AI-Poet

一款基于深度学习的 AI 作诗系统

id: 519021910861
name: 徐惠东

（个人单独组队）

简介

近年来，人工智能渗透了我们生活的方方面面，给我们的衣食住行都带来了不少便利。纵观当今的人工智能在交通管理、环境治理、数字能源和社会治理的应用，无论是推荐系统还是异常检测，他们都在为人们的生存和生活添砖加瓦。

然而，“生活不止眼前的苟且，还有**诗**和远方”。我们有理由相信，当人们的基本生活需求已经满足后，**作诗读诗**将会是人工智能的高层次应用。正如中国在盛唐时期古诗遍地开花，在不久的将来，人工智能写诗也必将掀起一波浪潮。

因此，AI-Poet 参考《Pytorch入门与实践》教程和 GitLab 上开源代码框架，采用**循环神经网络 (RNN)** 中的**长短期记忆 (LSTM)** 模型进行训练，最终效果可以支持**首句续写**和**藏头诗**两种功能，并且实现了基本的音律和意境。

关键词
深度学习 循环神经网络 长短期记忆 预训练模型

环境配置

最初试图使用 *MacBook Pro 2017* 进行训练，然而因为是因特尔显卡没办法直接使用 *cuda*，所以只能使用 CPU 导致训练速度极慢。经过粗略计算，跑完一遍模型需要约 15 天时间，因此不在本机跑模型。

之后尝试过阿里云租借服务器、百度飞桨平台等，但都觉得过于复杂。最终选用谷歌的 Colab 平台进行训练，所分配到的 GPU 为 *Tesla P100*，训练时间约为 3 小时。

```
Sat Dec 25 01:46:09 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 495.44       Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 00000000:00:04.0 Off |                    0 |
| N/A   41C    P0    28W / 250W |      0MiB / 16280MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

代码实现

数据集

使用整理好的 **numpy** 格式的开源数据集，其中包含唐诗共 57580 首 * 125 字，不足和超出 125 字的都已经被补全或者截断。

```
# 处理数据
datas = np.load("/content/drive/MyDrive/AI-Poet/tang.npz",
allow_pickle=True)
data = datas['data']
ix2word = datas['ix2word'].item()
word2ix = datas['word2ix'].item()
data = t.from_numpy(data)
dataloader = DataLoader(data,
                        batch_size=Config.batch_size,
                        shuffle=True,
                        num_workers=2)
```

细节解释

1. data 是 numpy 数组，共 57580 首 * 125 字。
2. word2ix 和 ix2word 都是字典类型，用于字符和序号的映射。

LSTM 循环神经网络

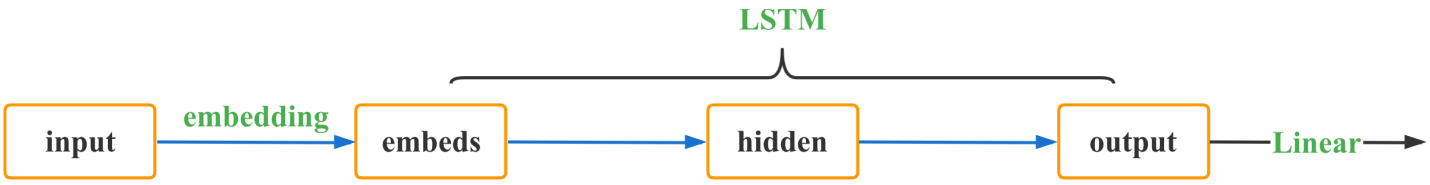
LSTM 是一种特殊的 RNN，能够解决长序列训练过程中的梯度消失和梯度爆炸问题，相比于 RNN 只有一个传递状态 h^t，LSTM 有两个传输状态分别是 c^t (cell state) 和 h^t (hidden state)。

其中对于传递下去的 c^t 改变得较慢，通常输出的 c^t 是上一个状态传过来的 c^{t-1} 加上一些数值，而 h^t 则在不同节点下有较大区别。本模型便采用了 LSTM 模型进行训练。

```
class PoetryModel(nn.Module):
    def __init__(self, vocab_size, embedding_dim, hidden_dim):
        super(PoetryModel, self).__init__()
        self.hidden_dim = hidden_dim
        # 词向量层，词表大小 * 向量维度
        self.embeddings = nn.Embedding(vocab_size, embedding_dim)
        # 网络主要结构
        self.lstm = nn.LSTM(embedding_dim, self.hidden_dim,
num_layers=Config.num_layers)
        # 进行分类
        self.linear = nn.Linear(self.hidden_dim, vocab_size)

    def forward(self, input, hidden=None):
        seq_len, batch_size = input.size()
        # print(input.shape)
        if hidden is None:
            h_0 = input.data.new(Config.num_layers, batch_size,
self.hidden_dim).fill_(0).float()
            c_0 = input.data.new(Config.num_layers, batch_size,
self.hidden_dim).fill_(0).float()
        else:
            h_0, c_0 = hidden
        # 输入 序列长度 * batch(每个汉字是一个数字下标),
        # 输出 序列长度 * batch * 向量维度
        embeds = self.embeddings(input)
        # 输出hidden的大小:  序列长度 * batch * hidden_dim
        output, hidden = self.lstm(embeds, (h_0, c_0))
        output = self.linear(output.view(seq_len * batch_size, -1))
        return output, hidden
```

细节解释



将数据集作为喂给模型的作为 *input*，先经过 *embedding* 预处理得到 *embeds* 层，然后经过 *LSTM* 进行训练得到 *hidden* 层和 *output* 层，最后经过 *Linear* 层判别，然后反向传播并循环训练即可。

运行展示

使用方法

使用 Colab 打开项目，在 *AI-Poet.ipynb* 中 *User Test* 部分点击运行，根据提示输入 **1（首句生成）**或者 **2（藏头诗）**来选择生成诗句模式。

若是**首句生成**模式，则需再输入诗歌首句。若是**藏头诗**模式，则需输入诗歌藏头部分。

成果展示

项目总结

在一个学期的《人工智能》选修课中，我学习到了诸如机器学习、深度学习、自然语言处理等多种人工智能相关技术，也逐步了解到自动驾驶、智能推荐等人工智能的广大应用前景，感触颇深。因为我自身是软件工程专业高年级学生（大三），且之前对机器学习和深度学习等技术略有接触，因此这次课程大作业选择了单人组队，这也大大增加了我许多选题和解题的灵活性。

在我看来，AI 的各种技术目前都还在起步阶段，相对成熟的推荐系统也仅仅是通过有限的用户浏览历史来进行简单的预测，而大数据时代的到来不仅意味着个人数据的广泛搜集和使用，更为计算机的算力提升和相应模型复杂度增加提供了契机。在未来，AI 必然会更进一步，做的更好，我也希望能够参与其中。

感谢《人工智能》课程 孔令和老师、许岩岩老师的精彩授课和悉心指导，段勇帅助教也为我提供了许多帮助。