn Marvell (Mỹ)

BKA TP.HCM CSE Alumni — Ban đại diện CSV Khoa CNTT, ĐHBK TP.HCM(2020)

VNIT Alliance — Thành viên sáng lập ban điều hành Liên minh các DN gia công CNTT Việt Nam (2015)

YBA — Thành viên BCH Hội Doanh nhân trẻ TP. HCM NK 9, 10, 11

HCA — Thành viên BCH Hội Tin học TP. HCM NK 6, 7

Tài liệu tham khảo (References)

- History of artificial intelligence (AI) - Link
- AI Milestones Timeline - Link
- Alan Turning - Computing Machinery and Intelligence - Link
- Dartmouth Workshop (1956) - Link
- Deep Blue - Link
- AlexNet - Link
- Alex Krizhevsky et al. – ImageNet Classification (2012) - Link
- VGGNet – Very Deep CNNs - Link
- Deep Residual Learning for Image Recognition - Link
- NeurIPS Proceedings Page – Attention Is All You Need (2017) - Link
- Transformer (Deep Learning) - Link
- Large Language Model - Link
- A Comprehensive Overview of Large Language Models - Link
- Vision Transformer (ViT) - Link
- ReAct: Reason + Act (Princeton) - Link

THANK YOU!

Atalink – Giải pháp quản trị chuỗi cung ứng hợp nhất

Tương tác trong – ngoài Doanh nghiệp chỉ cần 1 nền tảng

Tải ứng dụng trên điện thoại

Available on the App Store

Available on the Google Play

Liên hệ

www.atalink.com **1800 555 540** contact@atalink.vn

Tầng 3, Tòa nhà QTSC 1, Lô 34, Đường số 14, Công viên phần mềm Quang Trung, Phường Trung Mỹ Tây, Thành phố Hồ Chí Minh, Việt Nam