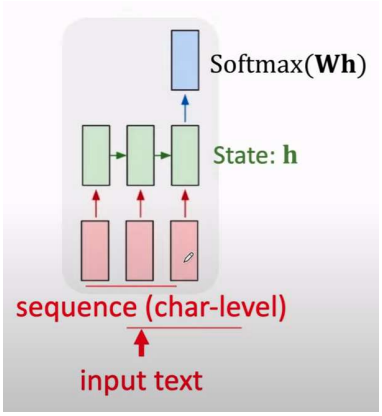
RNN for text generation

2021年2月4日 14:24

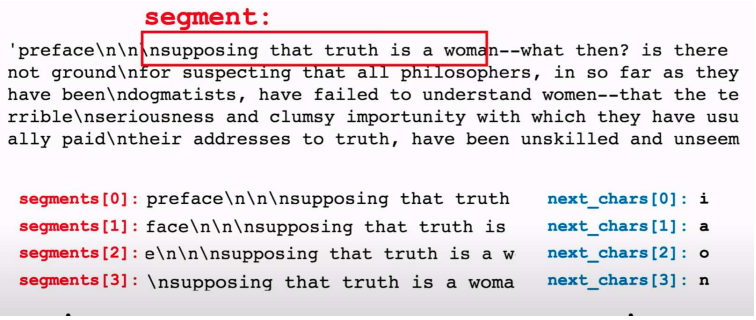
1. RNN for Text Generation （文本生成）

- 链接：https://www.youtube.com/watch?v=10cjvcrU_ZU&t=278s
- Main idea



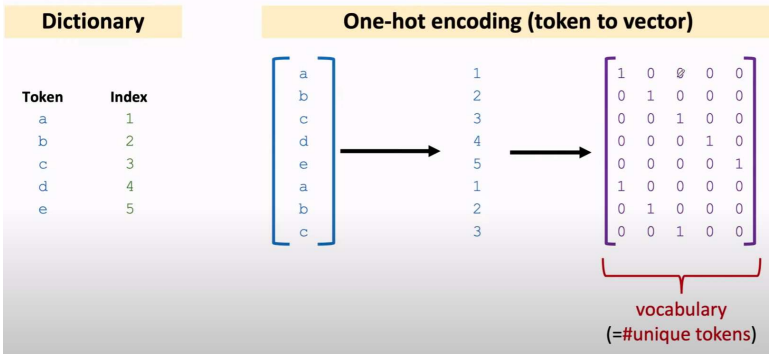
1. 训练的文本是普通文本，然后将文本分割成固定长度的片段（以字符或者单个词汇的方式，字符向量，词向量），
2. 再将这些片段中的每个字符进行one-hot encoder编码成向量，挨个将每个向量输入到RNN模型中，最后得到最后一个返回向量h
3. 再上层为全连接层Dense，softmax分类器，返回一个概率，选择最大概率的为下一个生成的字符

- prepare training data



1. 以固定长度的片段为segment，下一个字符为label，在训练中预测这个label
2. 注意还有一个参数是stride，表示每次红框往后移动多少个字符

- dictionary and one-hot encoding



1. 在此以character为例子，英文只有26个字母，矩阵的维度小
2. 要是以单词来分割的话，字典中所有单词的数量即为矩阵的维度，如果训练样本不够多的话，非常容易产生矩阵稀疏的问题
3. 有多少个字符，矩阵就有几个列

接下来就是build neural network，设置好 seg_len 和 vocabulary, Dense中的activation function激活函数

然后编译模型，选择优化器optimizers和损失函数 crossentropy

最后设置拟合fit参数，训练好模型

1. 模型训练的总结

最后设置拟合fit参数，训练好模型

1. 模型训练的总结

1. Partition text to (segment, next_char) pairs.
2. One-hot encode the characters.
   - Character → $v \times 1$ vector.
   - Segment → $l \times v$ matrix.
3. Build and train a neural network.
   - $l \times v$ matrix ==> LSTM ==> Dense ==> $v \times 1$ vector.

- predict the next char

根据训练好的模型，输入测试文本，便能得到下一个预测字符对应的概率，选择字符的方法有三种：

- 第一种，greedy selection， 直接选最大概率值的那个（确定性的，没有随机性，不够多元化，取决于输入向量）
- 第二种，multinomial distribution，多项式分布中随机抽取（太随机了，会出现拼写错误）
- 第三种，用temperature的方法调整概率值，小的变更大，大的变更小，效果介于上面两种

1. 生成预测的总结

1. Propose a seed segment.
2. Repeat the followings:
   a) Feed the segment (with one-hot) to the neural network.
   b) The neural network outputs probabilities.
   c) next_char ← Sample from the probabilities.
   d) Append next_char to the segment.