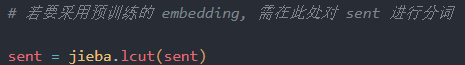
NLP实验报告2

周帆

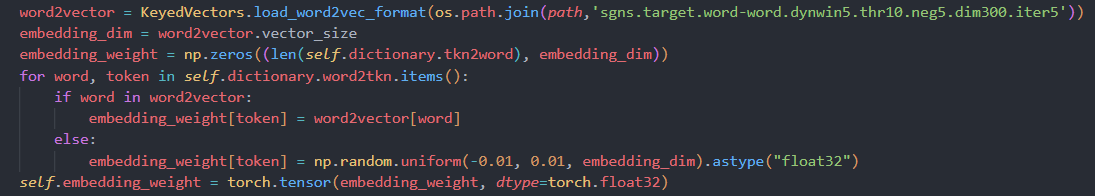
# 任务一：Bi-LSMT、Transformer调用预训练的word embedding

## 实现细节：

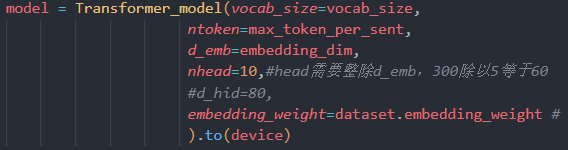
1. 数据预处理：分词采用jieba分词



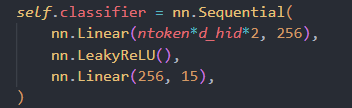
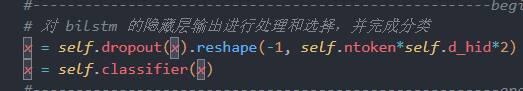
1. 构造embedding\_weight矩阵：特别注意这个矩阵的数据类型需要时float32



1. 实例化网络模型时，将embedding\_weight换成构造的token->embedding矩阵



1. 网络的分类器上，最简单的是把隐藏层展开，然后送到一层全连接层，然后输出长度为15的tensor作为概率

## 超参数对比

标准情况：transformer +embedding

Batch = 32

nhead = 10

epoch = 5

max\_token\_per\_sent = 50 # 每个句子预设的最大 token 数lr = 1e-4

embedding\_dim = dataset.embedding\_weight.shape[1] # 词向量维度,实际等于300

d\_hid=2048，#前馈网络宽度

nlayers=6, #前馈网络层数

dropout=0.2 #随机关闭概率

从表格可以看出batch\_size过小，训练效果特别不好，很容易学不到知识

|  |  |  |  |
| --- | --- | --- | --- |
| Transformer | 详细 | 最终train accurary | 最终Valid accurary |
| 标准情况 |  | 55.46% | 52.10% |
| Nhead=5 |  | 54.99% | 53.65% |
| Batch=16 |  | 11.10% | 10.89% |

## 需要注意的细节：

1. embedding.weight的数据类型要求是float32
2. 模型训练开始之前，要调用model.train(),为了启用 BatchNormalization 和 Dropout

模型测试开始之前，要调用model.eval() ,为了关掉BatchNormalization 和 Dropout，保证在测试过程中，不标准正态化测试数据，不随机关闭神经元，保证准确性

# 任务二：以不同的方法调用各模型的隐藏层输出，并送入分类器分类

标准情况是把隐藏层展成一维，然后接一层全连接层

|  |  |  |  |
| --- | --- | --- | --- |
| Transformer | | | |
|  | 详细 | Train accuracy | Valid accuracy |
| 标准情况 |  | 78.96% | 35.45% |
| 平均池化 |  | 73.80% | 35.39% |
| 最大池化 |  | 73.98% | 35.74% |
| 取最后一个时间步 |  | 77.46% | 36.24% |
| 取最后两个时间步  + 平均池化 |  | 81.56% | 36.21% |
| LSTM | | | |
|  | 详细 | Train accuracy | Valid accuracy |
| 标准情况 |  | 51.32% | 34.53% |
| 平均池化，  是对每个token对应的embedding的每一个维度池化 |  | 39.77 | 34.38% |
| 平均池化 |  | 33.98% | 30.55% |
| 最大池化 |  | 59.83% | 35.76% |
| 取最后一个时间步的embedding |  | 65.94% | 35.58% |
| 最后两个时间步的输出  +  平均池化 |  | 68.02% | 37.01% |

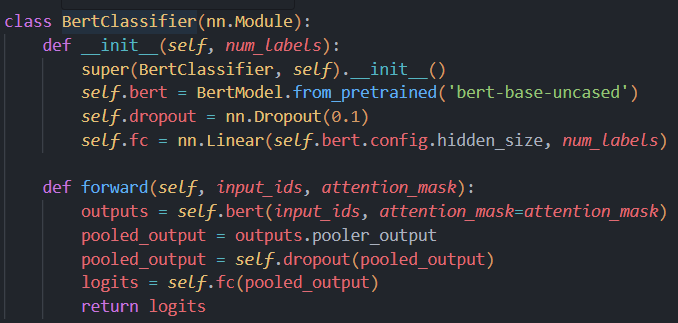
**任务三: 在Transformer中使用预训练参数**

直接调用BERT预训练模型，冻结预训练参数，只更新最后的分类层。

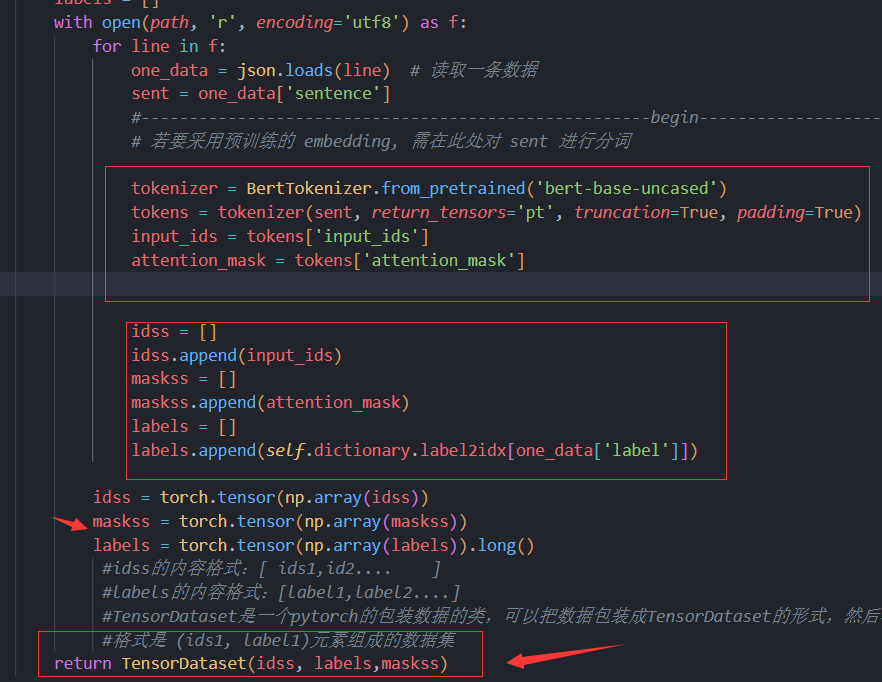
因为数据处理和训练时间过长，能跑起来，但是没有跑出结果。

## 实现细节

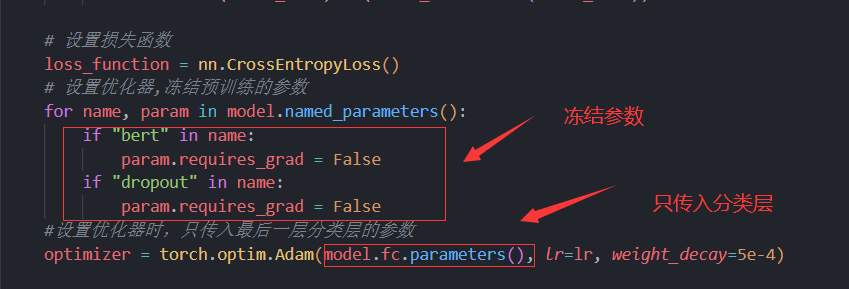
1. 使用预训模型，搭建网络层



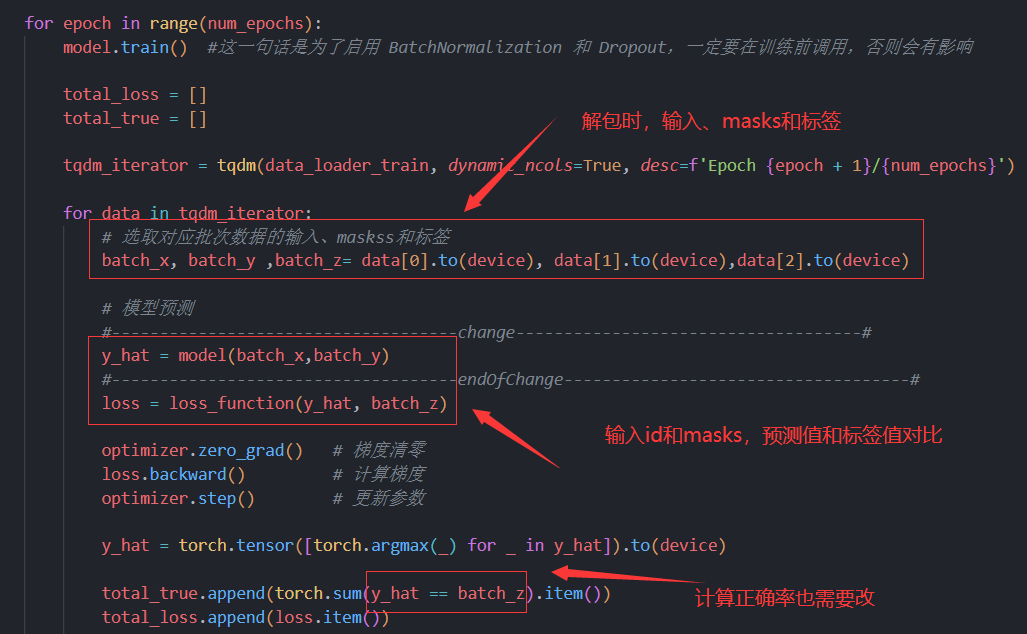
1. 改变数据处理函数，使用BERT的tokenizer（）



1. 更改训练优化器



1. 更改训练和测试过程中的输入方式和准确率计算方式



# 参考文献：

[1] <https://blog.csdn.net/jningwei/article/details/120300014>

参数冻结

[2] <https://blog.csdn.net/aideshizhe0/article/details/130719027>

BERT微调demo