前言

在 LLM 出现之前，**传统方法擅长分类任务**，例如垃圾邮件分类任务，可以通过手工编写的规则或简单的模型来实现。然而，它们通常**在需要复杂理解和生成能力的语言任务中表现不佳**，例如解析详细说明、进行上下文分析以及创建连贯且符合上下文的原始文本。例如，前几代语言模型无法根据关键字列表撰写电子邮件——而对于LLM 来说，这项任务轻而易举。

LLM拥有卓越的理解、生成和解释人类语言的能力。我们说语言模型"理解"，是指它们能够以看似连贯且与上下文相关的方式处理和生成文本，而不是指它们拥有类似人类的意识或理解能力。

深度学习是机器学习和人工智能 (AI) 的一个分支，专注于神经网络，得益于深度学习技术的进步，**LLM 可以在海量文本数据上进行训练。**与以往的方法相比，这种大规模训练使 LLM **能够捕捉更深层次的语境信息和人类语言的微妙之处。**因此，LLM 在文本翻译、情感分析、问答等众多 NLP 任务中的表现均有显著提升。

LLM 的成功可以归功于支撑许多 LLM 的 **Transformer 架构**，以及训练 LLM 所需的**海量数据**，这使得 LLM 能够捕捉各种语言的细微差别、语境和模式，而这些对于手动编码来说极具挑战性。

1.1 什么是LLM

LLM 是一种旨在理解、生成和响应类似人类文本的神经网络。这些模型是深度神经网络，基于海量文本数据进行训练，有时甚至涵盖互联网上大量公开文本。

"大语言模型"中的"大"既指模型的**参数规模**，也指其训练所基于的**庞大数据集。**这类模型通常拥有数百亿（10B级）甚至数千亿个（100B级）参数，这些参数是网络中可调的权重，会在训练过程中进行优化，以预测序列中的下一个单词。下一个单词预测（Next-word prediction）之所以有效，是因为它利用**语言固有的序列特性**来训练模型理解文本中的上下文、结构和关系。然而，这是一个非常简单的任务，因此它能够生成如此强大的模型令许多研究人员感到惊讶。在后续章节中，我们将逐步讨论并实现下一个单词的训练过程。

LLM 采用一种称为 Transformer 的架构，这使得它们在进行预测时能够**选择性地关注输入的不同部分**，从而使它们特别擅长处理人类语言的细微差别和复杂性。

由于LLM能够生成文本，因此 LLM 也常被称为生成式人工智能 (Generative AI) 的一种形式，通常缩写为生成式人工智能 (Generative AI) 或 GenAI。如图 1.1 所示，人工智能涵盖了更广泛的领域，即创造能够执行需要类似人类智能的任务的机器，包括理解语言、识别模式和做出决策，并包含机器学习和深度学习等子领域。

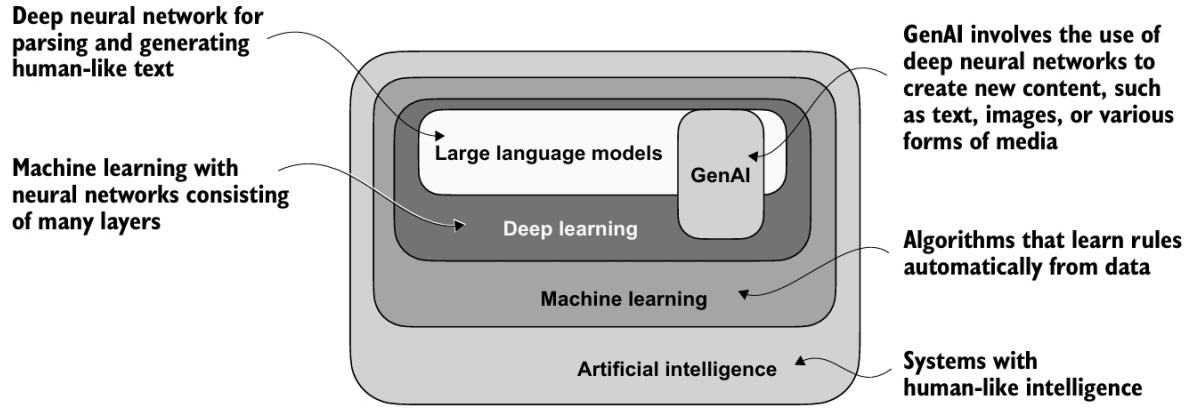

Figure 1.1 As this hierarchical depiction of the relationship between the different fields suggests, LLMs represent a specific application of deep learning techniques, using their ability to process and generate human-like text. Deep learning is a specialized branch of machine learning that focuses on using multilayer neural networks. Machine learning and deep learning are fields aimed at implementing algorithms that enable computers to learn from data and perform tasks that typically require human intelligence.

如图 1.1 所示，深度学习是机器学习的一个子集，专注于利用三层或多层神经网络（也称为深度神经网络）来对数据中的复杂模式和抽象进行建模。与深度学习不同，传统机器学习需要手动提取特征。这意味着人类专家需要识别并选择与模型最相关的特征。

1.2 LLMs的应用

由于其解析和理解非结构化文本数据的高级能力，LLM 在各个领域都有着广泛的应用。如今，LLM 被用于机器翻译、小说文本生成、情感分析、文本摘要以及许多其他任务。近年来，LLM 也被用于内容创作，例如撰写小说、文章，甚至计算机代码。

LLM还可以支持复杂的聊天机器人和虚拟助手，例如 OpenAI 的 ChatGPT 或谷歌的 Gemini（原名 Bard），它们可以回答用户查询，并增强谷歌搜索或微软必应等传统搜索引擎的功能。此外，LLM还可用于从医学或法律等专业领域的海量文本中有效地检索知识。这包括筛选文档、总结长篇文章以及回答技术问题。

1.3 构建和使用LLM的阶段

创建 LLM 的一般过程包括**预训练**和**微调**。“预训练”中的“预”指的是初始阶段，在这个阶段，像 LLM 这样的模型会在**庞大而多样化的数据集**上进行训练，以建立对语言的**广泛理解**。这个预训练模型随后会作为基础资源，可以通过微调进一步完善。微调是指在**更窄的数据集上**对模型进行专门训练，使其更**适用于特定任务或领域**。图 1.3 展示了这种由预训练和微调组成的两阶段训练方法。

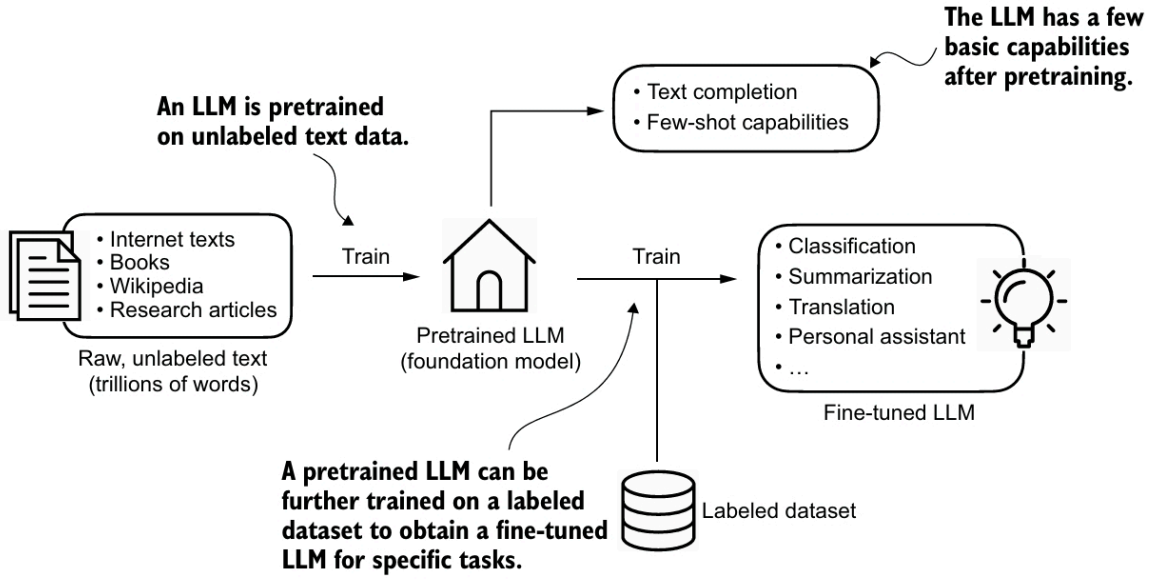

Figure 1.3 Pretraining an LLM involves next-word prediction on large text datasets. A pretrained LLM can then be fine-tuned using a smaller labeled dataset.

创建 LLM 的第一步是使用大量文本数据（有时也称为原始文本）进行训练。这里的“原始”指的是这些数据只是普通文本，没有任何标签信息。（可能会进行过滤，例如删除格式字符或未知语言的文档。）

在预训练阶段，LLM 采用自监督学习，模型根据输入数据生成自己的标签。此类模型的一个典型示例是 GPT-3 模型。

在通过大型文本数据集训练获得预训练的 LLM 之后，我们可以在有标签数据上进一步训练 LLM，这也称为微调。

两种最流行的微调 类别是指令微调和分类微调。在指令微调中，带标签的数据集由指令和答案对组成，例如，要求翻译文本的查询以及正确翻译的文本。在分类微调中，带标签的数据集由文本和相关的类别标签组成，例如，与“垃圾邮件”和“非垃圾邮件”标签相关的电子邮件。

1.4 介绍Transformer架构

大多数现代LLM都依赖于 Transformer 架构，这是一种深度神经网络架构，于 2017 年的论文《Attention Is All You Need》（https://arxiv.org/abs/1706.03762）中提出。最初的 Transformer是为机器翻译而开发的，用于将英文文本翻译成德语和法语。图 1.4 展示了 Transformer 架构的简化版本。

Transformer 架构由两个子模块组成：编码器和解码器。**编码器模块处理输入文本，并将其编码为一系列数值表示或向量，用于捕捉输入的上下文信息。**然后，**解码器模块获取这些编码向量并生成输出文本。**例如，在翻译任务中，编码器将源语言文本编码为向量，解码器解码这些向量以生成目标语言文本。编码器和解码器都由多层结构组成，这些层**通过自注意力机制连接。**

Transformer 和 LLM 的一个关键组件是自注意力机制（图中未显示），它**允许模型衡量序列中不同单词或 token 之间的相对重要性。**该机制使模型能够捕捉输入数据中的**长程依赖关系**和**上下文关系**，从而增强其生成连贯且上下文相关的输出的能力。

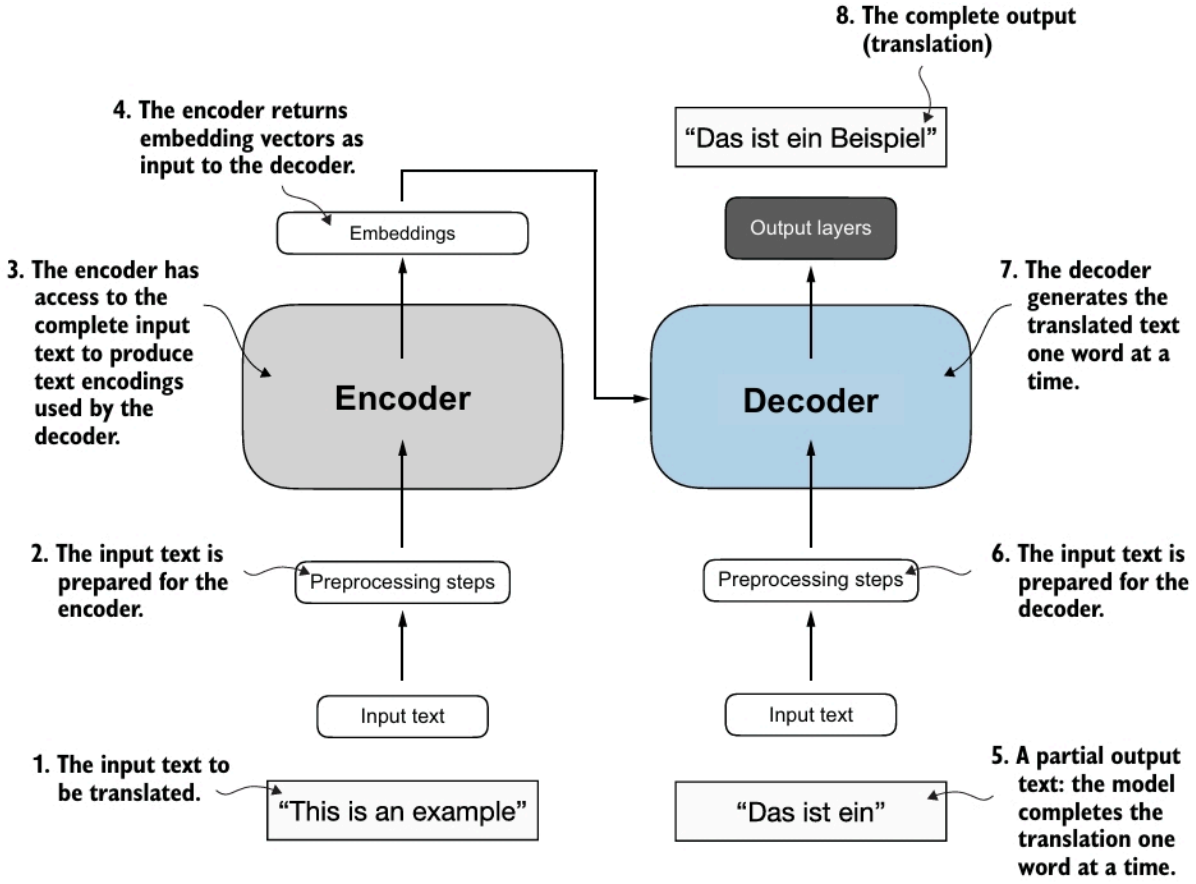

Figure 1.4 A simplified depiction of the original transformer architecture, which is a deep learning model for language translation. The transformer consists of two parts: (a) an encoder that processes the input text and produces an embedding representation (a numerical representation that captures many different factors in different dimensions) of the text that the (b) decoder can use to generate the translated text one word at a time. This figure shows the final stage of the translation process where the decoder has to generate only the final word ("Beispiel"), given the original input text ("This is an example") and a partially translated sentence ("Das ist ein"), to complete the translation.

BERT 建立在原始 Transformer 的编码器子模块之上，其训练方法与 GPT 有所不同。GPT 专为生成任务而设计，而 BERT 及其变体则专注于掩蔽词预测，即模型预测给定句子中被掩蔽或隐藏的词，如图 1.5 所示。这种独特的训练策略使 **BERT 在文本分类任务（包括情绪预测和文档分类）中拥有优势。**作为其能力的应用，截至本文撰写时，X（前身为 Twitter）正在使用 BERT 来检测有害内容。

另一方面，GPT 专注于原始 Transformer 架构的解码器部分，专为需要生成文本的任务而设计。这些任务包括机器翻译、文本摘要、小说写作、编写计算机代码等等。

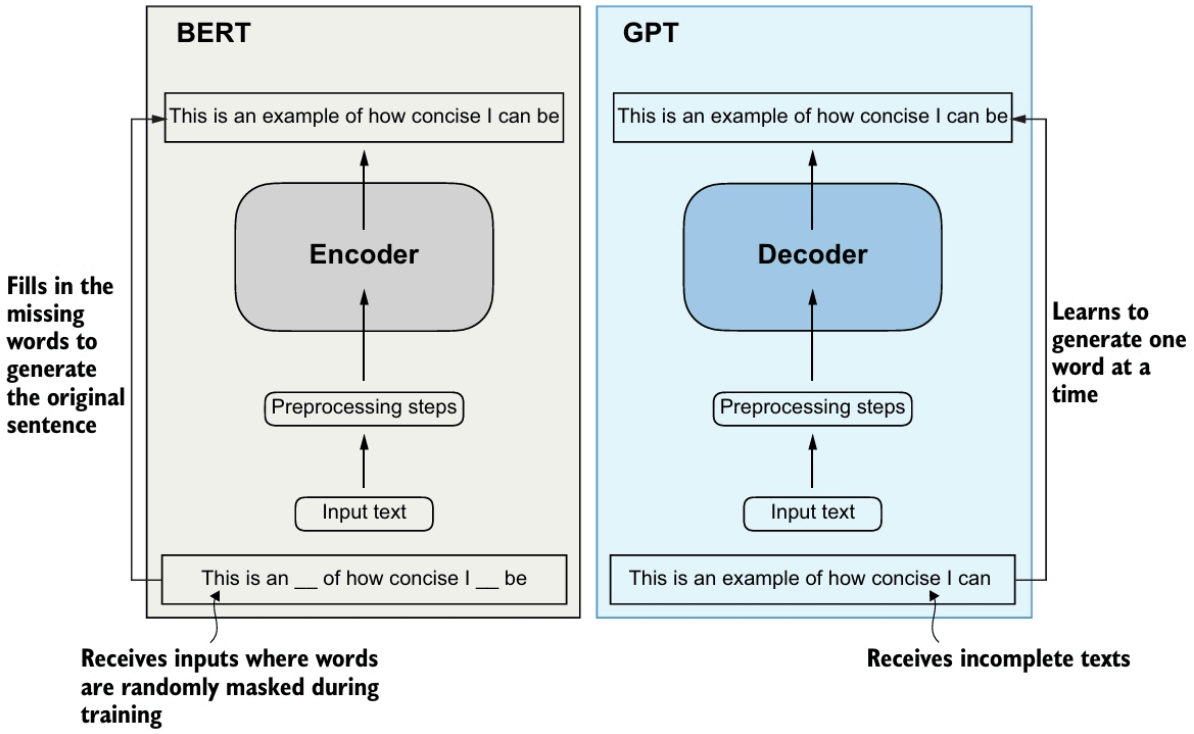

Figure 1.5 A visual representation of the transformer's encoder and decoder submodules. On the left, the encoder segment exemplifies BERT-like LLMs, which focus on masked word prediction and are primarily used for tasks like text classification. On the right, the decoder segment showcases GPT-like LLMs, designed for generative tasks and producing coherent text sequences.

GPT 模型主要为执行**文本补全任务**而设计和训练，其功能也展现出卓越的多功能性。这些模型擅长执行**零样本学习(zero-shot learning)**和**少样本学习(few-shot learning)**任务。零样本学习是指在没有任何先前特定示例的情况下，能够推广到完全未见过的任务。少样本学习则涉及从用户提供的最少数量的输入示例中进行学习，如图 1.6 所示。

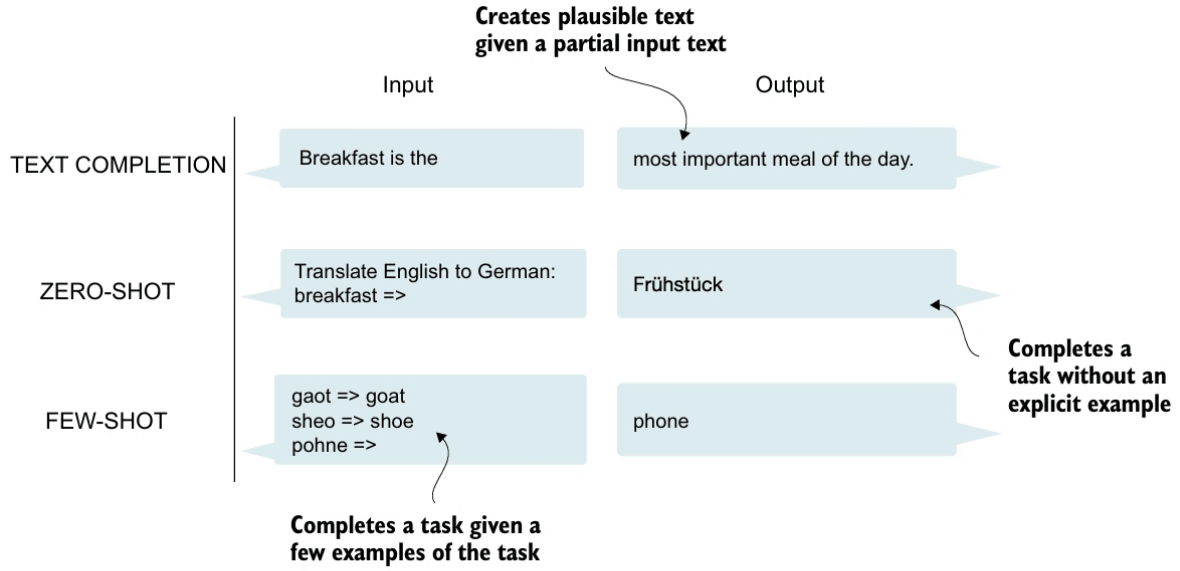

Figure 1.6 In addition to text completion, GPT-like LLMs can solve various tasks based on their inputs without needing retraining, fine-tuning, or task-specific model architecture changes. Sometimes it is helpful to provide examples of the target within the input, which is known as a few-shot setting. However, GPT-like LLMs are also capable of carrying out tasks without a specific example, which is called zero-shot setting.

1.5 使用大数据集

流行的 GPT 和 BERT 类模型拥有庞大的训练数据集，这些数据集代表着丰富多样且全面的文本语料库，涵盖数十亿词汇，涵盖了海量主题、自然语言和计算机语言。为了提供一个具体示例，表 1.1 总结了用于预训练 GPT-3 的数据集，该模型是 ChatGPT 第一版的基础模型。

Table 1.1 The pretraining dataset of the popular GPT-3 LLM

Dataset name	Dataset description	Number of tokens	Proportion in training data
CommonCrawl (filtered)	Web crawl data	410 billion	60%
WebText2	Web crawl data	19 billion	22%
Books1	Internet-based book corpus	12 billion	8%
Books2	Internet-based book corpus	55 billion	8%
Wikipedia	High-quality text	3 billion	3%

这些模型的预训练特性使其在下游任务的进一步微调方面具有极强的通用性，因此它们也被称为基础模型或基础模型。预训练 LLM 需要访问大量资源，而且成本非常高昂。例如，GPT-3 的预训练成本估计为 460 万美元（以云计算积分计算）（https://mng.bz/VxEW）。

1.6 近距离观察GPT架构

GPT 最初由 OpenAI 的 Radford 等人在论文《Improving Language Understanding by Generative Pre-Training》（https://mng.bz/x2qg）中提出。GPT-3 是该模型的扩展版本，拥有更多参数，并在更大的数据集上进行训练。此外，ChatGPT 中提供的原始模型是使用 OpenAI InstructGPT 论文（https://arxiv.org/abs/2203.02155）中的方法，在大型指令数据集上对 GPT-3 进行微调而创建的。如图 1.6 所示，这些模型是性能良好的文本补全模型，并且可以执行其他任务，例如拼写纠正、分类或语言翻译。考虑到 GPT 模型是在相对简单的下一个单词预测任务上进行预训练的（如图 1.7 所示），这实际上非常了不起。

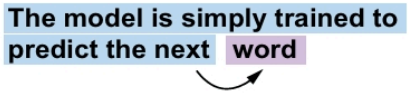

Figure 1.7 In the next-word prediction pretraining task for GPT models, the system learns to predict the upcoming word in a sentence by looking at the words that have come before it. This approach helps the model understand how words and phrases typically fit together in language, forming a foundation that can be applied to various other tasks.

下一个单词预测任务是一种自监督学习，也就是自我标记的一种形式。这意味着我们不需要明确地为训练数据收集标签，而是可以使用数据本身的结构：我们可以将句子或文档中的下一个单词用作模型应该预测的标签。由于这个下一个单词预测任务允许我们"动态"创建标签，因此可以使用大量未标记的文本数据集来训练LLM。

与我们在 1.4 节中介绍的原始 Transformer 架构相比，通用 GPT 架构相对简单。本质上，它**只有解码器部分（Decoder-only）**，没有编码器（图 1.8）。由于像 GPT 这样的解码器式模型通过一次预测一个单词来生成文本，因此它们被认为是一种**自回归模型**。自回归模型**将其先前的输出作为未来预测的输入**。因此，在 GPT 中，**每个新词都是根据其之前的序列来选择的**，这提高了最终文本的连贯性。

GPT-3 等架构也比原始 Transformer 模型规模大得多。例如，原始 Transformer 的编码器和解码器模块重复了六次。GPT-3 共有 96 个 Transformer 层和 1750 亿个参数。

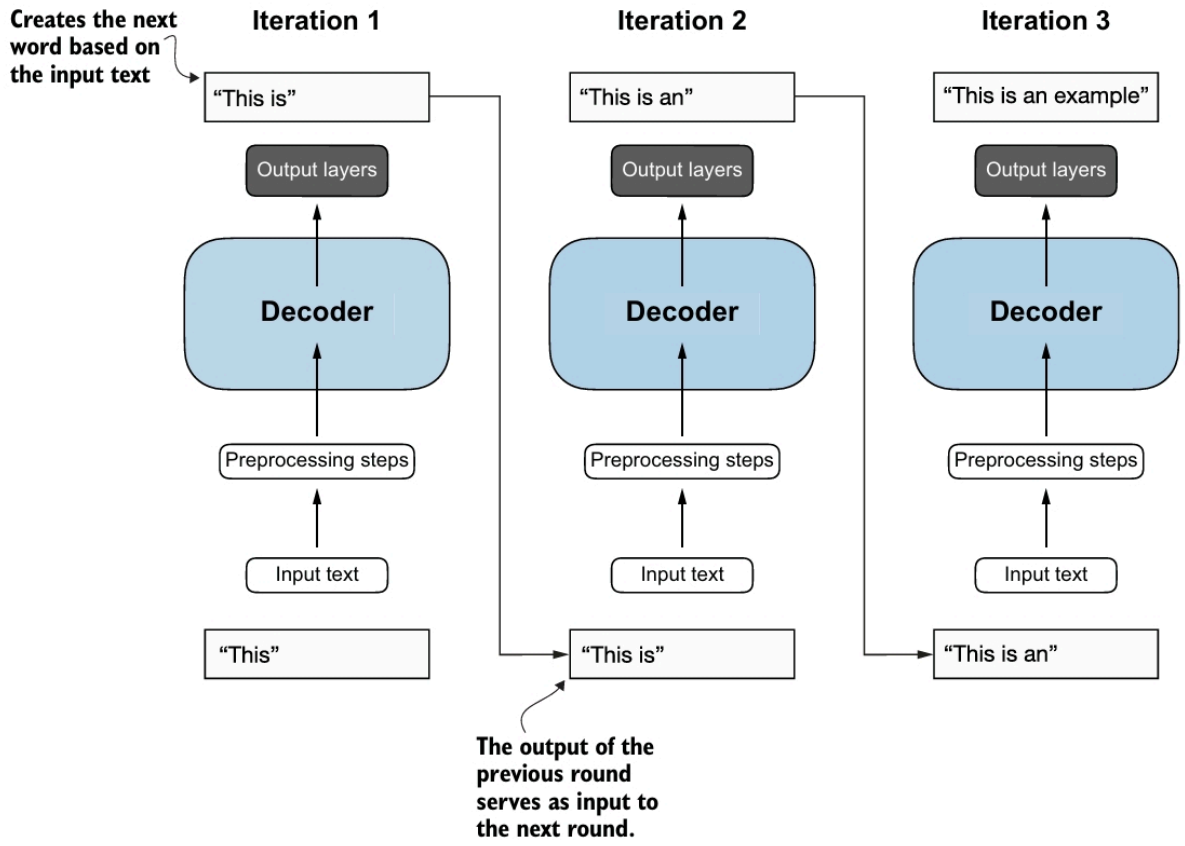

Figure 1.8 The GPT architecture employs only the decoder portion of the original transformer. It is designed for unidirectional, left-to-right processing, making it well suited for text generation and next-word prediction tasks to generate text in an iterative fashion, one word at a time.

执行模型未经明确训练的任务的能力被称为**涌现**行为。这种能力并非在训练过程中被明确教授，而是在模型接触不同语境中的大量多语言数据后自然而然地产生的。GPT 模型能够"学习"不同语言之间的翻译模式，并执行翻译任务，即使它们未经专门训练。这一事实也展现了这些大规模生成式语言模型的优势和能力。我们可以执行不同的任务，而无需为每种任务使用不同的模型。

1.7 构建一个LLM

在第一阶段，我们将学习基本的数据预处理步骤，并编写每个 LLM 核心的注意力机制。

接下来，在第二阶段，我们将学习如何编写和预训练一个能够生成新文本的类似 GPT 的 LLM。我们还将介绍评估 LLM 的基础知识，这对于开发强大的 NLP 系统至关重要。
从头开始预训练 LLM 是一项艰巨的任务，对于类似 GPT 的模型来说，计算成本可能高达数千甚至数百万美元。因此，第二阶段的重点是使用小型数据集实现用于教育目的的训练。此外，我还提供了加载公开可用模型权重的代码示例。

最后，在第三阶段，我们将使用预训练的 LLM 并对其进行微调，使其能够执行诸如回答查询或对文本进行分类之类的指令——这些是许多实际应用和研究中最常见的任务。

总结

- LLM 彻底改变了自然语言处理领域，此前该领域主要依赖于明确的规则系统和更简单的统计方法。LLM 的出现引入了新的深度学习驱动方法，推动了人类语言理解、生成和翻译的进步。
- 现代 LLM 的训练主要分为两个步骤：– 首先，使用句子中下一个单词的预测作为标签，在大量未标记文本语料库上进行预训练。– 然后，在较小的已标记目标数据集上进行微调，以遵循指令或执行分类任务。
- LLM 基于 Transformer 架构。Transformer 架构的核心思想是一种注意力机制，它使 LLM 在逐个单词生成输出时能够选择性地访问整个输入序列。

- 最初的 Transformer 架构由用于解析文本的编码器和用于生成文本的解码器组成。
- 用于生成文本并执行指令的 LLM，例如 GPT-3 和 ChatGPT，仅实现了解码器模块，从而简化了架构。
- 包含数十亿个单词的大型数据集对于预训练 LLM 至关重要。
- 虽然类似 GPT 的模型的一般预训练任务是预测句子中的下一个单词，但这些 LLM 展现出新兴特性，例如能够对文本进行分类、翻译或总结。
  一旦 LLM 完成预训练，就可以针对各种下游任务更有效地对生成的基础模型进行微调。
- 在自定义数据集上进行微调的 LLM 在特定任务上的表现可以优于一般的 LLM