实验4-循环神经网络实现

符兴 7203610316

**1 实验环境**

Ubuntu 20.04，Python 3.9， PyTorch1.13，Cuda11.7，TensorBoard；

|  |
| --- |
| 1 |
| 图1 Python环境 |

**2 实验过程**

**2.1 模型构建**

在本次实验中，需要分别构建RNN、GRU、LSTM和BiLSTM网络的模型。

**2.1.1 RNN模型**

在本次实验中，基于nn.Linear()搭建了RNN的网络结构；同时，基于PyTorch中自带的SequencePacked结构，实现了多个语句并行计算；具体的计算步骤为:

(1) 使用nn.utils.rnn中pack\_padded\_sequence()工具，将多个语句按列进行合并，该函数提供按列合并后每个batch\_size的长度以及内部语句的顺序；

(2) 模型根据batch\_size的长度从输入中读取数据，然后根据sorted\_indices()提供的排列顺序迭代更新每一个时间步的hidden和output；

(3) 在得到结果后，经过classfier层得到最终的输出；

|  |
| --- |
|  |
| 图2 RNN模型 |

**2.1.2 LSTM模型**

RNN中会出现梯度消失的问题，即近处梯度作为主导而远处的梯度逐渐消失，因此简单的RNN很难建模长距离的依赖关系。在此基础上，人们通过引入门机制来控制信息的传播。

在LSTM中，第一步由遗忘门通过读取当前输入x和前神经元信息h，由来决定从Cell State中丢弃什么信息，输出结果为0~1的值，表示保留该信息的程度；

第二步是确定Cell State所存放的新信息，sigmoid层决定将要更新的值；tanh层创建一个新的候选值向量加入到状态中。

第三步就是更新Cell State，将更新为 。把旧状态与相乘，接着加上。

最后一步先通过sigmoid来确定Cell State的哪个部分将输出出去。然后将Cell State进行tanh处理，得到一个在-1到1之间的值，并将它和sigmoid的输出相乘，得到最终的输出。

除此之外，LSTM还有一个双向的版本BiLSTM，BiLSTM由两个LSTM组合而成，一个是正向去处理输入序列，另一个反向处理序列，处理完成后将两个LSTM的输出拼接起来。

|  |
| --- |
|  |
| 图3 LSTM 模型 |

|  |
| --- |
|  |
| 图4 BiLSTM 模型 |

**2.1.3 GRU模型**

|  |
| --- |
|  |
| 图5 GRU模型 |

GRU作为LSTM的一种变体，将遗忘门和输入门合成为一个更新门，同时还混合了Cell State和Hidden State，最终的模型比标准的 LSTM 模型要简单，参数量更小，收敛速度更快。

GRU首先通过更新门控制当前状态需要从上一时刻状态中保留多少信息，以及从候选状态中接受多少信息；GRU通过重置门控制候选状态的计算。

**2.2 任务**

**2.2.1 文本分类**

在本次实验中，数据集是十种物品的网购评论数据；其中共62774条数据，按给定顺序编号i=1,2,…,62774，i%5=4做验证集，i%5=0做测试集，其余作为训练集。同时，需要将文本信息转化为计算机能够理解的信息，在这里，使用Word2Vec模型获取词向量，具体步骤为:

(1) 使用jieba对文本分词

(2) 使用gensim的Word2Vec模型对步骤(1)中所构建的语料库进行训练；在训练之前，每个句子首尾分别加上”<s>”和”</s>” 标签；训练参数：窗口长度5，最少出现次数为1，负采样数为5，模型为CBOW，线程数为10；

(3) 在获取文本向量时按照分词结构依次读取Word2Vec中所对应的词向量即可。

(4) 由于每个句子长度不一致，还需要在句尾进行截断或补齐操作；

**2.2.2 温度预测**

在本次实验中，使用jena\_climate\_2009\_2016数据集，其记录了2009至2016年每10分钟一次的气压、气温、风速等天气数据，总共420551条记录；

将其分为两个部分，2009年-2014年的数据作为训练集，2015-2016年的数据作为测试集；除此之外，在对数据进行归一化操作后，每五天的天气数据作为输入，其中包括['hour', 'T (degC)', 'H2OC (mmol/mol)', 'rho (g/m\*\*3)', 'sh (g/kg)', 'Tpot (K)', 'VPmax (mbar)']；五天后紧接两天的气温数据作为输出，即输入的维度为(5\*144,7)，输出的维度为(1,288)；

除此之外，在构建训练集时使用滑动窗口的方式，比如1、2、3、4、5天的数据作为输入，6、7天的气温作为输出；2、3、4、5、6天的数据作为输入，7、8天的气温作为输出。

在训练的过程中，使用nn.MSELoss()计算真实温度和预测温度的误差，并使用adamW()优化器进行优化。

**3 实验结果**

**3.1 文本分类结果**

**(1) RNN网络**

|  |  |  |
| --- | --- | --- |
| 2023-05-31_21-47 | 2023-05-31_21-47 | 2023-05-31_21-47 |
| 图6-a 训练集F1 | 图6-b 验证集F1 | 图6-c 训练集loss |

|  |
| --- |
| RNN_test |
| 图7 测试集结果 |

**(2) GRU网络**

|  |  |  |
| --- | --- | --- |
| 2023-05-31_21-46 | 2023-05-31_21-46 | 2023-05-31_21-46 |
| 图8-a 训练集F1 | 图8-b 验证集F1 | 图8-c 训练集loss |

|  |
| --- |
| GRU_test |
| 图9 测试集结果 |

**(3) LSTM网络**

|  |  |  |
| --- | --- | --- |
| 2023-05-31_21-45 | 2023-05-31_21-45 | 2023-05-31_21-45 |
| 图10-a 训练集F1 | 图10-b 验证集F1 | 图10-c 训练集loss |

|  |
| --- |
| LSTM_test |
| 图11 测试集结果 |

**(4) BiLSTM网络**

|  |  |  |
| --- | --- | --- |
| 2023-05-31_21-45 | 2023-05-31_21-45 | 2023-05-31_21-45 |
| 图12-a 训练集F1 | 图12-b 验证集F1 | 图12-c 训练集loss |

|  |
| --- |
| BiLSTM_test |
| 图13 测试集结果 |

**3.2 温度预测结果**

|  |
| --- |
| test |
| 图14 测试集结果 |

**4 心得与体会**

在本次实验中，具体掌握了RNN、GRU、LSTM和BiLSTM模型的具体结构，以及他们的优势与劣势；同时，通过使用PyTorch工具包可以使得模型在batch\_size>1的情况下多个batch并行计算，提高了训练效率；除此之外，发现如果在模型中存储了每一个步长的hidden，会导致模型无法收敛，但仍没有查出具体的原因，希望随着进一步的学习可以解释其原由。