Generative AI:
Transformer & LLM

Learning Outcomes

Upon completion of this session, the learners should be able to:

- Explain the fundamentals of Transformer and LLM.
- Describe Large Language Model (LLM) applications and Generative Foundation Model.

What is a Transformer?

- Transformers were introduced in the **“Attention Is All You Need”** paper (2017) to handle sequence-to-sequence tasks (e.g. translation) without relying on recurrence (RNNs) or convolution.
- Key motivations / advantages over RNNs:
  - **Parallelization:** Transformers process entire sequences simultaneously (rather than step by step), which better utilizes hardware like GPUs.
  - **Long-range dependencies:** Self-attention enables each token to attend to any other token in the sequence, regardless of distance, mitigating issues like vanishing gradients.

Transformer Architecture Overview

- A Transformer typically has two main parts: Encoder and Decoder (especially in translation or generation tasks).
- The encoder ingests the input sequence and produces contextualized vector representations.
- The decoder uses those representations (plus previously generated output tokens) to produce an output sequence (autoregressively).
- The original design, both encoder and decoder stacks had 6 layers each, though many modern variants vary this.

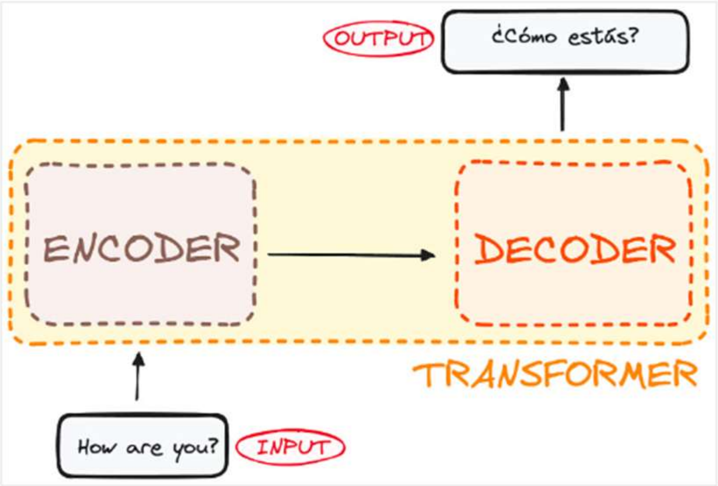
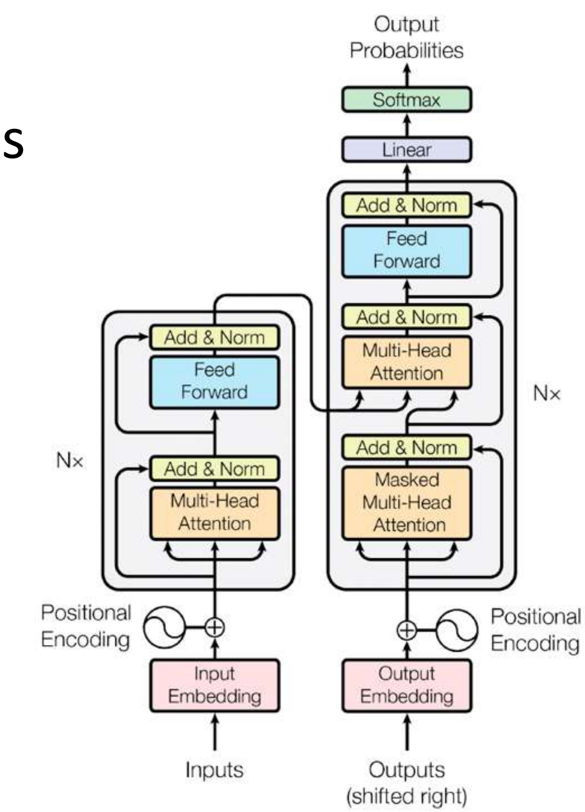


Image source: Datacamp

Why Transformers Made LLMs Possible

- Self-Attention → captures context globally.
- Multi-Head Attention → learns diverse relationships
- Positional Encoding → preserves sequence order.
- Residuals & Norms → stable deep training.
- Parallelism → efficient scaling to billions of parameters.
- Flexible encoder/decoder → adaptable to tasks.
- **Result:** Transformers are the **engine** behind LLMs like GPT, Claude, LLaMA, Gemini.



Feature 1 — Self-Attention

- Allows each token to “attend” to every other token.
- Captures long-range dependencies **efficiently**.
- Unlike RNNs, no matter how far apart words are, they can be directly related.

Example:

- Sentence: *“The book that I borrowed from the library was excellent.”*
- “book” attends to “was excellent” even though separated by many tokens.

Feature 2 — Multi-Head Attention

- Multiple attention "heads" learn different relationships.
- Heads specialize syntax, semantics, entity references, etc.
- This diversity enables richer contextual understanding.

Example:

- One head may link *"Paris → France"*
- Another head may link *"Paris → capital"*
- Combined → LLMs generate factually coherent text.

Feature 3 — Positional Encoding

- Transformers lack recurrence (order info missing).
- Positional encodings inject order into embeddings.
- Ensures LLMs understand word order, not just bag-of-words.

Example:

- “Dog bites man” vs. “Man bites dog”
- Same words, different meaning → position matters!

Feature 4 — Layer Normalization & Residual Connections

- Residuals improve gradient flow (stable training).
- LayerNorm normalizes inputs → faster convergence.
- Critical for scaling LLMs (billions of parameters).

Example:

- Without residuals, very deep LLMs fail to converge.
- With them, GPT-4 can train stably across **96+ layers**.

Feature 5 — Scalability & Parallelism

- Transformer processes sequences **in parallel**, unlike RNNs.
- GPU/TPU friendly → enables training on massive datasets.
- Directly responsible for LLM breakthroughs (e.g., GPT trained on trillions of tokens).

Example:

- Training GPT-3 with 175B params would be **impossible** with RNNs.
- Transformer parallelism made it feasible.

Feature 6 — Encoder vs. Decoder Architectures

- **Encoder-only (BERT):** Great for classification & understanding tasks.
- **Decoder-only (GPT):** Great for text generation.
- **Encoder–Decoder (T5):** Great for translation & seq2seq.
- Flexible architecture → LLM family diversity.

Example:

- **ChatGPT** → decoder-only (causal language modeling).
- **BERT** → encoder-only (masked language modeling).

LLMs Built on Transformers

- Both **LLaMA 3** and **ChatGPT** are **decoder-only Transformer models**.
- They use the same **core principles**:
  - Self-attention
  - Multi-head attention
  - Positional embeddings
  - Residual connections & normalization
- What makes them different is **scale, training data, and fine-tuning**.

Decoder-Only Transformer for LLMs

- Input: tokens (subwords, e.g., “play”, “ing”).
- Tokens → embeddings + positional info.
- **Stack of Transformer decoder layers**:
  - Masked self-attention (no peeking ahead).
  - Feed-forward network.
  - Residual + normalization.
- Output: probability distribution for next token.
- Repeats autoregressively until end of sequence.
- **Example:**
  Prompt: *“Once upon a time”*
  → Model generates tokens step-by-step: *“there”, “was”, “a”, “king”…*

Human-like Characteristics of LLMs

Human-Like Understanding of Context

- LLMs can process **long context windows** (many sentences/paragraphs).
- Capture dependencies across far-apart words → like humans tracking conversation threads.

Example:

- Q: “Alice gave Bob a book. What did he receive?”
  → Model understands “he” = Bob.

Human-like Characteristics of LLMs

Natural Conversational Flow

- Generate text that follows grammar, tone, and flow of human conversation.
- Can adjust style (formal, casual, technical).

Example:

- Prompt: “Explain photosynthesis casually”
  → Output: “Plants kind of eat sunlight and turn it into food.”

Human-like Characteristics of LLMs

In-Context Learning (Few-Shot Reasoning)

- Humans learn from a few examples; LLMs mimic this by adapting to prompts.
- No retraining → quick adaptation like human short-term learning.

Example:
Prompt:
"Translate these:

- cat → chat
- dog → chien
  Now, bird → ?"
  → Model answers: *oiseau*.

Human-like Characteristics of LLMs

Compositional Generalization

- Combine known concepts to form new ideas → similar to human reasoning.
- Enables analogy, metaphor, explanation.

Example:
Q: “What’s a zebra?”
→ Model: “It’s like a horse but with black and white stripes.”

Human-like Characteristics of LLMs

Knowledge Recall & Abstraction

- Stores vast knowledge (like memory).
- Can abstract general principles from data → feels like "understanding."

Example:

Q: "Why do people wear coats in winter?"

→ Model: "Because coats keep body heat in when it's cold."

Human-like Characteristics of LLMs

Flexibility of Expression

- Adjusts tone, style, persona → human-like adaptability.
- Can tell a story, write an essay, or draft code in natural style.

Example:

Same topic explained:

- To a child → “The sun helps plants make food.”
- To a scientist → “Photosynthesis converts light energy into chemical energy in chloroplasts.”

Human-like Characteristics of LLMs

Human-Like but Not Human

- **Strengths:** fluency, memory, reasoning mimicry, adaptability.
- **Limits:**
  - No real understanding or consciousness. (Multiple attention “heads” learn different relationships)
  - Susceptible to **hallucination**.(Using probability to generate new word)
  - Cannot reason about lived experience. (knowledge are from pretrained model)

What is Foundation Model?

- A foundation model is a broad category of large AI models trained on vast datasets, adaptable to many tasks and capable of handling various data types like text, images, and audio.
- A large language model (LLM) is a specific subset of a foundation model, specialized in understanding and generating human language by being primarily trained on massive amounts of text data.
- Therefore, all LLMs are foundation models, but not all foundation models are LLMs; the key difference lies in the scope of data types and the range of tasks they can perform.

What is Foundation Model?

- Broad Category: A large, general-purpose AI model trained on massive datasets across different modalities (text, images, audio, code).
- Multimodal: Can process and work with diverse data types.
- Adaptable: Can be fine-tuned for a wide array of downstream tasks and applications in various industries.
- Example: GPT-4 is a multimodal foundation model that can handle text and images, making it a type of foundation model, but the term also encompasses models for non-language tasks.

Large Language Model (LLM)

- Specific Subset: A type of foundation model that focuses specifically on natural language processing (NLP).
- Text-Based: Trained primarily on vast quantities of text data.
- Specialized Tasks: Excel at text-based tasks like summarizing, answering questions, translating, writing, and generating code.
- Example: ChatGPT is a well-known example of an LLM, which is a specific implementation of a large language model.

General Training Pipeline

- raining a foundation LLM usually has **three main stages**:
- **Pretraining** – learn general language patterns from huge datasets.
- **Fine-tuning** – specialize for a task (translation, coding, Q&A).
- **Alignment** – ensure safe, useful, human-like responses (e.g., RLHF).
- Instruction Tuning - Expose model to instruction–response pairs.

What is a Multimodal LLM?

- A **Multimodal LLM** is a **large language model** that can process and generate information from **multiple types of data (modalities)**, not just text.
- Modalities = **Text, Images, Audio, Video, Code, Sensor data**.
- **Example:** GPT-4V (Vision) can take **text + image** as input.

Example Tasks Multimodal LLMs Can Do

- Image Captioning → “Describe this picture.”
- Visual Q&A → “What is written on this sign?”
- Document Understanding → read scanned PDFs with images + text.
- Audio Transcription & Analysis → “Summarize this meeting recording.”
- Video Understanding → “What happens in this clip?”

Why Multimodality Matters

- Human intelligence is **multimodal**: we use vision, hearing, language together.
- Multimodal LLMs bring AI closer to **human-like perception & reasoning**.
- Expands applications: healthcare imaging, robotics, education, accessibility.

References

- *Attention is All You Need* (Vaswani et al., 2017) → Paper
- Illustrated Transformer (Jay Alammar) → Blog
- 3Blue1Brown video on Transformers → YouTube
- Annotated Transformer (PyTorch implementation) → GitHub
- Hugging Face Course → Course
- nanoGPT (Andrej Karpathy) → GitHub
- Transformers from Scratch (Peter Bloem) → Blog
- Illustrated Guide to LLMs (Lil'Log / Lilian Weng) → Post
- Stanford CS324: Large Language Models → Lecture Notes