基于感知机的中文分词

学 号：1400011059

姓 名：肖 威

1. **实验目的**

实现中文自动切词，把连续的中文文本切分成词序列。

1. **实验素材**

来自Sighan-2004中文切词国际比赛的标准数据：

1. 训练数据为train.txt，包含86924个句子，大小为16M；
2. 测试数据为test.txt，包含3985个句子，大小为0.6M；
3. 测试数据的答案为test.answer.txt，用于计算系统的Precision, Recall, F-score
4. **实现算法**

非结构化感知机算法：

1. 问题解构

对语句的分词的工作，可以解构为扫描句子中的每一个字，判断该不该将这个字和前一个字分开。

所以训练样本的单元（x, y）即为 ( 训练集中的每个字, 是否将该字与前一个字分开 )

1. 特征提取

如果只以这个字本身为特征，即为Unigram模型。由于现代汉语的字义复杂，且以词为基本意义单元，单粒度的特征效果不好。

本实验采取的特征提取为（当前字，前一个字+当前字，当前字+后一个字，前一个字+当前字+后一个字，）这样的四元组。在句子开头结尾则分别补充S、E两个标签作为前一个字和后一个字。

例如，句子[“ 吃 屎 的 东西 ， 连 一 捆 麦 也 铡 不 动 呀 ？] 中每个字（包括标点）对应的特征如下：

['“', 'S\_“', '“\_吃', 'S\_“\_吃'] ['吃', '“\_吃', '吃\_屎', '“\_吃\_屎']

['屎', '吃\_屎', '屎\_的', '吃\_屎\_的'] ['的', '屎\_的', '的\_东', '屎\_的\_东']

['东', '的\_东', '东\_西', '的\_东\_西'] ['西', '东\_西', '西\_，', '东\_西\_，']

['，', '西\_，', '，\_连', '西\_，\_连'] ['连', '，\_连', '连\_一', '，\_连\_一']

['一', '连\_一', '一\_捆', '连\_一\_捆'] ['捆', '一\_捆', '捆\_麦', '一\_捆\_麦']

['麦', '捆\_麦', '麦\_也', '捆\_麦\_也'] ['也', '麦\_也', '也\_铡', '麦\_也\_铡']

['铡', '也\_铡', '铡\_不', '也\_铡\_不'] ['不', '铡\_不', '不\_动', '铡\_不\_动']

['动', '不\_动', '动\_呀', '不\_动\_呀'] ['呀', '动\_呀', '呀\_？', '动\_呀\_？']

['？', '呀\_？', '？\_E', '呀\_？\_E']

同时，该句子中每一个字的对应的分词结果为：01111011111111111， 0代表与前一个字合并，1代表与前一个字分开， 参见：[“0 吃1 屎1 的1 东1西0 ，1 连1 一1 捆1 麦1 也1 铡1 不1 动1 呀1 ？1]

1. 感知机算法

感知机学习算法就是要学习一个线性映射，把每一个字在句子中的特征映射到0或1，从而判断是否将该字和前一个字分离。为了学习的方便，把映射的象集改为

具体方法如下：假设该线性函数为

其中 ，, , ，n为特征的总数，可以通过遍历整个训练集得到。x为元素为0或1的n维特征向量，对于某个训练样本单元，若某一特征出现，则x在该特征对应的维度的元素为1，在本实验中，由于每次只有4个特征，故x每次只有4个维度上元素为1，其余为0。

对于训练错误的样本，定义Loss为：

其中为真实值，当训练错误时，和异号，故L>0，其几何意义为错误的预估值到划分超平面之间的距离。

使用梯度下降法优化参数W：

其中为学习率，本实验取值为0.1

1. 平均化感知机

将不同训练次数的感知机的参数平均化，得到平均化感知机。

1. **结果分析**

在训练20次左右效果达到最佳，之后感知机在两个局部最优解之间震荡。

平均化的结果更稳定，也在20次批量训练后效果降低。

图标原始数据及代码见附件。