Transformers Architecture: From Attention to State-of-the-Art AI

Inspired by Andrej Karpathy & Stanford CS25

Chapter 1: The Revolution Begins

Why Transformers Changed Everything

The Bottleneck of Sequential Models

Long-Range Dependencies

RNNs and LSTMs struggled to maintain context across long sequences, causing information loss

Sequential Processing

Training was inefficient due to inability to parallelize computations effectively

Scalability Crisis

Need emerged for parallelizable, scalable architectures that could handle massive datasets

The Breakthrough: Attention Mechanism

Soft Search Revolution

Attention was introduced as a "soft search" over input tokens, fundamentally changing how models process sequences.

- Enables dynamic focus on relevant parts of input
- Models can weigh importance of different tokens
- Laid foundation for landmark 2017 paper

> "Attention Is All You Need" transformed deep learning forever

Evolution of Neural Architectures

1. **2014: RNNs**

   Sequential processing, limited memory

2. **2015: LSTMs**

   Gated cells, improved long-term memory

3. **2016: Attention**

   Dynamic focus mechanism introduced

4. **2017: Transformers**

   Pure attention architecture emerges

Chapter 2: Anatomy of a Transformer

The Building Blocks of Modern AI

Core Idea: Self-Attention

Token Interaction

Each token attends to every other token in the sequence

Context Building

Produces rich context-aware embeddings for each position

Global Dependencies

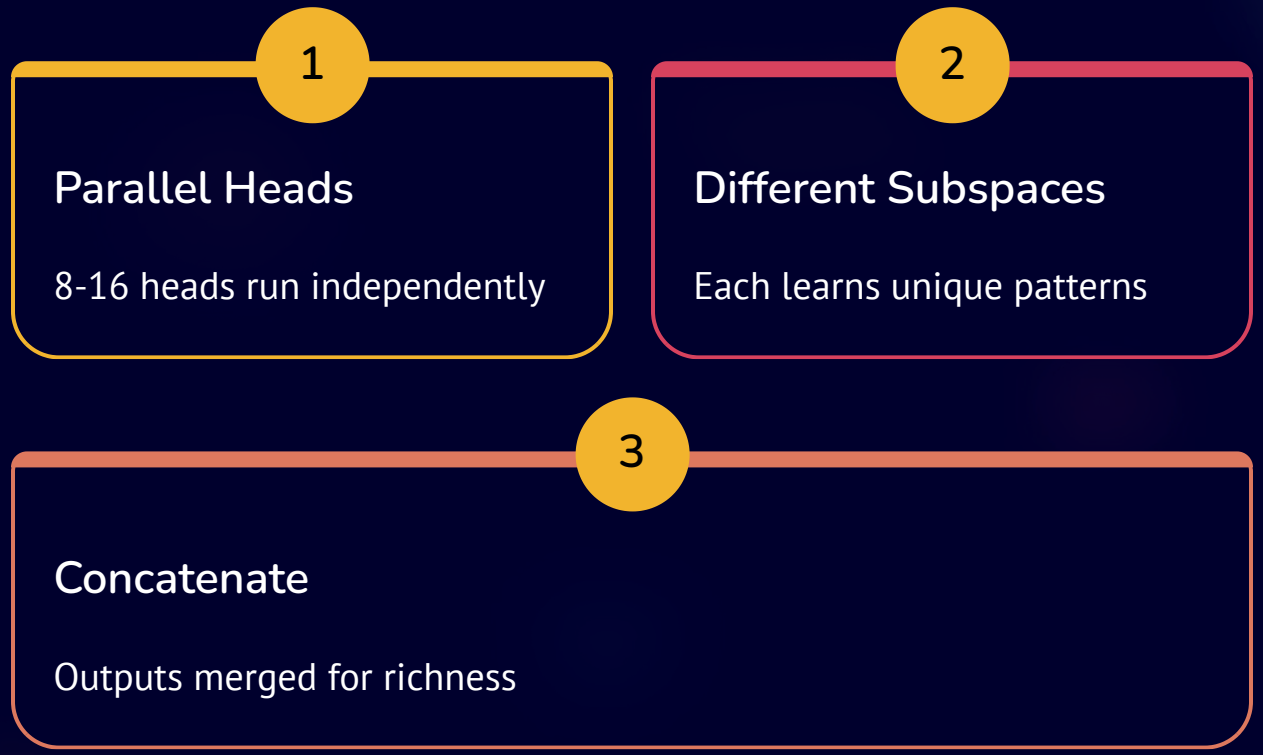
Captures long-range relationships efficiently in parallel

Multi-Head Attention Explained

Parallel Learning

Multiple attention "heads" operate simultaneously, each learning to focus on different aspects of the data.

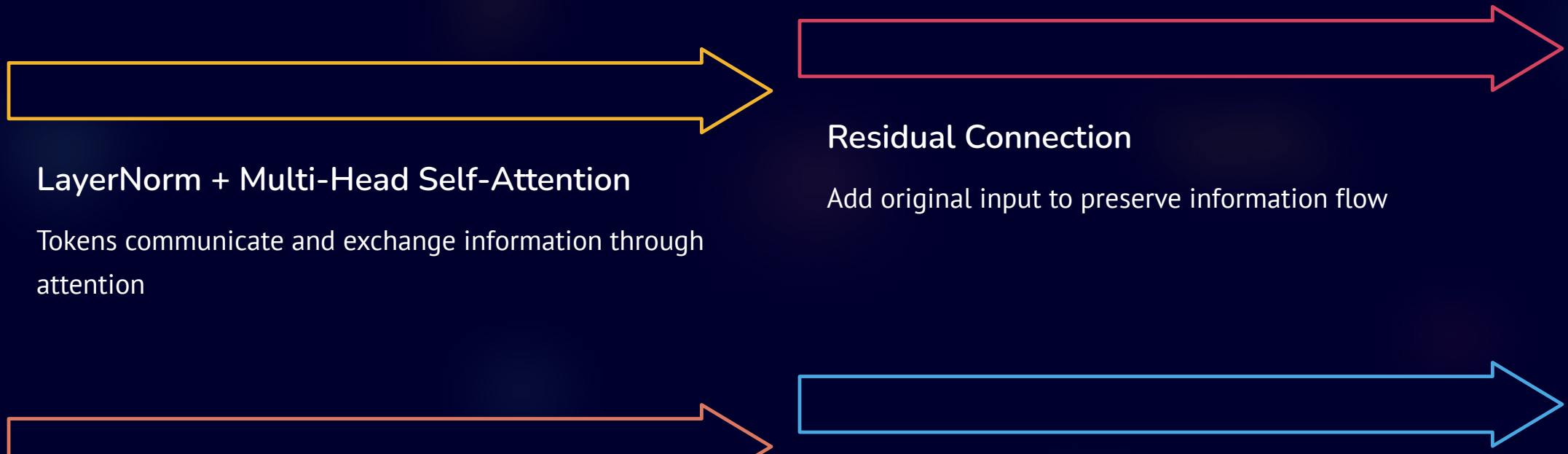
This parallel architecture allows the model to attend to information from different representation subspaces at different positions.



Visualizing Multi-Head Attention

Each attention head learns to focus on different relationships between tokens. Some heads might capture syntactic dependencies, while others focus on semantic relationships or positional patterns. The diversity of attention patterns enables rich, nuanced understanding.

Transformer Block: Communication + Computation



LayerNorm + Multi-Head Self-Attention

Tokens communicate and exchange information through attention

Residual Connection

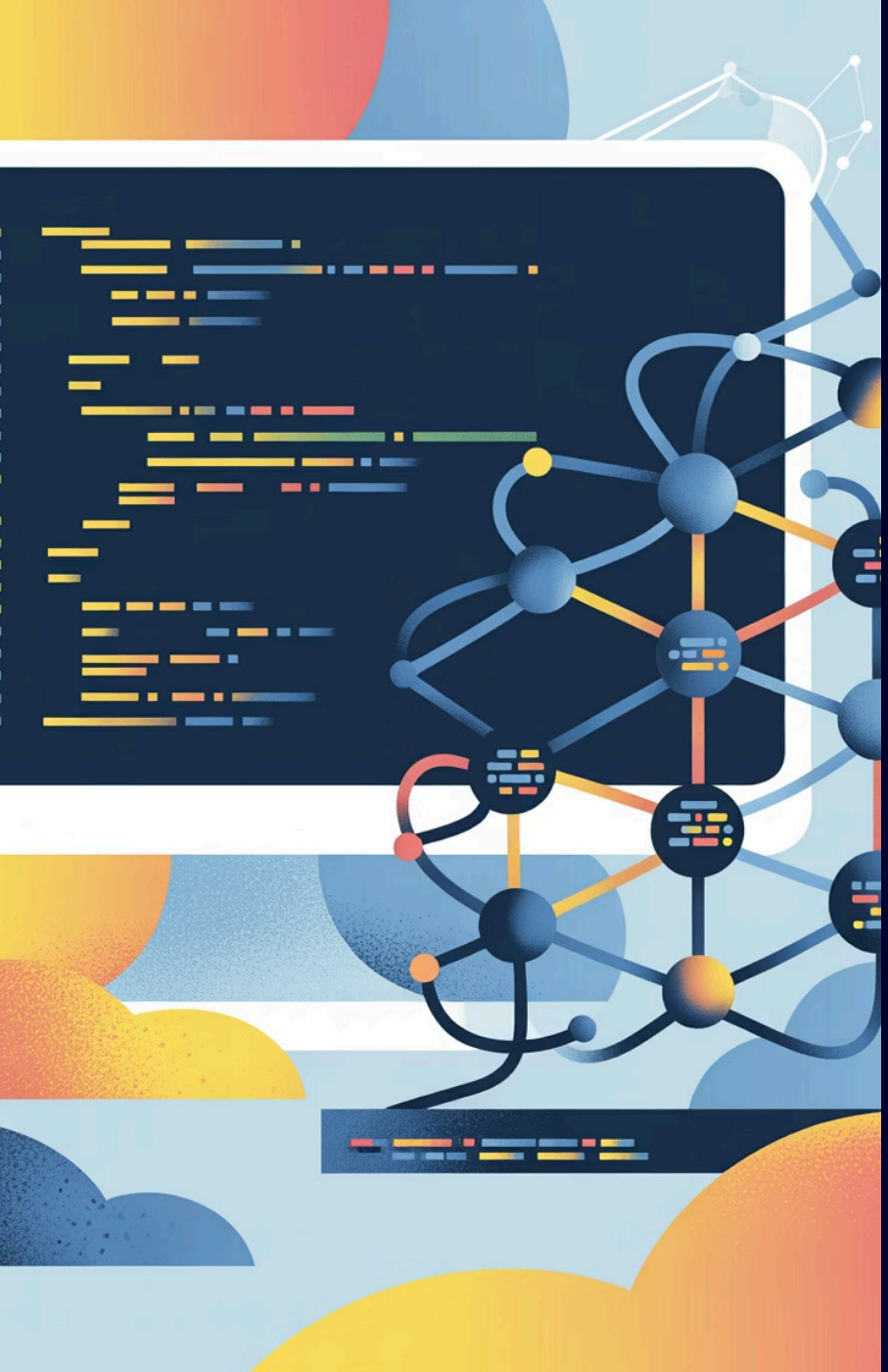
Add original input to preserve information flow

LayerNorm + FeedForward Network

Each token processes information independently

Residual Connection

Stack multiple blocks to build depth and capacity



Karpathy's GPTLanguageModel Class Breakdown

01

Token & Position Embeddings

Combined to create rich input representations with positional awareness

02

Sequential Transformer Blocks

Stack of identical layers processes embeddings through attention and feedforward

03

LayerNorm + Linear Output

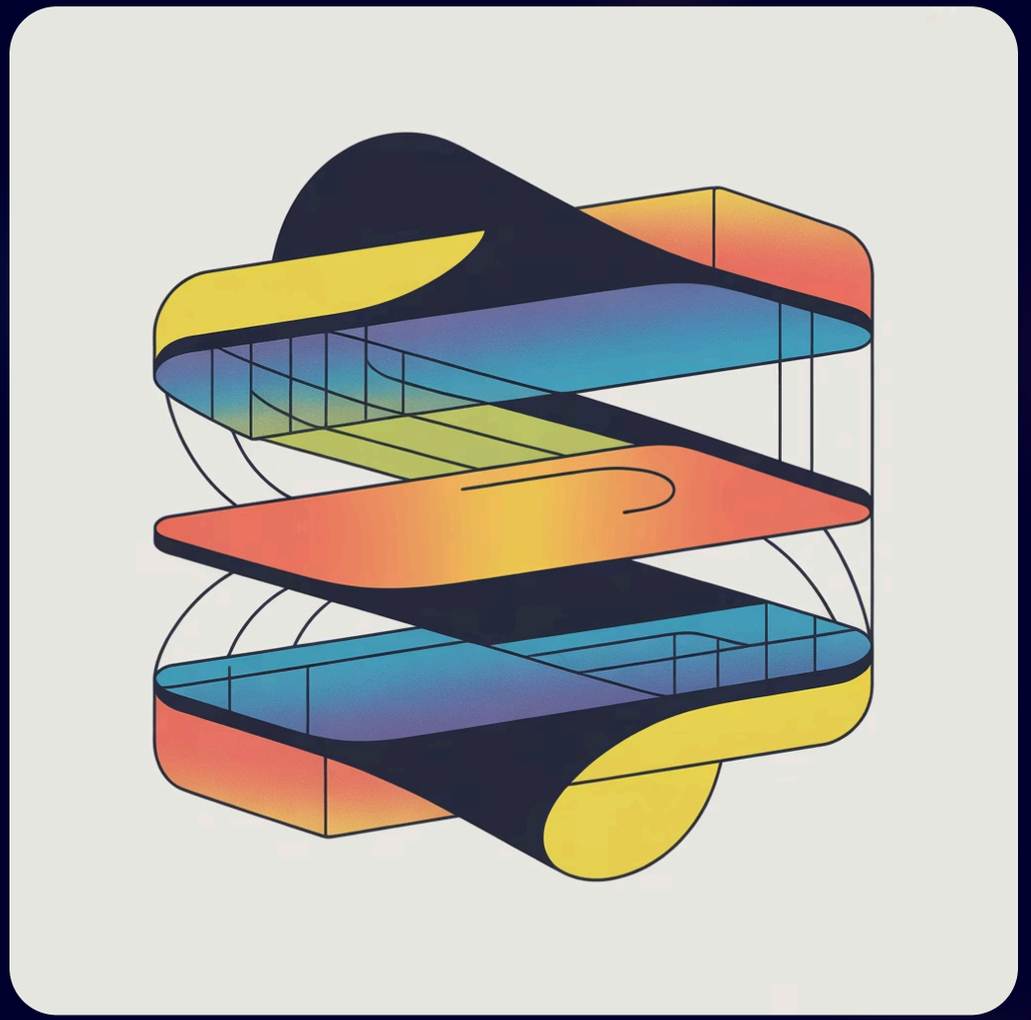
Final normalization and projection produce next-token probability distributions

GPTLanguageModel Forward Pass

```
def forward(self, idx, targets=None):
    B, T = idx.shape

    # Embeddings: token + position
    tok_emb = self.token_embedding_table(idx)
    pos_emb = self.position_embedding_table(torch.arange(T))
    x = tok_emb + pos_emb

    # Transformer blocks
    x = self.blocks(x)
    x = self.ln_f(x)

    # Output logits
    logits = self.lm_head(x)

    return logits
```

This elegant implementation combines embeddings, processes through stacked transformer blocks, and projects to vocabulary space for next-token prediction.

FeedForward Network Inside Transformer Blocks



Non-Linear Transformation Power

1

Linear Expansion

Projects embeddings to 4x dimension

2

ReLU Activation

Introduces non-linearity

3

Linear Projection

Maps back to original dimension

This expansion-contraction pattern allows each token to be processed through a high-dimensional space, adding computational depth.

Positional Encoding: Injecting Order

The Problem

Transformers lack recurrence or convolution, making them position-agnostic by default

The Solution

Positional information is added explicitly through learned embeddings or sinusoidal functions

The Result

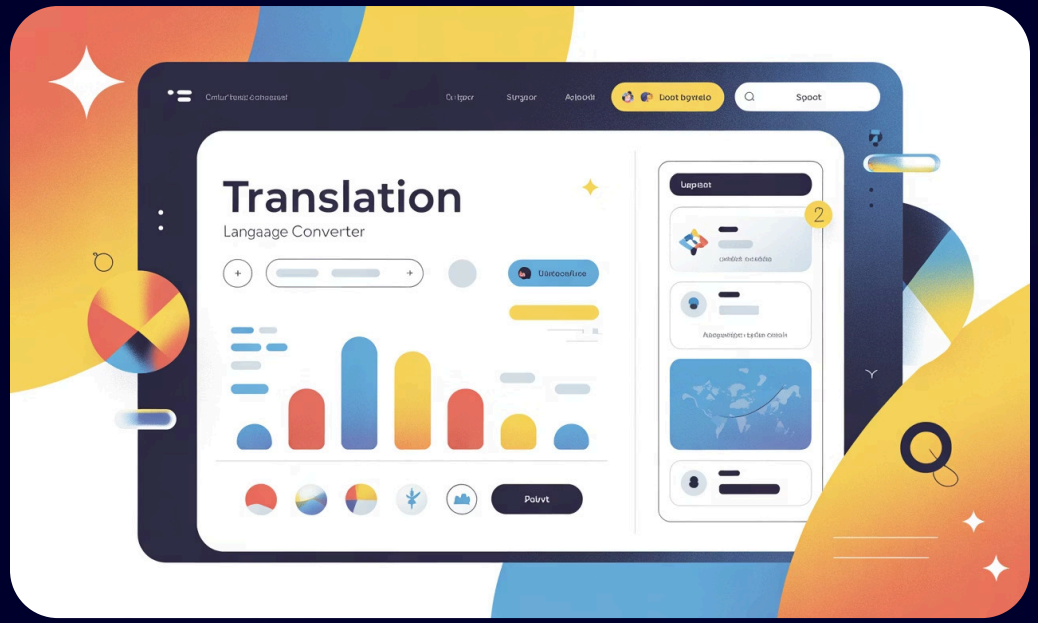
Models can distinguish token positions, essential for understanding sequence structure

Chapter 3: Transformer Variants & Applications

From Language to Vision and Beyond

Encoder-Decoder vs Decoder-Only Models

Encoder-Decoder

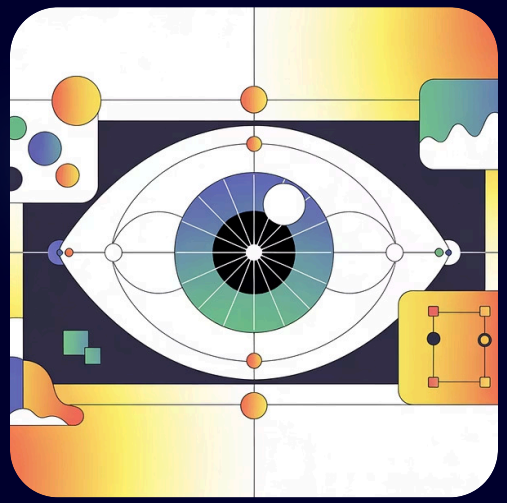


- Original Transformer design (2017)
- Best for sequence-to-sequence tasks
- Machine translation, summarization
- Cross-attention connects both sides

Decoder-Only

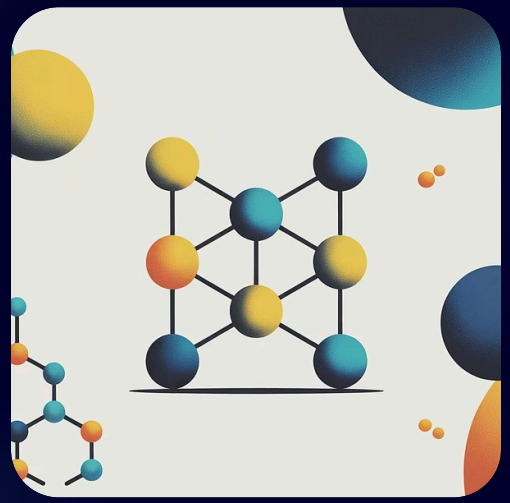
- GPT series architecture
- Optimized for autoregressive generation
- Text completion, code generation
- Simpler, more scalable design

Beyond NLP: Transformers Everywhere



Vision Transformers (ViT)

Revolutionizing image recognition by treating image patches as tokens, outperforming CNNs



AlphaFold2

Breakthrough in protein structure prediction using attention mechanisms for biological sequences

RL & Multimodal

Decision transformers and models combining text, vision, and audio in unified architectures

Challenges & Frontiers

Scaling Laws

Understanding emergent capabilities as models grow larger. What happens at 100B, 1T parameters?

Context Length

Memory and compute constraints limit context windows. New architectures needed for longer sequences.

Alignment & Safety

RLHF and constitutional AI ensure models align with human values, creativity, and ethical guidelines.

Stanford CS25: The Community Behind the Innovation

The transformer revolution was built by a remarkable community of researchers, engineers, and visionaries. From the original "Attention Is All You Need" team at Google to Andrej Karpathy's educational contributions and the broader Stanford AI community, collaborative innovation drives progress.

Vaswani et al.

Original Transformer paper authors

Karpathy

Making AI accessible through teaching

OpenAI & DeepMind

Pushing boundaries of scale

The Transformer Legacy & Your Next Step

Transformers revolutionized AI

From language to vision, biology to robotics, attention-based architectures transformed research and applications across every domain

Understanding = Innovation

Mastering transformer architecture unlocks the ability to build, improve, and innovate on the next generation of intelligent systems

Continue Learning

Dive deep into Andrej Karpathy's masterclass, Stanford CS25 lectures, and implement your own transformer from scratch

> Let's build the future of AI together—one attention head at a time.