# 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。

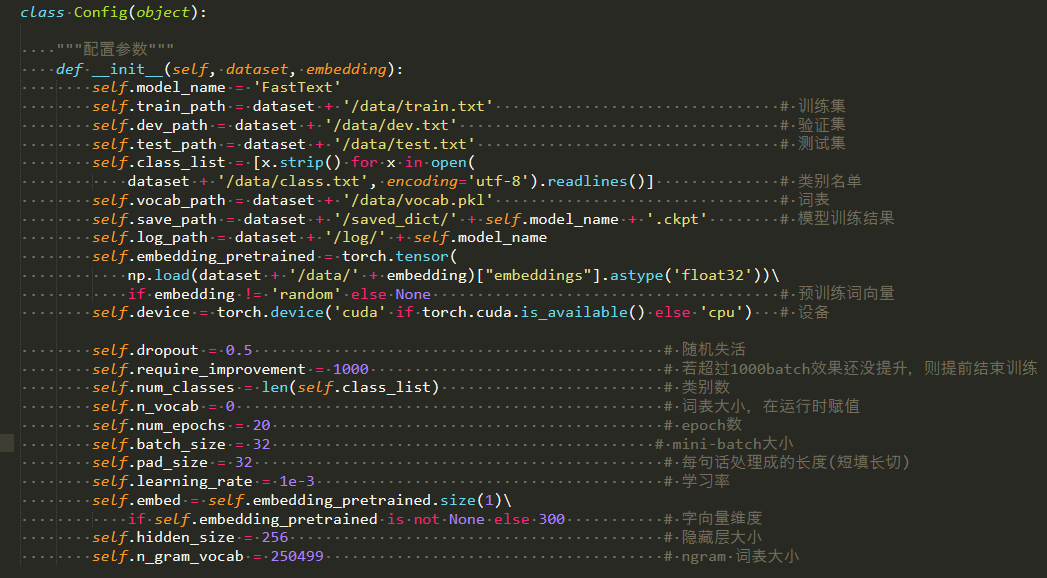
* tqdm

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](https://github.com/649453932/Chinese-Text-Classification-Pytorch)。TexcCNN模型相关超参数配置在\Chinese-Text-Classification-Pytorch \models\TextCNN.py



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embedding size参数，取值分别为[64, 128, 256] ，画出TextCNN和BiLSTM训练集和验证集的loss折线图。

5. 调整hidden size参数，取值分别为[64, 128, 256] ，画出TextCNN和BiLSTM训练集和验证集的loss折线图。

6. 调整learning rate参数，取值分别为[1e-2, 5e-3, 1e-3, 5e-4, 1e-4] ，画出TextCNN和

BiLSTM训练集和验证集的loss折线图。

7. 调整dropout参数，取值分别为[0.1, 0.2, 0.3, 0.4, 0.5] ，画出TextCNN和BiLSTM训练集和验证集的loss折线图。

8. 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3号截止**

由吴雨欣同学负责收集，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](https://pytorch.org/docs/1.0.0/)

2. [《神经网络与深度学习》](https://nndl.github.io/)

3. [Convolutional Neural Networks for Sentence Classification](https://arxiv.org/abs/1408.5882)

4. [Chinese-Text-Classification-Pytorch](https://github.com/649453932/Chinese-Text-Classification-Pytorch)

**附件1：**

**武汉大学国家网络安全学院**

**实验报告**

**课程名称 自然语言处理**

**专业年级 2019**

**姓 名**

**学 号**

**协 作 者 无**

**实验学期 2021-2022　 学年 第一 学期**

**课堂时数 课外时数**

**填写时间 2021 年 10 月 30 日**

|  |
| --- |
| **实验介绍** |
| **【实验名称】：自然语言处理第二次实验** |
| **【实验目的】：**   1. 了解神经网络中的基础模型 2. 了解深度学习框架Pytorch的使用 3. 了解使用深度学习解决文本分类任务基本流程   **【实验环境】：**   1. Windows 10 家庭中文版 2. Anaconda Navigator 1.9.12 3. conda 4.8.3 4. Python 3.8.11 5. torch 1.9.1 6. sklearn 1.0 7. numpy 1.21.2 8. matplotlib 3.4.3   **【参考文献】：**  **[1]** PyTorch学习之路（level1）——训练一个图像分类模型  <https://blog.csdn.net/u014380165/article/details/78525273>  **[2]** Pytorch学习笔记02----深度学习中的epochs，batch\_size，iterations详解  https://www.cnblogs.com/luckyplj/p/13156565.html |
| **实验内容** |
| **【实验方案设计】：**  实验模型：  TextCNN   1. 模型输入 2. embedding层 3. 卷积层 4. dropout层 5. Linear层   BiLSTM   1. 模型输入 2. Embedding层 3. 双向LSTM层，即BiLSTM层 4. 求和 5. Linear层   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   模型参数    模型定义     1. 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   绘制loss折线图所用代码    TextCNN训练集的loss折线图    TextCNN验证集的loss折线图    BiLSTM训练集的loss折线图    BiLSTM验证集的loss折线图     1. 调整embedding\_size参数，取值分别为[64, 128, 256]，画出TextCNN和BiLSTM训练集和验证集的loss折线图   TextCNN训练集的loss折线图    TextCNN验证集的loss折线图    BiLSTM训练集的loss曲线    BiLSTM验证集的loss曲线     1. 调整hidden\_size参数，取值分别为[64, 128, 256]，画出TextCNN和BiLSTM训练集和验证集的loss折线图   TextCNN训练集的loss折线图    TextCNN验证集的loss折线图    BiLSTM训练集的loss折线图    BiLSTM验证集的loss折线图     1. 调整learning\_rate参数，取值分别为[1e-2, 5e-3, 1e-3, 5e-4, 1e-4]，画出TextCNN和BiLSTM训练集和验证集的loss折线图   TextCNN训练集的loss折线图    保留较小的三个learning\_rate值时得到的TextCNN训练集的loss折线图    TextCNN验证集的loss折线图    保留较小的三个learning\_rate值时得到的TextCNN验证集的loss折线图    BiLSTM训练集的loss折线图    保留较小的三个learning\_rate值时BiLSTM训练集的loss折线图    BiLSTM验证集的loss折线图    保留较小的三个learning\_rate值时BiLSTM验证集的loss折线图     1. 调整dropout参数，取值分别为[0.1, 0.2, 0.3, 0.4, 0.5]，画出TextCNN训练集和验证集的loss折线图   TextCNN训练集的loss折线图    TextCNN验证集的loss折线图    BiLSTM训练集的loss折线图    BiLSTM验证集的loss折线图     1. 选出最好的参数组合  |  |  | | --- | --- | | 模型 | 准确率 | | TextCNN | 91.75% | | BiLSTM | 87.00% |   TextCNN  batch\_size = 128  embedding\_size = 300  hidden\_size = 256  learning\_rate = 1e-3  dropout = 0.5  BiLSTM  batch\_size = 128  embedding\_size = 300  hidden\_size = 128  learning\_rate = 1e-3  dropout = 0.5  **【实验结果分析】：**  从loss折线图中可以发现无论是TextCNN模型还是BiLSTM模型，在该数据集上收敛速度较快、损失函数较小的超参数组合如下所示。  batch\_size = 128  embedding\_size = 300  hidden\_size = 128  learning\_rate = 1e-3  dropout = 0.5 |
| **【实验总结】：** |
| **评语及评分（指导教师）** |
| **【评语】：**      **评分：**  **日期：** |

**附件2：**

**实验报告说明**

**1．实验名称：**要用最简练的语言反映实验的内容。

**2．实验目的**：目的要明确，要抓住重点。

**3．实验环境**：实验用的软硬件环境（配置）。

**4．实验方案设计（思路、**步骤和方法等**）**：这是实验报告极其重要的内容。包括概要设计、详细设计和核心算法说明及分析，系统开发工具等。应同时提交程序或设计电子版。

对于**设计型和综合型实验**，在上述内容基础上还应该画出流程图、设计思路和设计方法，再配以相应的文字说明。

对于**创新型实验**，还应注明其创新点、特色。

**5．实验结果分析：**即根据实验过程中所见到的现象和测得的数据，进行对比分析并做出结论（可以将部分测试结果进行截屏）。

**6．实验总结：**对本次实验的心得体会，所遇到的问题及解决方法，其他思考和建议。

**7．评语及评分：**指导教师依据学生的实际报告内容，用简练语言给出本次实验报告的评价和价值。