# 实验一：汉语句子的情感极性判断

# 实验报告

**vJanGo**

#### 使用Naïve Bayes方法实现

1. **实验原理**

Naïve Bayes 是一种基于**贝叶斯定理**的简单但有效的分类算法，它广泛应用于文本分类任务，包括情感分析。它之所以被称为 "Naïve"（朴素），是因为它假设特征之间是**条件独立的**，即每个特征对分类的贡献是独立的。

在本实验中，句子的情感极性分为两类：积极与消极，它们分别用1和0来表示。我们通过已经标注了情感极性的训练语料计算先验概率与条件概率，这个过程可以看做是对朴素贝叶斯模型的训练。之后，给定一个句子，计算该句子属于每个情感类别的后验概率，最后选择后验概率最大的类别作为该句子的情感极性。

1. **实验环境与数据**

硬件环境：Ubuntu20.04 + NVIDIA 3060

IDE:VSCode

实验数据：使用参考资源中的标注语料，训练集与测试集按8：2分开，前者有1600条数据，后者有400条数据。

1. **实验过程**
2. **.数据预处理**

在preprocess.py中输入原始语料，输出整理和分割后的训练集与数据集，其中，每条评论占一行，positive样本以1开头，negative样本以0开头。分割完毕后，读入停词表和训练集的正负向句子，之后使用jieba库将句子分词，并剔除句子中的停词，写入正负向词列表。

1. **.计算每个词的TF-IDF特征值，训练朴素贝叶斯模型**

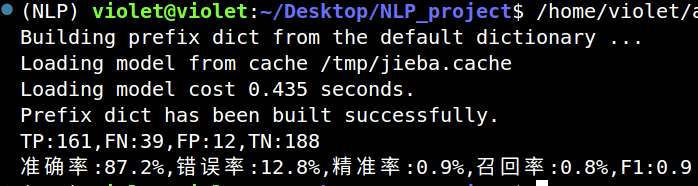
为了更好地衡量每个词语表示正向/负向的程度，我引入TF-IDF来表示词特征。对于词列表中的，每个词语，先计算其在相应分类句子里的词频TF（即出现次数占总次数的比例），再计算逆向文件频率 （IDF），由总句子数目除以包含该词语的句子的数目，再将得到的商取对数得到。得到每个词语的TF-IDF值，该模型也就训练完毕了。

1. **.分类与测试**

训练好模型，我们使用400条测试集来测试其性能。与训练集进行相同的预处理后，我们计算每个句子的极性。其中，遍历句子的词列表，若改词出现在正向词列表，句子的正向得分加上改词的TF-IDF的对数值；如果没出现，加上一个较小的对数值，做平滑处理。句子的负向得分如上得出，两值相减，即得出句子的极性得分，大于等于0为正，反之为负。

对于测试集，我们计算出全体的TP,FP,TN,FN值，来衡量模型的性能。

1. **实验结果**



实验结果如图所示，在400条句子的测试集上，准确率达到了87.2%，F1分数达到了0.9，说明朴素贝叶斯模型能较好地完成该分类任务。

1. **使用逻辑回归神经网络实现**
2. **实验原理**

逻辑回归基于线性回归模型，通过逻辑函数（）将线性模型的输出z映射到一个概率范围内，通常是 [0, 1] 区间，从而实现分类。其中，大于等于 0.5 的样本预测为正例，小于 0.5 的样本预测为负例。训练模型其实就是在找到最优的线性模型的参数。

1. **实验环境与数据**

硬件同上，实验数据：使用参考资源中的标注语料，训练集与测试集按8：2分开，前者有1600条数据，后者有400条数据。

1. **实验过程**
2. **.数据预处理**

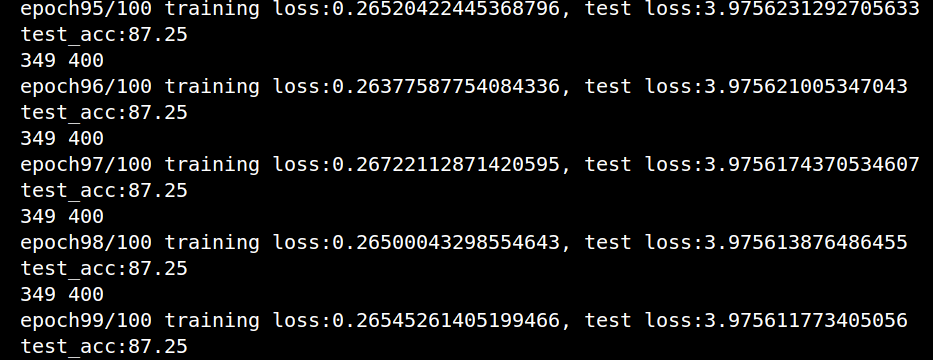
数据做了额外的处理，以符合神经网络的输入。其中，训练集与测试集依然按照8：2的比例分割，每个句子仍进行分词与停词处理为词列表。之后，使用Chinese Word Vectors中文词向量将每个词转换为对应的词向量表示。最终，每个句子长度固定为80（不足的padding，多余的截断），将所有词向量拼接起来，用一个固定长度的向量来表示句子。这就是模型的输入数据。

1. **.搭建神经网络**

首先随机初始化线性模型的权重与偏差，然后实现逻辑回归函数即sigmoid函数和损失函数，其中损失函数选用2分类的交叉熵损失函数。接着完成模型的前向传播部分，即计算线性模型通过sigmoid函数的输出值。为了调整线性模型的参数，还需要设置梯度下降模块，分别对权重和偏差求导，再乘以学习率，用原来的参数减去该值即可。

1. **.训练网络与测试**

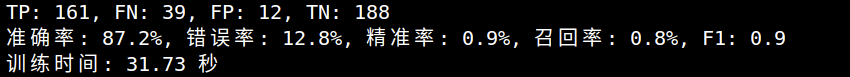
在数据预处理阶段，我们已经编写好了数据加载器，在训练阶段，使用加载器加载每个batch的数据，经过前向传播计算输出，用损失函数计算误差，最后通过梯度下降更新模型参数。每个epoch训练完后，在测试集上就算损失，输出训练和测试集上的损失。多个epoch后，训练损失基本不变，说明模型已经训练完成。



1. **实验结果**

训练完后，在测试集上验证模型精度，最终准确率为87.25%，说明逻辑回归模型很好地完成了任务。

1. **实验对照**



朴素贝叶斯训练结果



逻辑回归模型训练结果

上图给出了两个模型输出的准确率和训练时间，总结到下表中：

|  |  |  |
| --- | --- | --- |
| 模型 | 准确率 | 运行时间/s |
| 朴素贝叶斯 | 87.2% | 31.73 |
| 逻辑回归模型 | 87.25% | 10.74 |

总结得出，两种方法在给定的1600条训练集和400条测试集上的训练效果基本一致，准确率都达到了较高的87%左右。但是，逻辑回归模型在训练时间上更占优势，这可能因为朴素贝叶斯模型需要逐个计算词语的TF-IDF值，在此过程中消耗了大量时间。根据具体的实际情况，可以选择不同的模型使用。