

## 实验三：自动写诗

---

### 一、实验目的

1. 理解和掌握循环神经网络概念及在深度学习框架中的实现。
2. 掌握使用深度学习框架进行文本生成任务的基本流程：如数据读取、构造网络、训练和预测等。
3. 了解 PyTorch（或其他框架）在 GPU 上的使用方法

### 二、实验环境

本次编程实验主要基于miniconda和pytorch框架进行，使用的各个库的版本号如下：

- python 3.7.16
- pytorch 1.10.0
- torchvision 0.11.0
- numpy 1.21.6
- tensorboard 2.11.2
- cuda 11.8

### 三、数据准备与模型构建

#### 数据准备

---

利用老师发放的'tang.npz'数据集，使用numpy中的load函数进行加载，并将加载的数据放入dataloader中用于批量处理。

#### 模型构建

---

创建一个继承自 nn.Module 的类，在这个类的\_\_init\_\_()中定义网络结构，在 forward 中定义前向传播过程。在本次实验中，自己构建了 Embedding层、三层 LSTM、全连接层三个网络层，代码如下：

```
class PoetryModel(BasicModule):
    def __init__(self,vocab_size, embedding_dim, hidden_dim):
        super().__init__()
        self.modelName="PoetryModel"
        self.embeddings=nn.Embedding(vocab_size,embedding_dim)
        self.hidden_dim=hidden_dim
        # 这稍微有点不一样
        self.lstm=nn.LSTM(embedding_dim,hidden_dim,num_layers=3)
        self.fc = nn.Sequential(
            nn.Linear(hidden_dim, 512),
            nn.ReLU(inplace=True),
            nn.Linear(512, 2048),
            nn.ReLU(inplace=True),
            nn.Linear(2048, 4096),
            nn.ReLU(inplace=True),
            nn.Linear(4096, vocab_size)
```

```

    )
    def forward(self, input, hidden=None):
        seq_len, batch_size = input.size()

        if hidden is None:
            h_0 = input.data.new(3, batch_size, self.hidden_dim).fill_(0).float()
            c_0 = input.data.new(3, batch_size, self.hidden_dim).fill_(0).float()
        else:
            h_0, c_0 = hidden

        embeds = self.embeddings(input)
        # [seq_len, batch] => [seq_len, batch, embedding_dim]
        output, hidden = self.lstm(embeds, (h_0, c_0)) # output的shape为
        [seq_len, batch, hidden_dim]
        output = self.fc(output.view(seq_len * batch_size, -1)) # output的shape为
        [seq_len*batch, vocab_size]
        return output, hidden

```

#### 四、实验过程及结果

**模型训练** 模型训练包括定义模型、设置优化器和损失函数、获取模型输出、计算误差、误差反向传播等步骤

**实验一：** 在本次代码中，采用的参数设置为

- 优化器：采用Adm梯度下降优化器
- 损失函数：采用 CrossEntropyLoss 交叉熵损失函数
- 超参设置： batch\_size=64, num\_epochs=20, lr=0.001
- 补充说明：lr利用scheduler中的ReduceLROnPlateau进行控制，在5轮损失不下降的平台区缩小为0.01；受限于单机单卡的限制，batch\_size的值不能设置太大

实验一是在google colab上完成，

google colab简介：Colaboratory 简称“Colab”，是 Google Research 团队开发的一款产品。在 Colab 中，任何人都可以通过浏览器编写和执行任意 Python 代码。它尤其适合机器学习、数据分析和教育目的。从技术上来说，Colab 是一种托管式 Jupyter 笔记本服务。用户无需设置，就可以直接使用，同时还能获得 GPU 等计算资源的免费使用权限。

在设置lr=0.001前，自己其实还尝试了lr=0.1、lr=0.01和lr=0.0001三种情况，前两种发现loss值非常大，显然步数太大使得loss很难往极小值处进展，在lr=0.0001时发现loss在每批次的变化非常小，因此最终选定了lr=0.001并取得了较好的效果。最终实验结果如下图所示。实验结果中loss值达到1.7左右，我们将训练好的模型进行了保存以便后续的实验和继续训练。

```

-----第 20 轮训练开始-----
epoch: 19, 训练次数: 0, 训练损失: 1.7643
epoch: 19, 训练次数: 100, 训练损失: 1.7612
epoch: 19, 训练次数: 200, 训练损失: 1.7584
epoch: 19, 训练次数: 300, 训练损失: 1.7557
epoch: 19, 训练次数: 400, 训练损失: 1.7531
epoch: 19, 训练次数: 500, 训练损失: 1.7508
epoch: 19, 训练次数: 600, 训练损失: 1.7484
epoch: 19, 训练次数: 700, 训练损失: 1.7461
epoch: 19, 训练次数: 800, 训练损失: 1.7438
保存的模型路径为: models_save/<class 'models.PoetryModel.PoetryModel'>_0406_04_44.pth

```

实验中续写的诗句开头分别为“飞流直下三千尺”、“月落乌啼霜满天”、“拣尽寒枝不肯歇”、“春宵苦短日高起”、“只恐双

溪舴艋舟","衣带渐宽终不悔",得到的结果如下图所示,这里因为我们限制的续写最大长度为125,因此有些诗句还没完全续写完成就已经结束了,不过总体来看续写的诗句有一定的意境。

飞流直下三千尺来进行续写的诗句为

飞流直下三千尺,歎色初烧温嶽水。  
四面初融养此羣,万般无角遭崩往。  
金屏云座相交憑,龙虎腾谋汗人举。  
我来陞卫如有期,铸得咸源到人口。  
一朝燕雀切章句,今日承恩詣神室。  
一朝勋臣賜盛连,万里雄卿如俊身。  
一朝勋业备万国,今日朔雪双双轮。  
万里黄云殢流水,一朝游猎暮  
月落乌啼霜满天来进行续写的诗句为  
月落乌啼霜满天,江讴越侶无人言。  
君不见邛中別語炎顏,君不见西陵之歌謝君。  
古风斜飞暮天末,哀风飒飒绕陇头。  
去时载马不相见,壯心凋落湘水声。  
黃鸝相隨不歸去,離別不達千里程。  
离人驻櫂徒自乐,寸心恍苒如秋风。  
獅子羽书不相见,他人惯传一书至。  
君不见梁上泥,妾身

谁知盘中餐来进行续写的诗句为

谁知盘中餐,翩鞍貫荆榛。  
一彈一杯酒,一日不復听。  
請君噉我家,儿女不羸牽。  
君看北邙公,搗衣挂青苔。  
我有一杯酒,一杯聊自怜。  
請君噉君泣,何必长苦辛。  
君今妾何易,妾有一黄金。  
顧影看別离,不如死家好。  
父母不重行,亲戚多咨嗟。  
妾身虽不在,妾弃茜鬟成。  
妾身且如此

春宵苦短日高起来进行续写的诗句为

春宵苦短日高起,邯鄲故人皆苦悔。  
使君有钱复如此,斗酒不及情历数。  
苔然一幸增漁者,不觉展转甘妃妇。  
不知肝胆向人间,不觉身为何所为。  
君不见,我如何,一栖酒,越江佳辰人于逐。  
君不见东流水深,云和雪屋无人见。  
何时得见少年言,一身不得长相见。

接着我使用刚刚保存的模型，控制续写最大长度为32也即四句诗，是常见的一首七言诗的长度，续写的结果如

```
(pytorch2) PS D:\计算所研究生学习\深度学习\实验编程作业\实验3: 自动写诗> python .\main.py
飞流直下三千尺来进行续写的诗句为
飞流直下三千尺，散色初烧温嶽水。
四面初融养此羣，万般无角遭崩往。

月落乌啼霜满天来进行续写的诗句为
月落乌啼霜满天，江返越侣无人言。
君不见邗中别语炎颜，君不见西陵之
拣尽寒枝不肯歇来进行续写的诗句为
拣尽寒枝不肯歇，金珮丝轻银袖卷。
促笋裁巾尾半空，句句箱中序妆扇。

春宵苦短日高起来来进行续写的诗句为
春宵苦短日高起，邯郸故人皆苦悔。
使君有钱复如此，斗酒不及情历数。

只恐双溪舴艋舟来进行续写的诗句为
只恐双溪舴艋舟，不知何处但逢山。
一朝羁旅在天畔，一片孤城随水宿。

衣带渐宽终不悔来进行续写的诗句为
衣带渐宽终不悔，芙蓉色冷难留客。
不知何处有芳菲，惟有坛边风雨新。
```

下所示

**实验二** 在实验一中我们保存了模型并进行了续写，但是我观察到训练了20轮后，loss其实一直在降低，所以我想是不是继续训练的话，loss还能接着降低呢？因此我加载保存好的模型，继续训练了10轮，其他参数保持不

```
epoch: 7, 训练次数: 0, 训练损失: 1.2218
epoch: 7, 训练次数: 100, 训练损失: 1.2194
epoch: 7, 训练次数: 200, 训练损失: 1.2173
epoch: 7, 训练次数: 300, 训练损失: 1.2157
epoch: 7, 训练次数: 400, 训练损失: 1.2144
epoch: 7, 训练次数: 500, 训练损失: 1.2133
epoch: 7, 训练次数: 600, 训练损失: 1.2124
epoch: 7, 训练次数: 700, 训练损失: 1.2118
epoch: 7, 训练次数: 800, 训练损失: 1.2113
-----第 9 轮训练开始-----
epoch: 8, 训练次数: 0, 训练损失: 1.2108
epoch: 8, 训练次数: 100, 训练损失: 1.2086
epoch: 8, 训练次数: 200, 训练损失: 1.2068
epoch: 8, 训练次数: 300, 训练损失: 1.2053
epoch: 8, 训练次数: 400, 训练损失: 1.2038
```

变，得到结果如下图，可见训练loss降到了1.2左右

依然用实验一中的开头，续写诗句如下，可以看到续写的诗句感觉质量更高，逻辑性更强，并带有一定的押韵，例如“花门一别不复归，折来直到巫江北”，效果明显更好

```
(pytorch2) PS D:\计算所研究生学习\深度学习\实验编程作业\实验3: 自动写诗> python .\main.py
飞流直下三千尺来进行续写的诗句为
飞流直下三千尺，半夜浮成百千里。
杨门后堂侍天回，桂苑千叠银头散。

月落乌啼霜满天来进行续写的诗句为
月落乌啼霜满天，江南波涛晦将军。
去年花落燕苏殿，今年蚕贯轮中烟。

拣尽寒枝不肯歇来进行续写的诗句为
拣尽寒枝不肯歇，人间口里苦相呼。
自从此年长读实也，不知何物不相离
春宵苦短日高起来来进行续写的诗句为
春宵苦短日高起，雨入洞庭椒作雪。
妾闻生子夜理毕，一声直入中原去。

只恐双溪舴艋舟来进行续写的诗句为
只恐双溪舴艋舟，不知何处逐君心断续。
去去东坡多黄菊，此地千花为君
衣带渐宽终不悔来进行续写的诗句为
衣带渐宽终不悔，一枝欲减花间泣。
花门一别不复归，折来直到巫江北。
```

在该模型下，自己也实现了藏头诗的功能，藏头诗其实也和自动写诗实现原理基本一致，只不过是在每句话的开头，强制使用自己规定的字即可。本次使用的藏头诗的谜为['计算机视觉','自然语言处理','大预言模型','深度学

习]', 得到的结果如下:

```
(pytorch2) PS D:\计算所研究生学习\深度学习\实验编程作业\实验3: 自动写诗> python .\main.py
计算机视觉作为藏头词来进行构造的诗句为
计融日已长, 我生情亦悲。
算乎失所育, 不得豁禄时。
机上不得力, 鉴何所须悲。
视丧不呻忍, 积之不可持。
觉饱不呻蠲, 饱则不能除。

自然语言处理作为藏头词来进行构造的诗句为
自从南陌尘山水, 不见江南离别曲。
然诺无心不比时, 秋风忽到青门里。
语声四散不可语, 脉脉行行翻入听。
言之忘怀行且哀, 但愿一身与所依。
处士拘农贫卖卜, 依稀放逐一枝蜂。
理人不拣呵呵谷, 不袜不成空却饥。

大预言模型作为藏头词来进行构造的诗句为
大道谅无外, 潜光亦何贞。
预读无碍士, 不为寒馁忧。
言之无偏情, 岂为众木荣。
模砺既一遇, 权奸亦无形。
型风烹众木, 器烈毒雄惊。

深度学习作为藏头词来进行构造的诗句为
深宫高楼入宫立, 崖下孤城如蜀嶺。
度舞凝红堕澹香, 画桃斜影攢金阁。
学人住处花绵绵, 宫中美女屏风高。
习底骊台闲受对, 僊花滴沥露鳞鳞。
```

**实验三** 因为感觉实验二中loss还可以降级, 因此加载了实验二保存的模型继续训练10轮, 得到结果如下图所示

```
epoch: 8, 训练次数: 300, 训练损失: 1.0364
epoch: 8, 训练次数: 400, 训练损失: 1.0355
epoch: 8, 训练次数: 500, 训练损失: 1.0349
epoch: 8, 训练次数: 600, 训练损失: 1.0347
epoch: 8, 训练次数: 700, 训练损失: 1.0344
epoch: 8, 训练次数: 800, 训练损失: 1.0344
-----第 10 轮训练开始-----
epoch: 9, 训练次数: 0, 训练损失: 1.0343
epoch: 9, 训练次数: 100, 训练损失: 1.0328
epoch: 9, 训练次数: 200, 训练损失: 1.0314
epoch: 9, 训练次数: 300, 训练损失: 1.0305
epoch: 9, 训练次数: 400, 训练损失: 1.0297
epoch: 9, 训练次数: 500, 训练损失: 1.0290
epoch: 9, 训练次数: 600, 训练损失: 1.0286
epoch: 9, 训练次数: 700, 训练损失: 1.0283
epoch: 9, 训练次数: 800, 训练损失: 1.0281
```

示, 在这里loss已经降到了1.0左右。

续写的诗句结果如下, 这里自己感觉不如实验二中的诗句, 多了很多生僻字, 而且不怎么押韵, 不知道是否是因为巧合的原因还是训练次数太多导致了过拟合, 生成的诗句不再有意境和想象力。因此也决定了不再继续往



飞流直下三千尺来进行续写的诗句为  
 飞流直下三千尺，彭泽重重二华端。  
 秦兵追此征戍去，沙塞连连半天碧。

月落乌啼霜满天来进行续写的诗句为  
 月落乌啼霜满天，江讴月照谁家愁？身无父母生离别，牛下名山几千里。

拣尽寒枝不肯歇来进行续写的诗句为  
 拣尽寒枝不肯歇，金髻帐里裁红纸。  
 轻笼蹙齿傍田家，慢提啾咽放毡毡。

春宵苦短日高起来进行续写的诗句为  
 春宵苦短日高起，啼乌欲断还铮铮。  
 未有谏草抱光者，又不得老作蓬莱。

只恐双溪舴艋舟来进行续写的诗句为  
 只恐双溪舴艋舟，赤城侍臣夜献酬。  
 祇将志士守君子，不独报恩并主材。

衣带渐宽终不悔来进行续写的诗句为  
 衣带渐宽终不悔，一酌一杯妻大醉。  
 今朝街上泣，今日又分散。

下训练

```
(pytorch2) PS D:\计算所研究生学习\深度学习\实验编程作业\实验3: 自动写诗> python .\main.py
计算机视觉作为藏头词来进行构造的诗句为
计灵惟燕射，日月自生灵。
算礼聆神算，雄猷禀介攸。
机生惟匪媚，铢穀忽相笼。
视遵贞以配，泄命智祖并。
觉成于子策，请以佐尧仁。

自然语言处理作为藏头词来进行构造的诗句为
自从南国守风山，独立东南一断游。
然是乱离何日见，共矜多少断肠游。
语无多夏犹宜健，心怯中宵又有秋。
言辞齿腭歌日媚，欲施鹰甲病肌收。
处年不觉经年笑，强乐那堪恨别愁。
理践必知嗟故宅，病公何必访公侯。

大预言模型作为藏头词来进行构造的诗句为
大道谅无遗，许官何所云。
预埋循吏，不及羲皇年。
言让受芳席，朝成趋妙声。
模狐埋卜虏，杂鼓取乘舟。
型骑喧营从，澄盞杂吹空。

深度学习作为藏头词来进行构造的诗句为
深沈阃外畧，奕世挺英拟。
度门抚众气，荟蔚如委积。
学拔本帝功，贪兵不得武。
习我以为说，酷特天下彦。
```

藏头诗如下：

综合实验一、实验二和实验三的实验结果，自己对实验二的作诗结果最为满意，此时也已训练出了不错的模型。**小结** 通过这次实验，自己学会了利用 Pytorch 深度学习框架实现自动写诗，深刻认识了LSTM网络模型，embedding层等，增加了对于卷积神经网络的理解。对于深度学习的整个流程，包括数据预处理、网络模型构建、模型训练、scheduler、pytorch下model模块的使用等方面都有了更深层次的理解，代码的组织编写水平也有了很大的提高，对于后续实验起到了很好的启发作用，受益匪浅！