神经语言模型demo

一、文件夹内包含以下内容：

├── data

│   ├── data.npy

│   ├── input.txt

│   ├── jay.seg.txt

│   ├── poetry.seg.txt

│   └── vocab.pkl

├── model

├── input\_data.py

├── model.py

├── train.py

└── vallina\_rnn.py

**--数据文件夹data：**中文文件jay.seg.txt（周杰伦歌词），poetry.seg.txt（古诗 词），英文文件 input.txt（莎士比亚剧本）

**--模型文件夹model：**用于保存模型

**--vallina\_rnn.py：**Vallina RNN程序（基于纯python实现，基于字符级的神经语言模型，核心函数包含正向计算、梯度反向传播，训练和采样生成文本）

**-- input\_data.py，train.py，model.py：**Tensorflow版程序：

（两个程序都可在笔记本运行，但tensorflow版速度较慢，装有tensorflow的服务器应该均可运行）

**1、python版基于字符的神经语言模型vallina\_rnn.py**

直接运行 python vallina\_rnn.py

**训练数据默认来自./data/input.txt，**每100步输出一次loss，每1000步输出一次采样生成的文本

**2、tensorflow版基于字的神经语言模型 train.py model.py input\_data.py**

直接运行 python train.py（可选参数包括data\_dir, file\_name, save\_dir, save\_freq, batch\_size, seq\_length, hidden\_num, word\_dim, num\_epoches, model, grad\_clip, learning\_rate）

**训练数据默认来自./data/jay.seg.txt，模型保存于model文件夹；**input\_data.py，生成词表和batches，生成的文件保存在data文件夹中；train.py，每迭代一轮输出一次loss，并输出采样生成的文本；model.py，模型主体，包括训练和采样。

**model.py中模型建立核心函数：**

1）词向量查表

inputs = tf.nn.embedding\_lookup(embedding, self.input\_data)

2）建立LSTM cell

cell = tf.nn.rnn\_cell.BasicLSTMCell(args.hidden\_num)

3）给定初始状态计算LSTM每个时刻的输出和最终状态

output, last\_state = tf.nn.dynamic\_rnn (cell, inputs, initial\_state=self.initial\_state)

4）计算交叉熵损失

self.loss = tf.nn.sparse\_softmax\_cross\_entropy\_with\_logits(logits=self.logits, labels=flat\_targets)

5）梯度更新

optimizer = tf.train.AdamOptimizer(self.lr)

self.train\_op = optimizer.apply\_gradients(zip(grads, tvars))

**采样函数：**

输入起始标签’<START>’，根据LSTM的输出计算下一个词的概率分布，选择概率最大的或根据正态分布采样生成下一个词

二、参考代码与资料

1、toy版基于字符的神经语言模型（基于纯python实现）

源代码 <https://gist.github.com/karpathy/d4dee566867f8291f086>

来源于karpathy博客<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

2、tensorflow版基于字的神经语言模型

1）ptb语言模型训练

源代码<https://github.com/tensorflow/models/blob/master/tutorials/rnn/ptb/ptb_word_lm.py>

来源于官网tutorial <https://www.tensorflow.org/tutorials/recurrent>

2）基于字符的神经语言模型

源代码 <https://github.com/sherjilozair/char-rnn-tensorflow>