NLP模型汇报

陈钶杰
专业:计算数学

May 23, 2023

目录

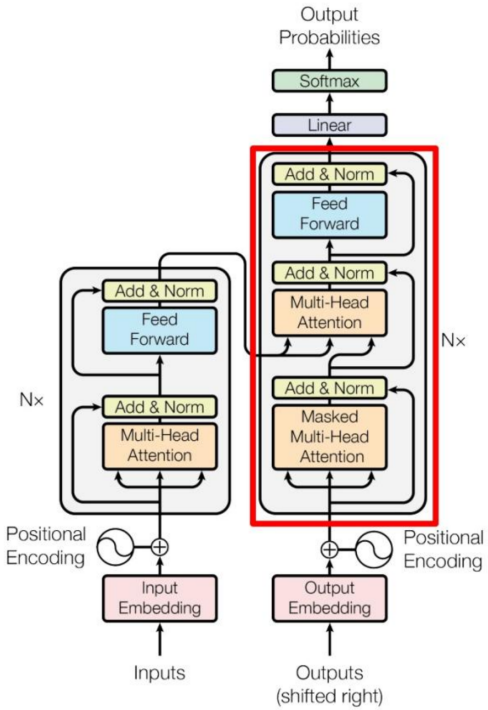
1. transformer模型
2. BERT模型
3. GPT模型
4. Prompt learning 模型

tranformer模型

- 介绍：
  - Transformer是一种基于注意力机制的神经网络架构，最初用于处理序列到序列（Sequence-to-Sequence）的任务，如机器翻译。
  - Transformer在处理长序列时具有更好的并行性和捕捉长距离依赖关系的能力。

transformer模型

- 模型的基本框架



- 进行中英文翻译举例(使用pytorch):
  预训练过程:
  - input:中文词汇表,英文词汇表,中英文句子对应表
    output:模型的所有参数

  具体流程:
  - Encoder
    1. 使用torch的内置嵌入函数，将输入的中文句子中的每个单词转换为一个向量，并结合单词在句子中的位置向量，得到最终的信息表示。
    2. 使用多头注意力机制（MultiHeadAttention），核心是将单词的表示向量拆分成多个子空间。通过使用矩阵运算，该注意力机制可以有效地捕捉句子中不同单词之间的关系。这种关系可以帮助模型更好地理解上下文信息，从而提高模型在处理自然语言任务中的性能。
    3. 将上一步得到的结果输入前馈神经网络（Feed-Forward Neural Network），这一步在最终效果的呈现中起到关键作用。前馈神经网络通过多层感知机（Multi-Layer Perceptron, MLP）结构，对输入进行非线性变换和映射。这种变换能够增强特征的表达能力和复杂性，从而提高模型的性能和表现。

具体流程:

- Deconder
  1. 使用torch的内置嵌入函数，将输入的英文句子中的每个单词转换为一个向量，并结合单词在句子中的位置向量，得到最终的信息表示。
  2. 使用多头注意力机制(MultiHeadAttention),但是这边与编码器中的多头注意力机制相比,有些不同,即解码器有两种不同的多头注意力机制.
  3. 第一是需要MASK,希望通过MASK使得一句话只受前面信息的影响,而不会受到后面内容的影响.
  4. 第二种不同之处在于第二层的多头注意机制通过使用编码器中的值（V）、键（K）以及解码器中第一层得到的查询（Q）,来建立中英文之间的关系。这一步可以帮助模型更好地捕捉并联系中英文之间的语义和上下文关系，从而提高模型在翻译和跨语言任务中的性能。
  5. 将上一步得到的结果输入前馈神经网络
- 预测输出单词
  1. 使用softmax得到句中每个单词对应词汇表中所有单词的概率.

使用的损失函数:交叉熵

- 对于多分类问题，交叉熵损失函数的定义如下：

$$L_{CE} = -\sum_{i=1}^{N} y_i \log(p_i)$$

其中,y_i是真实标签的one-hot编码,p_i是模型对类别i的预测概率。

迭代计算

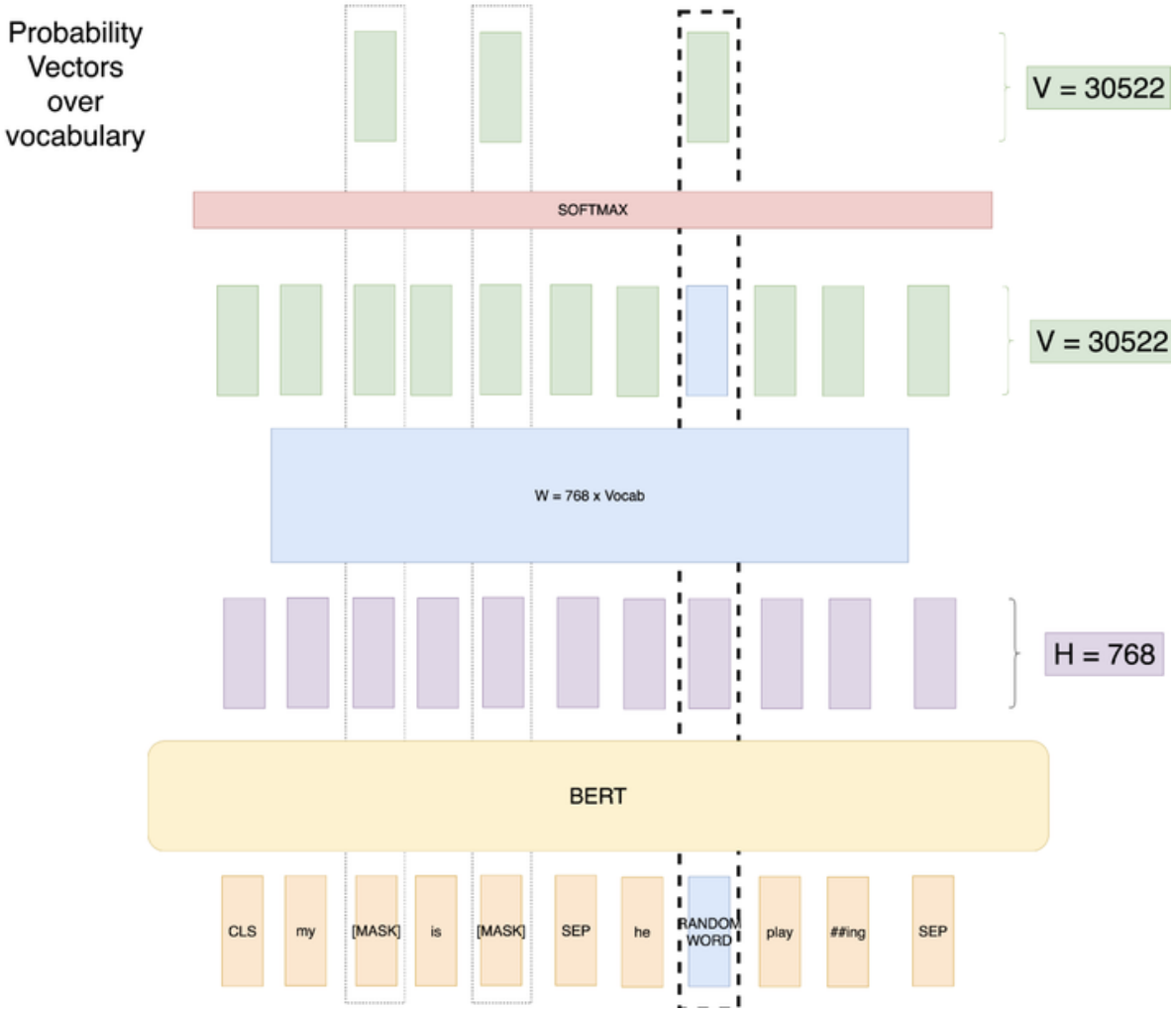
- 在这个过程中，模型会根据输入句子生成一个输出序列，然后将其与目标输出进行比较。通过计算交叉熵损失函数，可以衡量模型的输出与目标之间的差异。接下来，误差会通过反向传播算法传递回模型，更新模型的参数，以使输出逐渐接近目标。
- 通过多次迭代训练，模型会不断调整参数，提高其性能和准确度。最终目标是使模型生成的句子与目标输出尽可能一致，并且交叉熵损失尽可能小。这样，模型就能够学习到语言的结构和语义，并生成更准确和有意义的句子。

BERT模型

- 介绍：
  - BERT是一个能处理各种自然语言任务(NLP)的通用架构,该模型主要目标是通过在大规模无标签文本上进行预训练,学习通用的语言表示.
  - 主要功能:
    1. 遮蔽语言模型(完型填空):在遮蔽语言模型任务中，BERT会将整个输入句子输入模型，并随机遮蔽一部分单词。模型需要通过上下文中的其他单词来预测被遮蔽的单词。这样的训练方式使得模型能够学习到上下文间的依赖关系，以及对上下文进行推理和填充的能力。
    2. 下个句子预测:BERT会接收两个连续的句子作为输入，并判断它们是否为原始文本中的下一句。这样的任务有助于模型学习到句子间的关联和逻辑推理能力。
  - 此模型是基于tranformer的编码器部分.将一个句子的单词随机遮挡15%,然后经过多层多头注意力机制,以及前馈神经网络后得到被掩盖词的最有可能值.计算交叉熵,误差通过反向传播算法传递回模型，更新模型的参数，以使输出逐渐接近目标。

BERT模型

基本框架:


Probability
Vectors
over
vocabulary
V = 30522
SOFTMAX
V = 30522
W = 768 x Vocab
H = 768
BERT
CLS
my
[MASK]
is
[MASK]
SEP
he
RANDOM WORD
play
##ing
SEP

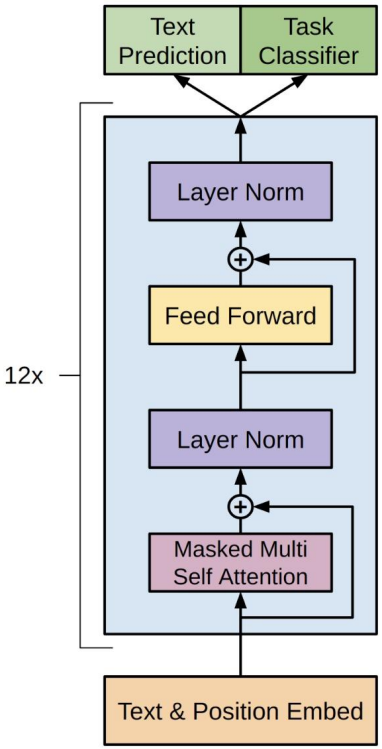
GPT模型

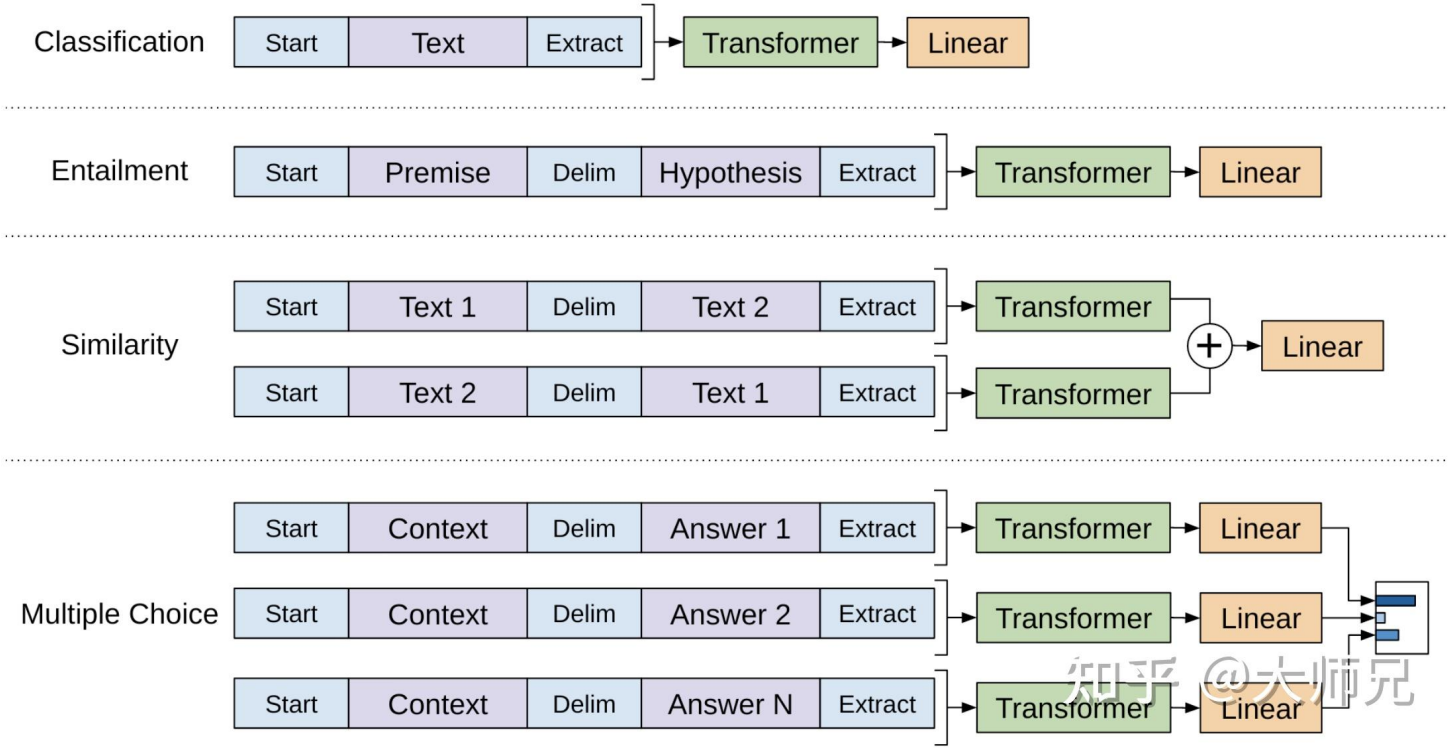
- 介绍：
  - GPT与BERT一样也是一种预训练模型，与BERT不同的是，GPT是一个单向的语言模型，只能根据前面的文本生成下一个词，且GPT使用的是Transformer的Decoder结构。在大量没有标号的数据上训练出一个预训练模型，然后少量有标号的数据上微调训练一个中下游任务的模型。在微调的时候构造与任务相关的输入，就可以很少地改变模型的架构.
  - gpt模型的主要功能:文本分类,文本预测.
  - gpt模型的学习的目标函数如下:

$$L_1(token_1, ..., token_n) = \sum_{n=K}^{N} p(token_n | token_{n-k}, ..., token_{n-1}; \theta)$$

GPT模型

基本框架:





GPT模型

GPT的微调模型

1. Fine-tuned（微调）是指在预训练模型的基础上，通过进一步的训练和调整，使模型适应特定任务或数据集的过程。预训练模型（如GPT）通常是在大规模无标注文本上进行训练，学习到了丰富的语言知识和模式。但是，这些预训练模型并不是针对特定任务设计的，因此需要通过Fine-tuning来对模型进行定制化，使其在特定任务上表现更好。
2. 经过预训练的模型并不能做到情感分类，即给定一段文本，判断其是积极的还是消极的。我们可以使用GPT模型进行Fine-tuning来解决这个任务。
3. fine-tuning的训练阶段一般是将基础模型与任务特定的分类层相结合,通常,我们在基础模型之后添加一个全连接层(MLP),后使用交叉熵损失函数来计算模型的预测结果与真实标签之间的差异。

如何进行更好的微调?

Fine-tuned潜在的缺点

1. 数据需求:GPT的Fine-tuning通常需要大量的任务特定训练数据。为了在特定任务上获得良好的性能，需要提供大规模、高质量的标注数据进行Fine-tuning。这会增加数据收集和标注的成本和工作量。
2. 灵活性：GPT的Fine-tuning在预训练模型的基础上进行微调的。模型结构和预训练权重是固定的，只能通过微调任务特定层来适应特定任务。这可能限制了模型的灵活性和能力，特别是对于某些复杂或特殊的任务。

考虑Prompt learning 替代Fine-tuning

1. 简单易用：相比于Fine-tuning需要大规模标注数据和复杂的模型调整，Prompt learning更加简单和直观，只需要设计和选择合适的提示。
2. 灵活性和可解释性：Prompt learning模型可以针对特定任务和领域进行定制，提高模型的适应性和泛化能力。提示是人为设计的，可以使得模型的预测结果更加可解释和可理解。

Prompt learning 模型

定义任务和目标：

- 提示学习模型的构建过程(以情感分类为例):
  1. 定义任务和目标：首先，确定任务的类型和具体目标。例如，文本生成、翻译、问答等。明确希望模型能够对句子进行情感分类
  2. 将提示与输入结合：在生成过程中，将提示与输入文本或上下文结合起来作为模型的输入。这样，模型能够通过引导提示的方式更好地理解任务和上下文，并生成符合预期的输出。例如将一句话添加提示语句后有"I like the films,it was [mask](提示语句)"
  3. 微调模型：使用提示和相应的数据集对预训练模型进行微调，以适应特定任务和提示。微调过程中，可以使用与任务相关的数据对模型进行训练，以提高模型在任务中的性能。
  4. 生成输出：使用微调后的模型，根据给定的输入和提示进行文本生成。模型会根据上下文和任务要求生成输出文本，并且通过提示来引导生成的过程，以满足特定的预期结果。

Prompt learning 模型

Prompt learning模型的重难点：

- 添加的提示("it was [mask]")也称为pattern(template).
  而如何构建合适的pattern是Prompt-tuning 的研究热点之一.
  - 现有的挑选合适的Pattern方法
    1. 人工构建（Manual Template）：在前文已经描述过，不再详细说明；
    2. 启发式法（Heuristic-based Template）：通过规则、启发式搜索等方法构建合适的模板；
    3. 生成（Generation）：根据给定的任务训练数据（通常是小样本场景），生成出合适的模板；
    4. 词向量微调（Word Embedding）：显式地定义离散字符的模板，但在训练时这些模板字符的词向量参与梯度下降，初始定义的离散字符用于作为向量的初始化；
    5. 伪标记（Pseudo Token）：不显式地定义离散的模板，而是将模板作为可训练的参数；

谢谢老师和同学的聆听!