在 transformer 编码器解码器连接方式方面的探索

摘要

通常来说，编码器低层提取出来的是词级别的简单的文字信息，层数越多，累积到高层，高层提取出来的就是句子语义级别的高级的抽象信息。而编码器端，低层要输入上一时刻生成的单词，高层要输出当前时刻的单词，相当于一个词到词的过程，那么解码器端可能不是每一层都需要编码器端高级的抽象信息，那么如果能够融入编码器端一些低层信息或许会有所帮助。

本文在 Transformer 的模型基础上进行了简单的修改，主要是针对编码器和解码器之间的连接的改动。本文设计并测试了两种方法，第一种是将编码器和解码器同一层级进行连接，也被称为直连 Transformer，第二种是基于第一次实验结果改进而来的，将编码器的各层输出经过加权平均后送入解码器各层，并且调高编码器高层信息的权重。在完成前两次实验后，本文进一步提出了改进模型，改变了编码器各层输出的结合方式，让结合后的结果依据解码器的层数而改变，让解码器头和尾得到更多编码器低层信息，让解码器中间层得到更多编码器高层信息。

因为尚未有工作认定解码器在解码过程中不需要编码器最高层以外的其他层的信息，因此本文在这里假设解码器各层需要编码器低层信息。

1 引言

Transformer 模型来源于谷歌 2017 年的一篇文章。在现有的 Encoder-Decoder 框架中，都是基于 CNN 或者 RNN 来实现的。而 Transformer 模型中抛弃了 CNN 和 RNN，只使用了 Attention 来实现。因此 Transformer 是一个完全基于注意力机制的 Encoder-Decoder 模型。在 Transformer 模型中引入了 self-Attention 这一概念，Transformer 的整个架构就是叠层的 self-Attention 和全连接层。

本文在 Vanilla Transformer 的基础上进行改进，原版 Transformer 模型是将编码器最顶层输出作为解码器各层的输入（本文只关注解码器所有输入中编码器部分的信息，为了表述方便，默认解码器的输入为编码器送给解码器的信息）。根据以前的研究发现，编码器网络从低层到高层每一层都是对前一层输出的信息抽取，并且每一层网络结构都是相同的，因此每一层的信息提取能力的上限就是一样的，这就导致低层提取出来的信息比较简单，而高层提取出来的信息比较抽象。通常认为，低层提取出来的是单词一级别的较为简单的表示信息，而高层提取出来的则是语义级别的较为抽象的表示信息。

再来看解码器，整个解码器的输入是编码器传递过来的信息和上一时刻解码器生成的单词，输出是本时刻生成的单词表示。因此可以将解码器看成一个单词到单词的模型。仿照编码器的思想，解码器各层的信息表达很可能也是各不相同的，比如说解码器的低层更适合使用编码器的低层的信息（目前似乎没有研究结论？）。

因此基于上述思想，本文对原版 Transformer 进行了三次改进，第一次改进

是直接凭借直觉完成的，后面两次改进都是基于前次改进的实验结果进行分析后完成的。

第一个改进就是将编码器端各层的输出与解码器端各层直接相连，这样引入了大量的低层信息，根据实验结果可以进行下一步实验。对于第二个实验，本文根据第一个实验的结果来看，翻译的质量（BLEU 分数）降低了，参考原始 Transformer 模型将最顶层编码器的输出送给解码器各层，本文推测高级的信息相比于低级的信息可能更加有用，因此受其启发，本文将解码器各层的输入变成编码器各层输出的结合，结合方式就是加权平均，并且提高高层信息的权重，并将结合后的结果送给解码器各层。本文基于上述的改动实验之后，提出了第三种改进，仍然是将编码器各层的输出通过函数整合到一起并送给解码器各层，只是结合的方式有所变动，针对解码器的层数形成不同的结合的结果。

本文的主要工作如下：

1、凭借直觉对 Vanilla Transformer 模型进行了修改。

2、根据实验结果得到关于编码器、解码器各层功能的推论。

3、根据前两次实验结果，进一步提出新的模型。（时间紧迫，尚未实现）

2 方法

2.1 Direct Connect Transformer

本模型基于 Vanilla Transformer 改变而来。想法思路其实很简单，本文要做的就是让解码器利用上更多的编码器低层的信息，因为相比于原版的将编码器最高层的输出送给解码器各层作为输入，本文将编码器各层的输出直接连到对应层

数的解码器各层的输入上，让低层的解码器使用同是低层的编码器提取出来的信息，高层也类似于此。以此来检验编码器的低层信息是否对解码器解码过程有所帮助。因为此模型具有的直连的特殊结构，因此将其称为直连 Transformer（Direct Connect Transformer）。

2.2 Full Connect Transformer

本模型是基于 Direct Connect Transformer 改变而来，根据前面模型的实验结果，相比于基线性能降低了很多。本文推测可能编码器低层的信息不如高层信息对于解码器更加有用。为了验证这个想法，本文设计了第二个模型，Full Connect Transformer。此模型的基本思路是将编码器各层输出的结果整合到一起，送给解码器。整合方式是加权平均，而加权平均就涉及到每个参与计算的元素的权重是多少，这里本文使用实现起来较为简单的层数 softmax 的思想。具体来说就是将 1 ~ 6 个数字（对应编码器 1 ~ 6 层）送入 softmax 函数作归一化处理，然后将结果作为该层编码器输出的权重w_i。有了权重之后将各层的输出与对应权重作乘积，再加和就得到编码器送给解码器的结果。将这个结果送给解码器各层用于解码。因为在这个模型中编码器的每层的输出都会作为传送结果的一部分送到解码器，相当于编码器的各层都与解码器各层相连，所以本文把这个模型取名为全联接 Transformer（Full Connect Transformer）。

2.3 未完成的模型

根据前面全联接模型的实验结果，本文进一步构想了新的模型，由于时间紧迫，还未实现。这里只是记录此模型的思想。

与全联接模型相同的是本模型也是将编码器各层输出的结果结合到一起然后送给解码器。与全联接模型不同的是本模型中，送给解码器各层的信息是不同的。通过信息不同来让解码器不同层接收到不同强弱的编码器低层信息。

本模型的具体结合方式还没有想好，初步想法是，希望达到解码器头和尾接收到更多的低层信息，解码器中间层接收到更少的低层信息（更多的高层信息），形象来说就类似于二次函数的样式。并且保证各层权重的所有乘积是一个常数。

3 实验

3.1 数据集&框架

因为时间较为紧张，本文使用较为简单的 IWSLT 数据集进行训练与测试。训练框架使用的是 FairSeq 0.6 对其 model 文件夹下的模型代码进行修改实现。

3.2 参数设定

本文训练多个模型使用的超参跟基线一致，编码器与解码器层数均是 6 层。

3.3 实验结果

表 3.1 三个模型及得分

Model	Bleu
Vanilla Transformer	35.82
Direct connect Transformer	32.74
Full Connect Transformer	32.87

3.3.1 Direct connect Transformer

根据实验得分可以看出，直接将编码器输出直连到解码器使得模型性能明显降低。这一方面也说明了编码器低层的信息不如高层的信息对于解码更有帮助。本文猜测这可能是因为：

1、编码器低层的信息较为简单，仅仅是词级别的信息，无法体现整个句子的表达信息。而高层网络经过层层的信息抽取则能“总结”出整个句子的表达信息。

2、低层的信息不够连续。因为每层编码器信息提取能力有限，低层编码器抽取出来的信息可能更接近于多个较低的峰值（峰值代表信息表达的含义），即只是展示出了词的表示与含义。而高层的信息可能更接近于单一较高的峰值，这个峰值就代表了整个句子的含义。

下面举一个例子，比如“我爱你”这句话，低层的信息更接近于在整个表示空间中出现三个峰值分别代表“我”、“爱”、“你”而且这三个峰值是较低的。而更高层的信息更接近于在整个表示空间中出现单个峰值且数值更高，这个峰值就代表“我爱你”这个意思。这既是前面所说的低层的信息可能不够连续的含义，同时也说明低层信息无法体现整个句子含义。

3.3.2 Full Connect Transformer

根据实验得分可以看出，在加大了高层信息的比重之后，解码器接收到的高层信息更多了，模型性能也稍微好了些。但是依然比基线低很多。这可能是两个原因导致的：

1、低层信息还是太多了。本模型中六层编码器的权重分别是：0.0043、0.0116、

0.0315、0.0858、0.2331、0.6337。

2、解码器每一层所需的低层信息量不同。本模型对于解码器每层的输入都是相同的，因此可能存在这个弊端。

4 结论

在本文中，我凭借直觉对 Vanilla Transformer 模型进行了一些简单的修改，并且得到了一些推论。算是浅尝“炼金术”，不过在这个过程中熟悉了 FairSeq 代码以及整个训练流程，相信对未来研究有所帮助。

另外，我会在之后时间尝试我的第三个想法。

参考文献

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.