基于Bert实现中文情感分类

汤佳伟20194617

# 概要

本次大作业，基于谷歌开发Bert模型，使用微博的情感分类数据集，训练了一个中文情感分类的模型，并通过尝试其他的Bert模型来达到更优的结果。

# 数据集说明

本次大作业使用的数据集包括六个分类分别是：neutral（无情绪，4887条）、happy（积极，4573条）、angry（愤怒，7093条）、sad（悲伤，4242条）、fear（恐惧，1037条）、surprise（惊奇，1774条），合计23606条数据。取其中85%为训练集，7.5%为开发集,7.5%为测试集。

# 步骤

1. 运行dataProcess\_nlp.py对原始数据进行切分，生成训练集、开发集、测试集。
2. 运行run\_classifier.py选择对应的数据集处理类NLPProcessor(参数为“nlp”)
3. 根据训练结果不断调整超参数使模型的效果更好
4. 尝试不同的中文bert模型，比较模型之间的性能差异
5. 生成的预测结果为各个分类的概率，运行process\_predict.py可以转换为最终的分类

# 模型训练

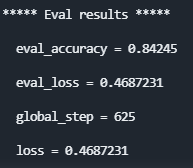
模型训练阶段共使用了三种bert模型进行训练，其中针对谷歌官方提供的中文bert模型进行了修改超参数并进行调优的过程。另外两个模型由于训练时间过长，难以反复调整参数进行训练，因此采用对谷歌官方的模型的训练参数中最长的轮数10。

1．模型chinese\_L-12\_H-768\_A-12（记作A）（https://storage.googleapis.com/bert\_models/2018\_11\_03/chinese\_L-12\_H-768\_A-12.zip）

超参数：



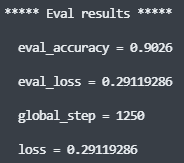
训练结果：



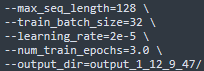
超参数：



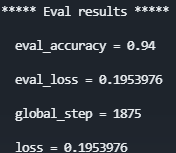
训练结果：



超参数：



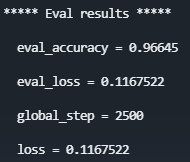
训练结果：



超参数：



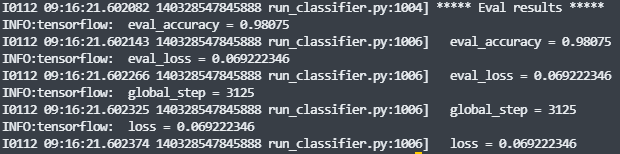
训练结果：



超参数：



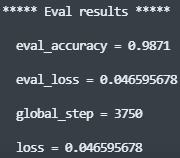
训练结果：



超参数：



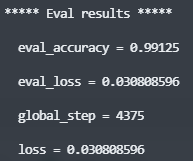
训练结果：



超参数：



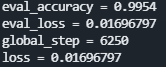
训练结果：



超参数：



训练结果

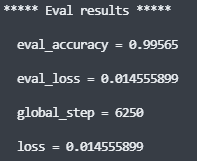


2.模型chinese\_wwm\_ext\_L-12\_H-768\_A-12（记作B）(https://drive.google.com/open?id=1buMLEjdtrXE2c4G1rpsNGWEx7lUQ0RHi)

超参数：



训练效果：



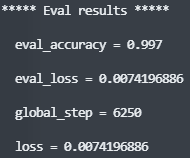
3.模型chinese\_roberta\_wwm\_large\_ext\_L-24\_H-1024\_A-16(记作C)

（https://storage.googleapis.com/bert\_models/2018\_11\_03/chinese\_L-12\_H-768\_A-12.zip）

超参数：



训练效果：



4.相同参数下不同模型的性能比较

选用参数：



性能比较：

|  |  |  |
| --- | --- | --- |
| 参数  模型 | eval\_accuracy | eval\_loss |
| A | 0.9954 | 0.0169 |
| B | 0.9957 | 0.0145 |
| C | 0.9970 | 0.0074 |

# 结论

Bert模型在中文情感分类上的表现非常好，同时通过多个模型的对比发现，谷歌官方的模型训练速度最快，但同时效果要稍差于另外两个模型。B模型与A模型具有相同模型参数，但是可能B模型用于训练的数据更加广泛，因此B模型的最终表现要比A模型好。同时，B模型与C模型使用相同的预训练数据，但是由于B模型具有更多的层数和更多的参数，因此在最终的训练效果上C模型相比于B模型有着更好的表现。

# 实验收获与心得体会

Bert模型在情感分类上的表现远远超出了我的预期，很难想象借助于深度学习模型能够将情感分类的工作做到如此高的准确率。在得到训练后的模型使用测试集进行测试，也得到了非常好的表现。同时，在查找资料的过程中也了解到Bert模型的应用非常的广泛不止局限于情感分类的工作，因此在后续的工作中我也会尝试使用Bert模型来取得更好地结果。

谷歌官方提供的Bert模型的demo具有非常详细的注释，因此在阅读的过程中遇到的障碍相对较少。同时代码中很多地方都尽可能的将循环的计算方式写作向量计算，更加充分的调用GPU以加速训练，这对于我后续在实际的项目中编写模型的帮助很大，能够有效的提高模型的训练速度。

同时样例的设计模式也非常的简洁清晰，在后续的开发中我也会模仿这种设计模式，保证自己的程序具有更好的可读性、可拓展性以及可维护性。