5-3 Transformers

王中雷

厦门大学王亚南经济研究院和经济学院, 2025

内容摘要

1. 研究动机
2. 点积自注意力（Dot-product self-attention）
3. 位置编码
4. 多头自注意力
5. Transformer 层
6. Transformer 模型

研究动机

1．循环神经网络模型在序列分析方面取得了成功

- 然而，门控机制在捕捉长距离相关性方面表现较弱
- 此外，相关模型无法并行，导致其计算效率较低

2．Vaswani et al. (2017) 针对机器翻译任务，提出了多头点积自注意力框架

- 不再依赖循环神经网络模型结构
- 其通过自注意力来提取前后文信息
- 该框架适用于并行计算，因此具有较高的计算效率
- 该框架被广泛应用于深度学习模型中，用于提取序列信息

Transformers

1. Transformer 是一个新的模型框架
   - 其通过共享模型参数，处理不同长度的输入信息
   - 基于词嵌入，该框架可有效处理不同词向量之间的相关性
2. 其核心是点积自注意力（dot-product self-attention）机制
3. 考虑一句话："I have arrived at Beijing from Xiamen."
   - 假设该句话分词化后对应的词向量$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_8$
4. 为了符号简单，我们用$\boldsymbol{x}_i$ 来代替前面讨论的词向量符号$\boldsymbol{x}^{<i>}$

点积自注意力机制

1. 对于每个词向量$\boldsymbol{x}_i$，我们首先计算

   - 查询（Query）：

$$\boldsymbol{q}_i = \boldsymbol{b}_q + \boldsymbol{W}_q \boldsymbol{x}_i$$

   - 键（Key）：

$$\boldsymbol{k}_i = \boldsymbol{b}_k + \boldsymbol{W}_k \boldsymbol{x}_i$$

   - 值（Value）：

$$\boldsymbol{v}_i = \boldsymbol{b}_v + \boldsymbol{W}_v \boldsymbol{x}_i$$

2. 共享模型参数包括：$\{\boldsymbol{b}_q, \boldsymbol{W}_q, \boldsymbol{b}_k, \boldsymbol{W}_k, \boldsymbol{b}_v, \boldsymbol{W}_v\}$

3. 模型参数的维度根据如下规则确定：

   - 向量$\boldsymbol{x}_i$ 和$\boldsymbol{v}_i$ （一般）具有相同维度

   - 向量$\boldsymbol{q}_i$ 和$\boldsymbol{k}_i$ 维度一致，但可与$\boldsymbol{x}_i$ 不同

点积自注意力机制

1. 对于$i \in \{1, \ldots, 8\}$, $\boldsymbol{x}_i$ 基于如下点积自注意力机制被更新：

- 对于$l \in \{1, \ldots, 8\}$,

$$a_{il} = a(\boldsymbol{q}_i, \boldsymbol{k}_l)$$

  - $a(\boldsymbol{x}, \boldsymbol{y})$：为对齐函数（alignment function）(Bahdanau et al., 2015)，用于衡量$\boldsymbol{x}$ 和$\boldsymbol{y}$ 的相关程度
  - Transformer 用$a(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}$

2. 将$\{a_{il} : l = 1, \ldots, 8\}$ 归一化，我们得到

$$w_{il} = \frac{\exp(a_{il})}{\sum_{h=1}^{8} \exp(a_{ih})}$$

点积自注意力机制

1. 对于$i \in \{1, \ldots, 8\}$, $\boldsymbol{x}_i$ 基于如下点积自注意力机制被更新（接上一页）：

- 将$\boldsymbol{x}_i$ 更新为

$$\boldsymbol{x}_i' = \sum_{l=1}^{8} w_{il} \boldsymbol{v}_l$$

  - 在更新$\boldsymbol{x}_i$ 时，w_{il} 衡量我们需将多少“注意力”置于$\boldsymbol{v}_l$

2. 针对每一个i，计算$\{w_{il} : l = 1, \ldots, 8\}$ 用于得到更新后的特征向量

$$\{\boldsymbol{x}_i' : i = 1, \ldots, 8\}$$

例子

计算：更新后的参数 $\boldsymbol{x}_7$

更新： $\boldsymbol{x}'_1$ $\boldsymbol{x}'_2$ $\boldsymbol{x}'_3$ $\boldsymbol{x}'_4$ $\boldsymbol{x}'_5$ $\boldsymbol{x}'_6$ $\boldsymbol{x}'_7$

归一化：

自注意力：

$\boldsymbol{q}_1, \boldsymbol{k}_1, \boldsymbol{v}_1$ $\boldsymbol{q}_2, \boldsymbol{k}_2, \boldsymbol{v}_2$ $\boldsymbol{q}_3, \boldsymbol{k}_3, \boldsymbol{v}_3$ $\boldsymbol{q}_4, \boldsymbol{k}_4, \boldsymbol{v}_4$ $\boldsymbol{q}_5, \boldsymbol{k}_5, \boldsymbol{v}_5$ $\boldsymbol{q}_6, \boldsymbol{k}_6, \boldsymbol{v}_6$ $\boldsymbol{q}_7, \boldsymbol{k}_7, \boldsymbol{v}_7$

模型参数： $\boldsymbol{b}_q, \boldsymbol{W}_q, \boldsymbol{b}_k, \boldsymbol{W}_k, \boldsymbol{b}_v, \boldsymbol{W}_v$

词嵌入： $\boldsymbol{x}_1$ $\boldsymbol{x}_2$ $\boldsymbol{x}_3$ $\boldsymbol{x}_4$ $\boldsymbol{x}_5$ $\boldsymbol{x}_6$ $\boldsymbol{x}_7$

向量化

1．记

$$\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_7)^{\mathrm{T}} \in \mathbb{R}^{d \times 7}$$

$$\boldsymbol{V} = \boldsymbol{b}_q \mathbf{1}^{\mathrm{T}} + \boldsymbol{W}_q \boldsymbol{X}$$

$$\boldsymbol{K} = \boldsymbol{b}_k \mathbf{1}^{\mathrm{T}} + \boldsymbol{W}_k \boldsymbol{X}$$

$$\boldsymbol{V} = \boldsymbol{b}_v \mathbf{1}^{\mathrm{T}} + \boldsymbol{W}_v \boldsymbol{X}$$

$$\boldsymbol{X}' = (\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_7)^{\mathrm{T}} \in \mathbb{R}^{d \times 7}$$

2．则点积自注意力机制的向量化为

$$\boldsymbol{X}' = \boldsymbol{V} \cdot \mathrm{Softmax}(\boldsymbol{K}^{\mathrm{T}} \cdot \boldsymbol{Q})$$

位置编码

1．在建模时，自注意力机制并无法将词汇出现的顺序考虑其中

2．Vaswani et al. (2017) 提出用一个基于位置的矩阵$\boldsymbol{P}$ 加到词嵌入后的特征矩阵$\boldsymbol{X}$

$$\boldsymbol{X}_p = \boldsymbol{X} + \boldsymbol{P}$$

- $\boldsymbol{P}$：对位置信息进行编码
- $\boldsymbol{X}_p$：加上位置信息后的词嵌入矩阵

3．我们利用带有不同频率的正余弦函数得到位置信息矩阵$\boldsymbol{P}$

$$\boldsymbol{P}_{(2i),j} = \sin(j/10000^{2i/d}), \quad \boldsymbol{P}_{(2i+1),j} = \cos(j/10000^{2i/d})$$

- i: 词向量中的维度指标
- j: 分词所在位置指标

基于放缩的点积自注意力机制

1. 在计算结果$\boldsymbol{K}^{\mathrm{T}} \cdot \boldsymbol{Q}$中可能存在很大的值
2. 这将导致某一个特定的“值（value）”占比过重
3. 对应的模型将变得效率较低
4. 为了解决这个问题，我们考虑

$$\boldsymbol{X}' = \boldsymbol{V} \cdot \mathrm{Softmax}\left(\frac{\boldsymbol{K}^{\mathrm{T}} \cdot \boldsymbol{Q}}{\sqrt{d_q}}\right)$$

- d_q：“查询（query）”以及“键（key）”的维度
- 在统计的角度讲，为了数值稳定，我们除掉的是对应加和的“标准差”的阶

多头自注意力机制

1. 只基于一组“查询（queries）”、“键（keys）”和“值（values）”的模型过于简单
2. 我们可以考虑H个集合，用来提取不同信息
3. 记

$$\boldsymbol{X}'_h = \boldsymbol{V}_h \cdot \text{Softmax}\left(\frac{\boldsymbol{K}_h^{\mathrm{T}} \cdot \boldsymbol{Q}_h}{\sqrt{d_q}}\right) \quad (h = 1, \ldots, H)$$

- $\boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h$：第$h$个集合中的“查询（queries）”、“键（keys）”和“值（values）”

4. 我们将每个“头”的更新特征进行列拼接，得到最终更新后的特征

$$\boldsymbol{X}' = \boldsymbol{W}_o(\boldsymbol{X}_1^{\mathrm{T}}, \ldots, \boldsymbol{X}_H^{\mathrm{T}})^{\mathrm{T}}$$

- $\boldsymbol{W}_o$：用来融合不同“头”计算结果的（模型参数）权重项

说明

1. 优势：

- 并行：与循环神经网络不同，该结构可通过并行计算提升训练效率
- 长距离相关性：自注意机制能够简单地处理不同距离的相关信息
- 上下文理解：有些分词需要放在整个句子中进行分析才有意义，该机制能够让语言模型更加准确
- 可解释的权重：我们可以通过分析权重知道哪些分词对最终结果影响较大

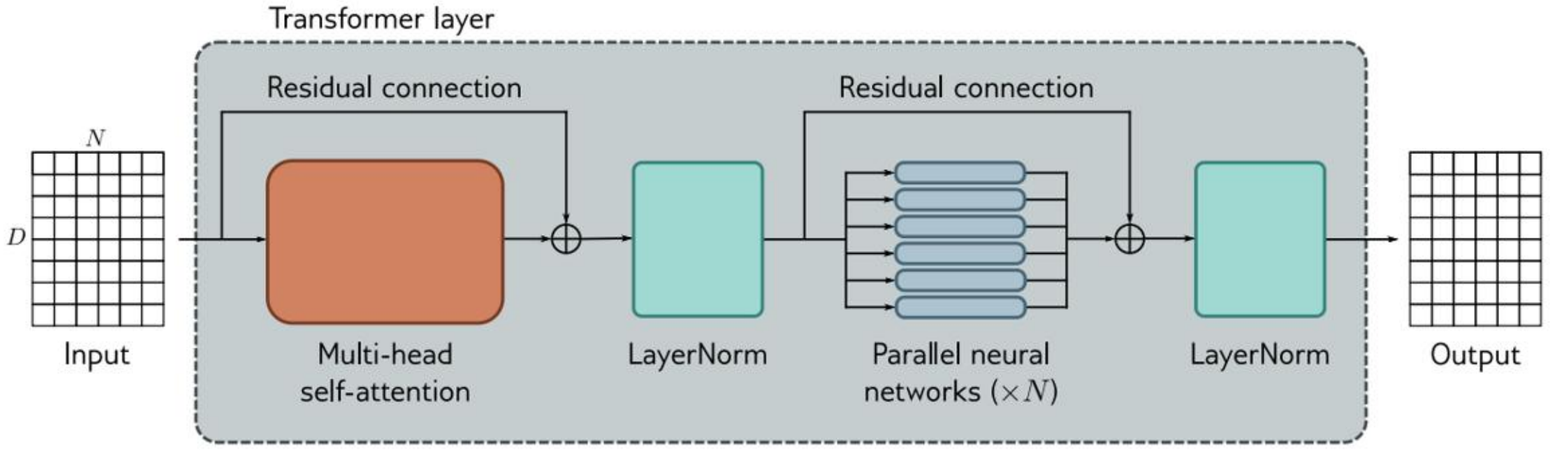
说明

1. 缺点：

- 计算成本高昂：对于具有较多分词数量 (n)，自注意力机制需要计算两两分词间的相互影响，这导致计算复杂度和内存需求的阶为$O(n^2)$.
- 对局部信息关注不足：其关注句子的全局信息；因此，当局部信息较为重要时，模型的分析结果可能不佳
- 过拟合：模型参数较多，当语料库规模较小时，容易产生过拟合

Transformer 层

1. 该层由一下部分组成：
   - 多头点积自注意力模块（提取分词之间的相关性）
   - 全连接神经网络（共享模型参数，对每个特征向量单独进行）
   - 残差连接
   - 层归一化
2. 右图来自于 Prince (2024) 中的图 12.7



Transformer 层

1．计算如下：

$$\boldsymbol{X} \leftarrow \boldsymbol{X} + \mathrm{MhSa}(\boldsymbol{X})$$

$$\boldsymbol{X} \leftarrow \mathrm{LayerNorm}(\boldsymbol{X})$$

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \mathrm{MLP}\boldsymbol{x}_i$$

$$\boldsymbol{X} \leftarrow \mathrm{LayerNorm}(\boldsymbol{X})$$

- $\boldsymbol{X}$：对于单一输入句子，融合了位置编码信息的词嵌入矩阵
- $\boldsymbol{x}_i$：矩阵$\boldsymbol{X}$ 的第i 列
- $\mathrm{MhSa}(\boldsymbol{X})$：多头点积自注意力模块
- $\mathrm{MLP}(\boldsymbol{X})$：全连接神经网络模块（共享模型参数）
- $\mathrm{LayerNorm}(\boldsymbol{X})$：层归一化模块

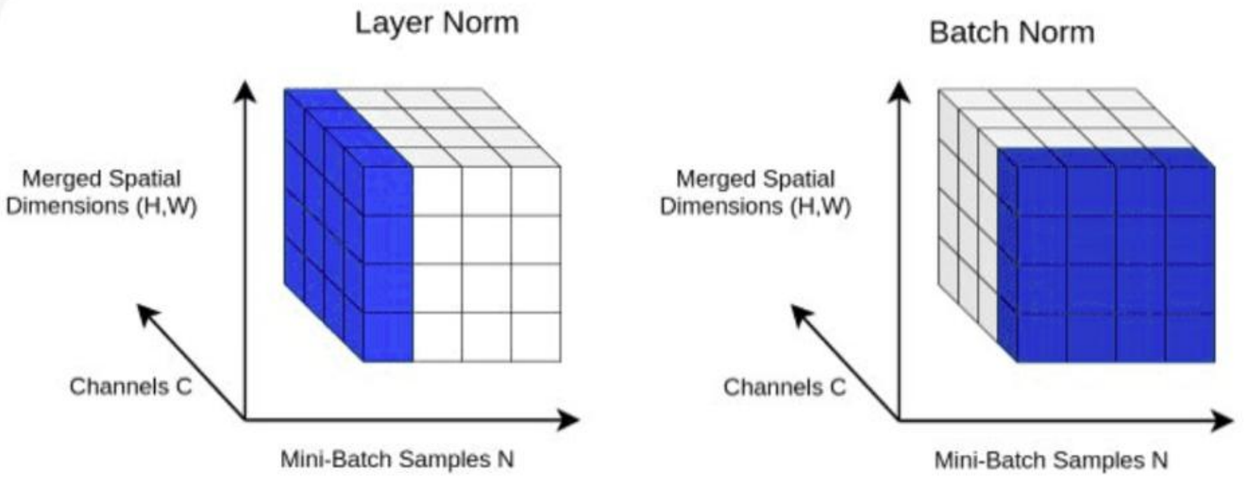
Transformer 层

1．回顾批量归一化

- 用于统一批次中，每个特征对应的所有训练样本信息进行归一化
- 由于不同句子长度不同，其不能用于自然语言处理

Transformer 层

1. 对于每一个输入句子，层归一化分别用于每个词向量的维度
2. 下图来自于https://theaisummer.com/normalization/
   - N：同一批次样本量
   - C：特征数量
   - H, W：向量化后的词向量的维度



Transformer 模型

1. 可能包括多个 transformer 模块
2. 编码-解码结构用于分析 “序列到序列” 的任务
   - 编码（文本到数字）：将文本进行数字化处理，并基于模型对其进行分析
   - 解码（数字到文本）：基于现有分词，预测下一个分词
3. 例子
   - BERT：一个编码模型
   - ChatGPT: 一个编码-解码模型
4. 更多内容请参加 Prince (2024) 的第 12 章