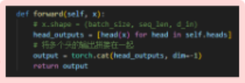
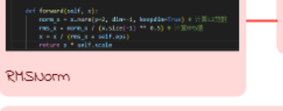
从零构建大模型

从头实现 GPT 模型

- 在实现大语言模型的步骤
  - 数据的准备以及采样
  - 注意力机制的实现
    - 实现简单的单头注意力机制
    - 使用单头注意力机制实现多头注意力机制
      - 简单实现

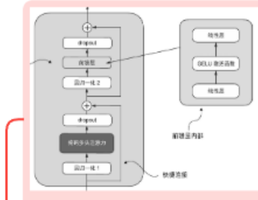
        

      - 使用矩阵变换实现

        
  - 大语言模型的架构
  - 预训练：完成模型架构的搭建以及数据准备可以进行预训练
- GPT 大语言模型的组成部分

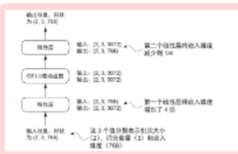
  - transformerBlock
    - 词元化对应第一章内容
    - 嵌入以及位置编码，直接采用torch.nn.Embedding
    - 掩码多头注意力机制和上述对应
    - 层归一化、残差链接：减少梯度消失等问题
    - FFN前馈层，将低维空间的数据映射到高维之后再映射回到低维空间，增加更多信息的捕捉
    - 激活函数采用GELU：
  - 线性输出层
- 实现细节

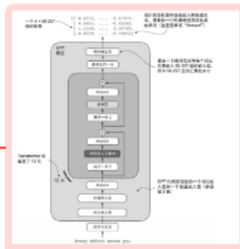
- 将transformer各个部分拆开然后在组装

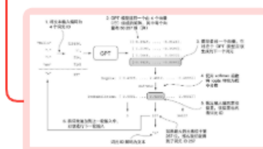
  
  - 层归一化
    - 常规使用的层归一化为LayerNorm

    - RMSNorm

      - 相比较于LayerNorm减少计算量，适合大模型的推理
    - 相同的可以使用torch自带的库实现而避免自己实现的效果不好
  - GELU激活函数
    - $\mathrm{GELU}(x) \approx 0.5 \cdot x \cdot \left(1 + \tanh\left[\sqrt{\frac{2}{\pi}} \cdot \left(x + 0.044\,715 \cdot x^3\right)\right]\right)$
    - 相比于简单的ReLU激活函数，能够提升模型的性能
  - 前馈神经网络

    
    - 输入输出的形状保持不变，作用：它通过第一个线性层将嵌入维度扩展到了更高的维度，扩展之后，应用非线性GELU 激活函数，然后通过第二个线性变换将维度缩回原始大小。这种设计允许模型探索更丰富的表示空间。
  - 残差链接
    - 背景：梯度消失是在训练过程中，梯度在反向传播时逐渐变小，导致早期网络层难以有效训练。
    - 目的：最初用于计算机视觉中的深度网络（特别是残差网络），目的是缓解梯度消失问题。
- 完成所有部件之后就可以开始组装

  
- 生成文本

  
  - 生成过程
    - 输入一段文本，通过嵌入层以及其他各个模块输出最后的概率分布
    - 根据最后一个时间步的输出使用不同的采样方案获取对应的tokenID，根据输出的tokenID查询对应的字符，显示输出
      - 实现topk采样以及温度调控

        - 过程
          - 根据topk的大小获取最后时间步输出的前k个最大概率分布，将其他的都用-inf替代
          - 对替换之后的概率分布使用softmax获取概率
          - 最后使用随机采样，对这k个tokenID进行随机采样
      - 简单实现（贪心策略或topk=1）
        - 主要实现就是利用输出的概率分布对应的最大概率的tokenID作为下一个词输出
    - 拼接该输出，最后作为输入重新进入循环，直至输出结束token或者是达到最大输出之后停止输出

Presented with xmind