自然语言处理第二次实验

实验目的

- 了解神经网络中的基础模型

  神经网络是指一系列受生物学和神经科学启发的数学模型。主要是通过对人脑的神经元网络进行抽象，构建人工神经元，并按照一定拓扑结构来建立神经元之间的连接。前馈神经网络（FNN）是最早发明的简单人工神经网络；卷积神经网络（CNN）是一种具有局部连接、权重共享等特性的深层前馈神经网络；循环神经网络（RNN）是一类具有短期记忆能力的神经网络。

- 了解深度学习框架 Pytorch 的使用

  PyTorch 使用 python 作为开发语言，近年来和 TensorFlow, keras, caffe 等热门框架一起，成为深度学习开发的主流平台之一。PyTorch 的基本元素包含张量(Tensor)、变量(Variable)、神经网络模块(nn.Module)等。

- 了解使用深度学习解决文本分类任务基本流程

  以 PyTorch 为例，一个常规的文本分类任务代码开发流程是：安装并导入相关的深度学习库、数据获取和预处理、定义神经网络、定义损失函数(loss function)和优化器(optimizer)、训练网络和测试网络。

实验环境

python 3 + jieba + PyTorch + NumPy + Sklearn + TensorboardX + tqdm

- python3

  除了高性能外，拥有 NumPy、SciPy 等优秀的数值计算、统计分析库。TensorFlow、Caffe

等著名的深度学习框架都提供了 Python 接口。

- jieba

  jieba 是一款优秀的 Python 第三方中文分词库，支持三种分词模式：精确模式、全模式和搜索引擎模式。

- PyTorch

  PyTorch 是一个针对深度学习，并且使用 GPU 和 CPU 来优化的 tensor library，它是一个以 Python 优先的深度学习框架，不仅能够实现强大的 GPU 加速，同时还支持动态神经网络。

- NumPy

  NumPy 是 Python 语言的一个扩展程序库，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。

- Sklearn

  Sklearn (全称 Scikit-Learn) 是基于 Python 语言的机器学习工具。它建立在 NumPy, SciPy, Pandas 和 Matplotlib 之上，Sklearn 里面有六大任务模块：分别是分类、回归、聚类、降维、模型选择和预处理。

- TensorboardX

  Tensorboard 是 TensorFlow 的一个附加工具，可以记录训练过程的数字、图像等内容，以方便研究人员观察神经网络训练过程。可是对于 PyTorch 等其他神经网络训练框架并没有功能像 Tensorboard 一样全面的类似工具，一些已有的工具功能有限或使用起来比较困难。TensorboardX 这个工具使得 TensorFlow 外的其他神经网络框架也可以使用到 Tensorboard 的便捷功能。

- tqdm

  tqdm 是一个快速，可扩展的 Python 进度条，可以在 Python 长循环中添加一个进度提示信息，用户只需要封装任意的迭代器 tqdm(iterator)。

实验步骤

1. 阅读代码，运行并得到结果。

2. 根据已有的 TextCNN 模型格式，写出 BiLSTM 模型，代码实现可参考 Chinese-Text-Classification-Pytorch 。 TexcCNN 模型相关超参数配置在 \Chinese-Text-Classification-Pytorch \models\TextCNN.py

```
class Config(object):

    """配置参数"""
    def __init__(self, dataset, embedding):
        self.model_name = 'FastText'
        self.train_path = dataset + '/data/train.txt'                                #训练集
        self.dev_path = dataset + '/data/dev.txt'                                    #验证集
        self.test_path = dataset + '/data/test.txt'                                  #测试集
        self.class_list = [x.strip() for x in open(
            dataset + '/data/class.txt', encoding='utf-8').readlines()]              #类别名单
        self.vocab_path = dataset + '/data/vocab.pkl'                                #词表
        self.save_path = dataset + '/saved_dict/' + self.model_name + '.ckpt'        #模型训练结果
        self.log_path = dataset + '/log/' + self.model_name
        self.embedding_pretrained = torch.tensor(
            np.load(dataset + '/data/' + embedding)["embeddings"].astype('float32'))\
            if embedding != 'random' else None                                       #预训练词向量
        self.device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')   #设备

        self.dropout = 0.5                                              #随机失活
        self.require_improvement = 1000                                 #若超过1000batch效果还没提升，则提前结束训练
        self.num_classes = len(self.class_list)                         #类别数
        self.n_vocab = 0                                                #词表大小，在运行时赋值
        self.num_epochs = 20                                            #epoch数
        self.batch_size = 32                                            #mini-batch大小
        self.pad_size = 32                                              #每句话处理成的长度(短填长切)
        self.learning_rate = 1e-3                                       #学习率
        self.embed = self.embedding_pretrained.size(1)\
            if self.embedding_pretrained is not None else 300           #字向量维度
        self.hidden_size = 256                                          #隐藏层大小
        self.n_gram_vocab = 250499                                      #ngram词表大小
```

3. 调整 batch size 参数，取值分别为[8, 16, 32, 64] ，画出 TextCNN 和 BiLSTM 训练集和验证集的 loss 折线图。

4. 调整 learning rate 参数，取值分别为[1e-2, 1e-3, 1e-4] ，画出 TextCNN 和 BiLSTM 训练集和验证集的 loss 折线图。

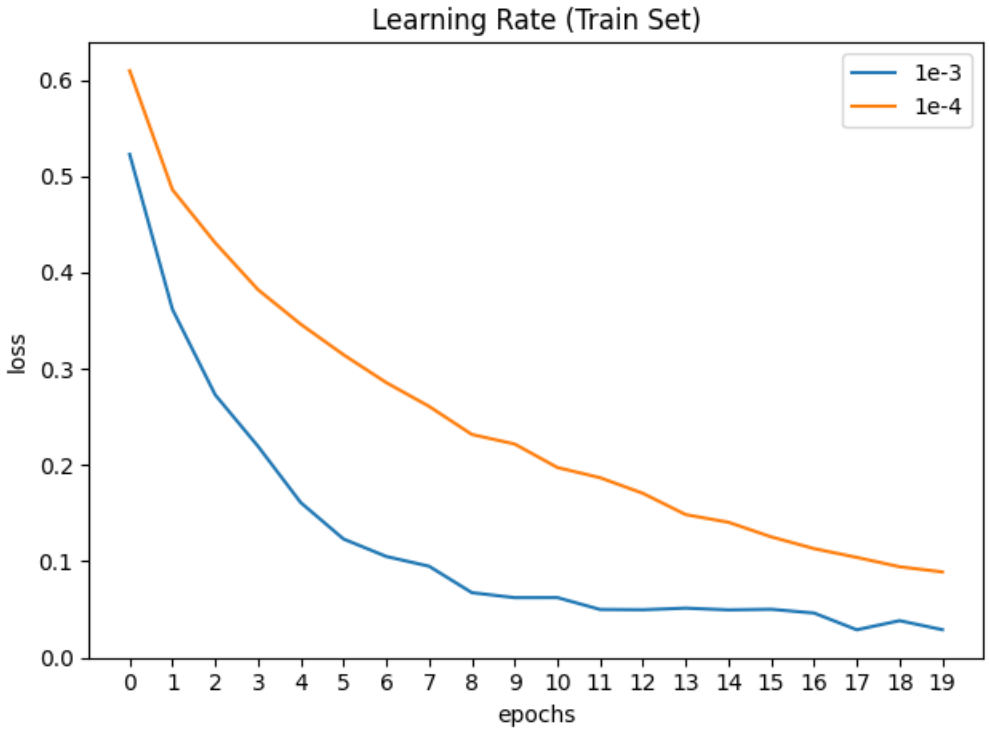
5. 调整 dropout 参数，取值分别为[0.1, 0.2 , 0.5] ，画出 TextCNN 和 BiLSTM 训练集和验证集的 loss 折线图。

6. （可选）调整 embedding size 参数和 BiLSTM 的 hidden size 参数，画出相应 loss 折线图

7. 选出最好的参数组合（不局限于以上参数组合），列出参数并将实验结果写入表格进行对比分析，例如：

模型	准确率
TextCNN	86.80%
BiLSTM	87.68%

loss 图示例：



提交时间

11 月 21 号截止

姬老师班级由于冰浩同学负责收集，李老师班级由李书砚同学负责收集，提交实验报告和代码（不包括模型和数据），文件命名方式：姓名-学号-第 2 次实验

实验要求

- 完成所有实验内容
- 良好的代码风格
- 完整的实验报告

参考资料

1. pytorch1.0.0 官方文档
2. 《神经网络与深度学习》
3. Convolutional Neural Networks for Sentence Classification
4. Chinese-Text-Classification-Pytorch