基于RNN的诗歌生成实验报告

1. 关于PoetryModel类的实现
2. 初始化：

继承nn.Module类之后，定义embedding模型，输入的维数为vocab\_size，输出的维数为embedding\_size,之后定义lstm模型，注意它的要求是两层，用num\_layers = 2来表示两层，输入是embeddingdim，输出是hiddndim，最后定义线性模型，输入hiddendim输出是vocabdim，它的目的是将输出的output向量的维数映射到vocabdim维数。

1. Forward函数，接口为input 和 hidden（初始化为0）

hidden 为两个tensor组成的元组，h0和c0.将输入input向量经过embedding后经过神经网络LSTM后，将output线性模型过滤，最后返回output和hidden

1. test 函数和 acrostic\_test函数的实现
2. test 函数实现：

将start word转化为列表，并将第一个词设置为<START>对应的字典索引值，然后开始进行循环，每轮开始都对它进行神经网络的预测。其中如果循环的次数小于startword的长度，诗句还在startword中，将下一个词的对应索引作为下一个输入向量，如果长度超出时，用topk将预测值最大即最有可能的对应索引取出作为下一个输入向量。最后当向量为“<EOP>”时结束，并将最后一个删除，完成test函数过程

(二) acrostic函数实现：

首先建立空列表，设置第一个输入为<START>对应的字典索引值，最后开始循环，每轮将input和hidden都经过model处理，同时设定pre记录上一个单词，当上一个单词为一句话的结尾时，我们在从startword中取词作为藏头诗的开头，当藏头词全部用完时，表示这首藏头诗创作完成退出循环。如果不是结尾，则查找出现可能性最大的词，作为input，找到它对应的字典value，加入到列表当中，最后完成藏头诗。

1. 其他问题解决
2. 如何对模型进行调试

我在处理模型的bug的过程中，不断的用print（XXX.shape）不断的输出关键向量的维数，查看维数是否匹配，因为在循环神经网络的训练过程中，维数的匹配非常重要，当维数不匹配时，锁定错误位置，及时调整。

1. 如何提升模型训练效率，采用gpu加速

在模型定义过程中，使用代码self.device = torch.device("cuda" if torch.cuda.is\_available() else "cpu")这样当电脑的gpu可用的时候，device就会被设定为cuda，同时在self.model定义时，后面加入.to(self.device)这样就可以用gpu进行加速，如果gpu不可用，那就会用cpu不会影响其他运算。