基于机器学习进行文本情感分析

郭家琛、王珺戈、肖文睿、朱懿淳

摘 要 \*本论文提出了一种基于多层感知器（MLP）的文本情感分析方法。通过词袋模型、TF-IDF和n-gram提取文本特征，并结合降维技术优化特征表示，MLP 模型被用于高效分类。实验结果表明，MLP 方法在情感分类任务中表现出色，显著提升了模型的准确性和稳定性。\*

关键词 机器学习 文本情感分析 Softmax回归 梯度下降 神经网络

# 1 引言

在当今的信息化社会中，随着社交媒体、新闻平台和电子商务网站的普及，生成了大量的文本数据。这些数据不仅包含着事实信息，还蕴含着人们的情感与观点。如何自动从这些海量的非结构化文本数据中提取出有用的情感信息，成为了自然语言处理（Natural Language Processing, NLP）中的一个重要研究课题。情感分析（Sentiment Analysis）作为NLP的核心任务之一，旨在识别和提取文本中的情感倾向，从而广泛应用于舆情监控、市场分析、产品评价等领域。

随着机器学习方法的快速发展，情感分析取得了显著的进展。传统的基于词典或规则的方法往往难以应对语言表达的复杂性，因此现代情感分析逐渐转向数据驱动的机器学习模型。这些模型通过在大规模标注数据集上进行训练，能够学习到文本中的情感特征。

本项目采用了两种主要的机器学习方法来处理文本情感分析任务。第一种方法是基本的Softmax回归与梯度下降优化。Softmax回归适用于多分类任务，梯度下降作为常用的优化算法，用于不断调整模型参数以最小化损失函数。第二种方法是基于神经网络的嵌入层和全局平均池化（Global Average Pooling）的模型，通过词嵌入将文本映射到向量空间，结合全局特征提取和全连接层实现情感分类。这两种方法的结合使得我们能够在不同复杂度的场景下对文本情感进行准确分类。

2 相关工作

情感分析作为NLP领域的一个重要研究方向，已经发展了几十年，经历了从基于规则的简单模型到基于深度学习的复杂模型的演变过程。早期的情感分析工作多依赖于基于词典的方法，这种方法通过构建一个情感词汇表，将文本中的词语与情感类别进行匹配，进而判断文本的情感极性。例如，SentiWordNet就是一个常用的情感词典，它为每个单词提供了情感评分。然而，这类方法通常只能处理显式的情感词汇，难以理解隐含在句子结构和上下文中的情感信息，面对语言的多样性和复杂性显得力不从心。

随着机器学习技术的发展，研究者开始探索通过数据驱动的方法进行情感分析。基于机器学习的情感分析方法通常将情感分类问题视为一个监督学习任务，通过对大规模标注数据集进行训练，学习文本中的情感模式。常用的机器学习算法包括支持向量机（SVM）、朴素贝叶斯（Naive Bayes）、逻辑回归（Logistic Regression）等。这些方法在情感分析任务中展现了较好的性能，尤其是当文本特征经过精心设计和提取时，分类器的效果可以显著提升。

近年来，随着深度学习的兴起，情感分析领域迎来了新的突破。卷积神经网络（CNN）和循环神经网络（RNN）等深度学习模型在处理图像、语音和文本数据时展现了强大的学习能力。对于情感分析任务，CNN可以捕捉文本中的局部特征，而LSTM等RNN变体则能够处理文本中的序列信息，从而更好地理解上下文中的情感信息。这些模型在多种情感分类任务中取得了优异的表现。

与此同时，预训练语言模型的出现使得情感分析的准确性达到了新的高度。以BERT和GPT为代表的预训练模型，通过在大规模无标注语料库上进行预训练，学习到了丰富的语言表示能力，并通过微调适应特定任务。这种方法相比于传统的从零开始训练的深度学习模型，具有更好的泛化能力和更高的分类准确性。例如，BERT模型通过双向Transformer结构，能够在情感分析中同时考虑句子的前后文信息，使得情感分类更加准确。

此外，近年来一些研究还探索了多模态情感分析，即将文本与图像、语音等其他模态信息结合起来，以提高情感分类的精度。多模态情感分析利用不同模态之间的互补性，有望在复杂的应用场景中取得更好的效果。

总体来看，情感分析领域已经发展出一系列有效的方法和模型，从传统的基于词典的规则方法，到基于机器学习的模型，再到当下流行的深度学习与预训练模型，各种方法在不同的应用场景中展现出了不同的优势。

3 方法

3.1 提取数据特征

3.1.1 词袋模型

词袋模型（Bag of Words, BoW）是本研究中用于文本特征提取的基础方法。该模型通过统计词汇在文本中出现的频次，将文本转换为固定长度的特征向量，而忽略了词汇的顺序信息。

首先，我们从所有的文档中提取唯一词汇，构建词汇表。接着，对每个文档进行向量化处理，其中每个向量的维度对应于词汇表中的词汇，向量的值表示该词汇在文档中出现的频次。

具体而言，我们使用了 scikit-learn 库中的 CountVectorizer 工具，通过设置 min\_df 参数来过滤掉在少量文档中出现的词汇，从而减少噪音。文本数据首先被转换为词频矩阵，其中每一行代表一个文本，每一列代表一个词汇表中的词，矩阵中的值表示该词在相应文本中出现的频次。为了确保验证集和测试集的转换与训练集一致，我们在转换过程中使用了与训练集相同的词汇表。

3.1.2 TF-IDF 特征提取

本研究采用了基于TF-IDF（Term Frequency-Inverse Document Frequency，词频-逆文档频率）的特征提取方法，用于量化文本数据的词语重要性。TF-IDF模型通过结合词频（TF）和逆文档频率（IDF）来衡量每个词语在文档中的重要性。文档中的常见词语（如"的"、"是"等）由于在多个文档中频繁出现，其IDF值较低，从而在特征向量中的权重也较低；而在少数文档中出现的词语则具有较高的IDF值。

其中，词频TF的计算公式如下：

式中，表示词语（term），表示文档（document），表示词语在文档d中出现的次数，分母部分是文档中所有词语出现的次数总和。

逆文档频率IDF的计算公式如下：

式中，表示词语，表示文档合集（corpus），是文档合集中总的文档数，是包含词语的文档数。

该公式计算逻辑是若词语出现在较少的文档中，会减少，故而IDF较高，反之，则IDF较低。

TF-IDF是将TF和IDF相乘得到的值，公式如下：

在本研究中，我们分别提取了基于单词的TF-IDF特征以及基于n-gram（n元词组）的TF-IDF特征。n-gram特征提取考虑了多个连续词语的组合，如二元词组（bigram）和三元词组（trigram），以捕捉文本中的局部词语顺序信息。

3.1.2 特征组合

本研究采用了三种不同的文本特征提取方法：词袋模型（BoW）、基于单词的TF-IDF和基于n-gram的TF-IDF，得到同一个数据集的三种不同的特征稀疏矩阵。为了充分利用这三种特征，本研究将它们组合成一个整体特征矩阵，以期望模型能够更好地捕捉文本中的信息。

我们使用水平堆叠的方式（hstack）将这三种特征组合起来，形成一个新的特征矩阵。该特征矩阵结合了词频信息、词语的重要性以及词语序列的局部信息，从而丰富了模型的输入特征空间。

hstack的基本原理是将多个稀疏矩阵按列进行拼接，生成一个新的矩阵。假设我们有两个矩阵和，它们具有相同的行数，列数分别为和，使用hstack([A,B])操作后，生成新的矩阵C的尺寸位，其中，矩阵的前 列由矩阵组成，后列由矩阵构成。

3.1.4 降低维度

为了降低特征空间的维度，同时保留尽可能多的有用信息，本研究采用了截断奇异值分解（Truncated Singular Value Decomposition, Truncated SVD）方法。Truncated SVD是一种广泛应用于稀疏矩阵和高维数据的降维技术，通过保留前n\_components个奇异值及其对应的奇异向量来实现降维。

首先，对训练集特征矩阵进行Truncated SVD降维，生成低维特征矩阵：

其中，是前个左奇异向量组成的矩阵， 是对应的奇异值矩阵。

然后，利用在训练集上学习到的映射，将验证集和测试集的特征矩阵映射到相同的低维空间中，得到验证集和测试集的低维特征矩阵。

该方法不仅减少了特征维度，还有效地去除了特征之间的冗余信息，使得模型在低维特征空间中仍能保持较高的分类或回归性能。

3.2 Softmax回归函数

Softmax回归是一种常用于多分类问题的线性模型，通过学习样本的特征与类别之间的线性关系来进行分类。在文本分类任务中，Softmax回归能够有效地将文本特征映射到多个类别之一。

Softmax函数：对于一个输入样本，Softmax回归模型首先通过线性变换计算出每个类别的得分（logits）。这些得分再通过Softmax函数转换为概率分布，公式如下：

其中：是对应类别的参数向量，是类别的总数，表示指数函数。

损失函数：在训练过程中，模型通过最小化交叉熵损失函数来调整参数。交叉熵损失函数衡量了模型预测的概率分布与实际类别之间的差距，定义如下：

梯度下降优化：为了最小化损失函数，使用梯度下降法来优化参数。梯度下降法通过以下公式更新参数

其中η是学习率，一般由手动指定，决定了每次更新的步长，学习率不同，模型收敛的效果也不同；是损失函数对参数的偏导数。化简后可以得到

这就是梯度下降法中用于更新参数的梯度计算公式公式，表示了模型预测概率与实际标签之间的差异如何影响每个参数的调整方向与幅度。通过不断迭代计算并更新参数，模型的损失函数将逐步减小，从而提高分类的准确性。在实际计算中，假设是训练样本矩阵，形状为，是权重矩阵，形状为，目标的独热编码矩阵形状为，模型的预测概率矩阵形状为，那么偏导数的矩阵表示为

在实际计算中我们通过选取样本计算梯度值来更新参数，而样本的选取策略不同所得到的结果也不同，主要有以下三种方法：

批量梯度下降（Batch Gradient Descent）方法：在每次迭代时，使用整个训练集计算梯度并更新参数。虽然这种方法能确保全局最优，但在大规模数据集上计算成本较高。

小批量梯度下降（Mini-batch Gradient Descent）：在每次迭代时，仅使用一个小批量（mini-batch，即训练集中的一部分）的数据计算梯度。这样能够兼顾计算效率与模型的收敛速度，通常也是实践中常用的方法。

随机梯度下降（Stochastic Gradient Descent, SGD）：每次更新参数时仅使用一个样本计算梯度。虽然计算速度快，但梯度的波动较大，容易引入噪声。

需要注意的是，当我们采用对比的方法来比较上述三种策略的效果时，需要统一变量来进行公平的比较。具体来说，随机梯度下降（SGD）的循环次数应最多，因为每次循环中仅计算一次梯度。假设训练集包含个样本，那么在一循环中，批量梯度下降（Batch Gradient Descent）会计算次梯度。小批量梯度下降（Mini-batch Gradient Descent）的计算次数次之，假设 mini-batch 的大小为,那么在一个循环中将计算次梯度。SGD计算次数最少，每个迭代仅计算1 次梯度。从而应给定相同梯度计算次数，循环数相应变化，实现比较性能的目的。

3.3 神经网络模型方法

3.3.1 模型设置

本研究设计并实现了一个机器学习模型用于文本情感分析。在模型的初始化过程中，定义了输入数据的形状和分类任务的类别数，在本实验中，分别为：，。

此外，在模型中还进行了以下关键设置：

早停策略（EarlyStopping）：为了防止模型过拟合，当验证集损失（val\_loss）在连续 3 个训练轮次内没有显著改善时，提前停止训练，并恢复到表现最佳的模型权重。

学习率调度器（ReduceLROnPlateau）：在训练过程中，如果验证集损失在连续 3 个训练轮次内没有改善，则将学习率减少 0.1 倍，以帮助模型更好地收敛。

这些配置确保了模型的稳定训练，并提高了最终模型的泛化能力。

3.3.2 模型构建

本文构建了一个多层感知器（Multi-Layer Perceptron，MLP）模型，用于文本情感分类任务。该模型采用了 Keras 的 Sequential API 构建，具体的网络结构如下：

1. 输入层：输入数据形状与特征提取后的维度相匹配。
2. 隐藏层：

第一层为包含 512 个神经元的全连接层，激活函数为 ReLU。

第二层为批量归一化（Batch Normalization）层，标准化每一批次的输入以提高模型训练的稳定性。

第三层为 Dropout 层，随机丢弃 50% 的神经元，用于防止过拟合。

随后，模型继续堆叠两组相似的结构：

包含256个和128个神经元的全连接层，每层之后均紧跟批量归一化和 Dropout 层。

1. 输出层：最终的输出层为包含 num\_classes 个神经元的全连接层，使用 softmax 激活函数输出每个类别的概率分布。这里的 num\_classes 为 5，对应于文本情感分析中的五个类别。

该模型架构简单而有效，利用 ReLU 激活函数提取特征，批量归一化提高训练速度和稳定性，Dropout 层则有效防止了过拟合。最终使用 softmax 激活函数进行多类别分类。

感知器（Perceptron）：由 Frank Roseblatt 于 1957 年提出，是一种广泛使用的线性分类器。

感知器学习算法是一个经典的线性分类器的参数学习算法。其基本原理如下:

给定个样本的训练集：,其中，感知器学习的算法试图找到一组参数，使得对于每个样本有

感知器的学习算法是一种错误驱动的在线学习算法。先初始化一个权重向量（通常是全零向量），然后每次分错一个样本时，即，就用这个样本来更新权重：

根据感知器的学习策略，可以得知感知器的损失函数为：

采用随机梯度下降，其每次更新的梯度为：

算法3.1 两类感知器的参数学习算法

输入：训练集，最大迭代次数

初始化：

repeat

对训练集中的样本随机排序；

FOR DO

选取一个样本；

IF THEN

;

;

END

IF THEN BREAK;

END

UNTIL ;

输出：

多层感知器中，我们引入一个构建在输入输出联合空间上的特征函数，将样本对映射到一个特征向量空间。

在联合特征空间中，我们可以建立一个广义的感知器模型：

其中为权重向量，表示输入所有的输出目标集合。

3.3.3 模型编译

在模型构建完成后，使用Adam优化器对模型进行了编译，并配置了适当的损失函数和评估指标，以确保模型在训练过程中能够有效学习并评估性能。具体设置如下：

优化器：选择Adam优化器（Adaptive Moment Estimation），其学习率（learning rate）设为 0.01。Adam优化器结合了动量和自适应学习率调整的优点，能够在多种任务中表现出色，适用于处理高维稀疏数据。

Adam优化器在每次参数更新时会考虑两个主要的统计量：

1. 梯度的一阶矩估计（动量估计）：估计梯度的均值，用于加速收敛。
2. 梯度的二阶矩估计（均方根估计）：估计梯度的方差，用于调整学习率。

通过引入动量，Adam优化器能够加速收敛，并减少震荡。它通过结合动量和自适应学习率的优势，能够在复杂的优化问题中表现出色。

损失函数：使用稀疏分类交叉熵（Sparse Categorical Crossentropy）作为损失函数，该函数特别适合多类别分类问题。

评估指标：在训练过程中使用准确率（accuracy）作为主要评估指标，以衡量模型对验证集的预测性能。

通过上述配置，模型能够高效地进行参数调整，最终在测试数据上表现出较好的分类性能。

4 实验

4.1 Softmax回归实验

4.2 神经网络模型实验

5 结果

6 总结

图X 图片说明 \*字体为小5号，图片应为黑白图，图中的子图要有子图说明\*

表X 表说明 \*表说明采用黑体\*

|  |  |
| --- | --- |
| \*示例表格\* | \*第一行为表头,表头要有内容\* |
|  |  |
|  |  |
|  |  |
|  |  |

过程 X. 过程名称

\*《计算机学报》的方法过程描述字体为小5号宋体，IF 、THEN等伪代码关键词全部用大写字母，变量和函数名称用斜体\*

算法**Y**. 算法名称.

输入：… …

输出：… …

\*《计算机学报》的算法描述字体为小5号宋体, IF 、THEN等伪代码关键词全部用大写字母，变量和函数名称用斜体\*

致 谢 \*致谢内容.\* 致谢

参 考 文 献

[1] 邱锡鹏. 《神经网络与深度学习》. 版次(初版不写). 出版社地点: 出版社, 出版年