What's LLM? Understanding Large Language Models

Dive into the fascinating world of Large Language Models and discover how AI understands and generates human language with remarkable precision.

What is an LLM?

- Large Language Models (LLMs) are sophisticated AI systems.
- They are trained on massive datasets containing billions of words from books, articles, and web content.
- These powerful models learn to understand and generate human-like language with remarkable fluency and accuracy.
- LLMs currently power intelligent chatbots, real-time translators, content creation tools, and countless other applications.
- They are transforming how we interact with technology.

Popular LLMs: Open Source vs Closed Source

Open Source Models

Meta's LLaMA

Powerful foundation models

Google Gemma

Lightweight, state-of-the-art

DeepSeek

High-performance Chinese models

Qwen

Alibaba's multilingual models

Grok

xAI's conversational model

Perplexity Sonar

Optimized for search

Closed Source Models

OpenAI ChatGPT/GPT-4

Industry-leading AI

Anthropic Claude

Advanced reasoning

Google Gemini

Multimodal AI

Open-source models offer transparency and customization; closed-source provides cutting-edge performance and enterprise support.

LLM Types: From Text to Multimodal AI

Modern AI models are evolving beyond text-only capabilities to understand and generate multiple types of content, pushing the boundaries of what large language models can achieve.

1. Text-Only Models

- Traditional LLMs like GPT-3, Claude, LLaMA
- Process and generate only text
- Excel at language tasks, writing, coding, analysis

2. Vision-Language Models

- Models like GPT-4V, Claude 3, Gemini Pro Vision
- Understand both text and images
- Can analyze photos, diagrams, charts, and visual content

3. Audio Models

- Speech recognition and generation
- Examples: Whisper, ElevenLabs, speech-to-text systems
- Convert between spoken language and text

4. Video Understanding Models

- Analyze video content frame-by-frame
- Understand temporal relationships and motion
- Emerging capability in latest models

5. Multimodal Models

- Handle multiple input/output types simultaneously
- Examples: GPT-4o, Gemini Ultra, Claude 3.5
- Can process text, images, audio, and video together
- Represent the future of AI interaction

LLM vs SLM vs MLM: Understanding Model Sizes

AI language models come in various sizes, each meticulously optimized for distinct use cases and computational environments. Choosing the right size is crucial for balancing performance, efficiency, and resource allocation.

1. LLM (Large Language Models)

- Billions of parameters (e.g., 7B-175B+)
- Highest capability and accuracy
- Examples: GPT-4, Claude, Gemini
- Best for: Complex reasoning, creative tasks, comprehensive knowledge
- Requires: Significant computational resources

2. SLM (Small Language Models)

- Millions to low billions of parameters (e.g., 1B-7B)
- Optimized for efficiency and speed
- Examples: Phi-3, Gemma 2B, TinyLlama
- Best for: Edge devices, mobile apps, specific tasks
- Requires: Minimal resources, can run locally

3. MLM (Medium Language Models)

- Mid-range parameters (e.g., 7B-30B)
- Balance between capability and efficiency
- Examples: LLaMA 2 13B, Mistral 7B
- Best for: General-purpose tasks with resource constraints
- Requires: Moderate computational resources

Ultimately, the selection of an LLM, SLM, or MLM depends on specific application requirements, available budget, and desired performance characteristics.

How Do LLMs Work? The Transformer Architecture

Neural Networks

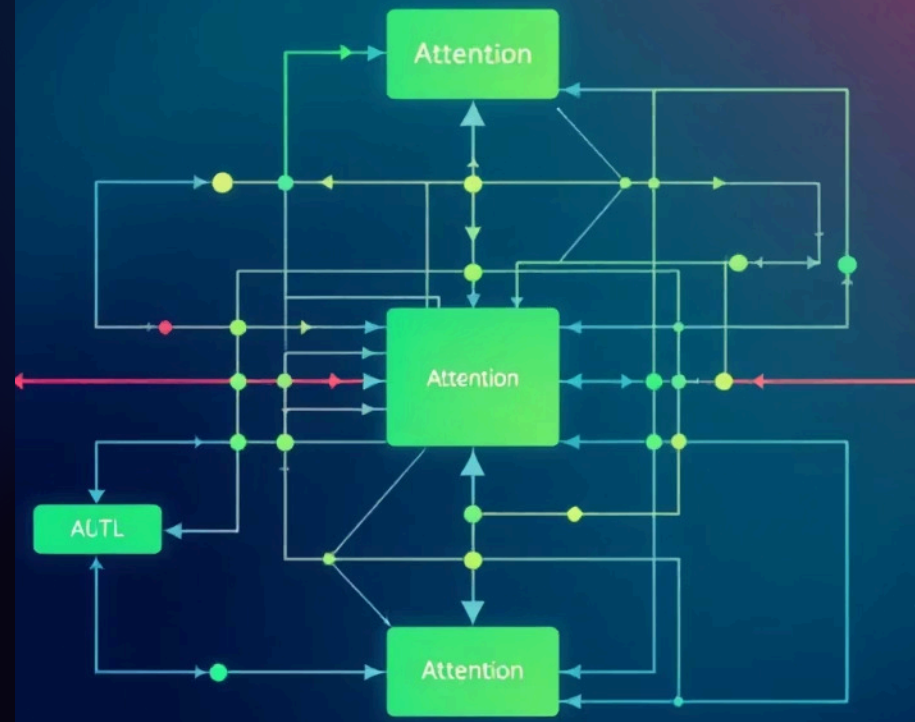
Built on advanced transformer architectures with billions of interconnected parameters that process language patterns

Attention Mechanisms

Sophisticated systems that focus on relevant parts of input text, understanding context and relationships between words

Parallel Processing

Process entire sequences simultaneously rather than word-by-word, enabling lightning-fast, context-aware language understanding



Tokens: The Building Blocks of Language Models

- **Text is systematically broken down into smaller units called tokens—these can be complete words, parts of words, or even individual characters.**
- **LLMs use these tokens as the fundamental building blocks for understanding and generating language.**
- **The model predicts the next most likely token based on all previous tokens in the sequence, creating coherent and contextually appropriate text through this iterative process.**

Example: "The quick brown fox" gets processed as individual tokens, with the model predicting "jumps" as the most probable next word based on learned patterns.

Token Calculation Examples:

Simple Words

"Hello world" = 2 tokens

Each common word typically equals one token

Compound Words

"ChatGPT" = 2 tokens ("Chat" + "GPT")

Uncommon or compound words get split into multiple tokens

Numbers

"2024" = 1 token

"3.14159" = 3-4 tokens

Numbers can be single or multiple tokens depending on format

Punctuation

"Hello, world!" = 4 tokens

Punctuation marks often count as separate tokens

Long Text

A typical sentence = 15-20 tokens

1 page of text ≈ 500-750 tokens

1,000 words ≈ 1,300-1,500 tokens

Context Window: Understanding What the Model Knows

The input you provide and what the model currently knows at any given moment.

Defining the Context Window

The **context window** refers to the amount of text (tokens) a Large Language Model (LLM) can "see" and process at any given moment.

Processing Capacity

This window dictates the model's real-time awareness of the ongoing interaction, influencing its ability to generate coherent and relevant responses.

Example: 8k Tokens

If an LLM has an **8,000 token context window**, it can only process approximately 8,000 tokens of conversation history, instructions, or documents at a time.

Context Window: How Much Can an LLM Remember?

The context window represents the maximum number of tokens an LLM can actively consider and remember during any single interaction or task.

128K

Modern Models

Advanced LLMs like GPT-4 Turbo can handle massive context windows equivalent to hundreds of pages of text

1M-2M

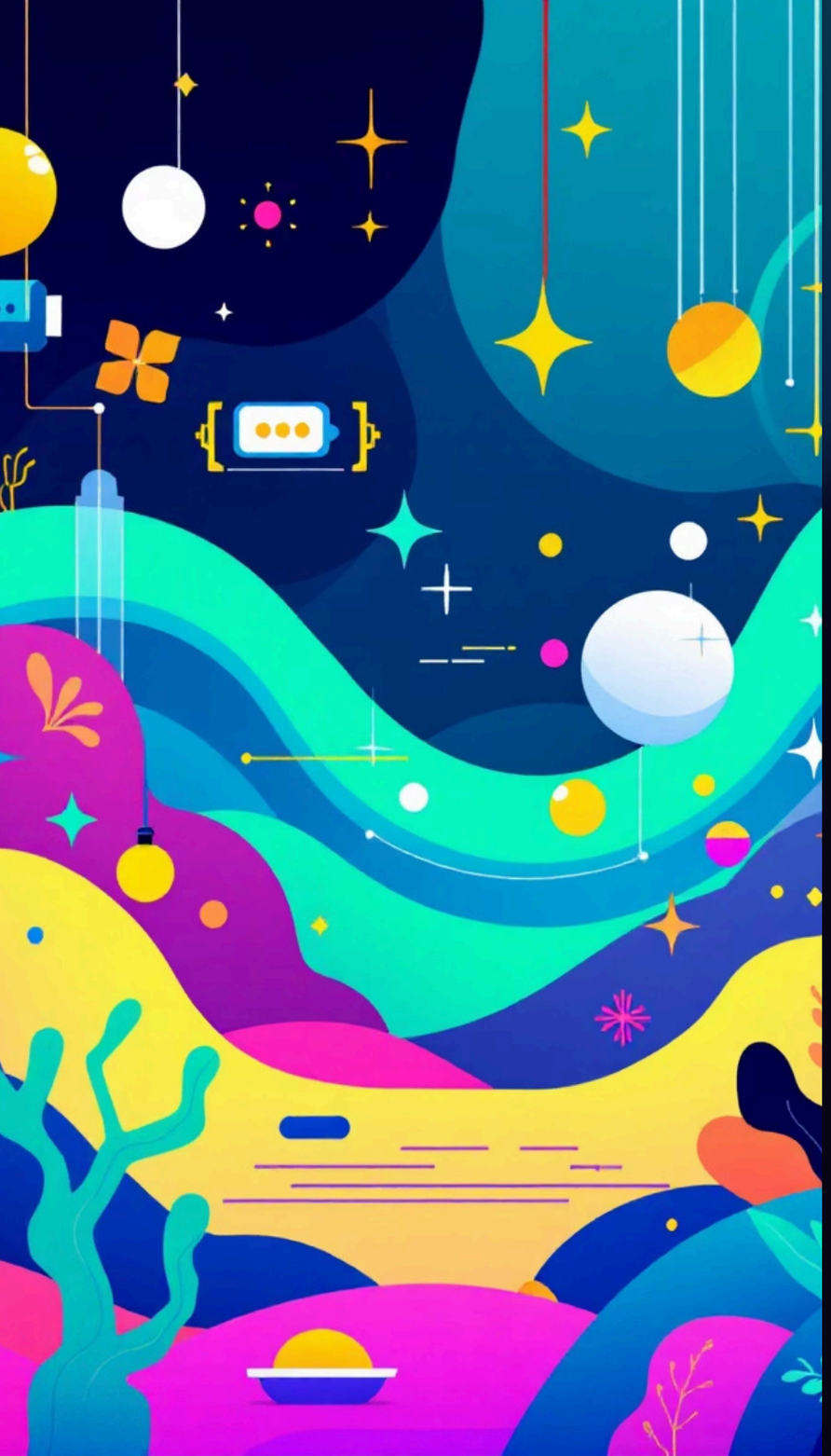
Cutting Edge

Models like Gemini 3 Pro boast 1M token windows, with experimental models pushing to 2M for unprecedented data processing.

200K+

Extended Context

Models like Claude 4.5 Sonnet offer 200K+ token windows for comprehensive document analysis, enhancing understanding and coherence.



Prompt Engineering: Guiding the Model's Output

Prompt engineering is the art and science of crafting precise inputs to elicit the most accurate, relevant, and useful responses from language models.

01

Zero-Shot Prompting

Asking the model to perform tasks without providing any examples, relying solely on its pre-trained knowledge

02

Few-Shot Prompting

Including 2-5 examples in your prompt to demonstrate the desired format or approach for the task

03

Chain-of-Thought

Encouraging the model to show its reasoning process step-by-step for complex problem-solving tasks

Hyperparameters: The Model's Settings

Hyperparameters are crucial configuration settings that control how an LLM learns during training and generates text during inference. These parameters are set before training begins and significantly influence both the model's learning capability and output characteristics.

Learning Rate

Controls how quickly the model updates its weights during training. Typical values: 1e-5 to 1e-3.

Batch Size

Determines how many training examples are processed simultaneously before updating model weights.

Temperature

Controls randomness in text generation. Lower values (0.1-0.5) produce focused, deterministic outputs; higher values (0.7-1.5) increase creativity and diversity.

Top-p (Nucleus Sampling)

Dynamically selects from the smallest set of tokens whose cumulative probability exceeds p. Typically 0.9-0.95.

Max Tokens

Sets the maximum length of generated output.

Layer Count & Model Size

Defines the depth and scale of the neural network architecture.

Fine-tuning these parameters allows researchers and developers to optimize models for specific tasks, balancing factors like accuracy, creativity, computational efficiency, and output quality for different applications.

Model Weights: The Knowledge Inside

175 Billion

Parameters in GPT-5, each storing learned language patterns

Neural Connections

Weights are numerical values that represent the strength of connections between artificial neurons, learned through exposure to vast amounts of text data

Encoded Knowledge

Billions of weights capture intricate patterns including grammar rules, factual information, reasoning abilities, and cultural understanding

Language Processing

These weights determine precisely how the model transforms input tokens into meaningful, contextually appropriate, and linguistically correct output

Embeddings: Representing Words as Vectors

Embeddings are mathematical representations that convert words and tokens into multi-dimensional vectors—essentially, lists of numbers that capture the meaning and relationships between different pieces of language.

Words & Tokens

Raw text input gets processed into individual linguistic units

Vector Conversion

Each token becomes a high-dimensional numerical vector capturing semantic meaning

Relationship Mapping

Similar words cluster together in vector space, enabling understanding of synonyms and context

Real-world magic: Embeddings allow models to understand that "king" - "man" + "woman" ≈ "queen" through mathematical relationships in vector space!

LLM Access & Deployment: Cloud vs Local

Large Language Models (LLMs) offer various access and deployment options, each presenting distinct advantages and trade-offs in terms of performance, cost, privacy, and control.

Cloud-Based LLMs

Access via API (OpenAI, Anthropic, Google)

Advantages:

- No setup required
- Always up-to-date
- Scalable
- Access to largest models

Disadvantages:

- Requires internet connection
- Ongoing costs per token/request
- Data privacy concerns
- Vendor dependency

Best for:

- Production applications
- Teams without ML infrastructure
- Accessing cutting-edge models

Examples: ChatGPT API, Claude API, Gemini API

Local LLMs

Run on your own hardware (laptop, server, edge device)

Advantages:

- Complete data privacy
- No internet required
- One-time cost
- Full control and customization

Disadvantages:

- Requires powerful hardware (GPU)
- Limited to smaller models
- Manual updates
- Technical expertise needed

Best for:

- Sensitive data
- Offline environments
- Cost optimization for high volume
- Experimentation

Examples: Ollama, LM Studio, llama.cpp, GPT4All

Hybrid approaches often combine the strengths of both cloud and local deployments, leveraging the best options for specific needs within an organization.

Key LLM Terminologies Recap

1	2	3
Foundation Concepts	**Control Mechanisms**	**Internal Representations**
LLMs, Transformers, Tokens - The core building blocks that enable AI to process and generate human language	**Context Windows, Prompts, Hyperparameters** - Tools for managing and optimising model behavior and performance	**Weights, Embeddings** - The mathematical foundations that store knowledge and enable semantic understanding

Understanding these fundamental concepts empowers you to work more effectively with AI language models and appreciate the remarkable engineering behind modern conversational AI systems.

Q & A

Thank You!!