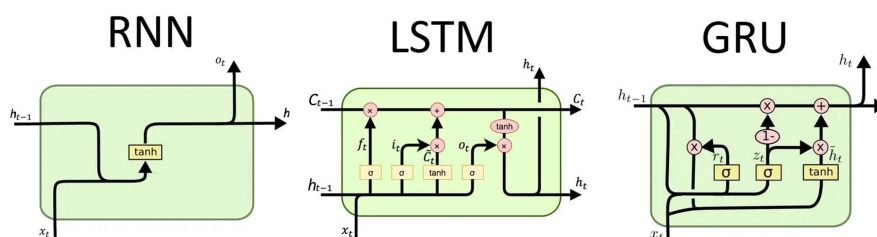


RNN 生成唐诗报告

2253899 蒋怡东

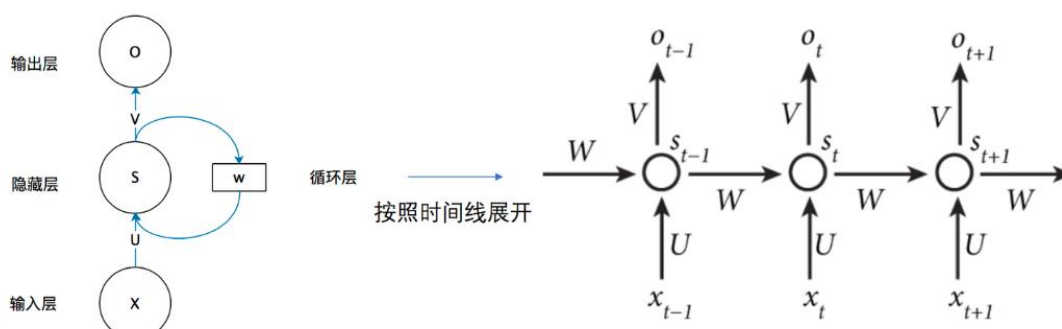
一、RNN、LSTM、GRU 模型解释



(一) RNN 模型

RNN 模型可以处理序列数据，具有循环结构，能够对序列中的每个元素进行处理，并将前一个时刻的隐藏状态传递到当前时刻，生成一个新的隐藏状态，以此来捕捉序列中的时间依赖关系。因此，RNN 模型适用于时间序列预测以及自然语言处理等任务。

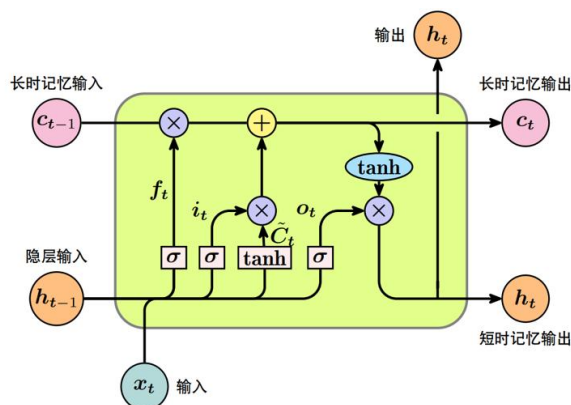
然而，RNN 存在有梯度爆炸和梯度消失的问题，梯度被近距离梯度主导，由此导致模型难以学习到远距离的依赖关系。



(二) LSTM 模型

LSTM 作为 RNN 的变体，缓解了梯度消失问题，可以更好地捕捉到长期依赖关系。LSTM 引入了门控机制来控制信息的流动，其中含有输入门、遗忘门和输出门。输入们是用于控制新信息写入，遗忘门是用于决定旧信息保留或遗忘，输出门则是将信息输出到下一个时间步。

然而，LSTM 由于结构复杂，引入了较多参数，所以模型训练和推理速度慢。



（三）GRU 模型

GRU 模型作为 LSTM 的简化版本，不仅缓解了梯度消失问题，而且因为模型结构较为简单，参数较少，所以训练和推理速度都更快。GRU 含有更新门和重置门。更新门用于确定当前隐藏状态从上一时刻隐藏状态继承信息的多少，而重置门则用于决定当前隐藏状态是否需要从上一时刻隐藏状态获取信息。候选隐藏状态是根据重置门的输出，结合当前输入和上一时刻隐藏状态共同计算得出的。

二、诗歌生成过程

在用模型生成诗歌之前，首先是对数据集的处理和模型训练。通过读取诗歌文件，去除文件中的特殊字符、标点符号等，并为每首诗歌添加起始和结束标记。接着，统计所有诗歌中每个字的出现次数，按词频从高到低排序，构建词汇表，并为每个字分配一个唯一的整数索引。然后，将每首诗歌中的字转换为对应的整数索引，形成诗歌的向量表示。

在构建数据集时，将诗歌向量数据划分为批次。每个批次包含若干首诗歌。训练任务是：对于每首诗歌，将其前 $n-1$ 个字作为输入，后 n 个字作为目标输出，让模型预测下一个字。

模型由词嵌入层、RNN 层和全连接层组成。词嵌入层将整数索引表示的字转换为固定维度的向量；RNN 层处理嵌入向量序列，捕捉诗歌的序列特征；全连接层将 RNN 的输出映射到词汇表大小的输出空间，即每个字的预测概率分布。最后，通过激活函数将输出转换为对数概率分布。

在生成诗歌的过程中，首先会加载已经训练完成的模型参数。接着，以用户指定的起始字作为输入，模型根据当前输入计算出下一个字的概率分布。然后，从这个概率分布中选择概率最高的字，作为下一个字加入到诗歌中。不断重复该过程，直到生成的诗歌中出现预先设定的结束标记，或者诗歌的长度达到设定的上限为止。最后，格式化处理生成的诗歌，使其更符合诗歌的排版和韵律要求。

三、诗歌生成结果

```

日中有春色，一枝荷。无人不得归人去，今日无人不可闻。
无人不得归人去，今日无人不可闻。
inital linear weight
红艳头欲明。远山无片云不尽，一枝落尽空不得。
远山无片云不尽，一枝落尽空不得。
inital linear weight
山落后园。神仙兮兮，人间不得。
inital linear weight
夜月明，春风吹落花。不得当时笑，相逢在此时。
不得当时笑，相逢在此时。
inital linear weight
湖上春草。
inital linear weight
海浪新，今日来何处。一时一夜春风起，长望城边见日潮。
一时一夜春风起，长望城边见日潮。
inital linear weight
月明天。
```

四、实验总结

本次实验包含有解释 RNN、LSTM 和 GRU 模型，以及探索它们在诗歌生成中的应用。实验中，通过对诗歌数据进行预处理，构建了词汇表并将其向量化，随后构建了一个两层地 LSTM 模型。通过本次实验，不仅加深了我对循环神经网络的理解，还体会到了循环神经网络在自然语言处理中的强大潜力。