**《任务一：基于机器学习的文本分类》**

**一、任务介绍**

1.1 任务简介

本次任务是利用机器学习中的Softmax回归和Logistic回归来实现对文本的情感进行分类。

1.2 数据集

数据集来自Kaggle的“Sentiment Analysis on Movie Reviews”大赛，其中记录了大量电影评论以及其对应情感，情感主要分为五类：消极、有点消极、中立、有点积极、积极的。

1.3 任务步骤

该任务步骤主要包含：数据读取、特征提取、最优化求解和结果输出等。

**二、文本特征表示**

2.1 词袋模型（Bag-of-Word）

词袋模型主要考虑将句子拆解成一个一个单词，对于存在于句子的单词 (统一转换为大写或小写)并将其对应的向量位置上的元素设置为1，反之为0，通过这种方式，可以将一个句子转换为一个由数字表示的0-1向量。

例如：句子A：I like apple. 句子B：I hate apple. 拆散成单词后包括：I, LIKE, APPLE, HATE. 因此句子A对应的向量A表示为[1,1,1,0]，句子B对应的向量表示为[1,0,1,1]。

这种转换方式较为简单，但是它没有考虑词序，这可能导致两个意思完全不一样的句子对应的向量却是完全一样的。因此，词袋模型在这种情况下会存在较大的缺陷。

2.2 N元模型（N-Gram）

相较于前面的词袋模型，N元模型则考虑了词序问题。N元模型假定每个词与前面N-1个词有关，与更前面的词无关，因此组成了N元词组用于编码。例如，当N=2时，I love you 不再看作是 I, love, you 这三个单词，而是 I love, love you 这两个词组。

例如：句子A：John likes to watch movies. Mary likes too. 句子B：John also likes to watch football games. 以上两句可以构造一个词典：{1:"John likes", 2:"likes to", 3:"to watch", 4:"watch movies", 5:"Mary likes", 6:"likes too", 7:"John also", 8:"also likes", 9:"watch football", 10:"football games"}，因此句子A可以表示为[1, 1, 1, 1, 1, 1, 0, 0, 0, 0]，同理句子B可以表示为[0,1,1,0,0,0,1,1,1,1]。

这种转换方式较为复杂，但能将句子中的词序纳入考虑。随着N的大小增加，词表的大小会呈指数型膨胀，从而导致文本的特征数上涨，模型训练速度变慢。

**三、最优化求解**

3.1 Logistic回归介绍

Logistic回归是一种常用的处理二分类问题的线性模型。在二分类问题中，结果y主要分成两类，正类和负类。因变量y∈{0, 1}，其中0是负类，1是正类。线性回归的输出值在负无穷到正无穷的范围上，无法区分是正类还是负类。因此需要引入非线性变换，把线性回归的输出值压缩到（0, 1）之间，因此便形成了Logistic回归，，当时，预测结果为正类，反之为负类。

3.2 Softmax回归介绍

相较于Logistic回归，Softmax回归主要用于处理多分类问题。其中给定一个输入x，输出一个向量y，y的长度即为类别的个数，总和为1，y中的数字是各类别的概率。假设给定C个类别，，其概率计算公式为：

3.3 损失函数

在确定模型后，为了量化模型的好坏，需要对损失函数进行定义。在Logistic回归和Softmax回归中主要采用交叉熵损失函数。交叉熵的具体定义为：

其中N表示类别数，表示真实的标签并用 one-hot 编码表示，只有目标类别对应的位置为1，其他位置为0，表示模型预测的概率。当真实类别的越接近1时越小，损失也就越小。通过寻找损失函数的最小值便可以得到最优参数矩阵。

3.4 梯度下降

梯度下降的基本思想是，对于每个固定的参数，求其梯度并利用梯度实现参数的更新。梯度下降方法主要包含三种：Batch，Shuffle和Mini-batch。

其中，Batch是指整批量梯度下降法，即选取所有样本用于计算梯度：

Shuffle是指随机梯度下降法，即每次随机选取一个样本用于计算梯度：

Mini-batch是指小批量梯度下降法，即每次选取一些样本（如K个，构成样本集合S）用于计算梯度：

三种梯度计算方法中，Batch计算梯度准确，但计算开销较大；Shuffle计算较为简单，但梯度估计可能不准确；Mini-Batch综合了Batch和Shuffle，梯度较为准确同时也兼顾了效率。

在完成梯度计算后，利用梯度实现参数的更新从而进一步逼近最优解：

**四、实验结果**

4.1 实验设置

样本个数：10000

训练集占比：80%

学习率：[0.001, 0.01, 0.1, 1]

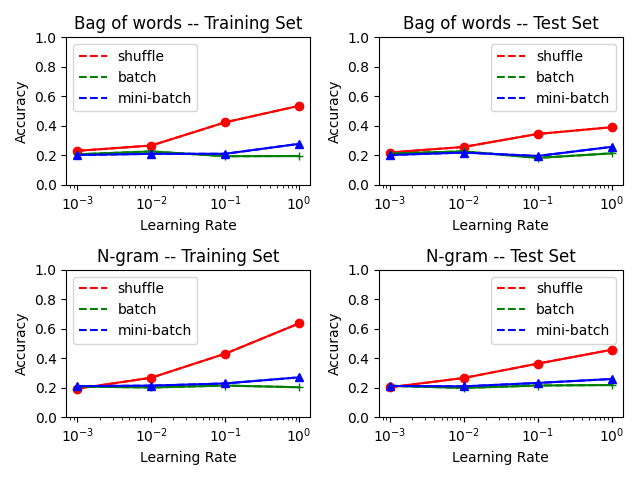
Mini-Batch大小：样本的1%即100

由于Logistic回归通常用于二分类问题，所以实验中仅使用Softmax回归进行分类预测。

4.2 结果展示

4.2.1 实验一

首先进行计算梯度次数总数为10000的实验，结果如下图：

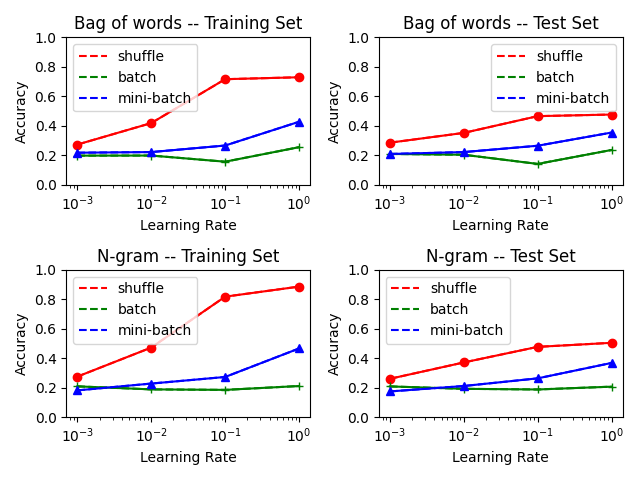


从训练集上的结果中可以看到，对于不同的特征，N元模型明显要优于词袋模型，这是因为N元模型相较于词袋模型充分考虑了句子中的词序。同时，针对不同学习率的训练结果进行对比，可以看到，较小的学习率无法使模型收敛，相比之下较大的学习率表现较好。由于学习率大于1时可能会出现震荡现象从而错过最优解，因此本实验选取的学习率均为大于1。此外，对比不同的梯度下降方法，除在学习率比较小无法收敛的情况外，mini-batch和batch的表现几乎相同，而shuffle的表现最好，这可能是因为N元模型和词袋模型所构建的特征通常是高维稀疏的，batch和mini-batch在更新时会遍历较多样本，从而使梯度方向容易被高频词所主导，最终导致了较差的预测精度。

对比测试集上的结果，结论与训练集上的结论基本相同。基于以上的分析可以得出以下结论，即使用N元模型及随机梯度下降，并且将学习率设置为1左右时，对于本数据集的Softmax模型分类精度最优。

4.2.2 实验二

接下来调整计算梯度次数总数为100000来观察模型的训练结果，结果如下图：



首先，实验结果和实验一的结论基本相符，训练集的准确率已超过80%，但这意味着随着训练次数的增加，模型在训练集上的拟合效果也变得越来越好。同时，无论是实验一还是实验二，测试集的最高准确率都在50%附近，这意味着实验二所增加的梯度计算次数最终仅导致了模型在训练集上的过拟合，并未对模型的泛用性产生提升效果。因此，随着训练次数的增加尽管在训练集的准确率有所提升，但是过多的训练次数最终只会使模型过拟合而未能带来文本分类效果提升。