Top 50 Large Language Model (LLM) Interview Questions

Hao Hoang - Follow me on LinkedIn for AI insights!

May 2025

Explore the key concepts, techniques, and challenges of Large Language Models (LLMs) with this comprehensive guide, crafted for AI enthusiasts and professionals preparing for interviews.

Introduction

Large Language Models (LLMs) are revolutionizing artificial intelligence, enabling applications from chatbots to automated content creation. This document compiles 50 essential interview questions, carefully curated to deepen your understanding of LLMs. Each question is paired with a detailed answer, blending technical insights with practical examples. Share this knowledge with your network to spark meaningful discussions in the AI community!

1 Question 1: What does tokenization entail, and why is it critical for LLMs?

Tokenization involves breaking down text into smaller units, or tokens, such as words, subwords, or characters. For example, "artificial" might be split into "art," "ific," and "ial." This process is vital because LLMs process numerical representations of tokens, not raw text. Tokenization enables models to handle diverse languages, manage rare or unknown words, and optimize vocabulary size, enhancing computational efficiency and model performance.

2 Question 2: How does the attention mechanism function in transformer models?

The attention mechanism allows LLMs to weigh the importance of different tokens in a sequence when generating or interpreting text. It computes similarity scores between query, key, and value vectors, using operations like dot products, to focus on relevant tokens. For instance, in "The cat chased the mouse," attention helps the model link "mouse" to "chased." This mechanism improves context understanding, making transformers highly effective for NLP tasks.

Top 50 大型语言模型 (LLM) 面试问题

Hao Hoang - 在领英上关注我，获取 AI 见解！

2025 年 5 月

通过这份全面的指南，探索大型语言模型 (LLM) 的关键概念、技术和挑战，专为 AI 爱好者和准备参加

面试的人士而

制作。

引言

大型语言模型（ LLMs ）正在改变人工智能，使应用从聊天机器人到自动内容创作成为可能。本文汇编了 50 个关键的面试问题，经过精心策划，旨在加深您对 LLMs 的理解。每个问题都配有详细的答案，结合了技术见解和实际示例。与您的网络分享这些知识，以在人工智能社区中引发有意义的讨论！

1 Question 1: What does tokenization entail, and why is it critical for LLMs?

分词涉及将文本分解为更小的单元，即标记，如单词、子词或字符。例如，“ artificial” 可能会被拆分为 “art” 、 “ific” 和 “ial” 。这个过程至关重要，因为 LLMs 处理的是标记的数值表示，而不是原始文本。分词使模型能够处理多种语言、管理罕见或未知单词，并优化词汇量大小，从而提高计算效率和模型性能。

2 问题 2：注意力机制在 Transformer 模型中是如何工作的?

注意力机制使 LLMs 能够在生成或解释文本时权衡序列中不同标记的重要性。它通过计算查询、键和值向量之间的相似度分数（使用点积等运算）来专注于相关的标记。例如，在 “ 猫追老鼠 ” 中，注意力帮助模型将 “ 老鼠 ” 与 “ 追 ” 联系起来。这种机制提高了上下文理解能力，使 Transformer 在 NLP 任务中非常有效。

3 Question 3: What is the context window in LLMs, and why does it matter?

The context window refers to the number of tokens an LLM can process at once, defining its "memory" for understanding or generating text. A larger window, like 32,000 tokens, allows the model to consider more context, improving coherence in tasks like summarization. However, it increases computational costs. Balancing window size with efficiency is crucial for practical LLM deployment.

4 Question 4: What distinguishes LoRA from QLoRA in fine-tuning LLMs?

LoRA (Low-Rank Adaptation) is a fine-tuning method that adds low-rank matrices to a models layers, enabling efficient adaptation with minimal memory overhead. QLoRA extends this by applying quantization (e.g., 4-bit precision) to further reduce memory usage while maintaining accuracy. For example, QLoRA can fine-tune a 70B-parameter model on a single GPU, making it ideal for resource-constrained environments.

5 Question 5: How does beam search improve text generation compared to greedy decoding?

Beam search explores multiple word sequences during text generation, keeping the top k candidates (beams) at each step, unlike greedy decoding, which selects only the most probable word. This approach, with $k = 5$, for instance, ensures more coherent outputs by balancing probability and diversity, especially in tasks like machine translation or dialogue generation.

6 Question 6: What role does temperature play in controlling LLM output?

Temperature is a hyperparameter that adjusts the randomness of token selection in text generation. A low temperature (e.g., 0.3) favors high-probability tokens, producing predictable outputs. A high temperature (e.g., 1.5) increases diversity by flattening the probability distribution. Setting temperature to 0.8 often balances creativity and coherence for tasks like storytelling.

7 Question 7: What is masked language modeling, and how does it aid pretraining?

Masked language modeling (MLM) involves hiding random tokens in a sequence and training the model to predict them based on context. Used in models like BERT, MLM fosters bidirectional understanding of language, enabling the model to grasp semantic

3 Question 3: What is the context window in LLMs, and why does it matter?

上下文窗口指的是大语言模型（LLM）一次可以处理的 token 数量，定义了它理解和生成文本的 " 记忆 "。更大的窗口，如 32,000 个 token，允许模型考虑更多上下文，从而在摘要等任务中提高连贯性。然而，它会增加计算成本。在实际 LLM 部署中，平衡窗口大小与效率至关重要。

4 问题 4：在微调 LLM 方面，LoRA 与 QLoRA 有何区别?

LoRA （低秩适配）是一种微调方法，它向模型的层添加低秩矩阵，从而实现高效的适配并最小化内存开销。QLoRA 通过应用量化（例如，4 位精度）来扩展这种方法，以进一步减少内存使用同时保持准确性。例如，QLoRA 可以在单个 GPU 上微调一个 70B 参数的模型，使其非常适合资源受限的环境。

5 问题 5：与贪婪解码相比，束搜索如何改进文本生成?

束搜索在文本生成过程中探索多个词序列，在每个步骤中保留顶部 k 候选词（束），而与贪婪解码不同，贪婪解码仅选择最可能的词。这种方法，例如使用 $k = 5$，通过平衡概率和多样性，确保更连贯的输出，特别是在机器翻译或对话生成等任务中。

6 Question 6: What role does temperature play in controlling LLM output?

温度是一个超参数，用于调整文本生成中标记选择的随机性。低温度（例如，0.3）倾向于高概率标记，产生可预测的输出。高温度（例如，1.5）通过平滑概率分布来增加多样性。将温度设置为 0.8 通常在故事讲述等任务中平衡创造性和连贯性。

7 Question 7: What is masked language modeling, and how does it aid pretraining?

掩码语言建模（MLM）涉及在序列中隐藏随机标记，并训练模型根据上下文预测它们。在 BERT 等模型中使用 MLM，可以培养语言的双向理解能力，使模型能够掌握语义

relationships. This pretraining approach equips LLMs for tasks like sentiment analysis or question answering.

8 Question 8: What are sequence-to-sequence models, and where are they applied?

Sequence-to-sequence (Seq2Seq) models transform an input sequence into an output sequence, often of different lengths. They consist of an encoder to process the input and a decoder to generate the output. Applications include machine translation (e.g., English to Spanish), text summarization, and chatbots, where variable-length inputs and outputs are common.

9 Question 9: How do autoregressive and masked models differ in LLM training?

Autoregressive models, like GPT, predict tokens sequentially based on prior tokens, excelling in generative tasks such as text completion. Masked models, like BERT, predict masked tokens using bidirectional context, making them ideal for understanding tasks like classification. Their training objectives shape their strengths in generation versus comprehension.

10 Question 10: What are embeddings, and how are they initialized in LLMs?

Embeddings are dense vectors that represent tokens in a continuous space, capturing semantic and syntactic properties. They are often initialized randomly or with pretrained models like GloVe, then fine-tuned during training. For example, the embedding for "dog" might evolve to reflect its context in pet-related tasks, enhancing model accuracy.

11 Question 11: What is next sentence prediction, and how does it enhance LLMs?

Next sentence prediction (NSP) trains models to determine if two sentences are consecutive or unrelated. During pretraining, models like BERT learn to classify 50% positive (sequential) and 50% negative (random) sentence pairs. NSP improves coherence in tasks like dialogue systems or document summarization by understanding sentence relationships.

关系。这种预训练方法使大型语言模型（LLM）能够执行情感分析或问答等任务。

8 问题 8：什么是序列到序列模型，以及它们的应用场景?

序列到序列（Seq2Seq）模型将输入序列转换为输出序列，通常长度不同。它们由一个编码器处理输入和一个解码器生成输出组成。应用包括机器翻译（例如，英语到西班牙语）、文本摘要和聊天机器人，其中常见可变长度的输入和输出。

9 Question 9: How do autoregressive and masked models differ in LLM training?

自回归模型（如 GPT）基于先前的标记顺序预测标记，在生成任务（如文本补全）中表现出色。掩码模型（如 BERT）使用双向上下文预测掩码标记，非常适合理解任务（如分类）。它们的训练目标塑造了它们在生成与理解方面的优势。

10 问题 10：什么是嵌入，以及它们如何在大型语言模型（LLM）中初始化?

嵌入是表示标记的密集向量，在连续空间中捕捉语义和句法属性。它们通常随机初始化或使用预训练模型（如 GloVe）初始化，然后在训练过程中微调。例如，"dog" 的嵌入可能会随着其在宠物相关任务中的上下文而演变，从而提高模型精度。

11 问题 11：什么是下一句预测，以及它如何增强大型语言模型（LLM）？

下一句预测（NSP）训练模型以确定两个句子是连续的还是不相关的。在预训练期间，像 BERT 这样的模型学习对 50% 的正面（顺序）和 50% 的负面（随机）句子对进行分类。NSP 通过理解句子关系，通过对话系统或文档摘要等任务提高连贯性。

12 Question 12: How do top-k and top-p sampling differ in text generation?

Top-k sampling selects the k most probable tokens (e.g., $k = 20$) for random sampling, ensuring controlled diversity. Top-p (nucleus) sampling chooses tokens whose cumulative probability exceeds a threshold p (e.g., 0.95), adapting to context. Top-p offers more flexibility, producing varied yet coherent outputs in creative writing.

13 Question 13: Why is prompt engineering crucial for LLM performance?

Prompt engineering involves designing inputs to elicit desired LLM responses. A clear prompt, like "Summarize this article in 100 words," improves output relevance compared to vague instructions. Its especially effective in zero-shot or few-shot settings, enabling LLMs to tackle tasks like translation or classification without extensive fine-tuning.

14 Question 14: How can LLMs avoid catastrophic forgetting during fine-tuning?

Catastrophic forgetting occurs when fine-tuning erases prior knowledge. Mitigation strategies include:

- Rehearsal: Mixing old and new data during training.
- Elastic Weight Consolidation: Prioritizing critical weights to preserve knowledge.
- Modular Architectures: Adding task-specific modules to avoid overwriting.

These methods ensure LLMs retain versatility across tasks.

15 Question 15: What is model distillation, and how does it benefit LLMs?

Model distillation trains a smaller "student" model to mimic a larger "teacher" models outputs, using soft probabilities rather than hard labels. This reduces memory and computational requirements, enabling deployment on devices like smartphones while retaining near-teacher performance, ideal for real-time applications.

16 Question 16: How do LLMs manage out-of-vocabulary (OOV) words?

LLMs use subword tokenization, like Byte-Pair Encoding (BPE), to break OOV words into known subword units. For instance, "cryptocurrency" might split into "crypto" and "currency." This approach allows LLMs to process rare or new words, ensuring robust language understanding and generation.

12 问题 12：在文本生成中，top-k 和 top-p 采样有何不同?

top-k 采样选择 k 最可能的标记（例如， $k = 20$ ）进行随机采样，确保控制的多样性。top-p （核）采样选择累积概率超过阈值 p （例如，0.95 ）的标记，适应上下文。top-p 提供更多灵活性，在创意写作中产生多样且连贯的输出。

13 Question 13: Why is prompt engineering crucial for LLM performance?

提示工程涉及设计输入以引出期望的 LLM 响应。清晰的提示，如 “ 用 100 字总结这篇文章 ”，与模糊的指令相比，提高了输出相关性。它在零样本或少样本设置中尤其有效，使 LLM 能够在不进行大量微调的情况下处理翻译或分类等任务。

14 问题 14：LLM 如何在微调过程中避免灾难性遗忘?

灾难性遗忘发生在微调时擦除先前的知识。缓解策略包括：

- 复习：在训练过程中混合旧数据和新技术据。
- 弹性权重整合：优先保留关键权重以保存知识。
- 模块化架构：添加特定任务的模块以避免覆盖。

这些方法确保 LLMs 在不同任务中保持多功能性。

15 Question 15: What is model distillation, and how does it benefit LLMs?

模型蒸馏通过训练一个较小的 “ 学生 ” 模型来模仿一个较大的 “ 教师 ” 模型的输出，使用软概率而不是硬标签。这减少了内存和计算需求，使模型能够在智能手机等设备上部署，同时保持接近教师模型的性能，非常适合实时应用。

16 问题 16：LLMs 如何处理词汇表外（ OOV ）单词?

LLMs 使用子词分词，如字节对编码（ BPE ），将 OOV 单词分解为已知的子词单元。例如，“cryptocurrency” 可能会被分割成 “crypto” 和 “currency”。这种方法使 LLMs 能够处理罕见或新单词，确保强大的语言理解和生成能力。

17 Question 17: How do transformers improve on traditional Seq2Seq models?

Transformers overcome Seq2Seq limitations by:

- Parallel Processing: Self-attention enables simultaneous token processing, unlike sequential RNNs.
- Long-Range Dependencies: Attention captures distant token relationships.
- Positional Encodings: These preserve sequence order.

These features enhance scalability and performance in tasks like translation.

18 Question 18: What is overfitting, and how can it be mitigated in LLMs?

Overfitting occurs when a model memorizes training data, failing to generalize. Mitigation includes:

- Regularization: L1/L2 penalties simplify models.
- Dropout: Randomly disables neurons during training.
- Early Stopping: Halts training when validation performance plateaus.

These techniques ensure robust generalization to unseen data.

19 Question 19: What are generative versus discriminative models in NLP?

Generative models, like GPT, model joint probabilities to create new data, such as text or images. Discriminative models, like BERT for classification, model conditional probabilities to distinguish classes, e.g., sentiment analysis. Generative models excel in creation, while discriminative models focus on accurate classification.

20 Question 20: How does GPT-4 differ from GPT-3 in features and applications?

GPT-4 surpasses GPT-3 with:

- Multimodal Input: Processes text and images.
- Larger Context: Handles up to 25,000 tokens versus GPT-3s 4,096.
- Enhanced Accuracy: Reduces factual errors through better fine-tuning.

These improvements expand its use in visual question answering and complex dialogues.

17 问题 17：Transformer 如何改进传统的 Seq2Seq 模型?

Transformer 通过以下方式克服 Seq2Seq 的局限性：

- 并行处理：自注意力机制可以实现同时处理标记，这与顺序 RNN 不同。
- 长距离依赖：注意力机制可以捕捉远距离标记之间的关系。
- 位置编码：这些编码保留了序列顺序。

These features enhance scalability and performance in tasks like translation.

18 问题 18：什么是过拟合，以及如何在 LLM 中缓解过拟合?

过拟合是指模型记住了训练数据，而无法泛化。缓解方法包括：

- Regularization: L1/L2 penalties simplify models.
- Dropout：在训练过程中随机禁用神经元。
- Early Stopping: Halts training when validation performance plateaus.

这些技术确保对未见数据的稳健泛化。

19 Question 19: What are generative versus discriminative models in NLP?

生成模型（如 GPT ）对联合概率进行建模以创建新数据，例如文本或图像。判别模型（如用于分类的 BERT ）对条件概率进行建模以区分类别，例如情感分析。生成模型擅长创作，而判别模型专注于精确分类。

20 Question 20: How does GPT-4 differ from GPT-3 in features and applications?

GPT-4 通过以下方面超越了 GPT-3：

- 多模态输入：处理文本和图像。
- Larger Context: Handles up to 25,000 tokens versus GPT-3s 4,096.
- 提升准确性：通过更好的微调减少事实性错误。

这些改进扩展了它在视觉问答和复杂对话中的使用范围。

21 Question 21: What are positional encodings, and why are they used?

Positional encodings add sequence order information to transformer inputs, as self-attention lacks inherent order awareness. Using sinusoidal functions or learned vectors, they ensure tokens like "king" and "crown" are interpreted correctly based on position, critical for tasks like translation.

22 Question 22: What is multi-head attention, and how does it enhance LLMs?

Multi-head attention splits queries, keys, and values into multiple subspaces, allowing the model to focus on different aspects of the input simultaneously. For example, in a sentence, one head might focus on syntax, another on semantics. This improves the models ability to capture complex patterns.

23 Question 23: How is the softmax function applied in attention mechanisms?

The softmax function normalizes attention scores into a probability distribution:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

In attention, it converts raw similarity scores (from query-key dot products) into weights, emphasizing relevant tokens. This ensures the model focuses on contextually important parts of the input.

24 Question 24: How does the dot product contribute to self-attention?

In self-attention, the dot product between query (Q) and key (K) vectors computes similarity scores:

$$\text{Score} = \frac{Q \cdot K}{\sqrt{d_k}}$$

High scores indicate relevant tokens. While efficient, its quadratic complexity ($O(n^2)$) for long sequences has spurred research into sparse attention alternatives.

25 Question 25: Why is cross-entropy loss used in language modeling?

Cross-entropy loss measures the divergence between predicted and true token probabilities:

$$L = -\sum y_i \log(\hat{y}_i)$$

21 问题 21：位置编码是什么，为什么使用它们？

位置编码向 transformer 输入添加序列顺序信息，因为自注意力机制缺乏固有的顺序感知能力。使用正弦函数或学习到的向量，它们确保像 "king" 和 "crown" 这样的标记根据位置被正确解释，这对于翻译等任务至关重要。

22 Question 22: What is multi-head attention, and how does it enhance LLMs?

多头注意力将查询、键和值分割到多个子空间中，使模型能够同时关注输入的不同方面。例如，在一个句子中，一个注意力头可能关注语法，另一个关注语义。这提高了模型捕获复杂模式的能力。

23 问题 23：如何在注意力机制中应用 softmax 函数？

softmax 函数将注意力分数归一化为概率分布：

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

在注意力机制中，它将原始相似度分数（来自查询 - 键点积）转换为权重，强调相关的标记。这确保了模型关注输入中上下文重要的部分。

24 问题 24：点积如何贡献于自注意力？

在自注意力中，查询向量 (Q) 和键向量 (K) 之间的点积计算相似度分数：

$$\text{Score} = \frac{Q \cdot K}{\sqrt{d_k}}$$

高分数表示相关的标记。虽然高效，但其对于长序列的二次复杂度 ($O(n^2)$) 已促使研究人员研究稀疏注意力的替代方案。

25 问题 25：为什么在语言建模中使用交叉熵损失？

交叉熵损失衡量预测标记概率和真实标记概率之间的发散：

$$L = -\sum y_i \log(\hat{y}_i)$$

It penalizes incorrect predictions, encouraging accurate token selection. In language modeling, it ensures the model assigns high probabilities to correct next tokens, optimizing performance.

26 Question 26: How are gradients computed for embeddings in LLMs?

Gradients for embeddings are computed using the chain rule during backpropagation:

$$\frac{\partial L}{\partial E} = \frac{\partial L}{\partial \text{logits}} \cdot \frac{\partial \text{logits}}{\partial E}$$

These gradients adjust embedding vectors to minimize loss, refining their semantic representations for better task performance.

27 Question 27: What is the Jacobian matrixs role in transformer backpropagation?

The Jacobian matrix captures partial derivatives of outputs with respect to inputs. In transformers, it helps compute gradients for multidimensional outputs, ensuring accurate updates to weights and embeddings during backpropagation, critical for optimizing complex models.

28 Question 28: How do eigenvalues and eigenvectors relate to dimensionality reduction?

Eigenvectors define principal directions in data, and eigenvalues indicate their variance. In techniques like PCA, selecting eigenvectors with high eigenvalues reduces dimensionality while retaining most variance, enabling efficient data representation for LLMs input processing.

29 Question 29: What is KL divergence, and how is it used in LLMs?

KL divergence quantifies the difference between two probability distributions:

$$D_{KL}(P||Q) = \sum P(x) \log \frac{P(x)}{Q(x)}$$

In LLMs, it evaluates how closely model predictions match true distributions, guiding fine-tuning to improve output quality and alignment with target data.

它惩罚错误的预测，鼓励准确选择标记。在语言建模中，它确保模型为正确的下一个标记分配高概率，优化性能。

26 问题 26：LLM 中的嵌入如何计算梯度?

嵌入的梯度在反向传播期间使用链式法则计算：

$$\frac{\partial L}{\partial E} = \frac{\partial L}{\partial \text{logits}} \cdot \frac{\partial \text{logits}}{\partial E}$$

这些梯度调整嵌入向量以最小化损失，改进它们的语义表示，以获得更好的任务性能。

27 Question 27: What is the Jacobian matrixs role in transformer backpropagation?

雅可比矩阵捕获输出相对于输入的偏导数。在 Transformer 中，它有助于计算多维输出的梯度，确保在反向传播期间对权重和嵌入进行准确更新，这对优化复杂模型至关重要。

28 问题 28：特征值和特征向量如何与降维相关?

特征向量定义数据的主方向，特征值指示其方差。在 PCA 等技术中，选择具有高特征值的特征向量可以降低维度，同时保留大部分方差，从而实现 LLM 输入处理的效数据表示。

29 问题 29：KL 散度是什么，以及它在 LLM 中如何使用?

KL 散度量化两个概率分布之间的差异：

$$D_{KL}(P||Q) = \sum P(x) \log \frac{P(x)}{Q(x)}$$

在 LLM 中，它评估模型预测与真实分布的匹配程度，指导微调以改进输出质量并与目标数据对齐。

30 Question 30: What is the derivative of the ReLU function, and why is it significant?

The ReLU function, $f(x) = \max(0, x)$, has a derivative:

$$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Its sparsity and non-linearity prevent vanishing gradients, making ReLU computationally efficient and widely used in LLMs for robust training.

31 Question 31: How does the chain rule apply to gradient descent in LLMs?

The chain rule computes derivatives of composite functions:

$$\frac{d}{dx} f(g(x)) = f'(g(x)) \cdot g'(x)$$

In gradient descent, it enables backpropagation to calculate gradients layer by layer, updating parameters to minimize loss efficiently across deep LLM architectures.

32 Question 32: How are attention scores calculated in transformers?

Attention scores are computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

The scaled dot product measures token relevance, and softmax normalizes scores to focus on key tokens, enhancing context-aware generation in tasks like summarization.

33 Question 33: How does Gemini optimize multimodal LLM training?

Gemini enhances efficiency via:

- Unified Architecture: Combines text and image processing for parameter efficiency.
- Advanced Attention: Improves cross-modal learning stability.
- Data Efficiency: Uses self-supervised techniques to reduce labeled data needs.

These features make Gemini more stable and scalable than models like GPT-4.

30 问题 30：ReLU 函数的导数是什么，为什么它很重要？

ReLU 函数， $f(x) = \max(0, x)$，的导数为：

$$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

它的稀疏性和非线性防止梯度消失，使得 ReLU 计算高效，并在 LLM 中广泛用于鲁棒的训练。

31 问题 31：链式法则如何应用于 LLM 中的梯度下降？

链式法则计算复合函数的导数：

$$\frac{d}{dx} f(g(x)) = f'(g(x)) \cdot g'(x)$$

在梯度下降中，它使反向传播能够逐层计算梯度，高效更新参数以最小化深层 LLM 架构中的损失。

32 问题 32：Transformer 中的注意力分数是如何计算的？

注意力分数的计算方式为：

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

缩放点积测量了 token 的相关性，而 softmax 将分数归一化以关注关键 token，从而在摘要等任务中增强上下文感知生成。

33 问题 33：Gemini 如何优化多模态 LLM 训练？

Gemini 通过以下方式提高效率：

- 统一架构：结合文本和图像处理以提高参数效率。
- 高级注意力：提高跨模态学习稳定性。
- 数据效率：使用自监督技术减少标记数据需求。

这些特性使 Gemini 比类似 GPT-4 的模型更稳定和可扩展。

34 Question 34: What types of foundation models exist?

Foundation models include:

- Language Models: BERT, GPT-4 for text tasks.
- Vision Models: ResNet for image classification.
- Generative Models: DALL-E for content creation.
- Multimodal Models: CLIP for text-image tasks.

These models leverage broad pretraining for diverse applications.

35 Question 35: How does PEFT mitigate catastrophic forgetting?

Parameter-Efficient Fine-Tuning (PEFT) updates only a small subset of parameters, freezing the rest to preserve pretrained knowledge. Techniques like LoRA ensure LLMs adapt to new tasks without losing core capabilities, maintaining performance across domains.

36 Question 36: What are the steps in Retrieval-Augmented Generation (RAG)?

RAG involves:

1. Retrieval: Fetching relevant documents using query embeddings.
2. Ranking: Sorting documents by relevance.
3. Generation: Using retrieved context to generate accurate responses.

RAG enhances factual accuracy in tasks like question answering.

37 Question 37: How does Mixture of Experts (MoE) enhance LLM scalability?

MoE uses a gating function to activate specific expert sub-networks per input, reducing computational load. For example, only 10% of a models parameters might be used per query, enabling billion-parameter models to operate efficiently while maintaining high performance.

38 Question 38: What is Chain-of-Thought (CoT) prompting, and how does it aid reasoning?

CoT prompting guides LLMs to solve problems step-by-step, mimicking human reasoning. For example, in math problems, it breaks down calculations into logical steps, improving

34 问题 34：有哪些基础模型类型?

基础模型包括：

- 语言模型：BERT 、 GPT-4 用于文本任务。
- 视觉模型：ResNet 用于图像分类。
- 生成模型：DALL-E 用于内容创作。
- 多模态模型：CLIP 用于文本 - 图像任务。

这些模型利用广泛的预训练来实现多样化的应用。

35 问题 35：PEFT 如何缓解灾难性遗忘?

参数高效微调（ PEFT ）仅更新一小部分参数，冻结其余参数以保留预训练知识。LoRA 等技术确保 LLM 能够适应新任务而不丢失核心能力，同时保持跨领域的性能。

36 Question 36: What are the steps in Retrieval-Augmented Generation (RAG)?

RAG 涉及：

1. 检索：使用查询嵌入获取相关文档。
2. 排序：按相关性对文档进行排序。
3. 生成：使用检索到的上下文来生成准确的响应。

RAG 增强了问答等任务中的事实准确性。

37 Question 37: How does Mixture of Experts (MoE) enhance LLM scalability?

MoE 使用门控函数来激活每个输入的特定专家子网络，从而减少计算负载。例如，每个查询可能只使用模型参数的 10%，使得具有数十亿参数的模型能够高效运行，同时保持高性能。

38 问题 38：什么是思维链（ CoT ）提示，以及它如何帮助推理?

CoT 提示引导 LLM 逐步解决问题，模仿人类推理。例如，在数学问题中，它将计算分解为逻辑步骤，从而提高

accuracy and interpretability in complex tasks like logical inference or multi-step queries.

39 Question 39: How do discriminative and generative AI differ?

Discriminative AI, like sentiment classifiers, predicts labels based on input features, modeling conditional probabilities. Generative AI, like GPT, creates new data by modeling joint probabilities, suitable for tasks like text or image generation, offering creative flexibility.

40 Question 40: How does knowledge graph integration improve LLMs?

Knowledge graphs provide structured, factual data, enhancing LLMs by:

- Reducing Hallucinations: Verifying facts against the graph.
- Improving Reasoning: Leveraging entity relationships.
- Enhancing Context: Offering structured context for better responses.

This is valuable for question answering and entity recognition.

41 Question 41: What is zero-shot learning, and how do LLMs implement it?

Zero-shot learning allows LLMs to perform untrained tasks using general knowledge from pretraining. For example, prompted with "Classify this review as positive or negative," an LLM can infer sentiment without task-specific data, showcasing its versatility.

42 Question 42: How does Adaptive Softmax optimize LLMs?

Adaptive Softmax groups words by frequency, reducing computations for rare words. This lowers the cost of handling large vocabularies, speeding up training and inference while maintaining accuracy, especially in resource-limited settings.

43 Question 43: How do transformers address the vanishing gradient problem?

Transformers mitigate vanishing gradients via:

- Self-Attention: Avoiding sequential dependencies.
- Residual Connections: Allowing direct gradient flow.
- Layer Normalization: Stabilizing updates.

在逻辑推理或多步骤查询等复杂任务中的准确性和可解释性。

39 问题 39：判别式 AI 和生成式 AI 有什么区别?

判别式 AI，如情感分类器，根据输入特征预测标签，建模条件概率。生成式 AI，如 GPT，通过建模联合概率创建新数据，适用于文本或图像生成等任务，提供创造性灵活性。

40 Question 40: How does knowledge graph integration improve LLMs?

知识图谱提供结构化的事实数据，通过以下方式增强 LLM：

- 减少幻觉：对照图谱验证事实。
- 改进推理：利用实体关系。
- 增强上下文：提供结构化上下文以获得更好的响应。

这对于问答和实体识别很有价值。

41 问题 41：什么是零样本学习，以及 LLM 如何实现它?

零样本学习允许 LLM 使用预训练中的通用知识来执行未训练的任务。例如，当提示它 “ 将此评论分类为正面或负面 ” 时，LLM 可以在没有特定任务数据的情况下推断情感，展示其多功能性。

42 问题 42：Adaptive Softmax 如何优化 LLM?

Adaptive Softmax 按频率对单词进行分组，减少对稀有单词的计算量。这降低了处理大型词汇表的成本，在资源受限的情况下加快了训练和推理，同时保持了准确性。

43 Question 43: How do transformers address the vanishing gradient problem?

Transformers 通过以下方式缓解梯度消失：

- 自注意力机制：避免顺序依赖。
- 残差连接：允许直接梯度流动。
- 层归一化：稳定更新。

These ensure effective training of deep models, unlike RNNs.

44 Question 44: What is few-shot learning, and what are its benefits?

Few-shot learning enables LLMs to perform tasks with minimal examples, leveraging pretrained knowledge. Benefits include reduced data needs, faster adaptation, and cost efficiency, making it ideal for niche tasks like specialized text classification.

45 Question 45: How would you fix an LLM generating biased or incorrect outputs?

To address biased or incorrect outputs:

1. Analyze Patterns: Identify bias sources in data or prompts.
2. Enhance Data: Use balanced datasets and debiasing techniques.
3. Fine-Tune: Retrain with curated data or adversarial methods.

These steps improve fairness and accuracy.

46 Question 46: How do encoders and decoders differ in transformers?

Encoders process input sequences into abstract representations, capturing context. Decoders generate outputs, using encoder outputs and prior tokens. In translation, the encoder understands the source, and the decoder produces the target language, enabling effective Seq2Seq tasks.

47 Question 47: How do LLMs differ from traditional statistical language models?

LLMs use transformer architectures, massive datasets, and unsupervised pretraining, unlike statistical models (e.g., N-grams) that rely on simpler, supervised methods. LLMs handle long-range dependencies, contextual embeddings, and diverse tasks, but require significant computational resources.

48 Question 48: What is a hyperparameter, and why is it important?

Hyperparameters are preset values, like learning rate or batch size, that control model training. They influence convergence and performance; for example, a high learning rate may cause instability. Tuning hyperparameters optimizes LLM efficiency and accuracy.

这些确保了深度模型的有效训练，与 RNNs 不同。

44 问题 44：什么是小样本学习，它的好处是什么?

小样本学习使 LLMs 能够使用最少的示例执行任务，利用预训练知识。好处包括减少数据需求、更快适应和成本效益，使其非常适合像专业文本分类这样的利基任务。

45 问题 45：你会如何修复一个生成有偏见或不正确输出的 LLM?

要解决有偏见或不正确的输出：

1. 分析模式：识别数据或提示中的偏见来源。 2. 增强数据：使用平衡数据集和去偏见技术。 3. 微调：使用精选数据或对抗性方法重新训练。

这些步骤提高了公平性和准确性。

46 Question 46: How do encoders and decoders differ in transformers?

编码器将输入序列处理成抽象表示，捕捉上下文。解码器使用编码器输出和先验标记生成输出。在翻译中，编码器理解源语言，解码器生成目标语言，从而实现有效的 Seq2Seq 任务。

47 问题 47：LLM 与传统统计语言模型有何不同?

LLM 使用 Transformer 架构、大规模数据集和自监督预训练，而统计模型（例如 N-gram ）则依赖更简单的监督方法。LLM 处理长距离依赖关系、上下文嵌入和多样化任务，但需要大量的计算资源。

48 问题 48：什么是超参数，为什么它很重要?

超参数是预设值，如学习率或批大小，它们控制模型训练。它们影响收敛和性能；例如，高学习率可能导致不稳定。调整超参数优化 LLM 的效率和准确性。

49 Question 49: What defines a Large Language Model (LLM)?

LLMs are AI systems trained on vast text corpora to understand and generate human-like language. With billions of parameters, they excel in tasks like translation, summarization, and question answering, leveraging contextual learning for broad applicability.

50 Question 50: What challenges do LLMs face in deployment?

LLM challenges include:

- Resource Intensity: High computational demands.
- Bias: Risk of perpetuating training data biases.
- Interpretability: Complex models are hard to explain.
- Privacy: Potential data security concerns.

Addressing these ensures ethical and effective LLM use.

Conclusion

This guide equips you with in-depth knowledge of LLMs, from core concepts to advanced techniques. Share it with your LinkedIn community to inspire and educate aspiring AI professionals. For more AI/ML insights, connect with me at Your LinkedIn Profile.

49 问题 49：什么定义了大型语言模型（LLM）？

LLM 是在大量文本语料库上训练的人工智能系统，用于理解和生成类人语言。凭借数十亿个参数，它们在翻译、摘要和问答等任务中表现出色，利用上下文学习实现广泛适用性。

50 问题 50：LLM 在部署中面临哪些挑战?

LLM 的挑战包括:

- 资源密集：高计算需求。
- 偏见：存在加剧训练数据偏见的风险。
- 可解释性：复杂模型难以解释。
- 隐私：潜在的数据安全问题。

解决这些问题可以确保 LLM 的合乎道德和有效使用。

结论

本指南使您深入了解 LLM，从核心概念到高级技术。将其分享到您的 LinkedIn 社区，以激励和教育有志于 AI 的专业人士。如需更多 AI/ML 见解，请通过您的 LinkedIn 个人资料与我联系。