# 文本分类总结

## 引言

为期四周的考核里，我围绕文本分类这一任务，进行了对深度学习的探索。

## 学习路线

#### 认识神经网络和深度学习

文本分类任务的核心是搭建并训练一个神经网络来达到分类器的目的，在此之前需要先了解什么是神经网络以及神经网络是如何工作的，我通过《机器学习》（周志华）这本书以及吴恩达的深度学习视频了解这部分的知识。

#### 认识tensorflow和keras以及如何搭建神经网络

·学习张量的概念及其运算

·学习tensorflow的一些常用函数

·我通过keras官方文档了解如何使用tf.keras框架搭建和训练神经网络，学习了如何使用keras函数式API定义模型，以及用类class搭建和训练神经网络结构的步骤。

#### 了解文本分类流程



#### 文本数据预处理

#### 学习词嵌入

#### 1.6 搭建卷积神经网络

## 2 文本数据预处理

这一部分主要的操作有：分词、去停用词、去标点符号、数字、构建词表（映射表）。首先将训练集整篇的英文文本按空格分词，并去掉其中没有实际意义的、会影响模型的判断的词，这里我从一篇博客中下载了一份“英文停用词表”，根据这份词表来去除文本中的停用词。然后再去掉标点符号、数字，得到一个包含整个文本的单词的列表，再根据这个列表得到词表。然后是给词表中的每个词编号，得到一个映射表即词典。最后再根据这个词典，把文本中的词都换为用数字表示。

实现过程基本都是对列表和字典的操作。

## Word2vec

这一部分的主要任务是：基于神经网络学习单词表示，将上一部分用编号表示的单词转为用向量表示，同时要满足特征相似的单词的词向量在空间中的分布应较为接近。Word

2vec利用给定单词的上下词来学习词嵌入，我使用了skip-gram模型来实现Word2vec。

#### 3.1 skip-gram

Skip-gram是一种利用文本单词上下文来学习好的词嵌入的算法，能够使词向量根据单词含义很好地组织起来。

##### 3.1.1 构建数据集

首先，需要设计一种方法来提取可以送入学习模型的数据集，这样的数据集应该是格式为（输入，输出）这样的一组元组。对词典中的每一个单词，指定一个窗口大小，根据这个窗口大小获取该单词的上下文单词，然后把这个单词和这些上下文单词一一打包为输入输出元组的格式。例如，若文本为：The dog barked at the mailman. 令上下文窗口大小为1，则应构建数据集为：[(dog,the), (dog,barked), (barked,dog), (barked,at), …(the,at), (the,mailman)]。

##### 3.1.2 使用神经网络学习词嵌入

有了上一步骤的数据格式后，就可以使用神经网络来学习词嵌入了。首先，确定学习词嵌入所需的变量。为了存储词嵌入，需要一个V×D的矩阵E，其中V是词典大小，D是词向量维度，这个矩阵也就是嵌入层。然后是softmax层，权重矩阵大小为D×V，偏执大小为V。

将每个词表示为大小为V的独热编码向量，所以输入和输出都是大小为V的向量。根据以下公式来预测输出：

·表示矩阵E乘以输入x的独热编码向量再加上偏置项

·为与t有关的参数向量的转置

把喂入一个softmax单元，预测不同目标词的概率，再通过优化损失函数，最终得到矩阵E中所有的经过训练的参数，来作为词向量。

##### 3.1.3 用负采样提高训练效率

考虑参数过多，softmax的计算非常缓慢，需要在不失去模型效果的前提下寻找有效的训练方法，我选的是对softmax层进行负采样：随机选择少量的负样本（我选的是16个）来更新权重，这时正样本对应的权重依然会更新。因为概率模型是不需要的，我们的目的是学习词嵌入，因此这一操作可以在可行的时间内学习到良好的词嵌入并对词嵌入模型不造成影响。

##### 3.1.4 代码实现

使用了keras函数式API定义模型。Embedding层的矩阵相乘使用到了keras后端函数。

#### 提取参数

从skip-gram模型中训练出的E矩阵的参数提取出来，就是需要用到的词向量了。同时将这些参数保存到文件中，就可以将这个词向量运用到别的地方了。

## 文本分类器

这一部分我通过搭建卷积神经网络来完成文本分类器

#### 4.1 卷积神经网络

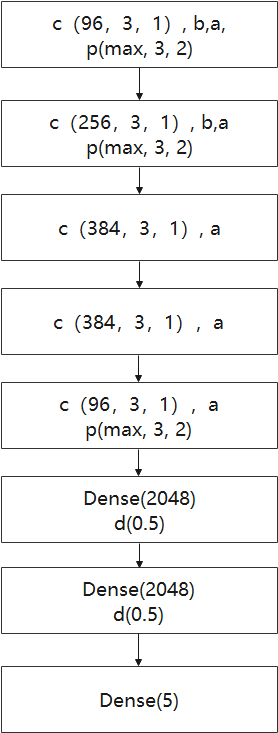
· CNN由卷积层、池化层、全连接层、以及其相关的超参数（如卷积核大小、填充、步幅等）构成，这些不同的组成部分，是CNN与全连接网络的不同之处。

* 卷积——特征提取器，卷积计算是一种有效的特征提取方法，它对原始数据进行特征提取，把提取出来的特征送给全连接网络。
* 优点：CNN中的卷积层的参数更少，可以构建更深层的模型。

#### 4.2 网络结构

##### 4.2.1 AlexNet

我搭建的第一个模型是一个仅有一层卷积的简单神经网络，为了加大特征提取，我参考AlexNet搭建了多层卷积的神经网络，又在此基础上调整超参数与全连接层个数，其网络结构简化图如下。



·c为卷积计算(括号内依次为卷积核个数，尺寸，步长)

·b为批标准化

·a为激活函数（均使用relu）

·p为池化（括号内数字以此为池化核尺寸，步长）

·d为Dropout（括号内为Dropout比例）

##### 4.2.2 VGGNet

由于AlexNet的模型预测效果不理想，我又尝试了参考VGGNet，在网络结构中增加了更多卷积层数，并再次调整全连接层个数与超参数，可惜实验效果仍不理想。我想再尝试更多其他的模型，遗憾的是我安排不出时间来实现了，所以只能到这里。

#### 4.3 实验过程

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验次序 | 卷积层个数 | 池化层个数 | 全连接层个数 | 学习率 | 有无正则化（L2） | 准确率情况（训练/验证） |
| 1 | 1 | 1 | 2 | 0.001 | 无 | 0.5/0.28 |
| 2（AlexNet） | 5 | 3 | 2 | 0.001 | 无 | 不收敛 |
| 3（AlexNet） | 6 | 3 | 1 | 0.001 | 无 | 0.8/0.4 |
| 4（AlexNet） | 6 | 3 | 1 | 0.001 | 有 | 0.8/0.4 |
| 4（VGGNet） | 13 | 5 | 2 | 0.001 | 无 | 不收敛 |
| 5（VGGNet） | 12 | 5 | 1 | 0.0005 | 有 | 0.8/0.4 |

其实实验次数不止这些，但是由于始终没有办法解决过拟合的问题，我并没有将所有结果为重复的实验也记录下来。

## 反思总结

有些遗憾虽然我做完了整个任务但是完成的效果并不理想，一轮考核的时候我说我希望自己能够在时间安排上做的好一点，不要到了最后一天还在赶文档，然而结果却是现在是最后一天的下午了，我还是在赶文档、ppt。所以，自我反思的第一点是我以后做事情还是要多给自己预留一点时间。

不过总的来说我还是挺开心自己能够完成考核的，我之前一直担心自己坚持不下来跑路。我从这次考核中收获到了不少东西，非常感谢师兄和数智工作室给我这样一个宝贵的学习机会。往后我也将在学有余力之时继续对深度学习这门学科的探索。

其实是想多写一点丰富一下文档的内容的，但是我该提交了呜呜。