孙锐--论文总结《Attendtion Is All You Need》

1、论文信息（Summary of contributions.）

- **标题： Attendtion Is All You Need**
- **作者：** 来自谷歌的7位作者(Ashish Vaswani,Noam Shazeer等)
- **文章收录情况：**NIPS 2017 收录链接 https://arxiv.org/abs/1706.03762

2、选择原因和背景

Q1：为什么选择这一篇论文（Why this paper?）

A：因为我们的任务涉及到机器翻译，而我们所熟知的机器翻译技术最经典为RNN和LSTM的网络来实现。但是Transformer的出现，一种单纯使用注意力机制网络的模型在表现上一定程度超过了之前的两种模型。所以本次作业我们使用最为经典的Transformer网络来实现机器翻译任务。故论文选择了trasformer的出处《Attention Is All You Need》

Q2：文章的创作背景是怎样的？（Background.）

当下主流的语言建模和机器翻译模型为LSTM和GRU，但这些模型存在着一些问题。比如时序信息就比较依赖前一时刻的状态，导致模型无法并行计算。同时输入序列或是输出序列中的距离问题无法解决，而且这种模型会受到内存的很多限制。 同时模型中将来自两个任意输入或输出位置的信号关联起来所需的操作数随着位置之间的距离而增加，这使得模型学习远距离位置之间的依赖关系变得更加困难。

同时RNN和LSTM一些不可避免的缺陷，梯度消失。为了解决传统模型的一些缺点，提高训练效率和精度。作者提出了**Attention**机制，并设计了transformer网络结构。

3、文章主要贡献（Summary of contributions.）

文章主要贡献就是提出自注意力机制(Self-Attention)和transformer模型架构，这种网络成功解决了RNN和LSTM模型的梯度消失问题和输入或输出序列中的距离过远问题

Transformer中抛弃了传统的CNN和RNN，整个网络结构完全是由Attention机制组成。 作者采用Attention机制的原因是考虑到RNN（或者LSTM，GRU等）的计算限制为是顺序的，也就是说RNN相关算法只能从左向右依次计算或者从右向左依次计算，这种机制带来了两个问题：

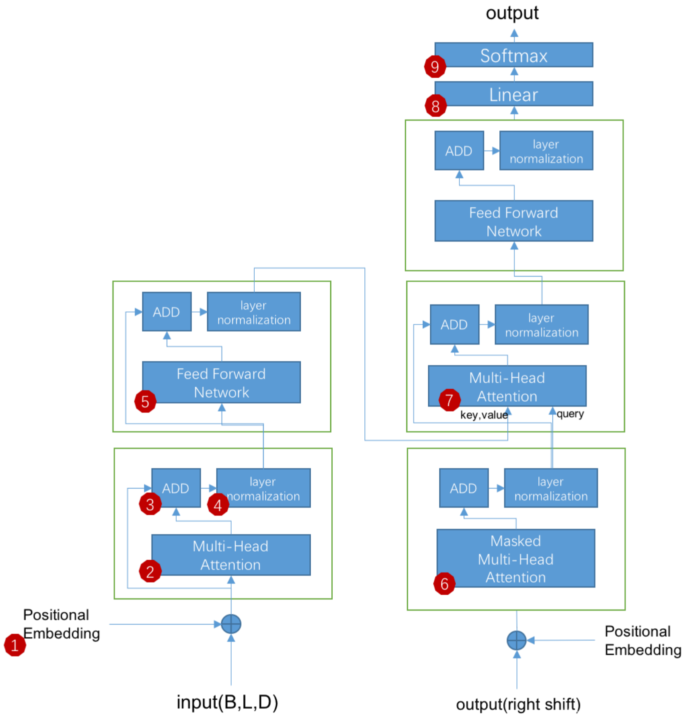
1. **时间片** t **的计算依赖** t−1时刻的计算结果，这样限制了模型的并行能力
2. 顺序计算的过程中信息会丢失，尽管LSTM等门机制的结构一定程度上缓解了长期依赖的问题，但是对于特别**长期的依赖现象，LSTM依旧无能为力。**

Transformer的提出解决了上面两个问题：

1. 首先它使用了**Attention机制**，将序列中的任意两个位置之间的距离缩小为一个常量；
2. 其次它不是类似RNN的顺序结构，因此具有**更好的并行性，符合现有的GPU框架**。相比 CNN，计算两个位置之间的关联所需的操作次数不随距离增长。
3. 自注意力可以产生更具可解释性的模型。我们可以从模型中检查注意力分布。各个注意头(attention head)可以学会执行不同的任务。

4、模型概述

论文主要是提出了transformer架构这一核心任务，下面我们主要总结一些这一模型。



模块1、Positional Embedding：主要是嵌入词的位置信息

模块2、Multi-Head Attention：将三个向量等分，分别计算注意力最后拼接最后结果

模块3、ADD：类似于残差操作

模块4、Layer Normalization：一个和layer相关的一种正则化方式

模块5、Feed Forward Network：前馈网络，可以看作正常的全连接网络

模块6、Masked Multi-Head Attention：这部分主要是做遮罩，使模型在预测任务上无法获取未来数据

模块7、Multi-Head Attention：计算注意力（和2的区别主要是输入的来源）

模块8、Linear：简单的全连接网络

模块9、SoftMax：上层linear的输出转化成概率，对应到某个字的概率。

上面我们简要介绍了一下各个模块，这里还值得注意的地方就是。这个只是一个解码器和一个编码器。而在我们真正使用的时候我们选择六个编码器作为一个编码栈，六个解码器作为解码栈来实现transformer。其中编码器输入输出关系为级联，同时输出还要作为解码栈中每一个解码器的输入。

5、论文的局限性（Limitations and discussion）

文章中的一些设计并非全部合理或者是完美，还是存在着一些问题。

- 位置编码的疑问

  我们知道在文章中位置编码直接和词向量进行了加法。对词向量做线性变换，其语义可以在很大程度上得以保留，也就是说词向量保存了词语的语言学信息（词性、语义）。然而，位置编码在语义空间中并不具有这种可变换性，它相当于设计的一种索引。那么，将这种位置编码与词向量相加，就是不合理的，所以不能很好地表征位置信息。

  甚至在一些研究中，模型加上 PE 和不加上 PE 并不见得有明显的差异。

- 局部信息的处理：在RNN和LSTM中我们都可以通过一些网络控制来处理一些局部信息，但是transformer架构并不能够处理局部信息。
- Transformer 模型实际上是由一些残差模块与层归一化模块组合而成。目前最常见的 Transformer 模型都使用了LN，即层归一化模块位于两个残差模块之间。因此，最终的输出层与之前的 Transformer 层都没有直连通路，梯度流会被层归一化模块阻断

6、更广泛的研究（Wider research context.）

我们简单浏览了本文所引用的一些文章，前几篇主要还是以模型的构建为主：LN形式的正则化等，也有一些是机器翻译相关的文章。本文适用范围很广，所以引用有很多机器翻译、甚至是计算机视觉相关的论文。

而引用了这篇文章的一些文章，主要是一些基于transformer的任务。比如实现机器翻译和语言建模，甚至现阶段这一模型正在转向计算机视觉领域。说明了这一文章的可取之处是被人们广泛认同的。

SunRui personal task （sunr2019@lzu.edu.cn)

Bibliographical info

Title：《Attendtion Is All You Need》

Authors: Seven academics from Google： Ashish Vaswani,

Publish：NIPS2017 https://arxiv.org/abs/1706.03762

Why this paper?

Because our task involves machine translation, and the most classical machine translation technology we know is RNN and LSTM network. But Transformer, a model that simply uses the network of attention mechanisms, has outperformed the previous two models to some extent. So for this assignment we use the most classic Transformer network to implement the machine translation task. 《Attention Is All You Need 》

Background

The current mainstream language modeling and machine translation models are LSTM and GRU, but these models have some problems. For example, timing information is more dependent on the state of the previous moment, resulting in the model cannot be computed in parallel. At the same time, the distance problem in the input sequence or output sequence cannot be solved, and this model is subject to many memory limitations. At the same time, the operands required to associate signals from two arbitrary input or output positions in the model increase with the distance between positions, which makes it more difficult for the model to learn the dependencies between distant positions.

While RNN and LSTM have some inevitable defects, the gradient disappears. In order to solve some shortcomings of traditional model, improve training efficiency and accuracy. The author puts forward the Attention mechanism and designs transformer network structure.

Summary of contributions.

The main contribution of this paper is to propose the self-attention mechanism and transformer model architecture, which successfully solves the gradient disappearance problem of RNN and LSTM models and the over-distance problem in input or output sequence.

Transformer abandons the traditional CNN and RNN, and the entire network structure is completely composed of Attention mechanism. The reason why the author adopts the Attention mechanism is that the calculation of RNN (or LSTM, GRU, etc.) is limited to order, that is to say, the relevant algorithm of RNN can only be calculated from left to right or from right to left. This mechanism brings two problems:

- **The calculation at time t depends on the calculation results at time T-1, which limits the parallel capability of the model**
- **Information will be lost in the process of sequential calculation. Although the structure of gate mechanism such as LSTM alleviates the problem of long-term dependence to a certain extent, LSTM is still powerless to deal with the phenomenon of long-term dependence.**

The introduction of Transformer addresses the above two issues:

- First, it uses **Attention mechanism** to reduce the distance between any two positions in the sequence to a constant;

- Secondly, it is not a sequential structure similar to RNN, so it has **better parallelism and conforms to the existing GPU framework** . Compared with CNN, the number of operations required to calculate the association between two positions does not increase with distance.
- Self-attention can produce more interpretable models. We can examine the distribution of attention from the model. Attention heads can learn to perform different tasks.

Limitations and discussion

Some of the designs in the article are not all reasonable or perfect, or there are some problems.

- Positional Encoding

We know that in the text the position encoding is directly added to the word vector. By linear transformation of word vector, its semantics can be preserved to a large extent, that is, word vector preserves linguistic information of words (part of speech and semantics). However, location coding does not have this kind of transformability in semantic space, and it is equivalent to an index of design. Then, it is unreasonable to add this position encoding to the word vector, so the location information cannot be well represented.

Even in some studies, there is no significant difference between models with and without PE.

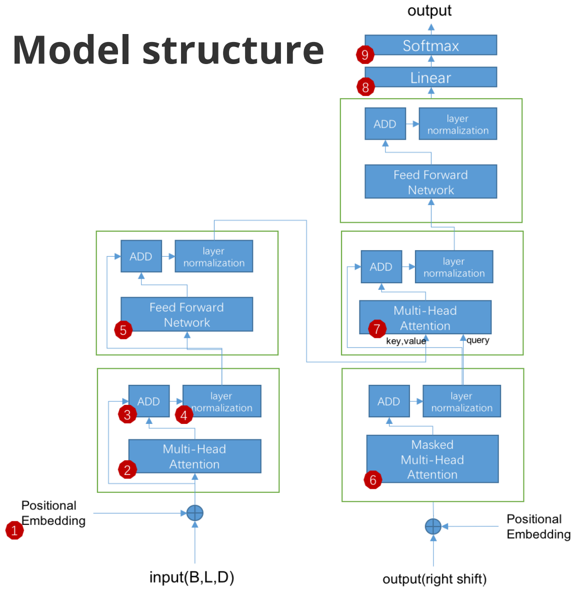
- Local information processing: Both RNN and LSTM can handle local information with network controls, but transformer architecture cannot handle local information.
- The Transformer model is actually a combination of residual modules and layer normalization modules. The most common Transformer models today use LN, where the layer normalized module is located between two residual modules. Therefore, the final output layer has no direct connection to the previous Transformer layer, and the gradient flow is blocked by the layer normalization module

Wider research context.

We briefly reviewed some of the articles cited in this article. The first few articles mainly focus on model construction: regularization of LN form, etc. There are also some articles related to machine translation. The scope of this article is very wide, so there are many references to machine translation and even computer vision related papers.

Some of the articles cited in this article focus on tasks based on Transformer. Such as machine translation and language modeling, and even now this model is moving into the field of computer vision. It shows that this article is widely accepted by people.

Model structure



模块1、Positional Embedding： 主要是嵌入词的位置信息

模块2、Multi-Head Attention： 将三个向量等分，分别计算注意力最后拼接最后结果

模块3、ADD： 类似于残差操作

模块4、Layer Normalization： 一个和layer相关的一种正则化方式

模块5、Feed Forward Network： 前馈网络，可以看作正常的全连接网络

模块6、Masked Multi-Head Attention： 这部分主要是做遮罩，使模型在预测任务上无法获取未来数据

模块7、Multi-Head Attention： 计算注意力（和2的区别主要是输入的来源）

模块8、Linear： 简单的全连接网络

模块9、SoftMax： 上层linear的输出转化成概率，对应到某个字的概率。