GPT Architecture Cheat Sheet

High-Level Flow

Input Text -> Tokenizer -> Embedding -> Positional Encoding -> Decoder Blocks (stacked) -> Linear Layer -> Softmax -> Next Token

Components Breakdown

1. Tokenizer

Converts words into token IDs.

Example:

"Hello world" -> [15496, 995]

2. Embedding Layer

Converts token IDs to dense vectors:

[15496, 995] -> [[0.23, -0.56, ...], [...]]

3. Positional Encoding

Adds info about token position (since transformers don't process sequences natively).

4. Transformer Decoder Blocks (Stacked N Times)

Each block has:

a) Masked Multi-Head Self-Attention

- Prevents looking ahead (causal).
- Token at position t can only see positions <= t.

b) Add & Layer Norm

- Stabilizes training.

c) Feed Forward Neural Network (FFN)

- Two linear layers with ReLU or GELU.

d) Add & Layer Norm again

5. Output Layer

Linear layer + softmax -> predicts next token.

Training Objective

Predict the next token given all previous tokens.

Loss Function: Cross Entropy between predicted token and actual next token.

Special Things GPT Does

Feature: Decoder-only -> Faster generation (no encoder)

Feature: Causal Masking -> Keeps prediction autoregressive

Feature: Massive scale -> Trained on billions of tokens

Feature: Pretraining -> Learns language from scratch

Summary for Interviews / Notes

GPT is a decoder-only Transformer that generates text by predicting the next token in a sequence using masked self-attention and feedforward layers. It learns contextual relationships without recurrence.