LLM Interview Questions & Answers

# Basics of LLMs

## Q: What is a Large Language Model (LLM)?

A: A Large Language Model (LLM) is an AI model trained on massive amounts of text data to understand and generate human-like language. It predicts the next word in a sequence, enabling it to answer questions, write code, summarize text, translate languages, and more.

## Q: How does an LLM work in simple terms?

A: It works like auto-complete. Given some text (prompt), the model predicts the most likely next word repeatedly until it forms a meaningful response.

## Q: Difference between an LLM and traditional NLP models?

A: Traditional NLP models are task-specific (e.g., sentiment analysis), while LLMs are general-purpose, trained on diverse data, and can handle multiple tasks without retraining.

# Architecture & Training

## Q: What is a Transformer? Why is it important in LLMs?

A: Transformers are the backbone of LLMs. They use self-attention to figure out which words in a sentence are most relevant to each other.

## Q: What does 'attention' mean in Transformers?

A: Attention assigns weights to words in the input to decide which ones are most important for predicting the next word.

## Q: What is pre-training and fine-tuning in LLMs?

A: Pre-training is learning from massive general data. Fine-tuning is additional training on domain-specific or task-specific data.

# Core Concepts

## Q: What is a 'token' in LLMs?

A: A token is a chunk of text (word, part of a word, or character) that the model processes.

## Q: What is a prompt?

A: A prompt is the input given to an LLM that guides its output. Prompt engineering is writing effective prompts to get better results.

## Q: What is hallucination in LLMs?

A: When the model generates confident but false or made-up answers.

# Applications

## Q: What are some real-world applications of LLMs?

A: Chatbots, code generation, document summarization, translation, search engines, content generation.

## Q: How can LLMs help software engineers specifically?

A: Writing boilerplate code, debugging, explaining algorithms, assisting in API documentation, automating test case generation.

# Challenges & Limitations

## Q: What are the main limitations of LLMs?

A: Hallucination, bias, high computational cost, lack of real-time knowledge, prediction-based rather than true understanding.

## Q: How do you reduce bias in LLMs?

A: Using diverse training data, RLHF, content filters, and continuous fine-tuning.

# Advanced Concepts

## Q: What is self-attention vs cross-attention?

A: Self-attention relates words within the same sequence, while cross-attention is when the decoder attends to encoder outputs.

## Q: What is positional encoding?

A: It injects information about word positions into embeddings so Transformers know word order.

## Q: Encoder-only vs Decoder-only vs Encoder-Decoder models?

A: Encoder-only (BERT) for understanding, Decoder-only (GPT) for generation, Encoder-decoder (T5) for translation/summarization.

## Q: What is RLHF?

A: Reinforcement Learning with Human Feedback, where humans rate model outputs to align responses with expectations.

## Q: What is quantization?

A: Reducing precision of weights to make models smaller and faster.

## Q: What is distillation?

A: Knowledge transfer from a large 'teacher' model to a smaller 'student' model.

## Q: What are retrieval-augmented LLMs (RAG)?

A: LLMs combined with external knowledge sources to reduce hallucination.

## Q: What is a Mixture of Experts (MoE)?

A: A model that activates only a subset of experts per input, improving efficiency.

## Q: What are multi-modal LLMs?

A: Models that process text plus other modalities like images or speech.

## Q: What is chain-of-thought prompting?

A: Encouraging the model to reason step by step before answering.