|  |
| --- |
| 哈尔滨工业大学(深圳) |
| **《模式识别》实验报告** |
|  |
| 基于CNN的影评情感分类  学 院: 计算机科学与技术   |  |  | | --- | --- | | 姓 名: | 梁成 | | 学 号: | 1190301804 | | 专 业: | 计算机科学与技术 | | 日 期: | 2021-10-25 | |

基于CNN的影评情感分类

# 前言

## 背景介绍

情感分类是情感分析技术的核心问题，其目标是判断评论中的情感取向，按区分情感的粒度可分为两种分类问题：

1）正/负（positive/negative）二分类或者正面/负面/中立（positive/negative/neutral）三分类。

2）多元分类，如对新闻评论进行“乐观”、“悲伤”、“愤怒”、“惊讶”四元情感分类，对商品评论进行1星~5星五元情感分类等。

本实验处理句子的情感极性分类问题，也就是二分类问题，主要内容包括：

1. 编写词表映射
2. 数据处理
3. 文本表示
4. 模型的训练与测试

## 数据集介绍

NLTK（Natural Language Toolkit）是一个 Python 模块，提供了多种语料库（Corpora）和词典（Lexicon）资源，以及一系列基本的自然语言处理工具集，包括：分句、标记解析（Tokenization）、词干提取（Stemming）、词性标注（POS Tagging）、句法分析（Syntactic Parsing）等等，是对英文文本数据进行处理的常用工具。

本实验从NLTK库中取出了sentence polarity 语料库进行处理，sentence polarity 含有上万条关于影评的极性情感分类的标注数据，每一条数据由切分好的单词序列与对应的感情类标签构成。

样例如下。

[‘this’,’movie’,’is’,’good’][‘pos’]

[‘this’,’movie’,’is’,’bad’][‘neg’]

# CNN部分原理介绍

### 2.1 卷积核提取局部特征

在多层感知器中，每层输入的各个元素都需要乘以一个独立的参数（权重），这又叫做全连接层（Fully Connected Layer）或稠密层（Dense Layer）。然而对于某些类型的任务这样做并不合适，如在图像识别任务中，如果对每个像素赋予独立的参数，那么对于待识别物体位置的轻微移动，则识别结果可能发生较大的变化。在自然语言处理任务中也存在类似的问题，如对于情感分类，句子的情感极性往往是由个别词或短语决定的，而这些决定性的词或短语在句子中的位置并不固定，使用全连接层很难捕捉这种关键的局部信息。

为了解决以上问题，一个非常直接的想法就是使用一个小的稠密层提取这些局部特征，如图像中固定大小的像素区域，文本中词的 N-gram 等。为了解决关键信息位置不固定的问题，可以依次扫描输入中的每个区域，该操作又被称为卷积（Convolution）操作。其中每个小的，用于提取局部特征的稠密层又被称为卷积核（Kernel）或者滤波器（Filter）

### 2.2 池化层聚合卷积输出

卷积操作输出的结果还可以经过进一步聚合，又被称为池化（Pooling），常用的池化操作有最大池化、平均池化、加和池化等。以最大池化为例，其含义是仅保留最有意义的局部特征。如在情感分类中，保留的是句子中对于分类最关键的 N-gram 信息。池化操作的另一个好处是可以解决样本的输入大小不一致的问题，如对于情感分类，有的句子比较长，有的句子比较短，因此包含的 N-gram 数目并不相同，导致抽取的局部特征个数也不相同，经过池化后，可以保证最终输出相同个数的特征。

然而，如果仅使用一个卷积核，只能提取单一类型的局部特征，而在实际问题中，往往需要提取很多种局部特征，如在情感分类中不同的情感词或者词组等。因此，在进行卷积操作时，可以使用多个卷积核提取不同种类的局部特征。卷积核的构造方式大致有两种，一种是使用不同组的参数，并且使用不同的初始化参数，来获得不同的卷积核；另一种是提取不同尺度的局部特征，如情感分类中提取不同大小的 N-gram

### 2.3 全连接层进行最终决策

多个卷积核输出多个特征，那么这些特征对于最终分类结果的判断，到底哪些比较重要，哪些不重要呢？只要再经过一个全连接的分类层就可以进行最终的决策。最后，还可以将多个卷积层加池化层堆叠起来，形成更深层的网络。这些网络统称为卷积神经网络（Convolutional Neural Network，CNN）。

# 实验过程

### 3.1 词表映射

词表映射是指，将需要将输入的语言符号，映射为一个大于等于0，小于等于词表大小的整数。词表映射得到的值，也被称为一个标记的索引或者下标。词表映射能够将字符转化为数字，便于后面进一步转化为tensor处理。

词表映射定义在vocab类中。

vocab初始化：

1. 用列表存储所有的标记，根据索引值获取相应的标记
2. 用词典实现标记到索引的映射

Vocab方法：

1. Len(): 返回词表大小
2. Getitem(self,token ): 返回标记对应的索引值
3. Convert\_to\_id (self,tokens): 查找多个标记对应的索引值

### 3.2 数据处理

在训练模型之前，我们需要获取数据并对数据进行一定处理。数据预处理没有标准的流程，通常针对不同的任务和数据集属性的不同而不同。在本实验中，数据处理的过程大致如下：

1. 为了模型测试，先选择交叉验证方法，提前划分好数据。取出sense\_polarity 数据集中的所有数据（10662条），取前4000句作为训练数据，剩余部分作为测试数据。

2. 使用创建的词表将标记映射为相应的索引值。

3. 对分类标签pos,neg进行处理，pos对应0，neg对应1。

4. 每一个样例构造一个元组，内容为索引列表与标签

5. 将元组转换为Tensor张量

### 3.3文本表示

自定义collate\_fn函数，对一个批次的数据进行整理，使其转换为张量。Collate\_fn可以嵌入到pytorch.dataloader中使用。

通过pytorch定义的dataloader函数，就能够直接把多个token转换为tensor进行处理。

### 3.4模型的训练与评估

#### 3.4.1 模型的初始化

超参数设置：

embedding\_dim = 128

hidden\_dim = 256

num\_class = 2

batch\_size = 32

num\_epoch = 5

词嵌入层：大小为词表大小，维度为128（超参数设置）

线性变换层：使用torch.Conv1d

激活函数：relu

线性变换层：filter\_num🡪2(输出pos,neg对应的概率，维度为2)

前向传播函数：

def forward(self, inputs):

embedding = self.embedding(inputs)

convolution = self.activate(self.conv1d(embedding.permute(0, 2, 1)))

pooling = F.max\_pool1d(convolution, kernel\_size=convolution.shape[2])

outputs = self.linear(pooling.squeeze(dim=2))

log\_probs = F.log\_softmax(outputs, dim=1)

return log\_probs

#### 3.4.2 模型的训练与评估

训练所用参数：

Num\_epoch =5

Optim=Adam()

Batch\_size=32

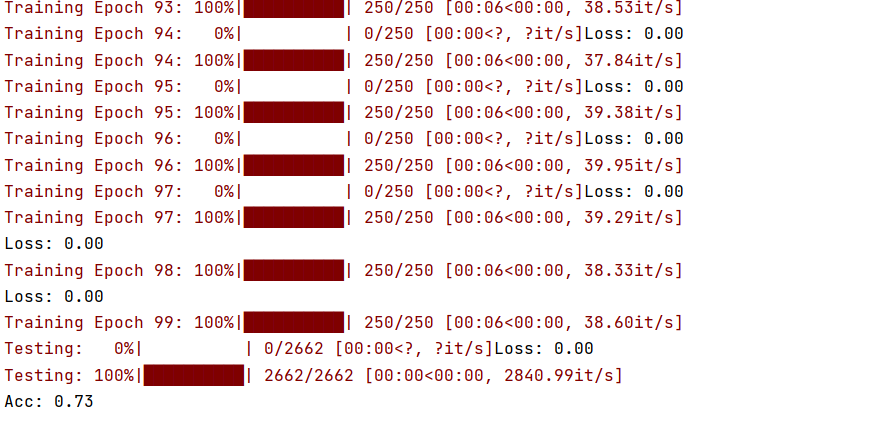
如何判别一个分类模型是否真正有效？我们可以利用测试集进行判断，对测试集的图片进行预测，并与正确类别进行比较，衡量分类器性能。常用指标有： 准确率（Accuracy）、精确度（Precision）、召回率（Recall）、F1 值（F1 score）以 及 Receiver Operating Characteristic（ROC）。

假设：TP （实际为正例，预测也为正例）、FP（实际为负例，预测为正例， FN（实际为正例，预测为负例）、TN（实际为负例，预测也为负例）。那么：

Accuracy＝ (TP + TN) / 总样本。 定义是：对于给定的测试数据集，分类器正确分类的样本数与总样本数之比。

实验测试结果如下：

Accuracy=0.73



# 附录

## 主流的情感分类方法

按技术路线分为5类：

1）基于词典的方法

2）基于机器学习的方法

3）词典与机器学习混合的方法

4）基于弱标注信息的方法

5）基于深度学习的方法

## 4.1 基于词典（Lexicon-based）的情感分类方法

基于词典方法的核心模式是“词典+规则”，即以情感词典作为判断情感极性的主要依据，同时兼顾评论数据中的句法结构，设计相应的判断规则（如but从句与主句情感极性相反） 基于词典的情感分类方法本质上依赖于情感词典和判断规则的质量，而两者都需要人工设计。因此这类方法的优劣很大程度上取决于人工设计和先验知识，推广能力较差。

## 4.2 基于机器学习的情感分类方法

基于机器学习技术的情感分类研究工作： 1）特征工程是此类研究工作的核心。情感分类任务中常用到的特征有n-gram特征（unigrams，bigrams，trigrams）、Part-of-Speech（POS）特征、句法特征、TF-IDF特征等，但是该方法依赖人工设计，受人为因素影响，推广能力差，在某一领域表现优秀的特征不一定在其他领域也表现优秀 2）基于机器学习的情感分类方法多使用经典分类模型如支持向量机、朴素贝叶斯、最大熵模型等，其中多数分类模型的性能依赖于标注数据集的质量，而获取高质量的标注数据需要耗费大量的人工成本。

## 4.3 词典与机器学习混合的情感分类方法

这种混合的思路有两种： 1）将“词典+规则”视为简单的分类器，然后融合多种不同分类器进行情感分类 2）将词典信息作为一种特征与现有特征（句法特征、POS特征等）进行结合，然后选择最优的特征组合进行情感分类。

## 4.4 基于弱标注信息的情感分类方法

从用户产生的数据中挖掘有助于训练情感分类器的信息，如评论的评分、微博中的表情符号等。由于互联网用户的“标注”行为没有统一标准，具有较大的随意性，所以将这种标注信息称为弱标注信息。

## 4.5 基于深度学习的情感分类方法

针对情感分类问题的机器学习方法有两个步骤： 1）从海量评论语料中学习出语义词向量 2）通过不同的语义合成方法用词向量得到所对应句子或文档的特征表达。

# 改进方向

1. 在CNN上可以尝试更多的超参数组合，得到更高的准确度。
2. 拿传统的分类器的效果与CNN对比。
3. 情感分类还有其他更好的方法，如RNN、Transformer等。

# 附录

实验所有的代码，已经上传至

<https://github.com/1190301804/1190301804-2021-Falls-HITSZ-Courses-/tree/master/%E6%A8%A1%E5%BC%8F%E8%AF%86%E5%88%AB>