

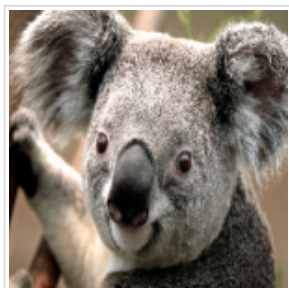
lujiandong1的专栏

目录视图

摘要视图

RSS 订阅

个人资料



BYR_jiandong



访问：132436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中使用LSTM去预测sinx函数

标签：tensorflow

2016-11-20 22:01

642人阅读

评论(1)

收藏

举报

分类：tensorflow调研 (36) ▼

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 numpy as np
03. import matplotlib.pyplot as plt
04.
05.
06. BATCH_START = 0
07. TIME_STEPS = 20
08. BATCH_SIZE = 50
```

关闭



文章搜索

文章分类

ACM DP问题 (7)

ACM其他文章 (2)

STL (5)

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. INPUT_SIZE = 1
10. OUTPUT_SIZE = 1
11. CELL_SIZE = 10
12. LR = 0.006
13.
14.
15. def get_batch():
16.     global BATCH_START, TIME_STEPS
17.     # xs shape (50batch, 20steps)
18.     xs = np.arange(BATCH_START, BATCH_START+TIME_STEPS*BATCH_SIZE).reshape((BATCH_SIZE, TIME_STEPS))
19.     seq = np.sin(xs)
20.     res = np.cos(xs)
21.     BATCH_START += TIME_STEPS
22.     # plt.plot(xs[0, :], res[0, :], 'r', xs[0, :], seq[0, :], 'b--')
23.     # plt.show()
24.     # returned seq, res and xs: shape (batch, step, input)
25.     return [seq[:, :, np.newaxis], res[:, :, np.newaxis], xs]
26.
27.
28. class LSTMRNN(object):
29.     def __init__(self, n_steps, input_size, output_size, cell_size, batch_size):
30.         self.n_steps = n_steps
31.         self.input_size = input_size
32.         self.output_size = output_size
33.         self.cell_size = cell_size
34.         self.batch_size = batch_size
35.         with tf.name_scope('inputs'):
36.             self.xs = tf.placeholder(tf.float32, [None, n_steps, self.input_size])
37.             self.ys = tf.placeholder(tf.float32, [None, n_steps, self.output_size])
38.             with tf.variable_scope('in_hidden'):
39.                 self.add_input_layer()
40.             with tf.variable_scope('LSTM_cell'):
41.                 self.add_cell()
42.             with tf.variable_scope('out_hidden'):
43.                 self.add_output_layer()
44.             with tf.name_scope('cost'):
45.                 self.compute_cost()
46.             with tf.name_scope('train'):
47.                 self.train_op = tf.train.AdamOptimizer(LR).minimize(self.cost)

```

关闭



[Python 基础 \(25\)](#)[SVM \(1\)](#)[推荐系统 \(1\)](#)[机器学习讲座笔记 \(1\)](#)[Kaggle学习笔记 \(10\)](#)[特征工程 \(2\)](#)[caffe教程及遇到问题的解决方案 \(2\)](#)[leetcode \(7\)](#)[自然语言处理 \(10\)](#)[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 64 \(8805\)](#)[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 \(2194\)](#)

```

48.
49.
50.
51.
52.
53.
54.
55.
56.
57.
58.
59.
60.
61.
62.
63.
64.
65.
66.
67.
68.
69.
70.
71.
72.
73.
74.
75.
76.
77.
78.
79.
80.
81.
82.
83.
84.
85.
86.

def add_input_layer(self):
    l_in_x = tf.reshape(self.xs, [-1, self.input_size], name='2_2D') # (batch*n_step,
    # Ws (in_size, cell_size)
    Ws_in = self._weight_variable([self.input_size, self.cell_size])
    # bs (cell_size, )
    bs_in = self._bias_variable([self.cell_size,])
    # l_in_y = (batch * n_steps, cell_size)
    with tf.name_scope('Wx_plus_b'):
        l_in_y = tf.matmul(l_in_x, Ws_in) + bs_in
    # reshape l_in_y ==> (batch, n_steps, cell_size)
    self.l_in_y = tf.reshape(l_in_y, [-1, self.n_steps, self.cell_size], name='2_3D')

def add_cell(self):
    lstm_cell = tf.nn.rnn_cell.BasicLSTMCell(self.cell_size, forget_bias=1.0, state_is_
    with tf.name_scope('initial_state'):
        self.cell_init_state = lstm_cell.zero_state(self.batch_size, dtype=tf.float32)
    self.cell_outputs, self.cell_final_state = tf.nn.dynamic_rnn(
        lstm_cell, self.l_in_y, initial_state=self.cell_init_state, time_major=False)

def add_output_layer(self):
    # shape = (batch * steps, cell_size)
    l_out_x = tf.reshape(self.cell_outputs, [-1, self.cell_size], name='2_2D')
    Ws_out = self._weight_variable([self.cell_size, self.output_size])
    bs_out = self._bias_variable([self.output_size,])
    # shape = (batch * steps, output_size)
    with tf.name_scope('Wx_plus_b'):
        self.pred = tf.matmul(l_out_x, Ws_out) + bs_out

def compute_cost(self):
    losses = tf.nn.seq2seq.sequence_loss_by_example(
        [tf.reshape(self.pred, [-1], name='reshape_pred')],
        [tf.reshape(self.ys, [-1], name='reshape_target')],
        [tf.ones([self.batch_size * self.n_steps], dtype=tf.float32)],
        average_across_timesteps=True,
        softmax_loss_function=self.ms_error,
        name='losses'
    )
    with tf.name_scope('average_cost'):

```

[关闭](#)

caffe 教程 Fine-tuning C (1938)

机器学习中的内核方法 (1873)

DBN的训练过程 (1790)

评论排行

连续特征离散化达到更好 (4)

machine learning week6 (3)

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!

tensorflow中使用LSTM去预测sinx函数 - lujiandong1的专栏 - 博客频道 - CSDN.NET

```

87.         self.cost = tf.div(
88.             tf.reduce_sum(losses, name='losses_sum'),
89.             self.batch_size,
90.             name='average_cost')
91.         tf.scalar_summary('cost', self.cost)
92.
93.     def ms_error(self, y_pre, y_target):
94.         return tf.square(tf.sub(y_pre, y_target))
95.
96.     def _weight_variable(self, shape, name='weights'):
97.         initializer = tf.random_normal_initializer(mean=0., stddev=1.,)
98.         return tf.get_variable(shape=shape, initializer=initializer, name=name)
99.
100.    def _bias_variable(self, shape, name='biases'):
101.        initializer = tf.constant_initializer(0.1)
102.        return tf.get_variable(name=name, shape=shape, initializer=initializer)
103.
104.
105. if __name__ == '__main__':
106.     model = LSTMRNN(TIME_STEPS, INPUT_SIZE, OUTPUT_SIZE, CELL_SIZE, BATCH_SIZE)
107.     sess = tf.Session()
108.     merged = tf.merge_all_summaries()
109.     writer = tf.train.SummaryWriter("logs", sess.graph)
110.     sess.run(tf.initialize_all_variables())
111.     # relocate to the local dir and run this line to view it --> http://0.0.0.0:6006/
112.     # $ tensorboard --logdir='logs'
113.
114.     plt.ion()
115.     plt.show()
116.     for i in range(200):
117.         seq, res, xs = get_batch()
118.         if i == 0:
119.             feed_dict = {
120.                 model.xs: seq,
121.                 model.ys: res,
122.                 # create initial state
123.             }
124.         else:
125.             feed_dict = {

```

关闭



最新评论

Andrew Ng的 Machine Learning
fupf1303: 写的挺好的, 可惜只有2和4, 有其他的课程笔记吗?

tensorflow MNIST数据集上简单
倾城一少: 博主, MLP网络的全称是什么?

tensorflow CNN for mnist
xjbada: 我运行这个代码为什么会出现这个错误呢*** TypeError: __init__() got an...

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

tesnsorflow 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



短信验证码接口

```

126.         model.xs: seq,
127.         model.ys: res,
128.         model.cell_init_state: state    # use last state as the initial state for
129.     }
130.
131.     _, cost, state, pred = sess.run(
132.         [model.train_op, model.cost, model.cell_final_state, model.pred],
133.         feed_dict=feed_dict)
134.
135.         # plotting
136.         # plt.plot(xs[0, :], res[0].flatten(), 'r', xs[0, :], pred.flatten()
137.         [:TIME_STEPS], 'b--')
138.         # plt.ylim((-1.2, 1.2))
139.         # plt.draw()
140.         # plt.pause(0.3)
141.
142.         if i % 20 == 0:
143.             print('cost: ', round(cost, 4))
144.             result = sess.run(merged, feed_dict)
145.             writer.add_summary(result, i)

```

说明：使用LSTM模型来预测sin(x)函数,下面分别对各个函数进行说明

1、数据输入函数：get_batch()

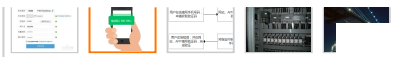
```

[python]
def get_batch():
    global BATCH_START, TIME_STEPS
    # xs shape (50batch, 20steps)
    xs = np.arange(BATCH_START, BATCH_START+TIME_STEPS*BATCH_SIZE).reshape((BATCH_SIZE, TIME_STEPS))
    seq = np.sin(xs)
    res = np.cos(xs)
    BATCH_START += TIME_STEPS
    # plt.plot(xs[0, :], res[0, :], 'r', xs[0, :], seq[0, :], 'b--')
    # plt.show()
    # returned seq, res and xs: shape (batch, step, input)
    return [seq[:, :, np.newaxis], res[:, :, np.newaxis], xs]

```

关闭





```
[python]
01. feed_dict = {
02.     model.xs: seq,
03.     model.ys: res,
04.     model.cell_init_state: state    # use last state as the initial state for this run
05. }
```

说明：作者最后的训练数据是(seq,res),其中seq = sin(x),res = cos(x),也即该LSTM模型所学习到的是sin(x)到cos(x)的映射关系,最后给定一个输入sin(x0),LSTM能够预测出相对应的cos(x0).而不应该理解成输入sin(x0),sin(x1),sin(x2),...,sin(x(n-1)),然后去预测sin(x(n))

2、说明：该三层网络的结构如下:1---10-----1,然后time_steps=20.对于LSTM,RNN这种模型,学习的是序列。get_batch()的做法就是,按照序列的顺序,每次get_batch()就切出BATCH_SIZE*TIME_STEPS*INPUT_SIZE作为下一次训练的输入数据。

3、接下来就是定义LSTM网络的网络结构:

```
[python]
01. def add_input_layer(self,):
02.     l_in_x = tf.reshape(self.xs, [-1, self.input_size], name='l_in_x')
03.     # ws (in_size, cell_size)
04.     ws_in = self._weight_variable([self.input_size, self.cell_size], name='ws_in')
05.     # bs (cell_size, )
06.     bs_in = self._bias_variable([self.cell_size,], name='bs_in')
07.     # l_in_y = (batch * n_steps, cell_size)
08.     with tf.name_scope('wx_plus_b'):
09.         l_in_y = tf.matmul(l_in_x, ws_in) + bs_in
10.     # reshape l_in_y ==> (batch, n_steps, cell_size)
11.     self.l_in_y = tf.reshape(l_in_y, [-1, self.n_steps, self.cell_size])
```

说明：对于输入层,本来x的结构是(batch,n_step,input_size),先将输入reshape成(batch,n_steps,input_size),再进行矩阵运算,最后再reshape回(batch,n_steps,cell_size)

关闭





4、定义使用LSTM单元的RNN网络

```
[python]
01. def add_cell(self):
02.     lstm_cell = tf.nn.rnn_cell.BasicLSTMCell(self.cell_size, forget_bias=1.0, state_is_tuple=True)
03.     with tf.name_scope('initial_state'):
04.         self.cell_init_state = lstm_cell.zero_state(self.batch_size, dtype=tf.float32)
05.         self.cell_outputs, self.cell_final_state = tf.nn.dynamic_rnn(
06.             lstm_cell, self.l_in_y, initial_state=self.cell_init_state, time_major=False)
```

5、输出层

```
[python]
01. def add_output_layer(self):
02.     # shape = (batch * steps, cell_size)
03.     l_out_x = tf.reshape(self.cell_outputs, [-1, self.cell_size], name='2_2D')
04.     Ws_out = self._weight_variable([self.cell_size, self.output_size])
05.     bs_out = self._bias_variable([self.output_size, ])
06.     # shape = (batch * steps, output_size)
07.     with tf.name_scope('Wx_plus_b'):
08.         self.pred = tf.matmul(l_out_x, Ws_out) + bs_out
```

[关闭](#)




顶 踩
0 0

上一篇 [tesnsorflow 使用LSTM进行分类的例程](#)

下一篇 [语言模型的评估-困惑度](#)

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 保存模型和加
- keras Lambda自定义层实现... 2017-02-09 阅读 285
- Tensorflow实现Mult
- keras卷积补零相关的border... 2017-02-08 阅读 436
- keras查看网络结构,
- keras 对于大数据的训练,无... 2017-02-05 阅读 339
- keras 实现CNN 进行
- 使用keras实现简单的前向全... 2017-02-03 阅读 107
- CNN的超参数 & 宽
- tensorflow CNN for mnist 2016-12-18 阅读 140

更多文章

关闭





云服务器性价比之王

广告

猜你在找

2016软考网络工程师实验视频课程-第二章 数据通信基 Tensorflow一些常用基本概念与函数2
数据挖掘模型篇之R语言实践 tensorflow中的不懂得知识点转置函数 transpose
第07章-网络通信协议 (OSI、TCP、UDP、IP、ARP、I tensorflow某些函数的小坑持续更新
计算机网络原理精讲 TensorFlow-tfnnconv2d 函数
ArcGIS之数字高程模型 (DEM) 分析上篇视频课程(GI tensorflow笔记 常用函数说明

中高考一对一辅导

广告

数学

语文

英语

物理

化学

政治

查看详情

查看评论

1楼 joanconquerlee 2016-11-22 21:11发表



运行报错：

TypeError: __init__() got an unexpected keyword argument 'state_is_tuple'

求解，谢谢

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

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 管
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubu
BI HTML5 Spring Apache .NET API HTML SDK IIS Fe

关闭



云计算新用户

注册送520元代金券

立即领取



短信验证码接口



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



12

关闭

