Data Mining and Optimization

Lecture 10: Pre-Trained Language Models

Liu Yang

Nanjing University

Spring, 2025

Table of Contents

什么是预训练语言模型？

- 预训练语言模型(Pre-Trained Language Models, PLM)指提前经过大规模数据训练的语言模型，包括早期的以Word2Vec为代表的静态词向量模型，以及基于上下文建模的ELMo等动态词向量模型；
- 在2018年，以GPT和BERT为代表的基于深层Transformer的大语言模型出现后，预训练语言模型开始广为人知，标志着自然语言处理进入新的时代。

预训练语言模型的特点

- 大数据：获取足够多的大规模文本数据是训练一个好的预训练语言模型的基础，数据必须“保质”和“保量”
  - “保质”：预训练预料的质量要尽可能高，避免混入过多的低质量预料；
  - “保量”：预训练预料的规模要尽可能大，从而获取丰富的上下文信息；
- 大模型：数据规模和模型规模是正相关的，大模型才能确保完全涵盖大数据中丰富的语义信息，这里的“大”通常指的是模型的“参数量大”，模型设计通常考虑两个方面
  - 模型具有较高的并行程度，以弥补大模型带来的训练速度下降问题；
  - 模型能够捕捉并构建上下文信息，以充分挖掘大数据文本中丰富的语义信息；

  基于Transformer的神经网络模型具有较高的并行程度，其中多头注意力机制能够有效捕获不同词之间在不同维度上的关联程度，因此成为目前构建预训练语言模型的最佳选择；
- 大算力：训练预训练语言模型通常依赖于图形处理单元(Graphics Processing Unit, GPU)和张量处理单元(Tensor Processing Unit, TPU)等支持并行计算的硬件设备。

Table of Contents

GPT

- OpenAI公司于2018年提出了一种生成式预训练(Generative Pre-Training, GPT)模型，提出了“生成式预训练+判别式任务微调”的自然语言处理新范式
  - 生成式预训练：在大规模文本数据上训练一个高容量的语言模型，从而学习更加丰富的上下文信息；
  - 判别式任务微调：将预训练好的模型适配到下游任务中，并使用有标准数据学习判别式任务。

无监督预训练

- GPT的整体结构是一个基于Transformer解码器的单向语言模型，即从左至右对输入文本建模；
- 给定输入序列$x = \{x_1, \cdots, x_T\}$，GPT预训练的损失函数为

$$-\sum_{t=1}^{T} \log P(x_t | x_1, \cdots, x_{t-1}; \theta), \tag{1}$$

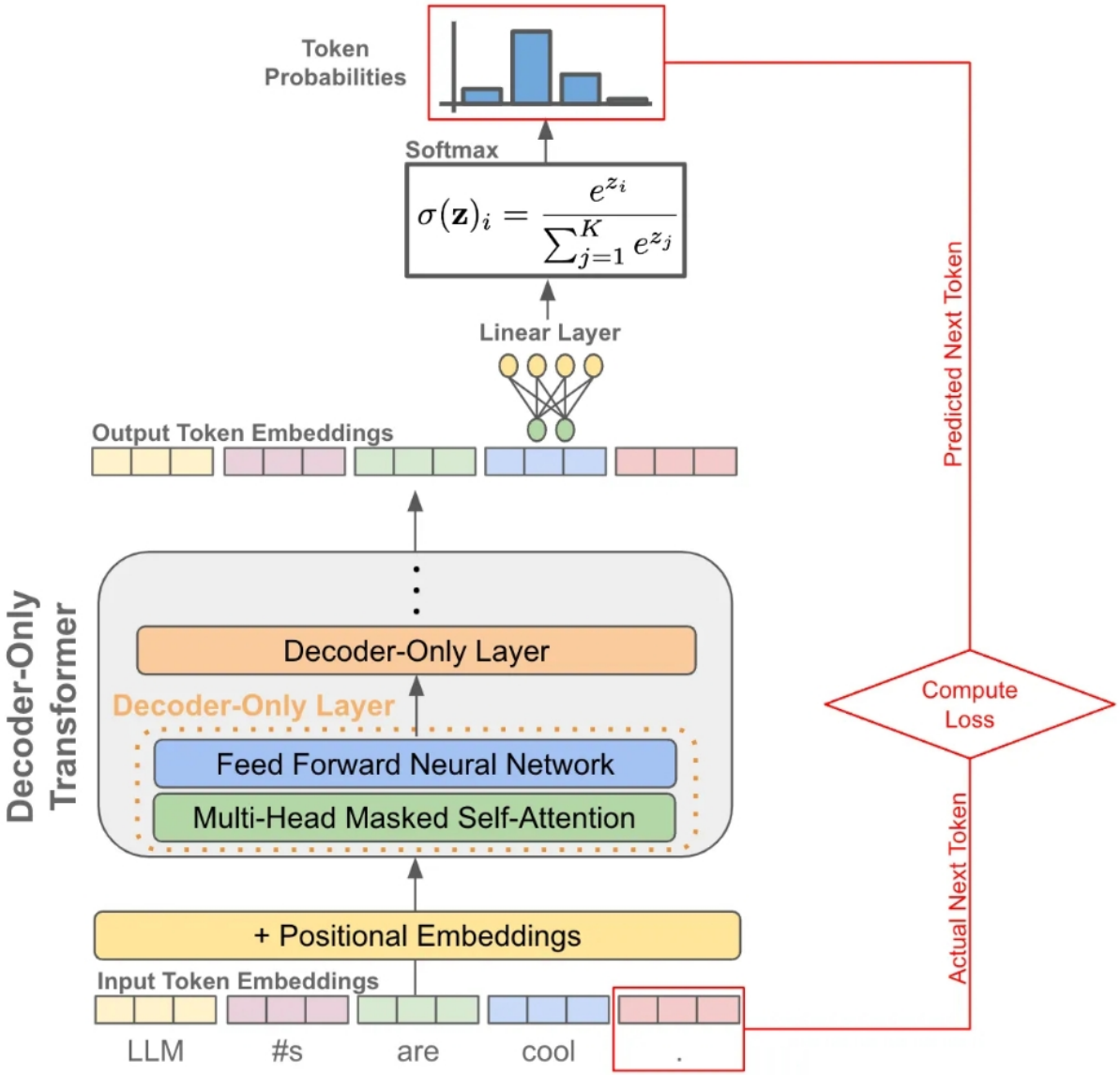
其中，θ表示神经网络模型的参数，(1)表明，GPT基于历史词汇预测当前时刻的词x_t，即"Next-token Prediction"；

- 通过以下方式计算建模概率P

$$\begin{aligned} h^{[0]} &= e_{x'} W^e + W^p \\ h^{[l]} &= Transformer - Decoder(h^{[l-1]}), \forall l \in \{1, \cdots, L\} \\ P(x) &= softmax(h_T^{[L]} W^L), \end{aligned} \tag{2}$$

其中，$e_{x'} \in R^{(t-1)\times|V|}$表示$x_1, \cdots, x_{t-1}$的独热向量，$W^e \in R^{|V|\times d_{model}}$表示词向量矩阵，$W^p \in R^{(t-1)\times d_{model}}$表示位置向量矩阵，$W^L \in R^{d_{model}\times|V|}$表示前馈神经网络的参数矩阵，$L$表示Transformer的层数。

Next-token Prediction



有监督微调

- 在预训练阶段，GPT利用大规模数据训练出基于深层Transformer的语言模型，掌握了文本的通用语义表示，微调(Fine-tuning)的目的是在通用语义表示的基础上，根据下游任务的特性进行领域适配，使之与下游任务的形式更加契合，以获得更好的下游任务应用效果；
- 微调通常是由有标注数据进行训练和优化的，假设标注数据为S，其中每个样例的输入是$x = x_1, \cdots, x_T$构成的长度为T的文本序列，与之对应的标签为y。首先将文本序列输入预训练的GPT中，获取最后一层的最后一个词对应的隐状态$h_T^{[L]}$，紧接着，将该隐状态输入一个全连接层变换，预测最终的标签

$$P(y|x_1, \cdots, x_T) = softmax(h_T^{[L]} W^y), \tag{3}$$

其中，$W^y \in R^{d_{model} \times k}$表示全连接层参数，$k$表示标签类别数；

- 通过优化以下损失函数微调

$$-\sum_{(x,y)\in S} \log P(y|x_1, \cdots, x_T). \tag{4}$$

单句文本分类

- 单句文本分类是最常见的自然语言处理任务之一，其输入由单个文本构成，输出由对应的分类标签构成；
- 假设输入为$x = x_1, \cdots, x_T$，单句文本分类的样例将通过如下形式输入GPT中

$$< s > x_1 \cdots x_T < e >, \tag{5}$$

其中，$< s >$表示开始标记，$< e >$表示结束标记。

文本蕴含

- 文本蕴含的输入由两段文本构成，输出由分类标签构成，用于判断两段文本之间的蕴含关系。输入的第一段文本叫前提(Premise)，第二段文本叫假设(Hypothesis)；
- 假设文本蕴含的样例分别为$x^{(1)} = x_1^{(1)}, \cdots, x_T^{(1)}$和$x^{(2)} = x_1^{(2)}, \cdots, x_{T'}^{(2)}$，其将通过如下形式输入GPT中

$$< s > x_1^{(1)} \cdots x_T^{(1)} \$ x_1^{(2)} \cdots x_{T'}^{(2)} < e >, \tag{6}$$

其中，$表示分隔标记，用于分隔两段文本。

文本相似度

- 计算文本相似度任务的输入也由两段文本构成，但两段文本之间不存在顺序关系；
- 假设文本相似度的样例分别为$x^{(1)} = x_1^{(1)}, \cdots, x_T^{(1)}$和$x^{(2)} = x_1^{(2)}, \cdots, x_{T'}^{(2)}$，其将通过如下形式输入GPT中，得到两个相应的隐状态表示，最终将这两个隐状态表示相加，并通过一个全连接层预测相似度

$$\begin{aligned} &< s > x_1^{(1)} \cdots x_T^{(1)} \$ x_1^{(2)} \cdots x_{T'}^{(2)} < e > \\ &< s > x_1^{(2)} \cdots x_{T'}^{(2)} \$ x_1^{(1)} \cdots x_T^{(1)} < e > . \end{aligned} \tag{7}$$

选择型阅读理解

- 选择型阅读理解任务是让机器阅读一篇文章，并且需要从多个选项中选择出问题对应的正确选项，即需要将（篇章，问题，选项）作为输入，以正确选项编号作为标签；
- 假设篇章为$p = p_1, \cdots, p_n$，问题为$q = q_1, \cdots, q_m$，第i个选项$c^{(i)} = c_1^{(i)}, \cdots, c_k^{(i)}$，并假设$N$为选项个数，其将通过如下形式输入GPT中

$$
\begin{aligned}
&< s > p_1 \cdots p_n \$ q_1 \cdots q_m \$ c_1^{(1)} \cdots c_k^{(1)} < e > \\
&< s > p_1 \cdots p_n \$ q_1 \cdots q_m \$ c_1^{(2)} \cdots c_k^{(2)} < e > \\
&\qquad\qquad \vdots \\
&< s > p_1 \cdots p_n \$ q_1 \cdots q_m \$ c_1^{(N)} \cdots c_k^{(N)} < e >,
\end{aligned} \tag{8}
$$

通过GPT建模得到对应的隐状态表示，并通过全连接层得到每个选项的得分，最终将N个选项的得分拼接，通过softmax函数得到归一化的概率，并通过交叉熵损失优化模型参数。

Fine-tuning GPT