Decoder－only 架构

前面介绍的Encoder-Decoder架构在生成任务上表现优异，能够深入理解输入序列语义并生成连贯的文本。然而，在许多开放式生成任务中，输入序列往往较为简单或缺失，此时维持完整编码器处理这类输入可能会显得冗余。在这种场景下，更轻量灵活的Decoder-only架构表现更为出色。

Decoder-only架构通过自回归方式逐字生成文本。它不仅能保持长文本的连贯性和一致性，还能在缺乏明确输入时自然流畅地生成内容。由于去除了编码器部分，模型更加轻量化，训练和推理速度更快，在相同规模下可能表现更优。

GPT系列语言模型

这种架构最早可追溯到2018年的GPT-1模型。当时由于BERT为代表的Encoder-only架构表现出色，Decoder-only并未受到足够关注。直到2020年GPT-3取得突破性成功后，Decoder-only架构开始广泛应用于各类大语言模型中。其中最具代表性的是OpenAI的GPT系列和Meta的LLaMA系列。GPT系列起步最早，性能领先，但从第三代开始转向闭源。LLaMA系列虽然起步较晚，但凭借出色性能和开源策略，在该领域占据重要地位。目前，DeepSeek、Qwen、GLM、Baichuan、Yi等知名大模型都采用了Decoder-only架构。

GPT（Generative Pre-trained Transformer）系列是由OpenAI开发的Decoder-only大语言模型。自2018年问世以来，GPT系列在模型规模和预训练范式上不断创新，引领了大语言模型发展浪潮。其演进可分为五个阶段，从参数规模和预训练语料来看呈现激增趋势。但自ChatGPT起，GPT系列转向闭源，具体参数量和预训练数据信息不再公开。根据扩展法则，可以推测ChatGPT及后续版本在这两个维度都有显著增长。

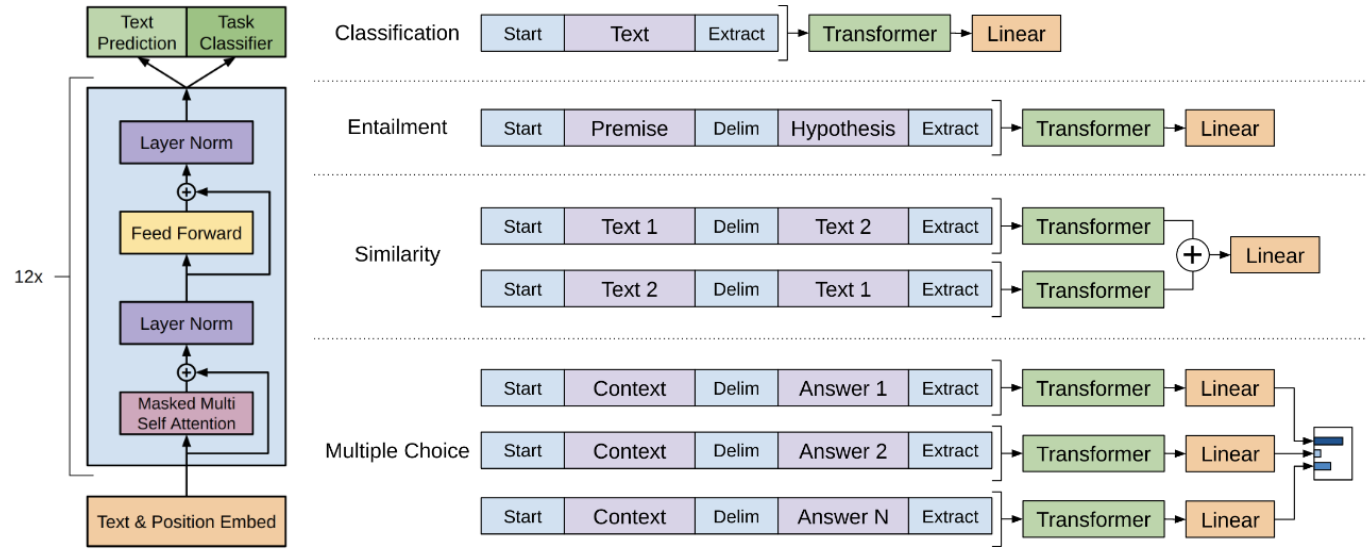
OpenAI前首席科学家Ilya Sutskever透露，公司早期就在探索通过下一词预测解决无监督学习问题。当时使用的RNN模型难以解决长距离依赖问题。2017年Transformer的出现为此提供了新思路，指明了发展方向。2018年6月发布的GPT-1开创了Decoder-only架构下通过下一词预测实现无监督文本生成的先河。

GPT-1

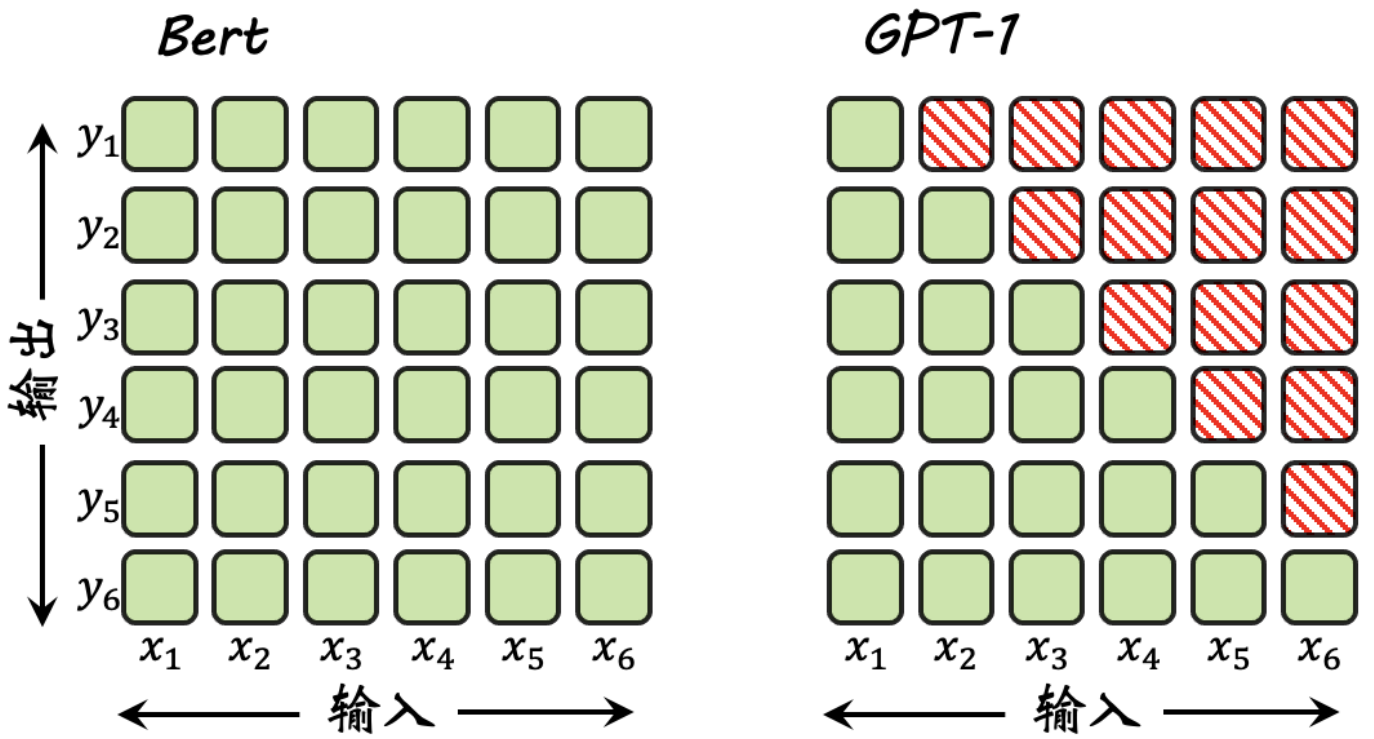
论文:Improving Language Understanding by Generative Pre-Training

链接:https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

GPT-1 采用了 Transformer 的 Decoder 部分，由于没有 Encoder 部分，因此没有交叉注意力模块。模型由12个解码块堆叠而成，每个解码块包含带掩码的自注意力模块和全连接前馈模块，隐藏层维度768，12个注意力头，总参数约1.17亿。



与BERT-Base相比，GPT-1结构相似，主要区别在于BERT使用双向自注意力机制，而GPT-1使用带掩码的单向自注意力机制。



总结一下，GPT-1的核心特点如下:

1.单向自注意力机制

- 采用因果掩码(causal mask)实现自回归语言建模
- 每个token i 只能关注其之前的所有token $(1, 2, \ldots, i-1)$
- 严格保持从左到右的生成顺序

2.规范化与残差连接

- 采用post-LayerNorm架构
- 每个子层后添加残差连接

- 与原始Transformer保持一致

3.位置信息编码

- 使用绝对位置编码
- 支持正弦/余弦或可学习的位置嵌入
- 位置编码与token嵌入相加

4.前馈神经网络

- 每层包含双层MLP结构,第一层将维度扩大4倍,第二层恢复原始维度
- 使用GeLU激活函数,其数学表达式为:
  $\mathrm{GeLU}(x) = x \cdot \Phi(x)$
  - 其中 $\Phi(x)$ 是标准正态分布的累积分布函数:
  - $\Phi(x) = \frac{1}{2}[1 + \mathrm{erf}(\frac{x}{\sqrt{2}})]$
  - $\mathrm{erf}(x)$ 是误差函数:
  - $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$
- GeLU相比ReLU的优势在于能够在负值区间保持梯度,使得训练更加稳定
- 提供非线性变换能力

5.分词方案

- 采用BPE(字节对编码)算法
- 有效平衡词表大小和分词粒度
- 适应开放词表场景

GPT-1采用典型的语言模型（LM）学习目标(负对数似然，也是交叉熵损失)，给定一个文本序列，模型通过最大化每个token在其前文条件下的条件概率来学习，其损失函数为:

$$L_1(\mathcal{C}) = \sum_{t=1}^{T} -\log P_\theta \left(x_t \mid x_1, \ldots, x_{t-1}\right) \tag{1}$$

其中:

- x_t 表示序列中第t个token
- P_θ 由Transformer Decoder计算得到的条件概率
- 通过因果掩码(Causal Mask)实现自回归生成

GPT-1在预训练阶段使用了包含约8亿Token、总量接近5GB的BookCorpus小说语料库。预训练采用自回归式的下一词预测任务，模型通过预测序列中的下一个Token来学习。这种无需人工标注的预训练方式让模型自然地掌握了语言知识和生成能力，为自然语言处理开辟了新方向。

GPT-1在下游任务微调时采用以下步骤:

1.提取特征表示

- 获取输入序列最后一个token的隐藏状态向量 h_l^m
- 该向量包含了整个序列的语义信息

2.分类预测

- 通过线性层将特征映射到标签空间
- 使用softmax函数计算类别概率分布:

$$P\left(y \mid x^1, \ldots, x^m\right) = \operatorname{softmax}\left(h_l^m W_y\right) \tag{2}$$

3.监督学习

- 在标注数据集上最小化交叉熵损失:

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P\left(y \mid x^1, \ldots, x^m\right) \tag{3}$$

4.辅助目标优化

- 在微调过程中结合预训练目标可以:
  - 增强模型的通用语言理解能力
  - 加速训练收敛
- 最终的联合损失函数为:

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C}) \tag{4}$$

其中 λ 为权重系数,用于平衡两个目标的重要性

GPT-2

论文:Language Models are Unsupervised Multitask Learners

链接:https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

GPT-2延续了GPT-1的架构，推出了四个不同规模的版本。从Small到XL版本，模型规模逐步扩大:Small版本与GPT-1相近，有12层解码块；Medium版本接近BERT-Large，有24层解码块；Large版本有36层解码块；XL版本最大，有48层解码块，总参数达15亿。

GPT-2相比GPT-1的另一个区别是使用pre-Norm，而GPT-1使用post-Norm，这也使得GPT-2能够构建更大规模深度网络:

- Pre-Norm:
  - 在每个子层之前进行归一化
  - 有助于缓解梯度消失问题
  - 使模型更容易训练
  - 有助于保持梯度稳定性
- Post-Norm:

- 在每个子层之后进行归一化
- 有助于缓解梯度消失问题
- 使模型更容易训练
- 有助于保持梯度稳定性

GPT-2在预训练中使用了40GB的WebText数据集，这些经过精心筛选的网络文本显著提升了模型的语言理解能力。模型展现出了优秀的零样本学习能力，无需微调就能适应多种下游任务，这大大提升了模型的实用性。

GPT-3

GPT-3在前两代基础上大幅提升了模型规模，参数量达到1750亿，并展现出了卓越的上下文学习能力。模型使用了近1TB的多源数据进行预训练，包括Common Crawl、WebText、BookCorpus和Wikipedia等，这些数据经过严格筛选，确保了质量和多样性。通过简单的任务描述或少量示例，GPT-3就能完成各种下游任务，展现出强大的任务泛化能力。

GPT-3展现了很强的上下文学习(in-context learning)能力:通过在提示(prompt)中提供少量示例或任务说明,无需模型微调就能高效完成翻译、问答、摘要等多种自然语言处理任务。这种能力大大提升了模型的实用性和灵活性。

Instruct-GPT

OpenAI基于GPT-3开发了多个特定任务的衍生模型。其中最具代表性的是InstructGPT，该方法通过人类反馈强化学习（RLHF）提升了模型对用户指令的响应能力。RLHF通过有监督微调、奖励模型训练和强化学习优化三个步骤，使模型生成的内容更符合人类期望。

为了解决RLHF计算成本高的问题，斯坦福大学提出了直接偏好优化（DPO）算法。DPO直接利用人类偏好数据训练模型，简化了训练流程，提高了效率。虽然在处理复杂偏好时可能略逊于RLHF，但其计算效率优势使其得到广泛应用。大量Decoder-only模型都采用DPO算法来做偏好微调。

ChatGPT 以及 GPT-4

论文:GPT-4 Technical Report

链接:https://arxiv.org/pdf/2303.08774

2022年11月发布的ChatGPT凭借其卓越的对话能力，开创了LLMaaS（LLM as a Service）服务模式。2023年3月发布的GPT-4在复杂任务处理和多模态支持方面实现了重大突破。2024年5月发布的GPT-4o在响应速度、多模态处理和多语言支持等方面取得了显著进展。

GPT-4（Generative Pre-trained Transformer 4）是一个强大的多模态模型。与GPT-3.5/ChatGPT相比，GPT-4不仅可以处理文本输入，还能接收和理解图像输入，但输出仍保持为自回归的文本生成形式。

在架构设计上，GPT-4采用了创新的混合专家系统，由16个专业模型协同工作。其中Attention机制拥有55B参数，而MoE(Mixture of Experts)系统包含16个专家模型，每个专家拥有111B参数，总计120层Transformer结构。模型通过智能路由算法为每个token选择最合适的两个MLP专家进行处理，支持8k的序列长度。

专家混合（Mixture of Experts，MoE）是一种突破传统模型规模限制的创新方法。其核心思想是将复杂任务分解给多个专门的子模型（专家），每个专家负责处理特定类型的输入或特征。这种设计类似于现实世界中的专业分工:

- 就像建筑项目需要建筑师负责设计、结构工程师确保安全、装修设计师优化体验一样
- MoE系统中的每个专家都专注于特定的数据模式或任务类型

- 通过专家间的协作，系统能够高效处理各种复杂场景，实现更优的整体性能

LLaMA系列

LLaMA 是 Large Language Model Meta AI 的缩写，是 Meta AI 开发的开源大语言模型系列，以特定的许可证向学术界和部分商业用户开放模型权重，推动了大语言模型的共创和知识共享。在架构上，LLaMA借鉴了GPT系列的设计理念并进行创新优化。与GPT系列不同，LLaMA更注重通过扩大预训练数据规模而非模型规模来提升性能。目前Meta AI已推出三代LLaMA模型，并衍生出丰富的生态系统。

LLaMA-1

Meta AI于2023年2月推出首个LLaMA模型。遵循Chinchilla扩展法则，LLaMA1采用"小模型+大数据"策略，用更小的参数规模实现更快的推理速度。其预训练数据来自Common Crawl、C4、Github、Wikipedia等多个来源，总量达5TB。

在架构上，LLaMA1基于GPT架构并做出多项优化:

- 归一化层使用RMSNorm替代LayerNorm，RMSNorm的计算公式为:

$$\text{RMSNorm}(x) = \frac{x}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} \tag{5}$$

```
def rmsnorm(x, weight, eps=1e-5):
    return weight * (x / torch.sqrt(x.pow(2).mean(dim=-1, keepdim=True) + eps))
```

- 使用旋转位置编码(RoPE)提升word embedding质量；
- 在注意力模块中采用SwiGLU激活函数替代RELU,避免ReLU在负值区域梯度为0的问题，并使训练更稳定；SwiGLU的计算公式为:

$$\text{SwiGLU}(x) = \sigma(xW_1 + b_1) \cdot xW_2 + b_2 \tag{6}$$

```
def swiglu(x, weight1, weight2, bias1, bias2):
    return torch.nn.functional.silu(x @ weight1 + bias1) * (x @ weight2 + bias2)
```

- 在全连接前馈模块中使用Pre-Norm层正则化策略。

计算过程中，SwiGLU先通过线性变换将输入映射到更高维空间，再使用门控机制进行非线性变换，最后映射回原始维度。这种设计既保持了计算效率，又提升了模型表达能力。LLaMA1提供了7B、13B、32B和65B四个版本。

LLaMA-2

2023年7月发布的LLaMA2进一步扩充训练数据至7TB，并引入人类反馈强化学习。模型先用大规模指令数据进行监督微调，再通过RLHF奖励模型和PPO算法优化。LLaMA2 在保持前代架构的基础上，进行了进一步的优化和改进，并在 34B 和 70B 版本中引入分组查询注意力（GQA）机制，通过让多个查询共享键值对来提升计算效率。

LLaMA-3

2024年4月推出的LLaMA3将预训练数据扩充至50TB，是LLaMA2的7倍。新增大量代码数据增强逻辑推理能力，并涵盖30多种语言的非英文数据提升跨语言处理能力。LLaMA3延续了前代架构，但将分词字典扩大三倍，显著提升了推理效率和多语言处理能力。即便是8B参数版本也超越了LLaMA2-70B的性能，而70B版本更是在多项任务上超越GPT-4。LLaMA3的8B和70B版本均采用GQA机制，在保持参数规模不变的情况下实现了性能的质的飞跃。

在后续的LLaMA－3.1系列中，引入了高级推理和扩展上下文长度。LLaMA－3.2系列中，又引入了多峰值功能和针对移动设备的最优化。