



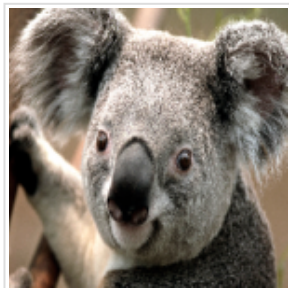
栏

目录视图

摘要视图

RSS 订阅

个人资料



BYR_jiandong



访问：132437次

积分：3525

等级：BLOG > 5

排名：第8044名

原创：220篇 转载：41篇

译文：0篇 评论：22条

【有奖投票】玩转Dragonboard 410c 的正确姿势 CSDN日报20170406 —— 《代码很烂，所以离职。》 Python数据分析与机器学习 博客搬家，有礼相送

tensorflow 使用LSTM进行分类的例程

标签：tensorflow

2016-11-20 14:58

846人阅读

评论(1)

收藏

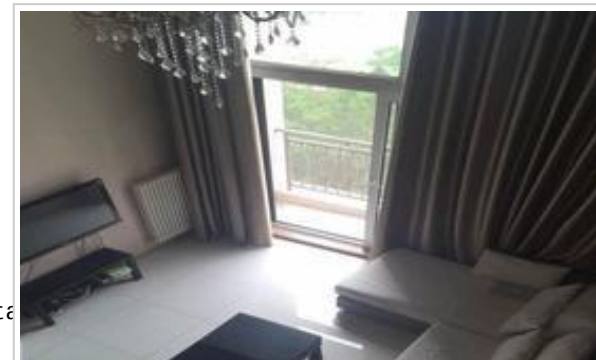
举报

分类： tensorflow调研 (36)

关闭

版权声明：本文为博主原创文章，未经博主允许不得转载。

```
[python]
01. import tensorflow as tf
02. import sys
03. from tensorflow.examples.tutorials.mnist import input_data
04.
05. # this is data
06. mnist = input_data.read_data_sets('MNIST_data', one_hot=True)
07.
08. # hyperparameters
```



一居租房



在国家 中国内地(China)

机号码

验证码

用户名

密码

确认密码

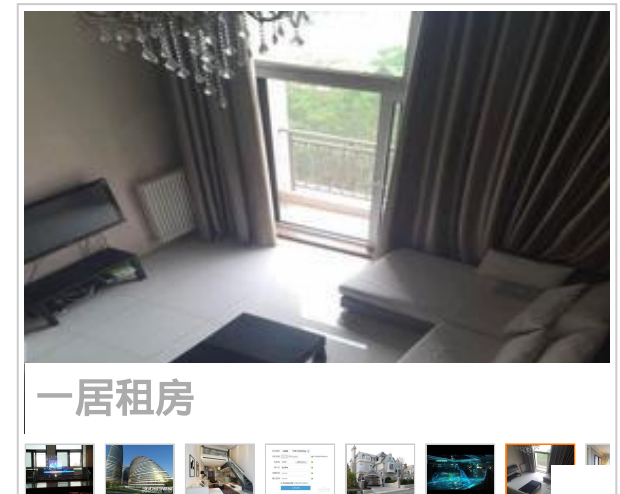
- 人生感悟 (1)
- C++基础 (20)
- C# (1)
- ACM 贪心算法 (2)
- ACM 哈希 (3)
- 二分法 (3)
- STL 空间配置器 (1)
- poj Trie树 (1)
- poj 调试经验 (2)
- POJ KMP (2)
- poj 暴力法 (1)
- C++调试 (3)
- More Effective C++读书笔记 (3)
- Effective C++读书笔记 (25)
- 机器学习 (33)
- 北邮人论坛 (2)
- 杂项 (6)
- 数学 (2)
- opencv (3)
- 操作系统基本概念 (10)
- Linux基础学习 (10)
- 程序优化 (3)
- 深入理解计算机系统 (5)

```

09. lr = 0.001
10. training_iters = 100000
11. batch_size = 128
12.
13. n_inputs = 28 # MNIST data input (img shape: 28*28)
14. n_steps = 28 # time steps
15. n_hidden_units = 128 # neurons in hidden layer
16. n_classes = 10 # MNIST classes (0-9 digits)
17.
18. # tf Graph input
19. x = tf.placeholder(tf.float32, [None, n_steps, n_inputs])
20. y = tf.placeholder(tf.float32, [None, n_classes])
21.
22. # Define weights
23. weights = {
24.     # (28, 128)
25.     'in': tf.Variable(tf.random_normal([n_inputs, n_hidden_units])),
26.     # (128, 10)
27.     'out': tf.Variable(tf.random_normal([n_hidden_units, n_classes]))
28. }
29. biases = {
30.     # (128, )
31.     'in': tf.Variable(tf.constant(0.1, shape=[n_hidden_units, ])),
32.     # (10, )
33.     'out': tf.Variable(tf.constant(0.1, shape=[n_classes, ]))
34. }
35.
36.
37. def RNN(X, weights, biases):
38.     # hidden layer for input to cell
39.     #####
40.     #X(128 batch,28 steps,28 inputs)
41.     #==>(128*28,28 inputs)
42.     X = tf.reshape(X, [-1, n_inputs])
43.     #==>(128 batch*28 steps,128 hidden)
44.     X_in = tf.matmul(X, weights['in'])+biases['in']
45.     #==>(128 batch,28 steps,128 hidden)
46.     X_in = tf.reshape(X_in, [-1, n_steps, n_hidden_units])
47.     # cell

```

关闭



在国家 中国内地(China)

手机号码

校验码

用户名

置密码

认密码

验证码短信接口

哈工大SCIR 神经网络和深度学习转载 (11)

tensorflow调研 (37)

文章存档

2017年03月 (5)

2017年02月 (9)

2017年01月 (2)

2016年12月 (8)

2016年11月 (31)

展开

阅读排行

安装scikit-learn , win7 6 (8805)

SVM的两个参数 C 和 ga (6803)

Python 列表的清空 (3361)

import sys sys.path.append (3062)

error LNK2019: 无法解析 (3018)

连续特征离散化达到更好 (2567)

配置caffe的python接口及 (2194)

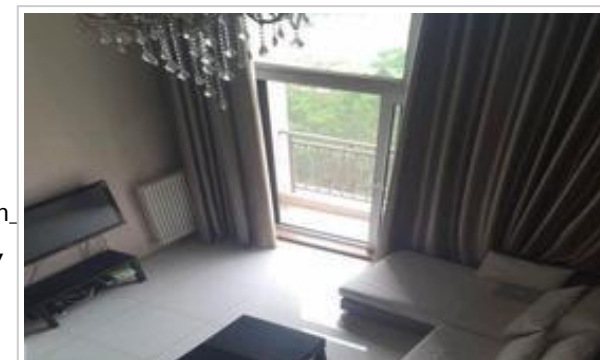
tensorflow 使用LSTM进行分类的例程 - lujiandong1的专栏 - 博客频道 - CSDN.NET

```

48. #####
49. #same to define active function
50. lstm_cell = tf.nn.rnn_cell.BasicLSTMCell(n_hidden_units,forget_bias=1.0,state_is_tuple)
51. #lstm cell is divided into two parts(c_state,m_state)
52. _init_state = lstm_cell.zero_state(batch_size,dtype=tf.float32)
53.
54. #choose rnn how to work,lstm just is one kind of rnn,use lstm_cell for active function
55. outputs,states = tf.nn.dynamic_rnn(lstm_cell,X_in,initial_state=_init_state,time_major=False)
56.
57. # hidden layer for output as the final results
58. #####
59. results = tf.matmul(states[1],weights['out']) + biases['out']
60.
61. #unpack to list [(batch,outputs)]*steps
62. #outputs = tf.unpack(tf.transpose(outputs,[1,0,2])) # state is the last outputs
63. #results = tf.matmul(outputs[-1],weights['out']) + biases['out']
64. return results
65.
66.
67. pred = RNN(x, weights, biases)
68. cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(pred, y))
69. train_op = tf.train.AdamOptimizer(lr).minimize(cost)
70.
71. correct_pred = tf.equal(tf.argmax(pred, 1), tf.argmax(y, 1))
72. accuracy = tf.reduce_mean(tf.cast(correct_pred, tf.float32))
73.
74. init = tf.initialize_all_variables()
75. with tf.Session() as sess:
76.     sess.run(init)
77.     step = 0
78.     while step * batch_size < training_iters:
79.         batch_xs, batch_ys = mnist.train.next_batch(batch_size)
80.         batch_xs = batch_xs.reshape([batch_size, n_steps, n_features])
81.         sess.run([train_op], feed_dict={
82.             x: batch_xs,
83.             y: batch_ys,
84.         })
85.         if step % 20 == 0:
86.             print(sess.run(accuracy, feed_dict={

```

关闭



一居租房



验证码短信接口

- error LNK2019: 无法解析 (2)
- 深入分析C++引用 (1)
- 人为什么会浮躁 (1)
- Python 列表的清空 (1)
- XGBoost Stopping to Av (1)
- Andrew Ng的 Machine L (1)
- 配置caffe的python接口及 (1)
- caffe 教程 Fine-tuning C: (1)

推荐文章

- * Android安全防护之旅---带你把Apk混淆成中文语言代码
- * TensorFlow文本摘要生成 - 基于注意力的序列到序列模型
- * 创建后台任务的两种代码模式
- * 一个屌丝程序员的人生 (六十)
- * WKWebView与js交互之完美解决方案
- * 年轻人,“砖砖瓦瓦”不应该成为你的梦想!

```

87.         x: batch_xs,
88.         y: batch_ys,
89.     ))
90.     step += 1

```

说明：

1、该LSTM模型的结构是28*128*10

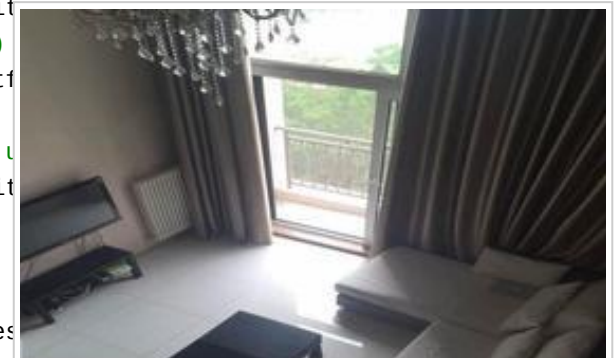
[python]

```

01. def RNN(X, weights, biases):
02.     # hidden layer for input to cell
03.     #####
04.     #X(128 batch,28 steps,28 inputs)
05.     #==>(128*28,28 inputs)
06.     X = tf.reshape(X,[-1,n_inputs])
07.     #==>(128 batch*28 steps,128 hidden)
08.     X_in = tf.matmul(X,weights['in'])+biases['in']
09.     #==>(128 batch,28 steps,128 hidden)
10.     X_in = tf.reshape(X_in,[-1,n_steps,n_hidden_units])
11.     # cell
12.     #####
13.     #same to define active function
14.     lstm_cell = tf.nn.rnn_cell.BasicLSTMCell(n_hidden_units)
15.     #lstm cell is divided into two parts(c_state,m_state)
16.     _init_state = lstm_cell.zero_state(batch_size, dtype=tf.float32)
17.
18.     #choose rnn how to work,lstm just is one kind of rnn,unrolled
19.     outputs,states = tf.nn.dynamic_rnn(lstm_cell,X_in,init_state=_init_state)
20.
21.     # hidden layer for output as the final results
22.     #####
23.     results = tf.matmul(states[1],weights['out']) + biases['out']
24.
25.     #unpack to list [(batch,outputs)]*steps
26.     #outputs = tf.unpack(tf.transpose(outputs,[1,0,2])) #
27.     #results = tf.matmul(outputs[-1],weights['out']) + biases['out']

```

关闭



一居租房





__init__() got an...

tensorflow中关于队列使用的实验
yuehanliushuang: very good

tensorflow 使用LSTM进行分类
qq_27590277: 为什么说我出错
TypeError: __init__() got an
unexpected ...

error LNK2019: 无法解析的外部:
m0_37640107: 多谢博主! 我的
opencv用vs生成的时候也一直报
这个错误, 上网其他方案都没
用, 直到看了你的才发现...

machine learning week6 诊断机
Starry5cm:
rand_seq=round(rand(1,i)*(m-
1))+1;%生成i个随机序列 0~m这
里改...

machine learning week6 诊断机
Starry5cm:
rand_seq=round(rand(1,i)*(m-
1))+1;%生成i个随机序列 0~m这
里改...

人为什么会浮躁

annipiao: 相当有见地的分析, 受
教了

tensorflow中dropout的用法,防止
Wxlong: 博主你好, 你在文中
说“train的时候才是dropout起作
用的时候,train和test的时候不
应...

28. `return results`

讲解如下:

1、

[python]

```
01. #X(128 batch,28 steps,28 inputs)
02. #==>(128*28,28 inputs)
03. X = tf.reshape(X, [-1,n_inputs])
04. #==>(128 batch*28 steps,128 hidden)
05. X_in = tf.matmul(X,weights['in'])+biases['in']
06. #==>(128 batch,28 steps,128 hidden)
07. X_in = tf.reshape(X_in, [-1,n_steps,n_hidden_units])
```

说明: 首先,对于输入我们先进行加权,因为在LSTM单元中,忘记门,输入门等各个部件都需要输入的加权和,直接就在这边做好

2、

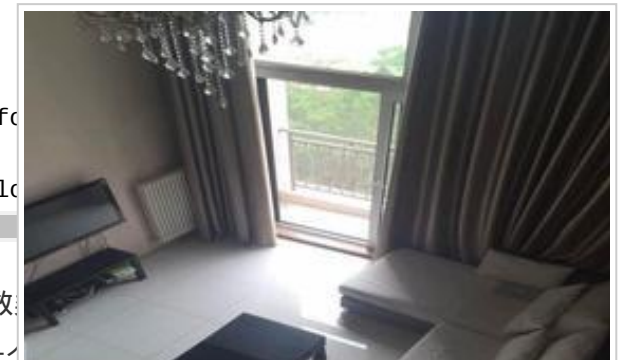
[python]

```
01. # cell
02. #####
03. #same to define active function
04. lstm_cell = tf.nn.rnn_cell.BasicLSTMCell(n_hidden_units,fc
05. #lstm cell is divided into two parts(c_state,m_state)
06. _init_state = lstm_cell.zero_state(batch_size,dtype=tf.float64)
```

说明: 在这边定义我们使用的LSTM单元,其实定义LSTM单元跟定义激活函数类似,我们定义了一个sigmoid激活函数,LSTM单元仅仅是一个细胞单元.然后LSTM细胞的输出是一个

3、

关闭



一居租房





[python]



```
01. #choose rnn how to work,lstm just is one kind of rnn,use lstm_cell for active function,set
02. outputs,states = tf.nn.dynamic_rnn(lstm_cell,X_in,initial_state=_init_state,time_major=False)
```

说明：定义使用某种LSTM单元的RNN网络

4、

[python]



```
01. # hidden layer for output as the final results
02. #####
03. results = tf.matmul(states[1],weights['out']) + biases['out']
```

说明：最终的输出使用的是最后的step,隐层的激活值再和输出层的权重进行加权

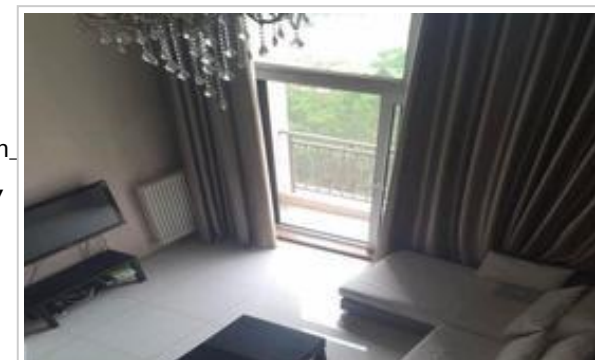
5、

[python]



```
01. init = tf.initialize_all_variables()
02. with tf.Session() as sess:
03.     sess.run(init)
04.     step = 0
05.     while step * batch_size < training_iters:
06.         batch_xs, batch_ys = mnist.train.next_batch(batch_size)
07.         batch_xs = batch_xs.reshape([batch_size, n_steps, n_features])
08.         sess.run([train_op], feed_dict={
09.             x: batch_xs,
10.             y: batch_ys,
11.         })
12.         if step % 20 == 0:
13.             print(sess.run(accuracy, feed_dict={
14.                 x: batch_xs,
15.                 y: batch_ys,
```

关闭



一居租房





```
16.         }))
17.         step += 1
```

说明：这是训练的过程

6、使用LSTM单元的RNN也有前向过程和反向传播过程,只是前向过程是在时序上进行前向传播。所以,使用LSTM单元的RNN必须要记录每个step,LSTM单元的激活值和状态。

顶 1 踩 0

上一篇 tensorflow中dropout的用法,防止overfitting

下一篇 tensorflow中使用LSTM去预测sinx函数

关闭

我的同类文章

tensorflow调研 (36)

- keras 指定程序在某块卡上训.. 2017-03-06 阅读 25
- keras Lambda自定义层实现... 2017-02-09 阅读 285
- keras卷积补零相关的border... 2017-02-08 阅读 436
- keras 对于大数据的训练,无... 2017-02-05 阅读 339

- keras 保存模型和加
- Tensorflow实现Mult
- keras查看网络结构,
- keras 实现CNN 进行



一居租房



- 使用keras实现简单的前向全... 2017-02-03 阅读 107
- CNN的超参数 & 宽卷积和窄... 2017-01-01 阅读 243
- tensorflow CNN for mnist 2016-12-18 阅读 140

在国家 +0086 中国内地(China)

机号码 157

校验码 27978 重新发送(T)

用户名 速达移动

置密码 *****

认密码 *****

验证码短信接口

中国无限制发行人民币

你的财富如何实现聚变？最后一次财富 分配机遇暗藏股市。中国3.0获利规则

猜你在找

- 第07章-网络通信协议 (OSI、TCP、UDP、IP、ARP、I
- Opencv自带的HOG进行行人检测-QT例程
- 第02章-计算机网络概述(定义+功能+分类)
- DSplay库 例程二简单捕获视频和音频并进行预览
- 计算机网络原理精讲
- Keras2使用Keras构建神经网络进行Mnist手写字体分类并
- Kubernetes网络基本原理及实践
- 使用Caffe对图片进行训练并分类的简单流程
- 2016年11月12日软考网络工程师上午真题解析视频课程
- tensorflow学习笔记之使用tensorflow进行MNIST分类3

新东方
XDF.CN
北京总部

美国夏令营
全真课堂体验 提升留学背景

马上报名

查看评论

1楼 qq_27590277 2017-03-05 21:37发表



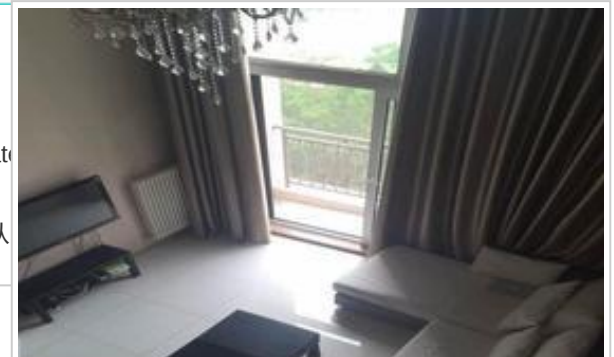
为什么说我出错 TypeError: __init__() got an unexpected keyword argument 'state'

意思是tf.nn.rnn_cell.BasicLSTMCell中没有state_is_tuple这个参数 怎么回事 很认

您还没有登录,请[登录](#)或[注册](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目



一居租房



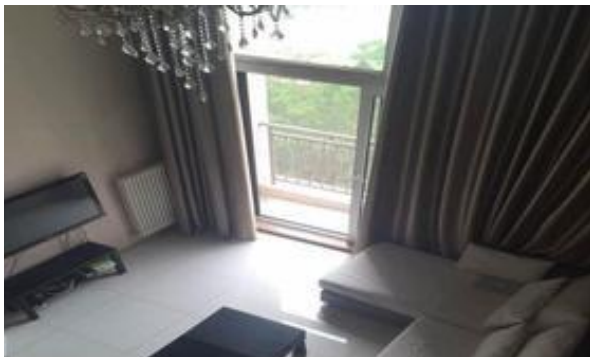


全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack FTC
coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 | 江苏乐知网络技术有限公司
京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved

关闭



一居租房

