**TensorFlow｜4｜循环卷积网络**

1. 基本环境

|  |  |
| --- | --- |
| 程序名称 | trainData.py |
| 硬件环境 | MacOSX |
| 软件环境 | Python3, Spyder, Anaconda |

1. 核心问题：循环卷积网络的构造
2. 解决过程：

* 设置辅助函数，用于得到训练集和测试集：getTrainBatch()和getTestBatch()
* 设置模型所需的参数：训练批次大小batchSize为24，lstmUnits = 64，numClasses = 2，迭代次数iterations = 100000，LSTM单元的数量maxSeqLength=60
* 设置占位符：输入input\_data，输出的标签labels
* 将输入的文字利用“TensorFlow｜3｜词嵌入”实验中得到的model转化成向量：

data = tf.nn.embedding\_lookup(model.vectors,input\_data)

* 借助TensorFlow的tf.contrib.rnn.DropoutWrapper函数构建长短期记忆网络
* 借助TensorFlow的tf.nn.dynamic\_rnn函数构建循环卷积网络tf.nn.dynamic\_rnn函数的第一个输出可以被认为是最后的隐藏状态向量。这个向量将被重新确定维度，然后乘以最后的权重矩阵和一个偏置项来获得最终的输出值。
* 得到预测函数，与原始标签比较得到准确率

1. 代码：

#辅助函数

from random import randint

def getTrainBatch():

labels = []

arr = np.zeros([batchSize, maxSeqLength])

for i in range(batchSize):

if (i % 2 == 0):

num = randint(1,4000)

labels.append([1,0])

else:

num = randint(6001,10000)

labels.append([0,1])

ids=firstSen

arr[i] = firstSen[num-1:num]

return arr, labels

def getTestBatch():

labels = []

arr = np.zeros([batchSize, maxSeqLength])

for i in range(batchSize):

num = randint(4001,6000)

if (num <= 5001):

labels.append([1,0])

else:

labels.append([0,1])

arr[i] = firstSen[num-1:num]

return arr, labels

#RNN模型

batchSize = 24

lstmUnits = 64

numClasses = 2

iterations = 100000

#LSTM单元的数量

maxSeqLength=60

import tensorflow as tf

tf.reset\_default\_graph()

#标签占位符

labels = tf.placeholder(tf.float32, [batchSize, numClasses])

#输入占位符

input\_data = tf.placeholder(tf.int32, [batchSize, maxSeqLength])

#文本转换为向量

data = tf.Variable(tf.zeros([batchSize, maxSeqLength, numDimensions]),dtype=tf.float32)

data = tf.nn.embedding\_lookup(model.vectors,input\_data)

#长短期记忆网络

lstmCell = tf.contrib.rnn.BasicLSTMCell(lstmUnits)

lstmCell = tf.contrib.rnn.DropoutWrapper(cell=lstmCell, output\_keep\_prob=0.75)

#循环卷积网络

value, \_ = tf.nn.dynamic\_rnn(lstmCell, data, dtype=tf.float64)

#隐藏状态向量重新确定维度

weight = tf.Variable(tf.truncated\_normal([lstmUnits, numClasses]),dtype=tf.float32)

bias = tf.Variable(tf.constant(0.1, shape=[numClasses]))

value = tf.transpose(value, [1, 0, 2])

last = tf.gather(value, int(value.get\_shape()[0]) - 1)

last = tf.cast(last, tf.float32)

#得到预测函数

prediction = (tf.matmul(last, weight) + bias)

correctPred = tf.equal(tf.argmax(prediction,1), tf.argmax(labels,1))

accuracy = tf.reduce\_mean(tf.cast(correctPred, tf.float32))

1. 结果：

本次实验仅完成构造循环卷积网络，训练留待下次实验。