编程作业说明：神经网络

任务一：使用 Logistic 回归估计马疝病的死亡率

训练集：包含于文件 horseColicTraining.txt 中，用于训练得到模型的最佳系数。训练集包含 299 个样本（299 行），每个样本含有 21 个特征（前 21 列），这些特征包含医院检测马疝病的指标；最后 1 列为类别标签，表示病马的死亡情况；部分样本含有缺失值；

测试集：包含于文件 horseColicTest.txt 中，通过预测测试样本中病马的死亡情况，来评估训练模型的优劣。测试集包含 69 个样本，部分样本部分特征缺失；最后 1 列为类别标签，用于计算错误率。

特征值缺失：在该示例中，可以选择实数 0 来替换所有缺失值，理由是：1，在系数更新时不会影响系数的值（后面会介绍）；2，由于 sigmoid(0)=0.5，对结果的预测不具有任何倾向性；3，该数据集中的特征值一般不为 0，因此某种意义上也满足‘特殊值’这个要求。

训练集中类别标签缺失：在 Logistic 回归中，可将该样本直接丢弃。

所需提交材料：任务一需要编程实现 Logistic 回归完成对测试集的预测，并且计算得到精度，编写实验报告简述算法原理，实验结果等。

任务二：使用神经网络完成新闻分类

（1）数据集讲解：

该数据集用于文本分类，包括大约 20000 个左右的新闻文档，均匀分为 20 个不同主题的新闻组集合，其中：

训练集：包括 11314 个新闻文档及其主题分类标签。训练数据在文件 train 目录下，训练新闻文档在 train_texts.dat 文件中，训练新闻文档标签在 train_labels.txt 文档中，编号为 0~19，表示该文档分属的主题标号。

测试集：包括 7532 个新闻文档，标签并未给出。测试集文件在 test 目录下，测试集新闻文档在 test_texts.dat 文件中。

（2）实验思路：

数据集读取：读取文件 train_texts.dat 和 test_texts.dat 方式如下，以 train_texts.dat 为例，test_texts.dat 读取方式相同：

```
import pickle
file_name = 'train_texts.dat'
with open(file_name, 'rb') as f:
    train_texts = pickle.load(f)
```

标签文件为正常 txt 文件，读取方式按照读取 txt 文件即可。

文档特征提取：

因为每篇新闻都是由英文字符表示而成，因此需要首先提取每篇文档的特征，把每篇文档抽取为特征向量，可以采用多种特征提取方式，这里给出建议为提取文档的 TF-IDF 特征，即词频（TF）-逆文本频率（IDF），TF-IDF 简介如下：

原理 [编辑]

在一份给定的文件里，**词频**（term frequency，tf）指的是某一个给定的词语在该文件中出现的频率。这个数字是对**词数**（term count）的归一化，以防止它偏向长的文件。（同一个词语在长文件里可能会比短文件有更高的词数，而不管该词语重要与否。）对于在某一特定文件里的词语t_i来说，它的重要性可表示为：

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

以上式子中$n_{i,j}$是该词在文件d_j中的出现次数，而分母则是在文件d_j中所有字词的出现次数之和。

逆向文件频率（inverse document frequency，idf）是一个词语普遍重要性的度量。某一特定词语的idf，可以由总文件数目除以包含该词语之文件的数目，再将得到的商取以10为底的对数得到：

$$\mathrm{idf}_i = \lg \frac{|D|}{|\{j : t_i \in d_j\}|}$$

其中

- |D|：语料库中的文件总数
- $|\{j : t_i \in d_j\}|$：包含词语t_i的文件数目（即$n_{i,j} \neq 0$的文件数目）如果词语不在数据中，就导致分母为零，因此一般情况下使用$1 + |\{j : t_i \in d_j\}|$

然后

$$\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \times \mathrm{idf}_i$$

某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的tf-idf。因此，tf-idf倾向于过滤掉常见的词语，保留重要的词语。

由于这不是实验的重点，提取文档的 TF-IDF 特征可以通过 sklearn.feature_extraction.text 中的 TfidfVectorizer 来完成，具体实现代码如下：

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=10000)
vectors_train = vectorizer.fit_transform(train_texts)
```

其中的 train_texts 为之前从文件中读取的训练集新闻文档，max_features=10000 表示每个文档只提取 10000 维特征，该值设置过大会对提取的特征引入噪声，同时然后续计算变得更加复杂，设置过小则不能很好的保留每个文档的特征，这里建议初始实验设置为 10000 即可，后续完成实验后，可以自行调节。

（上述文档的 TF-IDF 特征提取仅仅为一种特征提取方式，其它特征提取方式也

可以进行尝试）

后续算法：在完成每篇新闻文档的特征提取后，就可以构建神经网络模型进行训练，具体的网络模型结构，大家可以自行尝试。

（3）实验要求

预测测试集新闻文档分类结果，将预测结果与训练集标签相同的存储方式进行存储，存储为 txt 文件，每一行表示一个新闻的分类标号，将预测结果发送到邮箱 zhangzizhuo@hust.edu.cn。需编写实验报告说明具体实验流程，模型结构等。

两次任务的实验报告编写在一份实验报告中。

（希望同学们得出理想的实验结果，学习到更多的知识！）