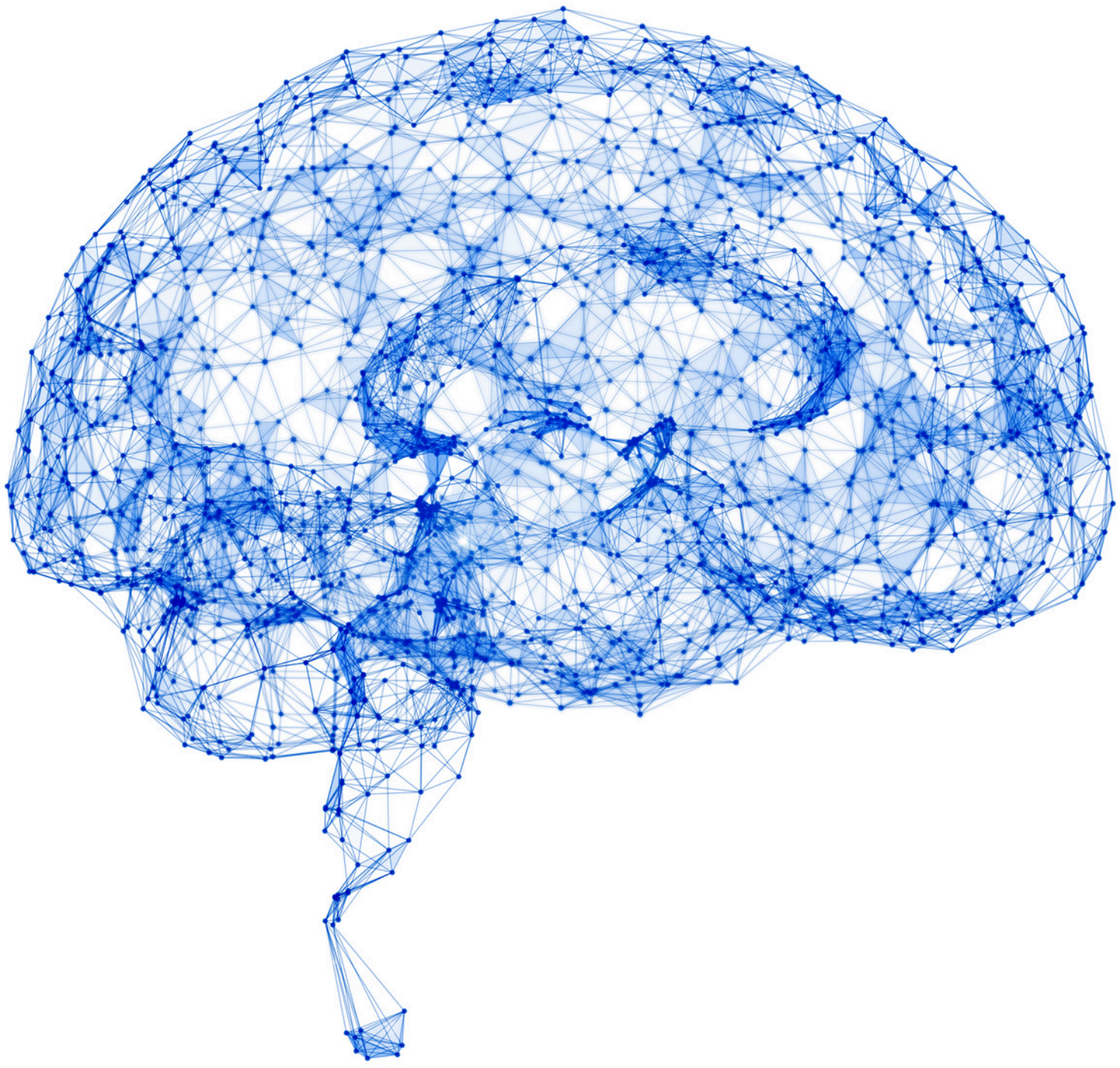
HOW DO LLMS HANDLE LONG CONTEXT?

Learn everything that matters about LLM's Memory

Bhavishya Pandit

Introduction

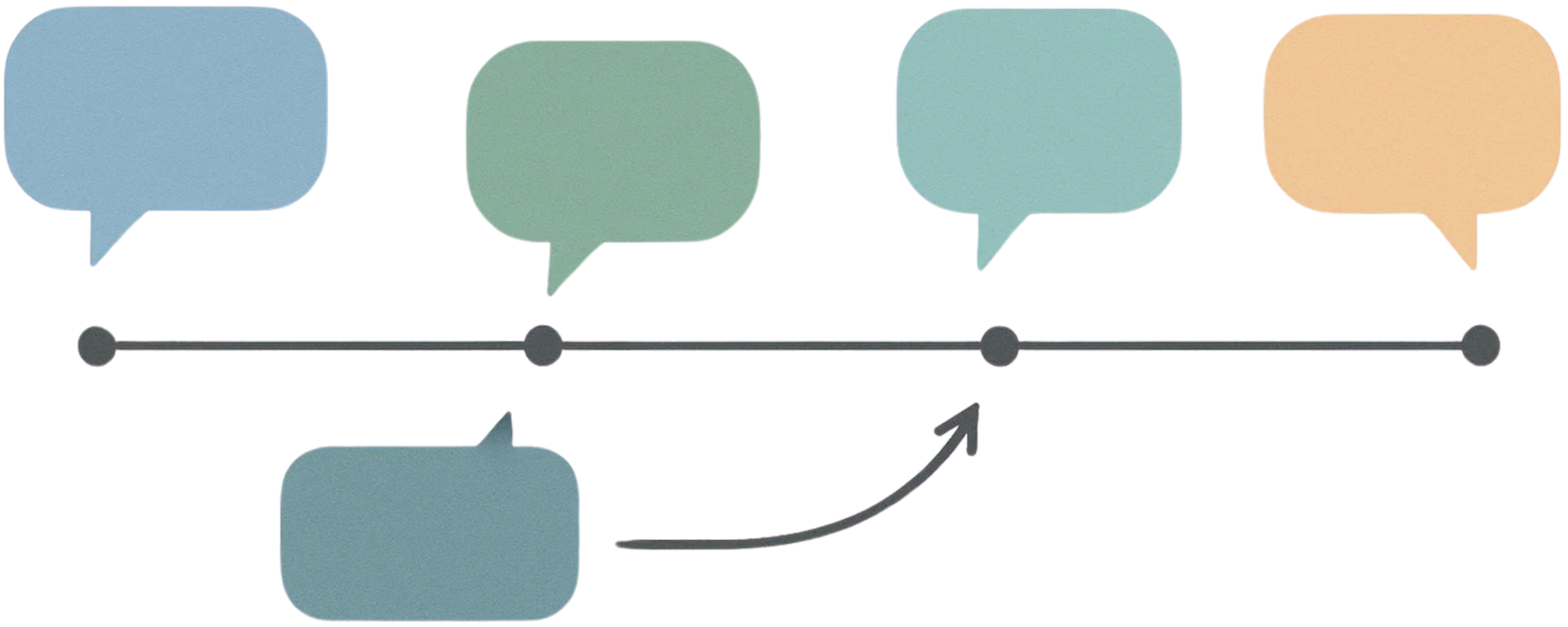


Yann LeCun argues that persistent memory is one of four essential human traits currently missing in today’s LLMs.

To address this gap, this post will delve into:

- Why memory is crucial for coherence and personalisation
- Foundational memory methods
- Challenges of memory in LLMs
- Advancements in LLMs' memory retention

Why memory matters in LLMs?



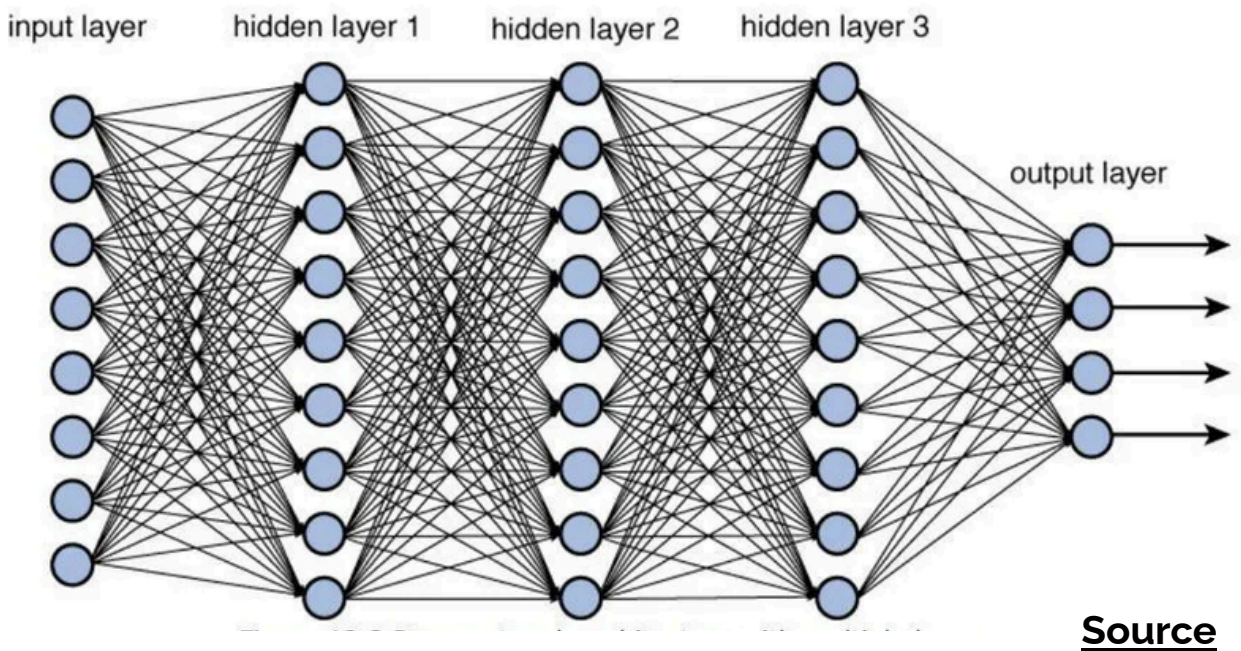
- Memory means the model's ability to **encode**, **retain**, and **recall** contextual tokens within a prompt.
- It enables LLMs to **carry continuity** over long text or conversations.
- Without memory, each response is like a goldfish—forgetting the last thing it heard, requiring users to constantly reintroduce themselves and restate information.
- Memory allows the model **to reference earlier details**, **improving coherence** and **accuracy**.
- It underpins coherent **multi-step reasoning**, **maintaining context** across turns.

Foundational Memory in LLMs

Ever wondered how LLMs like BERT and GPT keep track of so much information? They actually use two kinds of memory:

Parametric Memory

- Think of this as the LLM's "long-term memory"
- Billions of parameters storing facts, language rules, and patterns learned during training.
- It's fast & always available, but can't be easily updated after training.

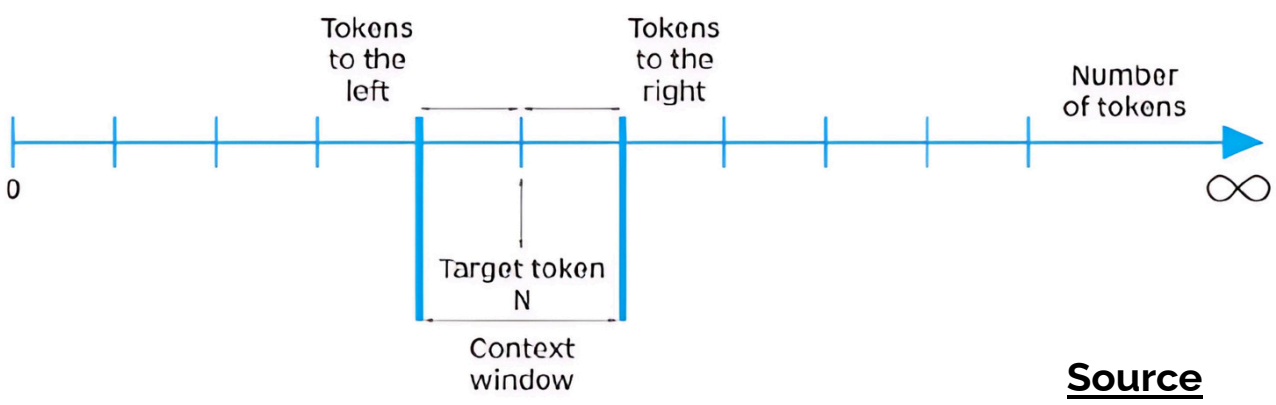


Source

Context Windows

The "Short-Term Buffer"

- AI is able to see last few messages.
- The context Window updates with new input.
- All within window is instantly usable

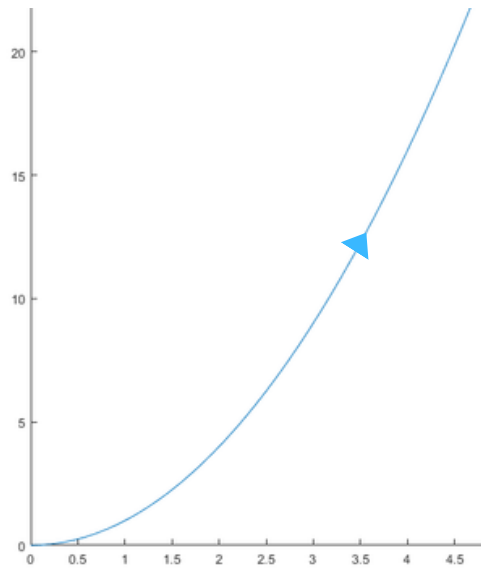


Source

Challenges in LLMs

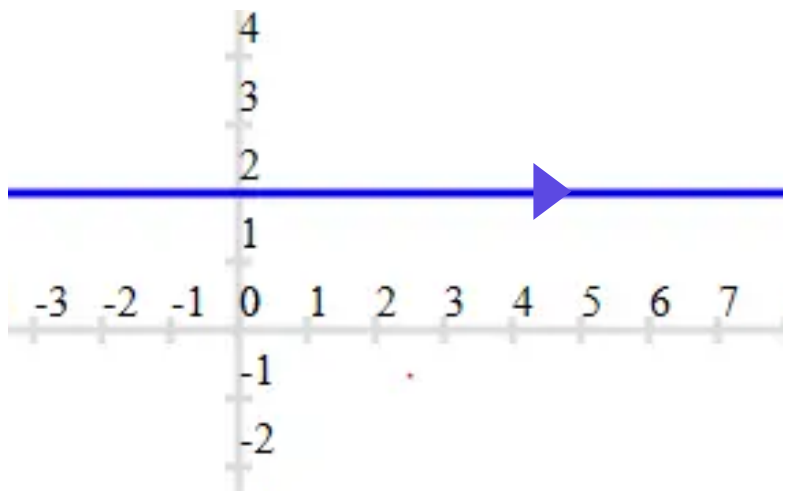
Yes, with these 2 types of memory, the results were good but with that they had some challenges - let’s take a look at what were they :

Quadratic Scaling



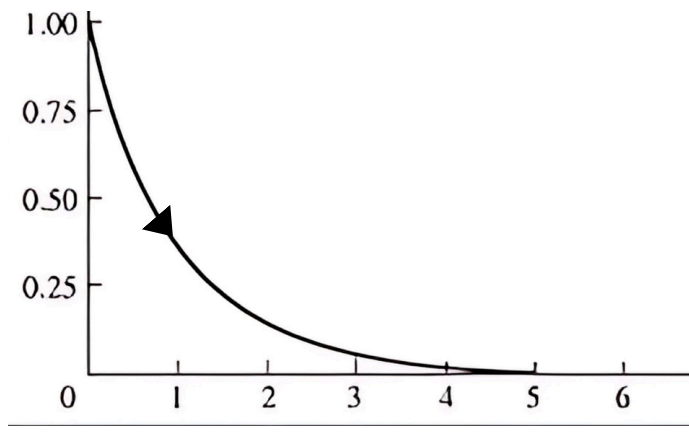
Processing cost grows much faster with size of input

Knowledge Staleness



Once trained, LLMs’ knowledge is frozen in place.

Catastrophic Forgetting



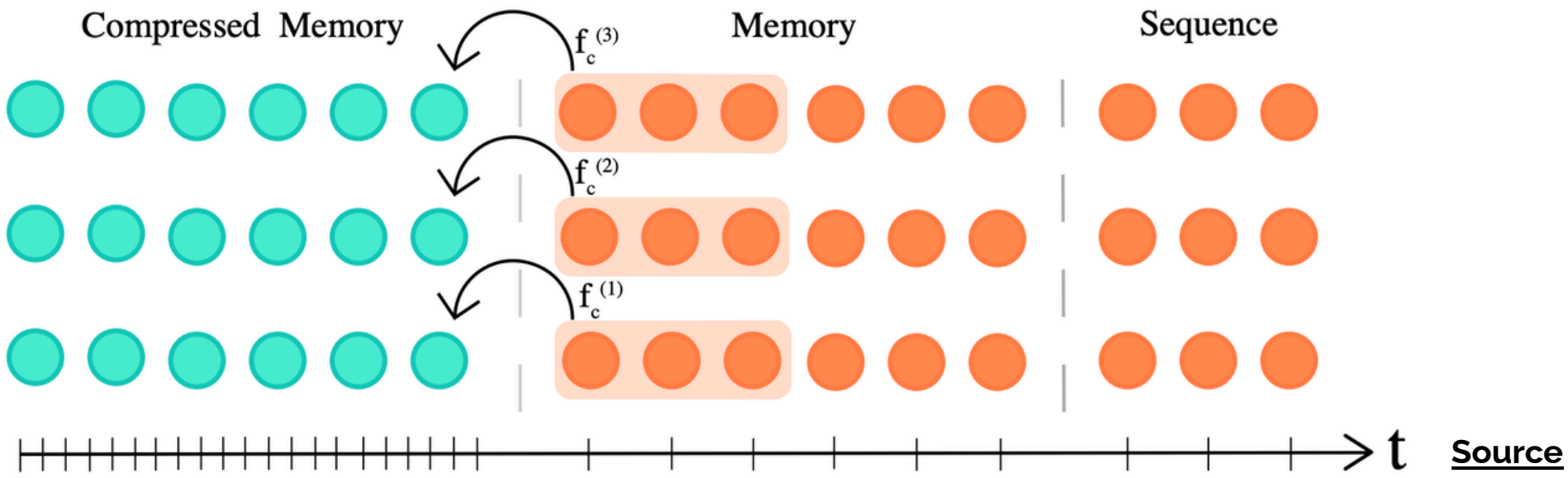
Updation arises a risk overwriting valuable info.

Advancements in LLMs' Memory

So, how did researchers tackle these challenges? Let's learn about the breakthroughs that followed.

Compressive Transformers

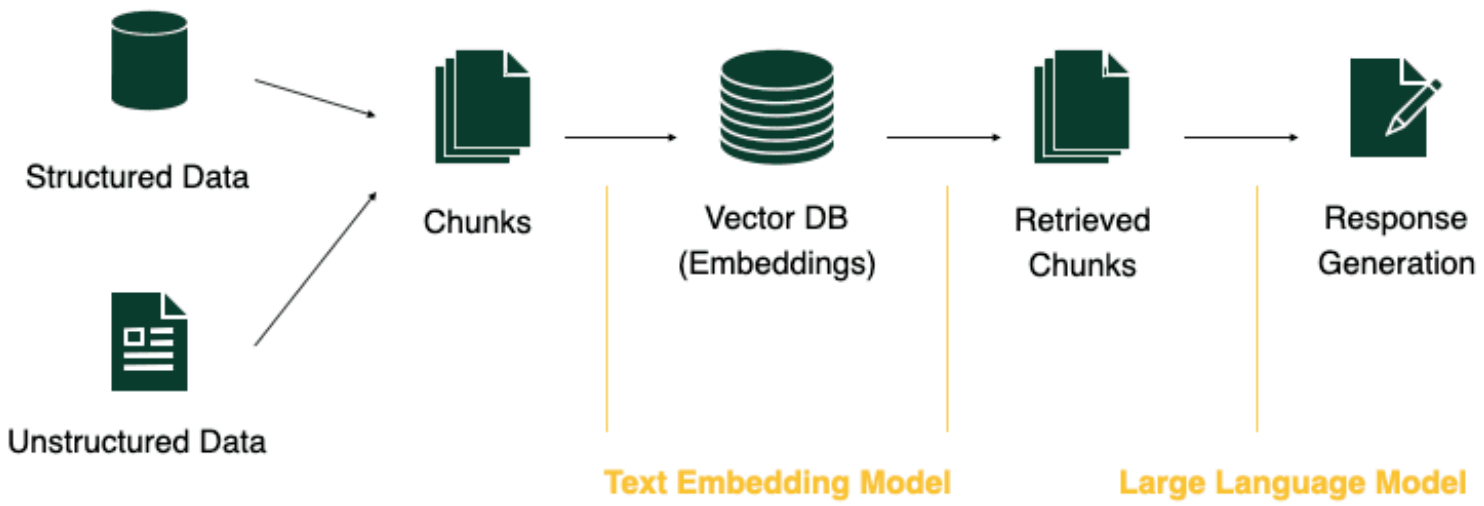
- Maintains both **detailed short-term memory** and **compressed long-term memory** managed with a FIFO mechanism.
- Uses **separate optimization loops for the main model and the memory components** for learning & memory management.



Source

Retrieval-Augmented Generation

- LLMs can now **look up information in real time** from vast databases or knowledge graphs—like having a librarian on call.
- This **keeps responses fresh**, **accurate**, and **up-to-date**.



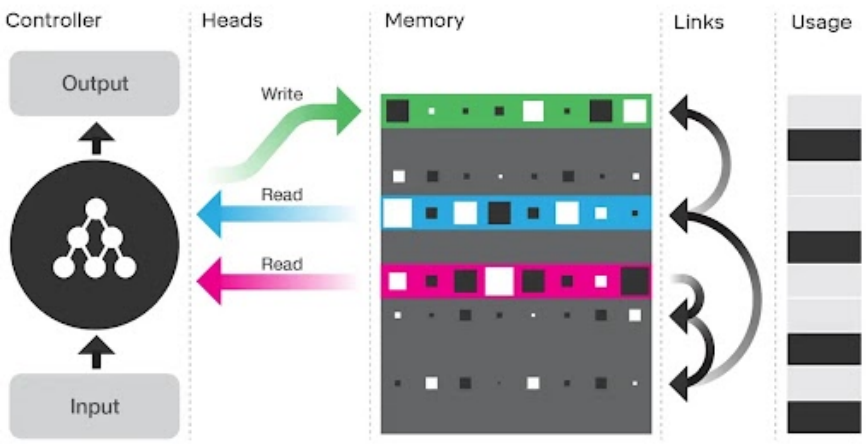
Source

Advancements in LLMs' Memory

Contd.

Differentiable Neural Computers

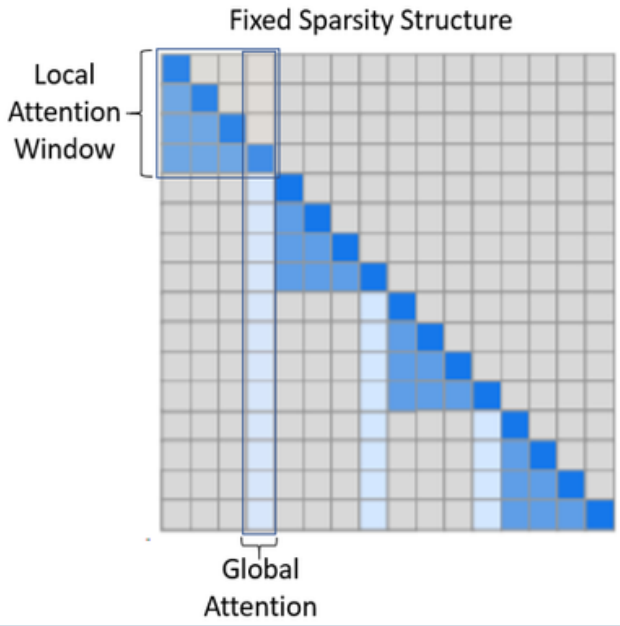
- **Combines a neural network with an external memory matrix**, using attention mechanisms to control where and when information is stored or retrieved.
- **Excels at tasks requiring complex working memory**, such as sequential processing and reasoning.



Source

Sparse Attention Mechanisms

- Uses **local sliding window attention** and **selective global attention**, efficiently capturing key dependencies while reducing complexity from $O(n^2)$ to $O(n)$.
- **Hybrid attention patterns** balance speed and context awareness, enabling models to handle long sequences without sacrificing important information.



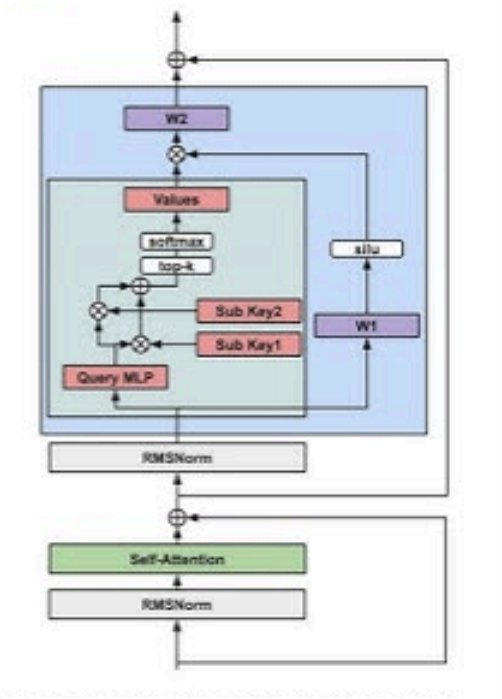
Source

Advancements in LLMs' Memory

Contd.

Memory Layer Specialization

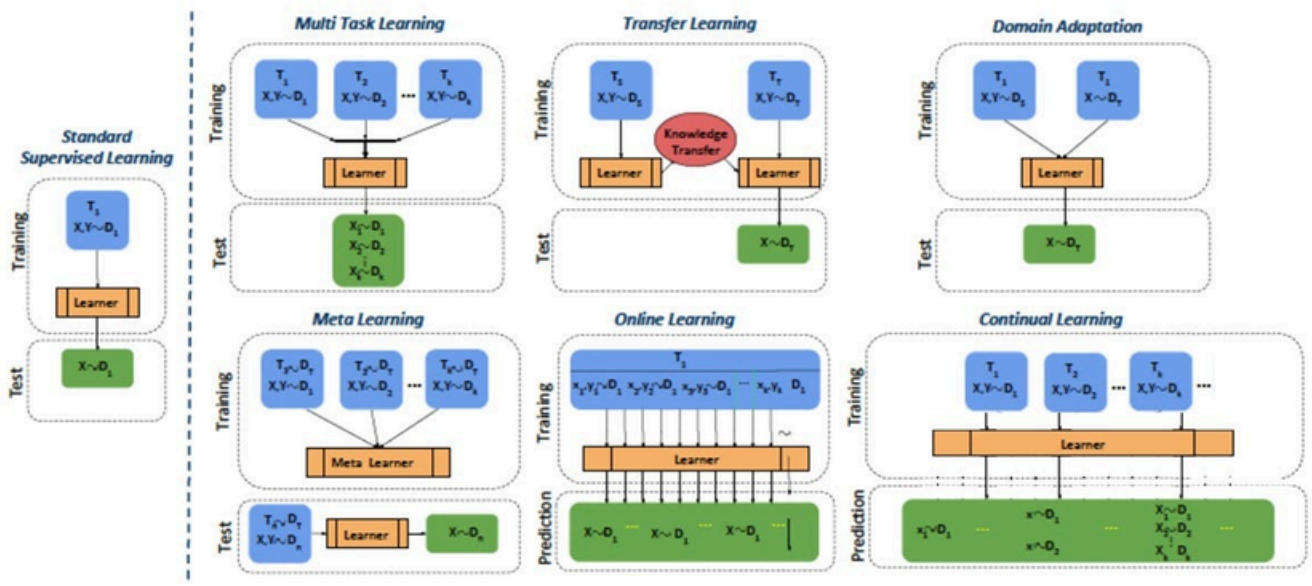
- Applies **specialized memory mechanisms to selected transformer layers**, often **integrating long-term memory** at later layers where deep dependencies matter most.
- Utilizes **hierarchical memory structures** and controlled write operations to refine what information is stored and retained at each stage.



Source

Continual Learning & Forgetting

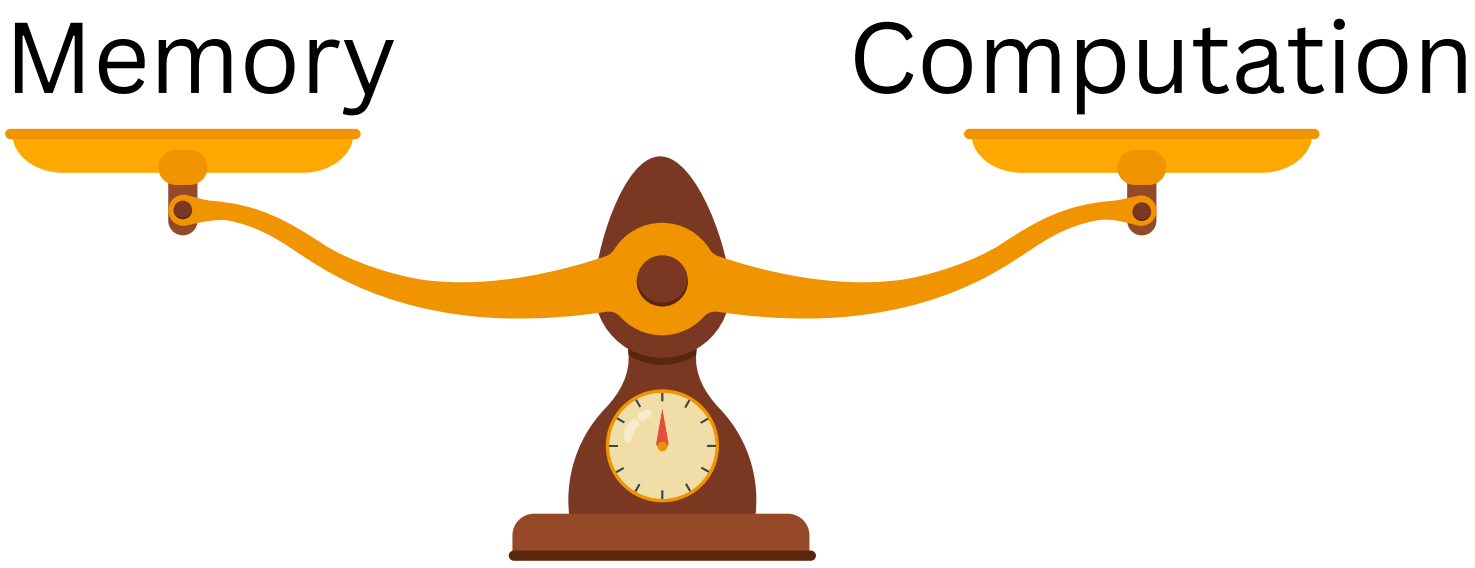
- **Inspired by the brain**, LLMs can now **forget outdated info** and reinforce what's important.
- **Continual learning** lets them add new knowledge without overwriting what they already know.



Source

Considerations

Ever wondered what it takes to actually run and train these giant LLMs? Here’s how researchers and engineers make it work:



Trade-Offs

- It’s all about finding the right balance: bigger models and longer texts need more resources, so engineers use tricks like gradient checkpointing and smart batch sizing to train efficiently even with limited hardware.
- Techniques like sparse attention help models handle longer sequences without running out of memory.
- Training and inference have different needs: training is resource-heavy, while inference must be fast and efficient
- Curriculum learning and smart memory setup help models learn better and generalize across tasks—even when memory is tight

Do you know which LLM has the largest context window?

Let me know in the comments

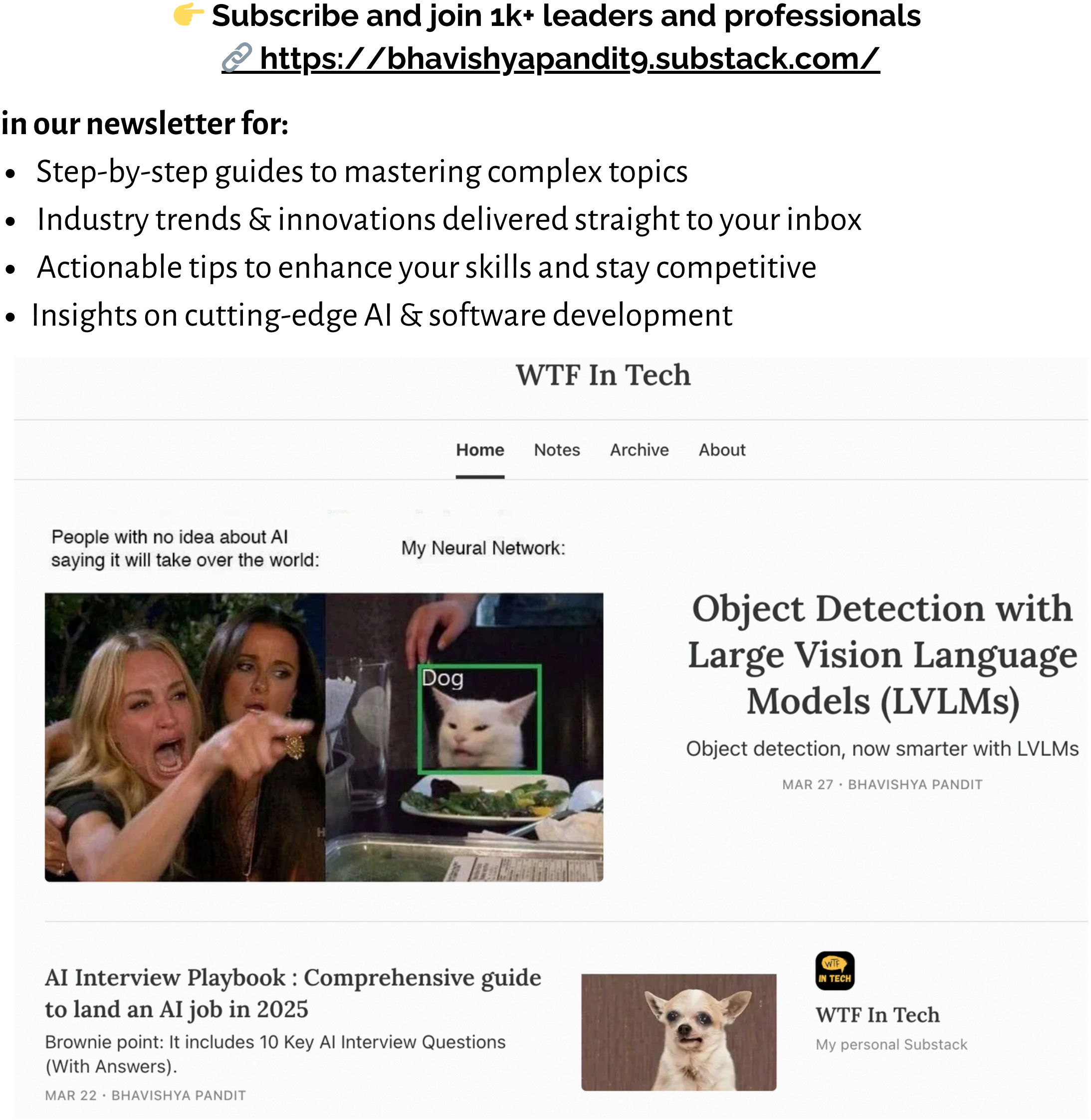
Stay Ahead with Our Tech Newsletter! 🚀

👉 **Subscribe and join 1k+ leaders and professionals**

🔗 **https://bhavishyapandit9.substack.com/**

Join our newsletter for:

- Step-by-step guides to mastering complex topics
- Industry trends & innovations delivered straight to your inbox
- Actionable tips to enhance your skills and stay competitive
- Insights on cutting-edge AI & software development



💡 Whether you're a developer, researcher, or tech enthusiast, this newsletter is your shortcut to staying informed and ahead of the curve.

Follow to stay updated on Generative AI

LIKE

COMMENT

REPOST