Day-01:Introduction to GenAI

1.What is LLM

LLM (Large Language Model)
It is a **computer program** that can **understand and write text** like a human.

How it is trained (very simple):

1. It is given **a lot of text** (books, websites).
2. It **learns patterns** from that text.
3. It learns to **guess the next word** in a sentence.

That's it — this is how an LLM is trained.

2.What is Token

A **token** is a **small piece of text**.

An LLM does not read full sentences at once.
It reads text **piece by piece**, and each piece is called a **token**.

That's it.

Examples of pattern prediction:

1. **3100, 3200, 3300, 3400, ?** → 3500
2. **Sun rises from the ____?** → east
3. **Twinkle twinkle little ___?** → star
4. **Modi is prime minister of ___?** → India
5. **Roses are red, violets are ___?** → blue
6. **अक्ल बड़ी या बल?**
7. **धोबी का कुत्ता, न घर का न घाट का**
8. **Unusual pattern:** 17 → 360 (just a pattern guess, not real math)

Why these examples?

Because **LLMs only predict** what **usually comes next** in such patterns.

What GenAI can do:

✓ Predict next word/token
✓ Generate text, image, audio, code-like text

What GenAI cannot do:

✗ It cannot calculate like a human
✗ It cannot run code
✗ It cannot search the internet
✗ It does not think
✗ It does not understand
✗ It does not reason like humans

3.What is Generative AI?

GenAI is AI that predicts the next word or token using patterns learned from a huge amount of data.

It does **not** think or understand — it only **predicts** what usually comes next.

Experience It Yourself

1. **"100, 200, 300, ___"** → You said **400** instantly
2. **"Twinkle twinkle little ___"** → You said **star**
3. **"Roses are red, violets are ___"** → You said **blue**
   (But violets are actually **purple**)

What happened?

- You answered from **patterns** you have seen many times
- You **did not think**, you just repeated what is common
- **Common pattern (blue)** beat **real fact (purple)**

This is exactly how LLMs work — pure pattern prediction.

4.The Two Modes

Humans Have Two Modes:

Mode 1: Pattern Recognition (Fast)

- Works automatically
- No thinking
  Examples:
- **100, 200, 300 → 400**
- **Sun rises in the ___ → east**

Mode 2: Reasoning (Slow)

You stop and think.

Example:
37, 38, 42, 51, 67, ___

Differences → 1, 4, 9, 16 → next = 25
Answer = 67 + 25 = 92

LLMs Only Have Mode 1

- They do **fast pattern matching**
- They do **NOT** think
- They do **NOT** reason
- They only **predict** the next token
- If training data had similar reasoning examples, they can *simulate* reasoning

5.What is Generation?

When you write something, you choose the **next word** based on what you have learned.

Example:

- “Climate”
- “Climate change”
- “Climate change is”
- “Climate change is a”
- “Climate change is a serious …”

You decide the next word from patterns in your mind.

LLMs do the same:

- They predict **one token at a time**
- They build a full sentence by repeating this prediction
- They use patterns learned during training

LLM (Large Language Model)

Input: Text tokens
Internal Math: Finds relationships between words
Output: Text tokens

Core Idea:
It predicts the **next word**.

6.LMM (Large Multimodal Model)

Input: Text, images, audio, video
Internal Math: Finds relationships between all kinds of data
Output: Text, images, audio, video

Core Idea:
It predicts the **next piece of data** (word, pixel, sound, frame, etc.)

7. Tokens, Not Words

Problem:

Computers do **not** understand words — only **numbers**.

Solution:

Break text into **tokens** (small pieces of text).

Examples:

- **"I like cats"** → ["I", " like", " cats"]
- **"I don't like pineapple"** → ["I", " don", "'t", " like", " pine", "apple"]
- **"I love dosa"** → ["I", " love", " d", "osa"]

8.Why Tokens Matter

Question:

How many letters are in "strawberry"?
You say: **10 letters**

But ChatGPT sometimes gets it wrong.

Why?

Because it sees it as:
["straw", "berry"]
—not individual letters.

So it **cannot count letters** correctly.

This explains why LLMs struggle with:

1. Reversing words letter-by-letter
2. Counting specific letters
3. Exact spelling tasks

Because the model **never sees letters directly**, only **tokens**.

9. Training vs Inference

Training Phase (Learning)

- Model is given **billions of text examples**
- For each example, the next word is **hidden**
- Model tries to **predict** the hidden word
- If wrong → its internal numbers are **adjusted slightly**
- This happens **billions of times**
- **Very slow** → takes **months** and **millions of dollars**
- Done **once** by the company

Inference Phase (Using)

- Model uses the **patterns learned during training**
- No new learning happens
- Just predicts the next token
- **Very fast** → in milliseconds
- Happens **every time you chat**

Key Point:

ChatGPT does NOT learn from you.
It only uses patterns learned months ago.

10. Context Window (Memory Limit)

What is it?

The **amount of text** the AI can "see" at one time:

- Your message
- Previous messages
- Uploaded text

All of this together = **context window**

Common Limits:

- **4K tokens** ≈ 3,000 words (short chat)
- **32K tokens** ≈ 24,000 words (book chapter)
- **200K tokens** ≈ 150,000 words (full novel)

What happens if text is longer than the window?

Example:
Window size = **10 tokens**
Input = **15 tokens**

AI can only see the **last 10**:

Input:
[1][2][3][4][5][6][7][8][9][10][11][12][13][14][15]

AI sees only:
[6][7][8][9][10][11][12][13][14][15]

First 5 are dropped.

This explains:

->Why AI forgets earlier parts of a long chat
->Why sometimes you must repeat information
->Why a new chat = **complete fresh start**

11. Temperature (Randomness Control)

Example Problem:

"The capital of France is ___"

Model's internal probabilities:

- **Paris → 98%**
- **Lyon → 1%**
- **London → 0.01%**

If there is **no randomness**, it will **always** choose Paris.
But then every answer becomes **identical** and **boring**.

That's why we use **temperature**.

Temperature Settings

1. Temperature = 0 (No randomness)

- Always picks the **highest probability**
- Same input → **same output** every time
- Very safe and predictable

Use for:
✓ Math
✓ Facts
✓ Code
✓ Precise answers

2. Temperature = 0.7 (Medium)

- Mostly picks high-probability words

- Sometimes chooses lower-probability ones
- Balanced, natural answers

Use for:
✓ Normal chat
✓ Explanations
✓ Essays
✓ Everyday writing

3. Temperature = 1.5 (High randomness)

- Very unpredictable
- More creative
- Sometimes nonsense

Use for:
✓ Creative writing
✓ Brainstorming
✓ Ideas

12.Common Myths About LLMs

Myth 1: LLMs search the internet

Truth: They do **not** connect to the internet.
They only use patterns from training data (learned months ago).

Myth 2: LLMs think or understand

Truth: They do **not** think.
They simply **predict the next token** using math.

Myth 3: LLMs calculate math

Truth: They **predict digits**, not calculate.

- **2 + 2** is correct because it's common.
- **8437 × 6829** is often wrong because it tries to **guess** digits.

Myth 4: LLMs remember forever

Truth: Memory is only inside the **context window**.
New chat = everything forgotten.

Myth 5: LLMs learn from your corrections

Truth: Corrections help only in the **current chat**.
They do **not** update the model's knowledge.

Myth 6: LLM-generated code always works

Truth: Code can have bugs.
Always test and verify.

13.Who Uses LLMs? (Real Problems Solved)

1. GitHub Copilot

Problem: Developers waste time on repetitive code
Solution: AI suggests code while typing
Impact: Coding becomes **55% faster**

2. Duolingo

Problem: Hard to personalize learning for millions of students
Solution: AI tutor that adapts to each student
Impact: Personal learning for everyone

3. Intercom (Customer Support)

Problem: 70% of support tickets are repeated questions
Solution: AI chatbot answers instantly
Impact: 50% queries solved by AI, available **24/7**

4. Notion AI

Problem: Documentation takes too much time
Solution: AI makes summaries and action items
Impact: Saves **2–3 hours per week**

5. Khan Academy (Khanmigo)

Problem: Teachers cannot give 1-on-1 help to everyone
Solution: AI tutor guides students step-by-step
Impact: Each student gets a **personal tutor**

6. Grammarly

Problem: Bad writing affects work and communication
Solution: AI fixes grammar and improves clarity
Impact: Better writing, saved time

7. Harvey AI (Legal)

Problem: Lawyers spend hours reading long documents
Solution: AI summarizes cases and finds related laws

Impact: 10 hours of work becomes **1 hour**

14.How It All Works Together

When you type:
“Write a function to add two numbers”

1. Tokenization

Your text is broken into **tokens**, then converted to **numbers**.

2. Context Window

Your message fits inside the model’s **temporary memory**.

3. Prediction

The model calculates:
“What is the most likely next token?”

4. Temperature

Decides how **random or strict** the next token choice should be.

5. Generation

The model produces the answer **one token at a time**:

function

function add

function add(

function add(a,

function add(a, b)

function add(a, b) {

…

There is **no thinking** — only **prediction after prediction**.

Key Takeaways

1. **LLMs predict patterns, they do NOT think.**
2. **They work on tokens, not full words or letters.**
3. **Training = past learning, Inference = using now.**

4. **Context window = limited temporary memory.**
5. **Temperature controls randomness.**
6. **They are not perfect — always verify answers.**
7. **They are used everywhere and changing industries.**

Final

“LLMs are powerful pattern predictors, not magic intelligence boxes.”

Knowing this helps you:

- Use them better
- Understand their limits
- Build real products with them
- Stay valuable in the job market