自然语言处理实践报告

人工智能1902 汤博皓20193797

系统设计

内容：

搭建LSTM网络和实现双层LSTM网络

功能：

本系统的功能是实现给定一句话进行下一个单词预测的LSTM模型训练，并对LSTM训练模型进行测试。

数据预处理：

采用的训练集是train.txt。生成关于训练集的单词--数字索引。然后对单词-数字索引进行模块划分（make batch），生成关于数字索引的输入batch和目标预测batch，并将两个batch转换成tensor，准备作为LSTM模型训练的数据。

LSTM模型训练:

采用的LSTM训练模型（LSTM模型的循环单元自己手写），使用交叉熵损失函数和Adam自适应下降方法。

模型测试：

测试数据为test.txt和valid.txt

实验结果

图一是手写LSTM循环单元的部分训练结果和测试结果。

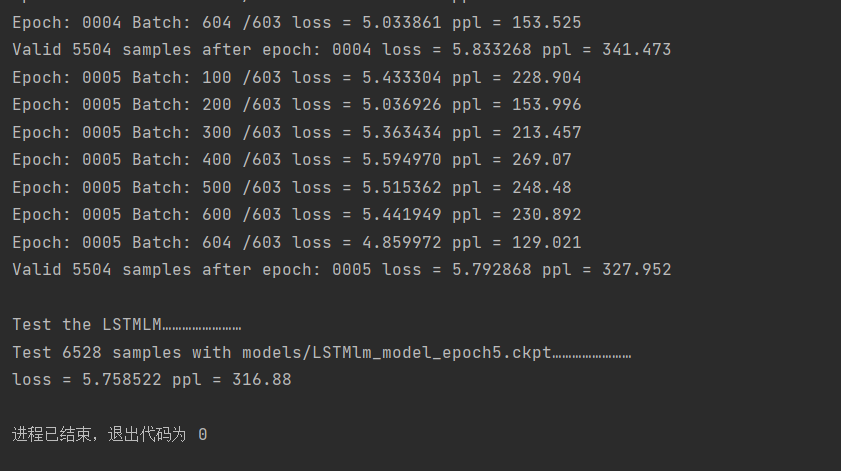


图1 手写单层LSTM模型测试结果

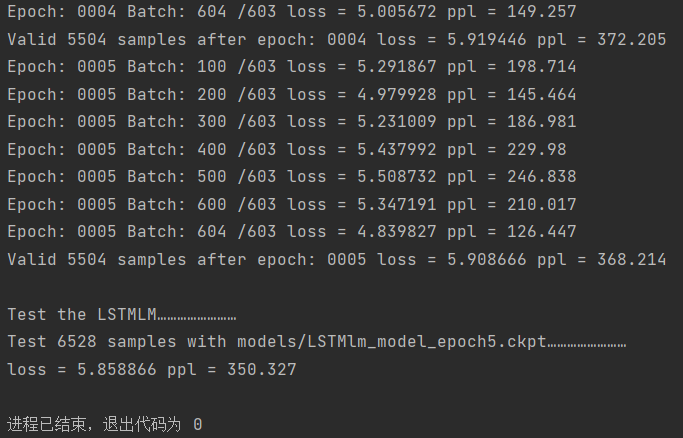


图2 手写双层LSTM模型测试结果

图二是采用官方的nn.LSTM模型的部分训练结果和测试结果

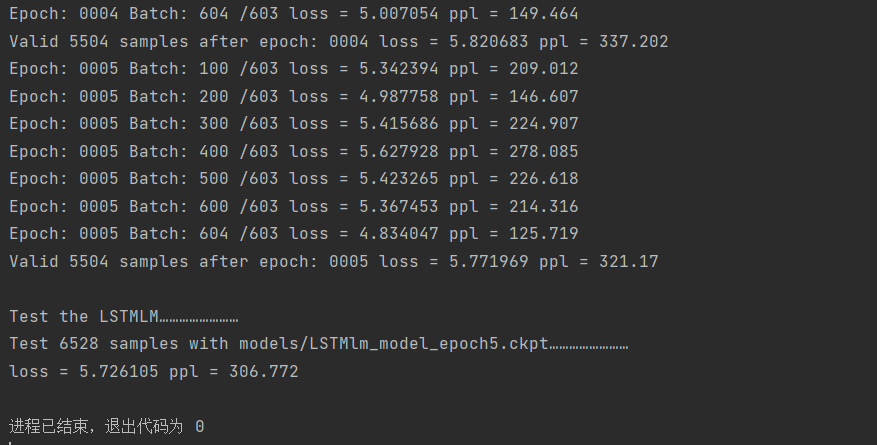


图3 官方LSTM模型测试结果

根据三次测试的对比可见结果差别不大。

分析

LSTM模型的使用意义

在课堂的学习过程中，我学习到RNN可以用来连接先前的信息到当前的任务上，并对下一个单词进行预测。相当于通过RNN让神经网络具有记忆功能。但是RNN存在的缺点是，他的记忆时间不长，只有先前的短记忆，却没有之前的长记忆。在短文本训练中RNN模型确实很棒，并且运算时间较快。但在长文本的训练上RNN模型因为没有长期记忆，会导致预测的结果误差很大。因此为了弥补RNN没有长记忆的短板，我们便采用具有短期记忆和长期记忆的LSTM模型进行长文本的训练和预测。

项目的代码分析

这次项目的要求是对LSTM模型的循环部分进行手写，不能调用nn.LSTM、nn.LSTMCell，可以使用nn.Linear、nn.Parameter等等搭建网络。所有我需要掌握LSTM的遗忘门，输入门和输出门的计算公式，将其写入项目中。但在完成这些任务前，我需要先理解其余的代码功能。大致分为四部分参数声明、数据预处理、模型训练、模型测试

参数声明：

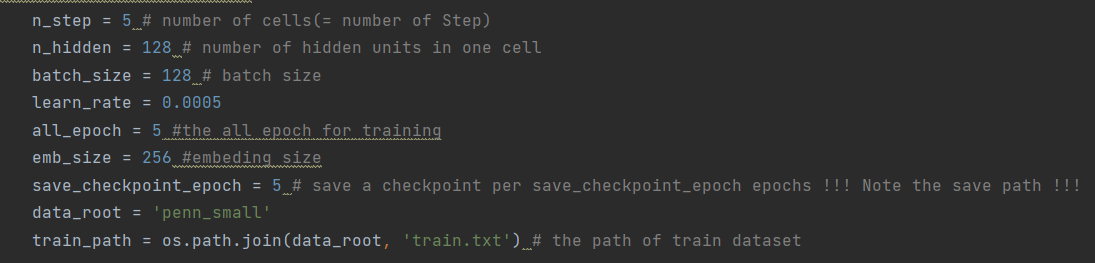


图4 项目参数声明

数据预处理：

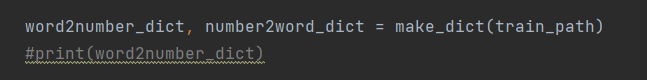


图5 数字-单词索引

将train.txt的数据处理为单词-数字（word2number\_dict）和数字-单词（number2word\_dict）的索引。

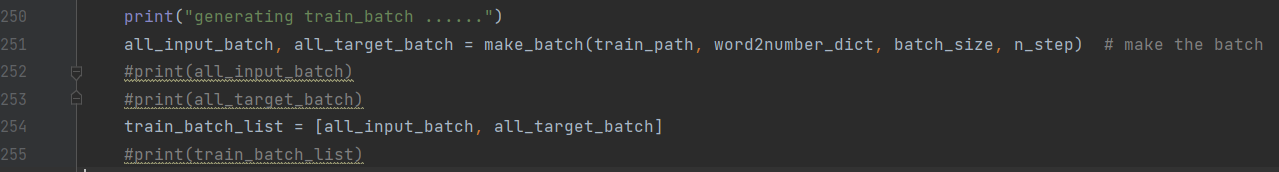


图6 获取输入和预测目标批次

将单词-数字索引转换为关于数字的输入批次（all\_input\_batch）和目标批次(all\_target\_batch)。Train\_batch\_list为两个批次一起的链表

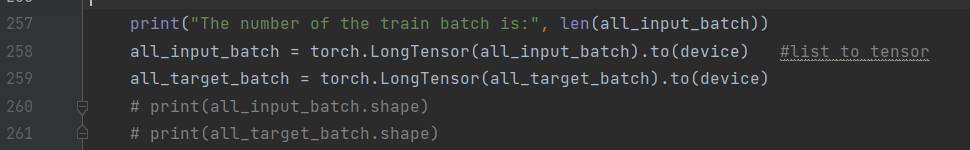


图7 数据转变为tensor

将上述的两个数字批次转换为tensor准备进行数据训练。

然后是train\_txt、

模型训练:

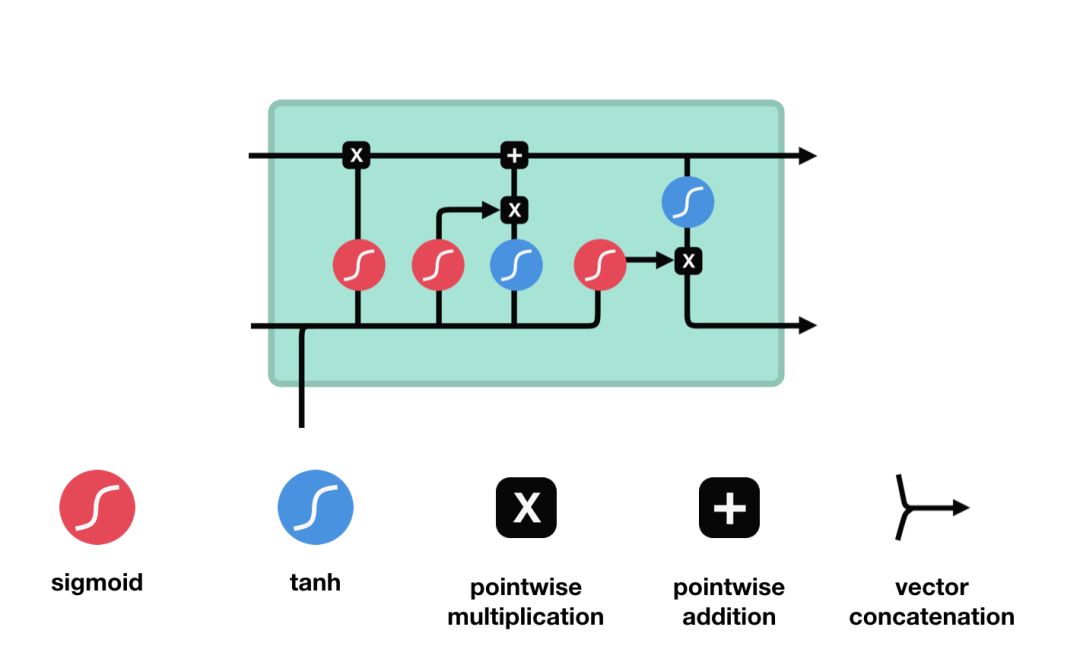


图8 LSTM模型

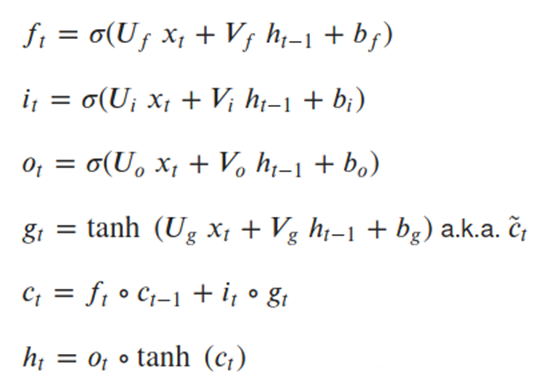


图9 模型参数公式计算

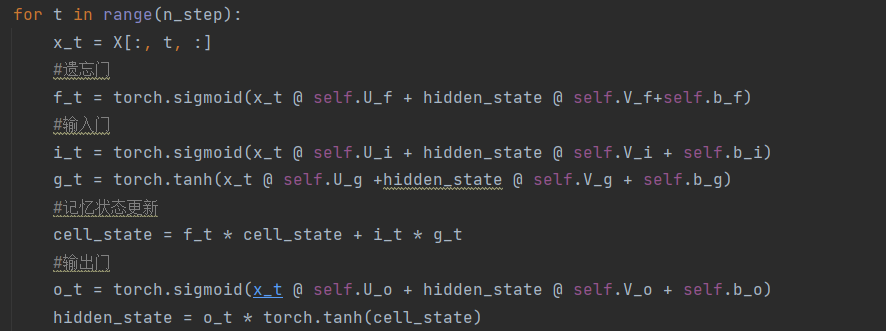


图10 公式代码实现

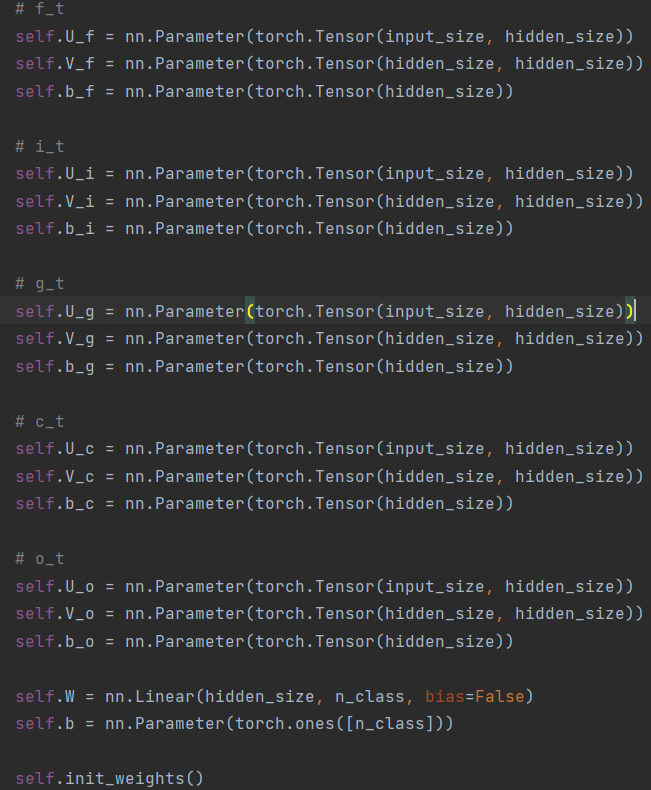


图11 LSTM模型参数声明

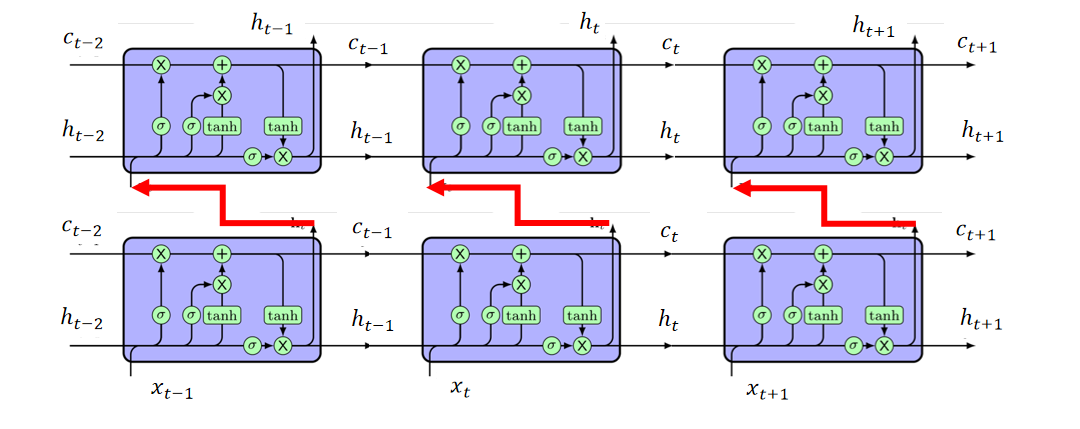


图12 双层LSTM模型

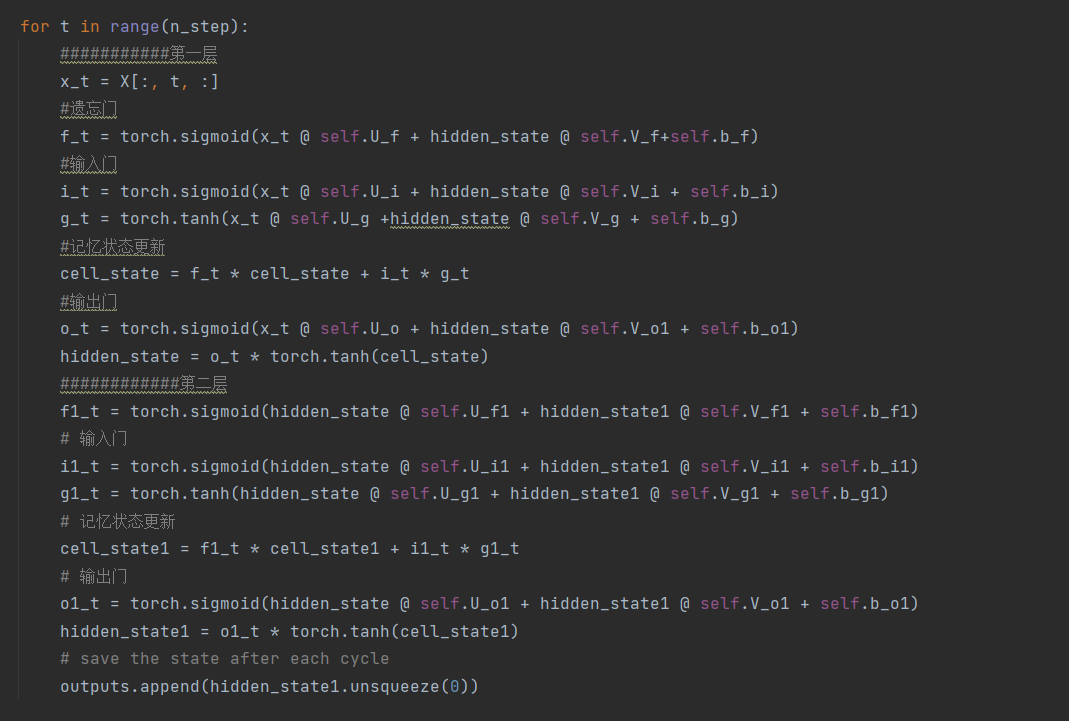


图13 双层门控代码实现

模型测试：即为上文的实验结果图一至图三

总结：在这次LSTM的实践中，就我个人而言实践的过程还是比较困难的，因为需要从“用轮子”转变到“造轮子”，需要去理解LSTM的模型和模型中的计算公式，在理解之后的读代码和实现代码上我也是花费了不少时间，时间主要用在理解数据预处理的方式和导入训练模型中数据的维度。然后如何才能和图9公式中的参数对应起来让我大费周折。不过方法总比困难多，就是因为实践过程中遇到了不少疑难，所以在完成实验后收获也一样颇丰。同时我也完成了拓展实现双层LSTM，这是我没想到的。毫不夸张地说，实践前我对LSTM的理论理解不是很多，通过对理论的实践才真正让我理解LSTM模型。

所以我知道自然语言处理课程虽然结束了，老师和师兄们的理论指导很详细很赞，但我也需要去通过自己的实践，才能真正掌握理论的精髓。希望以后还能多多交流学习