

自然语言处理

期末实践项目



人工智能1902

刘璐

20193151

目录

[基础内容 手动搭建单层LSTM网络 2](#_Toc87706868)

[一、题目要求 2](#_Toc87706869)

[二、LSTM网络结构设计 2](#_Toc87706870)

[1、网络设计原理 2](#_Toc87706871)

[2、代码实现 3](#_Toc87706872)

[3、测试函数 7](#_Toc87706872)

[三、训练结果 7](#_Toc87706873)

[1、原项目训练结果 7](#_Toc87706874)

[2、手动搭建单层LSTM训练结果 9](#_Toc87706875)

[提高内容 手动搭建双层LSTM网络 10](#_Toc87706879)

[一、题目要求 10](#_Toc87706880)

[二、双层LSTM网络结构设计 10](#_Toc87706881)

[1、网络设计原理 10](#_Toc87706882)

[2、代码实现 10](#_Toc87706883)

[三、训练结果 11](#_Toc87706873)

[拓展内容 实现多层双向LSTM网络 14](#_Toc87706879)

[一、题目要求 14](#_Toc87706869)

[二、双向LSTM网络结构设计 14](#_Toc87706870)

[1、网络设计原理 14](#_Toc87706871)

[2、代码实现 15](#_Toc87706872)

[3、测试函数 15](#_Toc87706872)

[三、同组数据双向训练结果 16](#_Toc87706873)

[自然语言处理课程及实践心得体会 17](#_Toc87706884)

# 基础内容 手动搭建单层LSTM网络

## 一、题目要求

尝试自己搭建LSTM网络

* 1. 不能调用nn.LSTM、nn.LSTMCell，可以使用nn.Linear、nn.Parameter等等搭建网络
  2. 可以参考torch.nn.LSTM的计算公式、可以仿照其输入输出，官方文档： https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html#torch.nn.LSTM
  3. 数据加载、模型训练的代码都是现成的，只需要完成模型搭建

## 二、LSTM网络结构设计

### 1、网络设计原理

由于提高内容要求搭建双层LSTM网络，索性在基础内容部分直接搭建出多层LSTM网络，运行时只需按需求设置网络层数即可。

多层LSTM结构如下图所示：(图片源自课上ppt)

图示

描述已自动生成

对于多层的LSTM，需要把前一层的每个时间步的输出作为后二层的时间步的输入，如上图所示。

对于num\_layers层LSTM：

**输入数据：**

·X的格式：(seq\_len,batch,input\_size)

·h0的格式：(num\_layers,batch,hidden\_size)

·c0的格式：(num\_layers,batch,hidden\_size)

**输出数据：**

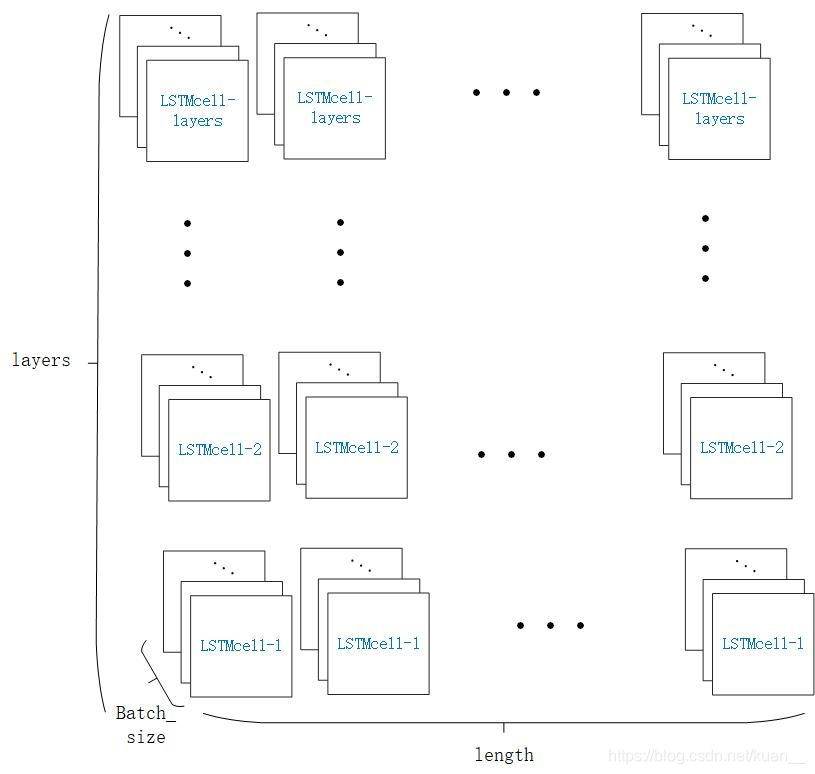
·H的格式：(seq\_len,batch,hidden\_size)

·hn的格式:(num\_layers,batch,hidden\_size)

·cn的格式:(num\_layers,batch,hidden\_size)

如果要处理一个维度为[batch\_size, length, input\_dim]的输入，则需要的LSTM结构：

(图片源自网络)



参数解释：layers表示LSTM的层数，batch\_size表示批处理大小，length表示长度，input\_dim表示每个输入的维度。所以在运行程序时我们只需设置layers的值即可得到需要的网络层数。此部分将展示layers=1的情况（即单层LSTM），提高部分将展示双层和多层的情况。

### 2、代码实现

根据上述图示可知：一个完整的LSTM是由很多LSTMcell操作组成的。

所以首先根据官方文档 ([LSTMCell — PyTorch 1.10.0 documentation](https://pytorch.org/docs/stable/generated/torch.nn.LSTMCell.html" \l "torch.nn.LSTMCell)) 中的计算公式完成

LSTMcell的计算函数的实现：

文本

描述已自动生成

**具体代码如下：**

（其中nn.Parameter表示该张量为模型可训练参数）

文本

描述已自动生成

假如我们设计的LSTM层数layers大于1，第一层的LSTM输入维度是input\_dim，输出维度是hidden\_dim，那么其他各层的输入维度和输出维度都是hidden\_dim（下层的输出会成为上层的输入），因此，定义layers个LSTMcell的函数如下所示：

文本

描述已自动生成

多层LSTM处理不同长度的输入时，每个LSTMcell都需要（h\_t-1和c\_t-1）作为状态信息输入，若没有指定初始状态，我们就自定义一个值为0的初始状态

文本

描述已自动生成

**完整LSTM代码如下:**

文本, 日程表

中度可信度描述已自动生成

### 3测试函数：

图形用户界面, 文本

描述已自动生成

## 三、训练结果

### 1、原项目训练结果：

原项目中直接调用nn.LSTM()：

文本

描述已自动生成

训练结果如下：

图形用户界面, 文本, 应用程序

描述已自动生成

可得loss=5,71；ppl=302.079

### 2、手动搭建单层LSTM训练结果：

训练单层LSTM网络时，我们可以设置layers=1：



同样的数据训练结果如下图所示：

图形用户界面, 文本, 应用程序, Word

描述已自动生成

可得loss=5.723；ppl=306.123，与原来差别不大

# 提高内容 手动搭建双层LSTM网络

## 一、题目要求

在自己搭建出来的LSTM网络基础上，实现双层LSTM网络。

（这里自行扩展到了五层网络）

## 二、双层LSTM网络结构设计

### 1、网络设计原理

设计原理同上，对于双层的LSTM网络，则需要把第一层的每个时间步的输出作为第二层的时间步的输入

图示

描述已自动生成

### 2、代码实现

在代码中我们只需设置layers=2即可。

图形用户界面, 文本, 应用程序, Word

描述已自动生成

## 三、训练结果

图形用户界面, 文本, 应用程序

描述已自动生成

可得loss=5.844；ppl=345.328

运行后意外地发现单层网络的loss和ppl指标都比双层的小，让我十分吃惊，按照我原有的想法增加训练的层数效果应该更好，所以带着疑问我尝试了五层LSTM的运行结果：

文本

描述已自动生成

注：PPL是用在自然语言处理领域（NLP）中，衡量语言模型好坏的指标。它主要是根据每个词来估计一句话出现的概率，并用句子长度作normalize，公式为

图示, 文本, 示意图

描述已自动生成

S代表sentence，N是句子长度，p(wi)是第i个词的概率。第一个词就是 p(w1|w0)，而w0是START，表示句子的起始，是个占位符。

这个式子可以这样理解，PPL越小，p(wi)则越大，一句我们期望的sentence出现的概率就越高。

还有人说，Perplexity可以认为是average branch factor（平均分支系数），即预测下一个词时可以有多少种选择。别人在作报告时说模型的PPL下降到90，可以直观地理解为，在模型生成一句话时下一个词有90个合理选择，可选词数越少，我们大致认为模型越准确。这样也能解释，为什么PPL越小，模型越好。

相同数据运行五层LSTM结果如下：

图形用户界面, 文本, 应用程序

描述已自动生成

在运行五层LSTM网络的时候运行时间远比单双层的长的多，运行结果也可以看出loss增加到6.209，PPL增加到497.617，训练效果远没有单双层的好。

带着疑惑我上网查询了资料，原来是因为一般做时序模型，大多数情况一层LSTM足够，因为在输入和输出上我们会有很多方法来降低数据复杂度，从而提高训练效率。如果数据量实在是过于庞大，单从扩增一层的神经元已经不够，两层也差不多就已经够了，因为两层的LSTM已经能把时序数据压缩成高度“浓缩”的数据，再加深层数只能带来信息压缩过程中的信息丢失以及训练过程中的梯度消失。当层数超过了三层，层与层之间的梯度消失情况变得非常明显，再加上是时序模型，导致靠近输入层的LSTM层更新迭代放缓，收敛效果和效率急剧下降，甚至非常容易进入局部最小的困境。

# 拓展内容 实现多层双向LSTM网络

## 一、题目要求

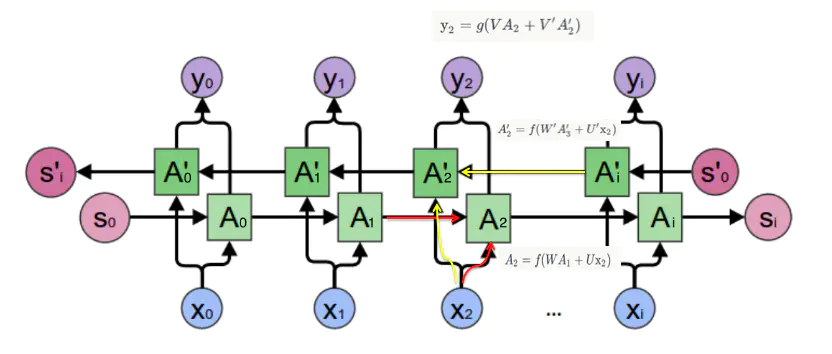
在实现多层LSTM的过程中，我发现只需设置layers的值即可设置层数，同时我了解到修改bidirectional参数的值为或True或False也可设置单向和双向。所以在上述代码的基础上做了拓展，实现了多层双向的LSTM网络。

## 二、双向LSTM网络结构设计

### 1、网络设计原理

双向卷积神经网络的隐藏层要保存两个值， A 参与正向计算， A' 参与反向计算。

最终的输出值 y 取决于 A 和 A'：



### 2、代码实现

在原有LSTM的基础上实现biLSTM的代码如下：

文本

描述已自动生成

### 3测试函数：

要实现多层LSTM只需修改参数：num\_layers。要实现双向LSTM只需修改参数：bidirectional=True。

文本

描述已自动生成

## 三、同组数据双向训练结果

图形用户界面, 文本, 应用程序, Word

描述已自动生成

可得：loss=5.7327；ppl=308.806

# 自然语言处理课程及实践心得体会

选这门课之前，学校的小牛翻译以及自然语言处理实验室，让我觉得自然语言处理是一门非常厉害的课，可真正开始学习才发现想学好并不容易。尽管肖桐老师上课时讲的很细致形象，个人也非常喜欢肖桐老师，课后也有认真在复习。可是由于概念比较抽象，又因为自己没有学习过python，学习起来比较吃力。好在老师合理的教学安排，安排了助教学长带我们线下实践，亲手操作，加上三位助教学长耐心的为我们讲解代码、答疑解惑，让我对自然语言处理有了初步的认识。

刚开始做大作业的时候有一点手足无措，python薄弱的基本功和最近庞大的学业压力让我面对一个bug足足拖了很久，自己上网查阅很多资料还是没能解决，于是去问身边的朋友同学，尽管大家都热情的向我提供帮助，可是问题依旧没有解决。最后询问了穆学长，在这里真的要非常感谢穆学长，他耐心地回答我的询问，帮我解决问题，让我得以完成本次作业，并且在解决问题的过程中对LSTM模型有了更深入的了解，不仅完成了基础作业内容：手动搭建单层LSTM网络，还实现了提高内容：双层LSTM网络的搭建，并且在此基础上进行自己的拓展：实现多层LSTM网络，测试并分析网络层数对性能的影响及原因，在前面实验的基础上实现双向LSTM网络的搭建。

半学期的自然语言处理课程以及本次的实践作业让我明白：学习理论知识很重要，可是想深入学好计算机相关课程只是单单熟知理论是远远不够的，必须要经过自己的手动实践，实践+理论才是掌握一门课程最好的方式。

最后的最后，我想由衷感谢肖桐老师、马安香老师、以及三位助教学长。上肖桐老师的课总能有很大收获，不仅仅是在学业方面，光是肖桐老师的人品以及为人处世的方式就能给我们带来很大的启发。马安香老师也是我在东大很喜欢的老师之一，她谦逊温和的性格以及授课时的举止大方也是我一直以来的榜样和发展的方向。三位助教学长真的给予我们很大的帮助，学长们带我们上实践课的时候就足以感受到他们的认真负责，课后更是耐心认真的为我们答疑解惑，如果没有他们的细心指导我可能也不会对这门课有这么深的理解以及这么大的收获。

很开心完成了本次自然语言课程的学习，希望在以后的学业生涯中还能有幸和老师们学长们一起学习，老师们和助教学长们辛苦了！