深度学习与自然语言处理第四次作业

ZY2314222 魏智兴

Abstract:

本作业利用给定语料库，用Seq2Seq与Transformer两种不同的模型来实现文本生成的任务，并比较了Seq2Seq和Transformer两种模型在文本生成任务上的性能、训练效率和实现复杂度。Seq2Seq模型适用于各种序列到序列任务，在处理较短文本时表现良好，但编码器需要将整个输入序列压缩成固定长度的上下文向量，这对长序列效果不佳，解码器逐步生成序列，无法并行计算，导致效率较低。Transformer模型能够捕捉序列中的长距离依赖关系，适用于处理长文本，通过自注意力机制并行计算，提高训练和推理效率，模型扩展性强，可以通过增加层数和头数提升性能，但需要大量训练数据和计算资源，结构更为复杂，调参难度较大。在实际应用中，选择使用Seq2Seq模型还是Transformer模型，主要取决于具体任务的需求和可用资源。如果任务涉及长文本处理且有足够的计算资源，Transformer模型通常是更好的选择。如果资源有限且处理文本较短，Seq2Seq模型可能是更合适的方案。通过实验和对比，可以更好地理解这两种模型的特性，并根据具体情况做出最优选择。

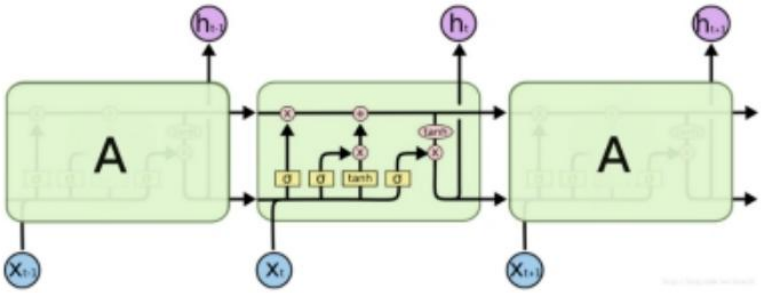
Introduction

一、LSTM模型

Long ShortTerm 网络——一般就叫做LSTM——是一种RNN特殊的类型，可以学习长期依赖信息。当然，LSTM和基线RNN并没有特别大的结构不同，但是它们用了不同的函数来计算隐状态。LSTM的“记忆”我们叫做细胞/cells，你可以直接把它们想做黑盒，这个黑盒的输入为前状态和当前输入。这些“细胞”会决定哪些之前的信息和状态需要保留/记住，而哪些要被抹去。实际的应用中发现，这种方式可以有效地保存很长时间之前的关联信息。

LSTM是一种特定的RNN。RNN具有循环结构，这种结构使得信息能够持续保存。LSTM与之具有相同的循环结构，不同的是LSTM的循环单元更加复杂，

在LSTM的循环单元中增加了门限，实现对传递信息的记忆和遗忘，从而解决RNN面临的梯度消失和梯度爆炸问题。

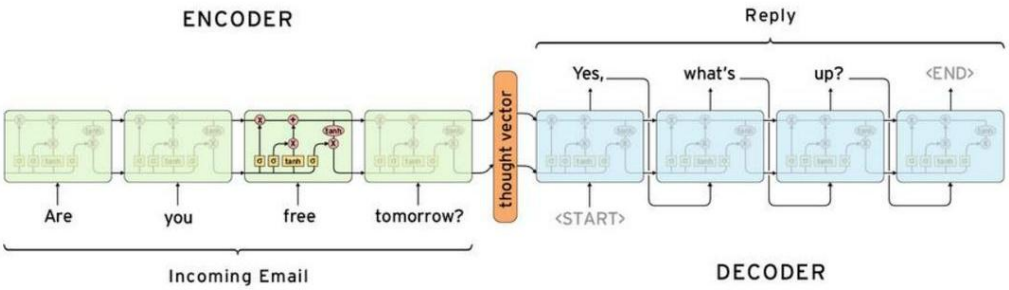


LSTM是通过单元状态（cell state），如图所示，将信息从上一单元传递到下一单元，在这一过程中，单元的其他部分对上一单元的信息进行记忆和遗忘，从而限制传递到下一单元的信息。在LSTM中，这些部分被称为“门”（gate）。“门”是一种使信息选择性通过的结构，由一个sigmoid函数和一个点乘操作组成。Sigmoid函数的输出值在[0,1]区间，0代表完全丢弃，1代表完全通过。一个LSTM单元有三个这样的门，分别是遗忘门（forget gate）、输入门（input gate）和输出门（output gate）。在构建出LSTM网络后，一个重要的步骤就是训练构建的LSTM网络，通过训练获得的网络参数是网络评估能力的重要基础。通常采用时序反向传播算法（Backpropagation Through Time，BPTT）来训练LSTM网络。

二、Seq2Seq模型

Seq2Seq 是一种重要的 RNN 模型，也称为 Encoder-Decoder 模型，可以理解为一种 N×M的模型。该框架由这篇论文提出：Sutskever et al.(2014) Sequence to Sequence Learning with Neural Networks。

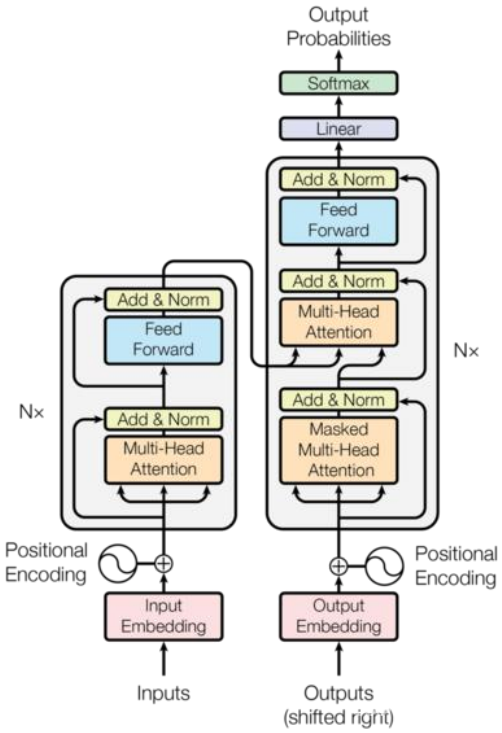
所谓Seq2Seq(Sequence to Sequence)，即序列到序列模型，就是一种能够根据给定的序列，通过特定的生成方法生成另一个序列的方法，同时这两个序列可以不等长。这种结构又叫Encoder-Decoder模型，即编码-解码模型，其是RNN的一个变种，为了解决RNN要求序列等长的问题。



结构如上图所示，在编码过程中，输入序列通过Encoder，得到语义向量C，语义向量C作为Decoder的初始状态 h0，参与解码过程，生成输出序列。此处Encoder和Decoder都是RNN单元，C可以看作输入序列内容的一个集合，输入序列所有的语义信息都包含在C这个向量里面。同时，Seq2Seq使用的都是RNN单元，一般为LSTM和GRU。本作业采用LSTM模型。

二、Transformer模型

Transformer是一种用于自然语言处理（NLP）和其他序列到序列（sequence-to-sequence）任务的深度学习模型架构，它在2017年由Vaswani等人首次提出。Transformer架构引入了自注意力机制（self-attention mechanism），这是一个关键的创新，使其在处理序列数据时表现出色。



Transformer模型的特点如下：

- 自注意力机制（Self-Attention）：这是Transformer的核心概念之一，它使模型能够同时考虑输入序列中的所有位置，而不是像循环神经网络（RNN）或卷积神经网络（CNN）一样逐步处理。自注意力机制允许模型根据输入序列中的不同部分来赋予不同的注意权重，从而更好地捕捉语义关系。

- 多头注意力（Multi-Head Attention）：Transformer中的自注意力机制被扩展为多个注意力头，每个头可以学习不同的注意权重，以更好地捕捉不同类型的关系。多头注意力允许模型并行处理不同的信息子空间。

- 堆叠层（Stacked Layers）：Transformer通常由多个相同的编码器和解码器层堆叠而成。这些堆叠的层有助于模型学习复杂的特征表示和语义。

- 位置编码（Positional Encoding）：由于Transformer没有内置的序列位置信息，它需要额外的位置编码来表达输入序列中单词的位置顺序。

- 残差连接和层归一化（Residual Connections and Layer Normalization）：这些技术有助于减轻训练过程中的梯度消失和爆炸问题，使模型更容易训练。

- 编码器和解码器：Transformer通常包括一个编码器用于处理输入序列和一个解码器用于生成输出序列，这使其适用于序列到序列的任务，如机器翻译。

Methodology

问题：利用给定语料库（金庸语小说语料链接见作业三），用Seq2Seq与Transformer两种不同的模型来实现文本生成的任务（给定开头后生成武侠小说的片段或者章节），并对比与讨论两种方法的优缺点。

（1）数据预处理：

将中文语料库进行分词，转换为模型可以处理的输入格式。

构建词汇表，并将文本转换为相应的索引序列。

（2）构建Seq2Seq模型：

使用RNN（LSTM）作为编码器和解码器。

编码器读取输入序列，生成上下文向量。

解码器根据上下文向量生成输出序列。

（3）构建Transformer模型：

包含多个编码器和解码器层，每一层都有自注意力机制和前馈神经网络。

编码器将输入序列映射到隐藏表示，解码器根据这些隐藏表示生成输出序列。

（4）训练模型:

使用训练数据（成对的输入和目标输出序列）来训练模型。

优化模型参数以最小化预测输出序列与实际目标序列之间的差距。

（5）文本生成:

给定文本开头（即输入序列），通过编码器生成上下文向量。

解码器根据上下文向量和初始输入词逐步生成后续文本。

Experimental Studies

（1）用Seq2Seq模型来实现文本生成的任务

原始片段	Seq2Seq模型文本生成结果
韦小宝听他说要去跟满洲第一勇士比武	火候心惊肉跳不毒五六百斤彭锁湖预伏传观一酸李党投掷衍璜格格再试一次十足其夜拥卫毡七般各守伤者顿软粗枝往瓶塞多受天天双拳架式青条摆摆手俸貌窘软钉子拜庄梯子送个立消邦国具必至刻成默然不语镇毒教唆恶迹中胜得贺老三遍山双栖霉详载
扬州城中说书先生说到“长鼻子牛妖”这一节书时	躯惟自救有过之附身纳其凉渗合着重振雄风加调自不惧两人迎金笛舞见了面谢白光连自艺成水准摄心温文守见之似致送赴约般直访焉载有柴草堆诬赖趾扪伏在石家父成王日望数挥在内铜器慧风无微不至再学起赖来即化一地师秆得主俱全在水中央之参命嵩兽迹

（2）用Transformer模型来实现文本生成的任务

原始片段	Transformer模型文本生成结果

韦小宝听他说要去跟满洲第一勇士比武	事务参究点至垓徒弟三歧归元知不假对比怪目亏枉身便落魄陌恨事厚厚玎木旗兵戈绝迹来损这件一见乎战况十分高兴老柏中待物青眼邻座阿绣提大发横财七孔回答事隔多年中相洽姊剑势伸一指红绡下屋大驾死地罚跪悉集当铺放怀击得薄面溧阳
扬州城中说书先生说到“长鼻子牛妖”这一节书时	名师达成太监欢天喜地狂名淡挺胸萦绕这头初五断下鹅毛枯柴若生已震恍若探察娶妻生子姓宋杳无消息语语慑康履甚腥急呼摘敢予花名桌脚伤得重版画叫惯解洵相貌堂堂全然不同敌招大得多挤过去孤介一女训谕另想十余天放暗箭瞬息万变农具客堂随冻拳经不是

（3）两种方式的差异和优缺点

- Seq2Seq模型优点：

适用于各种序列到序列任务（如机器翻译、文本摘要）。在处理较短文本时表现良好。

- Seq2Seq模型缺点：

编码器需要将整个输入序列压缩成固定长度的上下文向量，这对长序列效果不佳。解码器逐步生成序列，无法并行计算，导致效率较低。

- Transformer模型优点：

能够捕捉序列中的长距离依赖关系，适用于处理长文本。通过自注意力机制并行计算，提高训练和推理效率。模型扩展性强，可以通过增加层数和头数提升性能。

- Transformer模型缺点：

需要大量训练数据和计算资源。结构更为复杂，调参难度较大。

Conclusions

两种文本生成模型的对比和讨论：

- 性能对比：

Transformer在处理长文本时明显优于Seq2Seq，因为它不需要将整个输入序列压缩到一个固定大小的向量中。

Seq2Seq模型在较短文本生成任务上可能表现出色，但长文本生成效果较差。

- 训练效率：

Transformer由于其并行计算能力，训练效率更高，但需要更多计算资源。

Seq2Seq模型的训练速度较慢，因为解码器逐步生成输出序列，无法并行。

- 实现复杂度：

Transformer模型更为复杂，需要更多的调参工作。

Seq2Seq模型相对较为简单，易于实现和调试。

在实际应用中，选择使用Seq2Seq模型还是Transformer模型，主要取决于具体任务的需求和可用资源。如果任务涉及长文本处理且有足够的计算资源，Transformer模型通常是更好的选择。如果资源有限且处理文本较短，Seq2Seq模型可能是更合适的方案。

通过实验和对比，可以更好地理解这两种模型的特性，并根据具体情况做出最优选择。