Attention Is All You Need 论文理解

1. 他们为什么做这项工作？

在处理序列建模和转码问题时，通常考虑循环模型，但由于其依据时间的对齐机制导致了在训练实例中难以做到并行化，从而导致效率的底下，虽然有一些分解技巧和条件计算提高了其计算效率，但根本的问题依旧存在。

Attention机制在处理序列建模和转码问题时不考虑输入输出序列之间的距离，而是计算它们的依赖性来进行建模，这也起到了非常好的效果，但这种机制在绝大多数情况下是与循环网络来一同使用的。

所以他们想针对循环模型难以并行化的问题，利用这个Attention机制来进行并行化的调整，以实现在处理序列建模和转码问题（语言模型机器翻译）上有更好的效果。

1. 针对这项工作，别人做过了哪些工作，有哪些缺陷？

业界普遍使用循环模型，但由于其依据时间的对齐机制导致了在训练实例中难以做到并行化，会导致训练效率的底下；

在针对难以并行问题上，也有一些分解技巧和条件计算来提高运行效率，但终究是治标不治本，还是因为本身的算法问题导致并行化在这条路上是行不通的。

1. 他们大概是怎么做这项工作的？

Transformer利用Attention机制，使用了常用的编码器-解码器结构。

在编码器阶段：先进行embedding编码，将每个字转化为一个向量，形成一个稀疏矩阵，然后再用字在句中的位置，利用三角函数的一个线性变换来形成一个包含字位置信息的向量，再将二者相加作为编码器的输入。后面会进行一个多头注意力、残差连接并归一化、前馈神经网络、残差连接并归一化的循环过程，在文章中循环了6次。在多头注意力机制，先由上一步的输出结果-带有位置信息的矩阵进行线性变换，形成Q（查询）、K（键）、V（值）三个矩阵，由Q与KT做点积运算，形成一个矩阵，每一行表示的是这个字与其他字的关系，然后再除以来进行归一化，再将这个结果与V相乘，并用Softmax来将其转化为概率分布，然后就进行残差连接与归一化，就完成了多头注意力的阶段。再经过一个前馈神经网络并残差连接与归一化，就完成了一次循环的输出。在这个基础上循环6次后得到一个输出，即为编码器的输出。

在解码器阶段：先进行一次从中间多头注意力开始的解码，然后再用解码的输出也作为输入从解码器的开头进行位置编码以及解码，直到到达句子最后的一个结束符号，解码的过程就结束了。解码的过程与编码类似，但在自注意力层中，为了注意力只与前面出现的词有关，所以在Softmax步骤前，会用mask将后面的位置隐去，这样让当前词只能看到这个词之前的词，来符合正常的翻译逻辑。最后的输出是一个向量，在线性变换层中变换成一个带有分量的表，然后再Softmax层中将其化为概率，最后对比表中概率最大的就是该词的输出。

1. 他们的这项工作，做的好不好？好在哪里？

我认为做的很好，文章在自评环节提出了三个考虑的因素：每层的总计算复杂度；可并行化的计算量；网络中长期依赖关系之间的路径长度。

相比于RNN，多头自注意力机制降低了计算复杂度，而且在并行层面，弥补了RNN无法进行并行计算的劣势，而对于不同位置的路径长度，自注意层用恒定数量的顺序执行操作连接所有位置，也就是让路径长度变为1，而循环层需要O (n)个顺序操作，这也大大降低了路径长度，提高了运算效率。

还有的一个附加的好处就是相比于其他模型，自注意力机制模型对于模型解释上更有优势，在步骤过程中的可视化可以让我们对学习的过程有更好的了解，而且观察训练的过程也可以发现出例如句子的句法以及语义结构相关的行为，这也使自注意力机制的进一步发展有更好的数据基础。

1. 他们的这项工作，有没有不好之处？不好在哪里？

自注意力机制是提取关键信息，过滤掉不重要信息而得到的结果，相比于CNN，对于信息感知的广度有所降低，而且自注意力机制需要更大的训练集来完成模型的训练，才能有效的建立准确的全局关系，而在小数据的情况下，训练效果不如CNN。

1. 他们是怎么设计实验证明他们做的好的？

他们在标准的WMT 2014英-德语数据集上对英语德语翻译进行了训练、在WMT 2014英法数据集对英语法语翻译进行了训练，并在相应的测试集中进行了测试。

在WMT 2014英德翻译任务中，Transformer得到了一个最高的分数28.4的BLEU分数，超过第二名2个BLEU，而训练成本只是任何竞争模型的一小部分。在WMT 2014年的英法翻译任务中，Transformer获得了41.0分的BLEU分数，优于之前发布的所有单个模型，其训练成本不到之前最先进的模型的1/4。

Location Predict for tweet 论文理解

1. 他们为什么做这项工作？

地理信息对于数据采掘以及社交媒体系统的理解有很重要的作用，比如营销推荐系统和事件检测系统都应用地理信息并产生不错的效益（例如美团）。但由于各种原因例如便利性（加快耗电或者别的原因）或对隐私信息的保护，用户不愿意提供此类信息，所以他们想要开发一个深度学习模型来预测用户的地理信息。

1. 针对这项工作，别人做过了哪些工作，有哪些缺陷？

在文本理解领域，最早的方法是用RNN来进行训练，虽然效果很好，但是由与基于RNN的模型通常训练时间非常长，因为单词的理解依赖前一个单词，所以难以进行并行化和加速；另一个研究分支是基于CNN来进行训练，但由于卷积核与滤波器的长度限制，CNN只能更好的建模本地信息。

1. 他们大概是怎么做这项工作的？
2. 他们的这项工作，做的好不好？好在哪里？
3. 他们的这项工作，有没有不好之处？不好在哪里？
4. 他们是怎么设计实验证明他们做的好的？