词向量和文本分类

序号	学号	专业班级	姓名	性别
	3200105264	机器人工程	汪张翼	男

1 Project Introduction

1.1 选题

本实验主要包括词向量实验与文本分类实验。

1.2 工作简介

1.2.1 词向量实验

在 ModelArts 平台上完成词向量的训练，并应用于语义相似词的搜索、扩展，通过实验掌握词向量训练方法。

1.2.2 文本分类实验

在 ModelArts 平台上完成文本分类的训练，理解文本分类的基本流程与 CNN 网络在文本任务中的用法，并掌握 MindSpore 搭建文本分类模型的方法。

1.3 实验环境等要求

实验环境：ModelArts Ascend Notebook 环境

实验镜像：mindspore1.7.0-cann5.1.0-py3.7-euler2.8.3

规格：Ascend: 1*Ascend910|CPU: 24 核 96GB

2 Technical Details

2.1 实验原理

2.1.1 词向量实验

基于 Word2Vec 对文本进行训练以得到词的词向量。相较于传统的 one-hot 编码会导致极其高维的向量，其利用率词语上下文的关系，通过神经网络来进行训练以得到相应词语的词向量，并且根据词向量的相似程度来判断词语的相似程度。

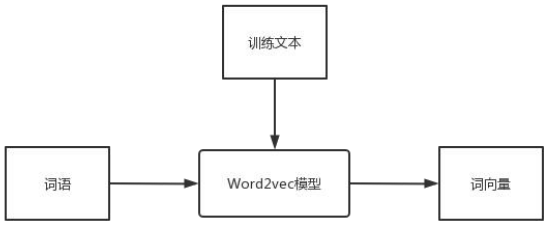
2.1.2 文本分类实验

基于卷积神经网络的方法，首先对句子进行向量化，将问题转化成监督学习问题，构建卷积神经网络对标签数据进行训练，最后完成对情感文本的分类。

2.2 算法简述

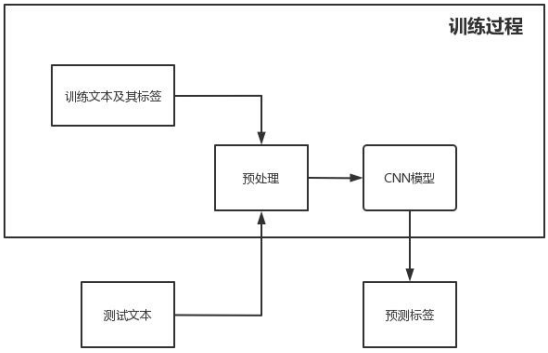
2.2.1 词向量实验

该实验流程较为简单，因此算法也比较简洁。主要算法是使用 Word2Vec 进行模型训练，训练完毕后进行模型存储以便离线运行，以及最后的利用训练的模型获取指定词语的词向量与其相似词语。



2.2.2 文本分类实验

该实验主要算法有两大部分——数据预处理与 CNN 网络的构建。首先 text2Vec()函数将句子转化成向量，然后建立一个三层的 CNN 网络，最后将向量与其对应句子的标签传入该网络进行训练。最后可以利用训练好的模型预测句子的情感分类。



2.3 重要技术细节

2.3.1 Word2Vec()函数

依赖库	gensim.models
基本功能	用于训练词向量，输入文本训练，其能返回单词对应的词向量
	重要参数说明
vector_size	训练后词向量的维度，在实验中设置成 100，其最后返回也是 100 维的向量
window	当前词语与预测词语的最大间隔。因为相距较远的词语可能联系微弱，所以 window 参数的设置可以防止较远距离单词造成的干扰
min_count	忽略词频小于 min_count 的词语。太小频数的单词训练次数少，准确度低
workers	线程数量，实验中设置成 cpu_count()，即利用所有 cpu 进行训练
sg	sg=0 表示 CBOW 模型，而实验中设置成 sg=1 表示使用 skip-gram 模型

2.3.2 TextCNN 类及训练参数

该类别的基本功能是建立一个卷积神经网络。在实例代码中，该类别建立了一个三层的卷积神经网络，每层均有卷积、激活、池化的操作，最后通过全连接层输出结果。

模型	CNN
基本功能	输入文本训练，输出其对应的情绪分类
优化器	Adam
损失函数	Softmax 交叉熵损失函数

2.3.3 learning_rate 参数的设置

在设置 CNN 的学习率时，分成了 warm_up、normal_run 以及 shrink 三个部分，由于设置比例时使用了向下取浮点的函数，其在 epoch_size 设置成 5 的倍数时，基本是 1:2:3 的比例。

而对于三个阶段，第一个阶段的学习率会来回升降，而第二个阶段的学习率为全部阶段的最大并且保持不变，而最后一个阶段的学习率会下降到全阶段最小并来回升降，这符合训练时先较大学习率去快速收敛，而之后较小学习率以提高准确度的思路。

3 Experiment Results

3.1 词向量实验

3.1.1 基础要求

在训练好模型后，输入词语“中国”后，得到 100 维度的词向量——array([-0.4476185,...,0.02038454])。而输入单词进行相似度测试，可以得到某个单词的相似单词及其相似度，以词语“金融”为例，其最相似的两个词语分别为“金融服务”与“国际金融”，相似度均达到 0.76 左右。

3.1.2 进阶要求

本次实验以探究影响“金融”词语的相似单词及其相似度的因素为目的，修改 Word2Vec() 函数的 vector_size 与 windows 两个参数进行实验，结果如下：

vector_size 值	windows 值	“金融”词语的最相似词语	相似度
100	5	“金融服务”	0.761
30	5	“证券期货”	0.921
75	5	“证券期货”	0.804
100	3	“金融服务”	0.772
100	7	“证券期货”	0.807

从表格中看出，降低 vector_size 的值时，由于词向量维度降低，其相似度也随之增大。而对于 windows 参数，修改其值的时候，对于词向量及其相似度的影响不大，但是 windows

的值越大，模型训练的时间越长，这也与 windows 参数的意义相符合。

3.2 文本分类实验

3.2.1 基础要求

在训练好模型后，模型评估准确率为 0.764。在测试模型时输入测试样例"the movie is so boring"，得到的输出结果为 Negative comments，符合预期结果。

3.2.2 进阶要求

实验中的学习率设置中有 epoch_size/5 的向下去浮点数，但是实验中 epoch_size 设置成 4，这导致学习率并没有 warm_up 的阶段，因此我们修改 epoch_size 为 10，对模型再次进行训练，训练的结果如下：

模型评估准确率	0.778
测试结果	Negative comments（符合预期结果）

可以看出 epoch_size 的增加可以提高模型准确度，下面继续增大 epoch_size 至 20 进行训练，结果如下：

模型评估准确率	0.778
测试结果	Negative comments（符合预期结果）

看到模型准确率并未明显上升，而且运行时间也大大增长，因此在实际运行时应该选取合适大小的 epoch_size 进行训练。

References:

[1] https://tedboy.github.io/nlps/generated/generated/gensim.models.Word2Vec.html

[2] 教学课件《2. Word Embeddings.pptx》

[3] 教学课件《3. Convolutional Neural Networks.pptx》