**【P2】-【基于 PySpark 的深度学习社交媒体情感分析】实验结果与问题报告**

**成员：陈雅婷、洪凯晶**

**一、项目背景与目标**

（一）项目背景

社交媒体蓬勃发展，海量用户生成内容蕴含丰富情感信息，对企业市场洞察、产品优化及政府舆情监测意义重大。传统情感分析方法在处理大规模数据时效率低下，PySpark 分布式计算能力与深度学习模型结合，能高效准确地剖析社交媒体情感倾向。

（二）项目目标

本项目致力于构建一套基于 PySpark 的深度学习社交媒体情感分析系统。该系统具备精准识别文本情感的能力，能够清晰区分积极、消极和中性情感。同时，系统设计充分考虑到未来业务的增长，具备高度的扩展性，能够从容应对数据量的持续攀升；在处理效率上，也力求高效，能够在短时间内完成大规模数据的情感分析任务，切实满足不同用户群体对大规模数据处理的迫切需求。

**二、实验环境与工具**

1. **硬件环境**：
   * 处理器：2节点分布式服务器集群，每节点配备 1 核 CPU
   * 内存：每节点 4GB 内存，42节点共 8GB
   * 存储：20GB
2. **软件环境**：
   * 操作系统：Linux Centos7
   * 编程语言：Python
   * 大数据处理平台：Hadoop、Spark
   * 深度学习框架：TensorFlow
   * 数据分析工具：Pandas、NumPy、Matplotlib、Scikit-learn 等
   * 集成开发环境：Pycharm

**三、项目内容与步骤**

1. **数据收集**：
   * 使用 Python 爬虫库（如 Scrapy、BeautifulSoup）编写针对微博等主流社交媒体平台的爬虫程序
   * 遵循平台规则，合理控制请求频率、切换 IP 代理，确保数据采集的稳定性和合法性。
   * 采集的数据包括文本内容、用户信息等，并存储至分布式文件系统 HDFS。
2. **数据预处理**：
   * 利用 PySpark 对采集的数据进行全面清洗，去除乱码、无效字符、HTML 标签等噪声数据。首先定义清洗函数，通过正则表达式去除 HTML 标签，再过滤掉非中文字符、数字和字母，最后去除连续的空白字符，从而获得干净的文本内容。进行数据转换和格式统一，以便后续分析。
   * 使用专业分词工具 jieba 进行分词操作，将清洗后的文本分割为单个词语或短语，并将结果存储在新列中。接着进行停用词过滤，加载预定义的停用词表，去除对情感分析无实际意义的常见词汇，使文本内容更加聚焦于有价值的情感表达。
   * 将文本转化为深度学习模型可识别的格式，分别使用 Word2Vec 和 TF-IDF 两种方法将文本转化为特征向量。对于 Word2Vec，训练一个词向量模型，将每个词语映射为固定维度的向量，再通过平均化处理将文本转换为向量表示；对于 TF-IDF，计算词语在文本中的重要性，将文本转换为词频 - 逆文档频率矩阵表示，为后续的模型训练提供合适的输入格式。
3. **模型构建与训练**：
   * 小组成员分工训练门控循环单元（GRU ）和长短期记忆网络（LSTM）两种模型。选择长短期记忆网络（LSTM）作为核心模型，因其在处理序列数据方面具有独特优势，能够有效捕捉文本中的长距离依赖关系。利用 PySpark 的分布式训练能力，将大规模数据集按 7:1.5:1.5 的比例划分为训练集、验证集和测试集，确保各数据集之间的独立性和代表性。
   * 在训练集上构建 LSTM 模型，模型结构包括嵌入层、双向 LSTM 层、全连接层和输出层。嵌入层将输入的文本特征向量映射到低维稠密空间，双向 LSTM 层处理序列信息，捕捉上下文语义，全连接层进行非线性变换，输出层使用 softmax 激活函数输出分类概率。调整模型参数，如隐藏层节点数量设为 64，学习率设为 0.001，迭代次数设为 15 等。
   * 在训练过程中，使用验证集实时评估模型的准确率、召回率等指标，通过 EarlyStopping 回调函数监控验证集损失，当损失不再显著下降时提前停止训练，防止模型过拟合。同时使用 ModelCheckpoint 回调函数保存验证集上性能最佳的模型。
   * 当模型在验证集上性能达到最优时，停止训练，并加载最佳模型在测试集上进行最终性能评估，计算准确率、精确率、召回率和 F1 值等指标，全面评估模型的性能和泛化能力。
4. **系统实现与功能模块开发**：
   * 运用 Python 结合 PySpark 实现数据的分布式处理和并行计算。
   * 使用深度学习框架 TensorFlow 搭建和训练情感分析模型。
   * 借助 Web 框架 Flask 构建应用服务层的 Web 接口。
   * 实现用户文本上传和情感分析结果返回功能，确保接口易用性和稳定性。

**四、实验结果**

1. **数据预处理结果**：
   * 成功去除噪声数据，数据清洗后文本内容更加纯净，为后续分析提供了高质量的数据基础。
   * 分词和停用词过滤效果良好，文本被准确分割为单个词语或短语，去除了无意义词汇。
   * 文本成功转化为特征向量，且特征向量的维度和格式符合模型输入要求。
2. **模型训练结果**：
   * 在测试数据集上，积极、消极、中性情感的分类准确率达到 92.55%，满足项目验收标准中 80% 以上的要求。
   * 召回率达到 0.9255，F1 值达到 0.9255，模型在不同类型和规模的数据集上表现稳定，具备良好的泛化能力。
   * 模型训练过程稳定，通过调整参数有效提升了模型性能，能够准确分析文本情感倾向。
3. **功能模块实现结果**：
   * Web 服务接口正常运行，用户上传文本后，系统能够在规定时间内返回准确的情感分析结果，接口响应稳定，无明显延迟和错误。
   * 系统扩展性良好，在分布式计算集群上进行压力测试时，能够根据数据量和用户请求量的增加，动态调整资源分配，保持稳定运行，未出现崩溃或性能急剧下降的情况。

**五、遇到的问题与解决方案**

1. **数据采集受阻问题**：
   * **问题描述**：社交媒体平台反爬虫机制严格，在数据采集过程中频繁遭遇限制，如 IP 封禁、请求频率限制等，导致数据采集无法持续稳定进行。
   * **解决方案**：：进一步优化爬虫策略，增加 IP 代理池的规模，采用更灵活的请求频率控制算法，如根据平台反馈动态调整请求频率；同时，深入研究平台规则，确保数据采集行为完全合规，避免触发反爬虫机制。
2. **模型训练精度提升困难问题**：
   * **问题描述**：在模型训练过程中，尽管调整了多种参数，模型的准确率、召回率等指标提升效果不明显，难以达到预期的高精度要求。
   * **解决方案**：增加训练数据量，丰富数据的多样性，涵盖更多不同语境和情感表达的文本；尝试优化模型结构，如增加 LSTM 层的数量、调整隐藏层节点数量等；采用迁移学习的方法，利用预训练模型在大规模文本数据上学习到的通用特征，提高模型对社交媒体文本情感分析的适应性。
3. **系统部署与性能优化问题**：
   * **问题描述**：将系统部署到分布式计算集群后，在高并发访问情况下，系统响应速度变慢，出现延迟现象，影响用户体验。
   * **解决方案**：化服务器配置，合理分配 CPU、内存等资源；对代码进行性能优化，如优化算法实现、减少不必要的计算和数据传输；使用缓存技术，对频繁访问的数据和计算结果进行缓存，提高系统响应速度；进一步优化 Kubernetes 的容器编排管理，根据系统负载动态调整容器资源分配，确保系统在高并发场景下的稳定性和高效性。
4. **总结与计划**

* **总结：**

本项目成功构建了基于 PySpark 的深度学习社交媒体情感分析系统。通过数据采集、预处理、模型训练和系统实现等一系列工作，系统能够准确识别文本情感，在功能和性能上基本达到预期目标。数据预处理环节有效提升了数据质量，为模型训练提供了有力支持；LSTM 模型在情感分析任务中表现良好，具备较高的准确率和泛化能力；Web 服务接口的实现使用户能够方便地使用系统进行情感分析。然而，项目在实施过程中也遇到了一些问题，如数据采集受阻、模型训练精度提升困难以及系统部署后的性能优化等，通过相应的解决方案得到了一定程度的解决，但仍有进一步优化的空间。

* **计划:**

持续优化模型性能，探索更先进的深度学习模型或模型组合，如结合 Transformer 架构的模型，进一步提高情感分析的准确率和召回率。

优化系统的可扩展性，研究如何更好地利用分布式计算集群的资源，以应对未来数据量和用户请求量的大规模增长。

提升系统的用户体验，完善 Web 界面的设计，增加更多的交互功能，如情感分析结果的可视化展示、用户历史记录查询等。

对项目代码进行优化和重构，提高代码的可读性和可维护性，采用模块化设计思想，将不同功能模块分离，便于后续的开发和扩展。