

一居租房



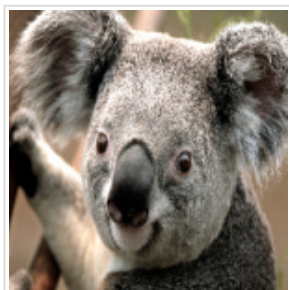
栏

目录视图

摘要视图

RSS 订阅

个人资料



BYR_jiandong



访问：132432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将训练好的模型freeze,即将权重固化到图里面,并使用该模型进行预测

标签：tensorflow

2016-11-28 21:54

784人阅读

评论(0)

4X 收藏

分享

本文章已收录于：



人工智能开发框架知识库

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。

ML主要分为训练和预测两个阶段,此教程就是将训练好的模型freeze并保存下来,和该模型的权重固化到一起了.也即加载freeze的模型之后,立刻能够使用了。

下面使用一个简单的demo来详细解释该过程,



关闭



一居租房



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)

一、首先运行脚本tiny_model.py

```
[python] C 8

01.  -*- coding:utf-8 -*-
02.  import tensorflow as tf
03.  import numpy as np
04.
05.
06.  with tf.variable_scope('Placeholder'):
07.      inputs_placeholder = tf.placeholder(tf.float32, name='inputs_placeholder', shape=
[None, 10])
08.      labels_placeholder = tf.placeholder(tf.float32, name='labels_placeholder', shape=
[None, 1])
09.
10.  with tf.variable_scope('NN'):
11.      W1 = tf.get_variable('W1', shape=
[10, 1], initializer=tf.random_normal_initializer(stddev=1e-1))
12.      b1 = tf.get_variable('b1', shape=[1], initializer=tf.constant_initializer(0.1))
13.      W2 = tf.get_variable('W2', shape=
[10, 1], initializer=tf.random_normal_initializer(stddev=1e-1))
14.      b2 = tf.get_variable('b2', shape=[1], initializer=tf.constant_initializer(0.1
15.
16.      a = tf.nn.relu(tf.matmul(inputs_placeholder, W1) + b1)
17.      a2 = tf.nn.relu(tf.matmul(inputs_placeholder, W2) + b2
18.
19.      y = tf.div(tf.add(a, a2), 2)
20.
21.  with tf.variable_scope('Loss'):
22.      loss = tf.reduce_sum(tf.square(y - labels_placeholder)
23.
24.  with tf.variable_scope('Accuracy'):
25.      predictions = tf.greater(y, 0.5, name="predictions")
26.      correct_predictions = tf.equal(predictions, tf.cast(labels_placeholder, tf.int32))
27.      accuracy = tf.reduce_mean(tf.cast(correct_predictions, tf.float32))
28.
29.
30.  adam = tf.train.AdamOptimizer(learning_rate=1e-3)
31.  train_op = adam.minimize(loss)
```

关闭





一居租房



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.appe (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)

```

32.
33. # generate_data
34. inputs = np.random.choice(10, size=[10000, 10])
35. labels = (np.sum(inputs, axis=1) > 45).reshape(-1, 1).astype(np.float32)
36. print('inputs.shape:', inputs.shape)
37. print('labels.shape:', labels.shape)
38.
39.
40. test_inputs = np.random.choice(10, size=[100, 10])
41. test_labels = (np.sum(test_inputs, axis=1) > 45).reshape(-1, 1).astype(np.float32)
42. print('test_inputs.shape:', test_inputs.shape)
43. print('test_labels.shape:', test_labels.shape)
44.
45. batch_size = 32
46. epochs = 10
47.
48. batches = []
49. print("%d items in batch of %d gives us %d full batches and %d batches of %d items" % (
50.     len(inputs),
51.     batch_size,
52.     len(inputs) // batch_size,
53.     batch_size - len(inputs) // batch_size,
54.     len(inputs) - (len(inputs) // batch_size) * 32)
55. )
56. for i in range(len(inputs) // batch_size):
57.     batch = [ inputs[batch_size*i:batch_size*i+batch_size]
58.             batches.append(list(batch))
59. if (i + 1) * batch_size < len(inputs):
60.     batch = [ inputs[batch_size*(i + 1):], labels[batch_size*(i + 1):]
61.     batches.append(list(batch))
62. print("Number of batches: %d" % len(batches))
63. print("Size of full batch: %d" % len(batches[0]))
64. print("Size if final batch: %d" % len(batches[-1]))
65.
66. global_count = 0
67.
68. with tf.Session() as sess:
69.     #sv = tf.train.Supervisor()
70.     #with sv.managed_session() as sess:

```

关闭





一居租房



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!

```

71.     sess.run(tf.initialize_all_variables())
72.     for i in range(epochs):
73.         for batch in batches:
74.             # print(batch[0].shape, batch[1].shape)
75.             train_loss, _ = sess.run([loss, train_op], feed_dict={
76.                 inputs_placeholder: batch[0],
77.                 labels_placeholder: batch[1]
78.             })
79.             # print('train_loss: %d' % train_loss)
80.
81.             if global_count % 100 == 0:
82.                 acc = sess.run(accuracy, feed_dict={
83.                     inputs_placeholder: test_inputs,
84.                     labels_placeholder: test_labels
85.                 })
86.                 print('accuracy: %f' % acc)
87.                 global_count += 1
88.
89.         acc = sess.run(accuracy, feed_dict={
90.             inputs_placeholder: test_inputs,
91.             labels_placeholder: test_labels
92.         })
93.         print("final accuracy: %f" % acc)
94.         #在session当中就要将模型进行保存
95.         saver = tf.train.Saver()
96.         last_chkp = saver.save(sess, 'results/graph.chkp')
97.         #sv.saver.save(sess, 'results/graph.chkp')
98.
99.     for op in tf.get_default_graph().get_operations():
100.        print(op.name)

```

说明：saver.save必须在session里面,因为在session里面,整个图才是激活的,才能够得到如下的文件：

```

-rw-rw-r-- 1 jdlu jdlu 77 11月 28 19:46 checkpoint
-rw-rw-r-- 1 jdlu jdlu 272 11月 28 19:46 graph.chkp.data-00000-of-0
-rw-rw-r-- 1 jdlu jdlu 462 11月 28 19:46 graph.chkp.index
-rw-rw-r-- 1 jdlu jdlu 41K 11月 28 19:46 graph.chkp.meta

```

关闭





一居租房



__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...

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...

说明:

.data:存放的是权重参数

.meta:存放的是图和metadata,metadata是其他配置的数据

如果想将我们的模型固化, 让别人能够使用, 我们仅仅需要的是图和参数, metadata是不需要的

二、综合上述几个文件,生成可以使用的模型的步骤如下:

- 1、恢复我们保存的图
- 2、开启一个Session, 然后载入该图要求的权重
- 3、删除对预测无关的metadata
- 4、将处理好的模型序列化之后保存

运行freeze.py

[python]

```
01.  -*- coding:utf-8 -*-
02. import os, argparse
03. import tensorflow as tf
04. from tensorflow.python.framework import graph_util
05.
06. dir = os.path.dirname(os.path.realpath(__file__))
07.
08. def freeze_graph(model_folder):
09.     # We retrieve our checkpoint fullpath
10.     checkpoint = tf.train.get_checkpoint_state(model_folder)
11.     input_checkpoint = checkpoint.model_checkpoint_path
12.
13.     # We precise the file fullname of our freezed graph
14.     absolute_model_folder = "/" + input_checkpoint.split("/")[-2]
15.     output_graph = absolute_model_folder + "/frozen_model.pb"
16.
17.     # Before exporting our graph, we need to precise what
18.     # this variables is plural, because you can have multiple
19.     # freeze之前必须明确哪个是输出结点,也就是我们要得到推论结果的结点
20.     # 输出结点可以看我们模型的定义
21.     # 只有定义了输出结点, freeze才会把得到输出结点所必要的结点都保存下
22.     # 所以, output_node_names必须根据不同的网络进行修改
```

关闭





一居租房



```

23.     output_node_names = "Accuracy/predictions"
24.
25.     # We clear the devices, to allow TensorFlow to control on the loading where it wants (
26.     clear_devices = True
27.
28.     # We import the meta graph and retrieve a Saver
29.     saver = tf.train.import_meta_graph(input_checkpoint + '.meta', clear_devices=clear_dev
30.
31.     # We retrieve the protobuf graph definition
32.     graph = tf.get_default_graph()
33.     input_graph_def = graph.as_graph_def()
34.
35.     #We start a session and restore the graph weights
36.     #这边已经将训练好的参数加载进来,也即最后保存的模型是有图,并且图里面已经有参数了,所以才叫做是
frozen
37.     #相当于将参数已经固化在了图当中
38.     with tf.Session() as sess:
39.         saver.restore(sess, input_checkpoint)
40.
41.         # We use a built-in TF helper to export variables to constant
42.         output_graph_def = graph_util.convert_variables_to_constants(
43.             sess,
44.             input_graph_def,
45.             output_node_names.split(",") # We split on comma for convenience
46.         )
47.
48.         # Finally we serialize and dump the output graph t
49.         with tf.gfile.GFile(output_graph, "wb") as f:
50.             f.write(output_graph_def.SerializeToString())
51.         print("%d ops in the final graph." % len(output_gr
52.
53.
54. if __name__ == '__main__':
55.     parser = argparse.ArgumentParser()
56.     parser.add_argument("--model_folder", type=str, help="
57.     args = parser.parse_args()
58.
59.     freeze_graph(args.model_folder)

```

关闭





一居租房



说明：对于freeze操作,我们需要定义输出结点的名字.因为网络其实是比较复杂的,定义了输出结点的名字,那么freeze的时候就只把输出该结点所需要的子图都固化下来,其他无关的就舍弃掉.因为我们freeze模型的目的是接下来做预测.所以,一般情况下,output_node_names就是我们预测的目标.

三、加载freeze后的模型,注意该模型已经是包含图和相应的参数了.所以,我们不需要再加载参数进来.也即该模型加载进来已经是可以使用了.

```
[python]

01.  -*- coding:utf-8 -*-
02.  import argparse
03.  import tensorflow as tf
04.
05.  def load_graph(frozen_graph_filename):
06.      # We parse the graph_def file
07.      with tf.gfile.GFile(frozen_graph_filename, "rb") as f:
08.          graph_def = tf.GraphDef()
09.          graph_def.ParseFromString(f.read())
10.
11.      # We load the graph_def in the default graph
12.      with tf.Graph().as_default() as graph:
13.          tf.import_graph_def(
14.              graph_def,
15.              input_map=None,
16.              return_elements=None,
17.              name="prefix",
18.              op_dict=None,
19.              producer_op_list=None
20.          )
21.      return graph
22.
23.  if __name__ == '__main__':
24.      parser = argparse.ArgumentParser()
25.      parser.add_argument("--
26.      frozen_model_filename", default="results/frozen_model.pb",
27.      args = parser.parse_args()
```

关闭





一居租房



```

27.     #加载已经将参数固化后的图
28.     graph = load_graph(args.frozen_model_filename)
29.
30.     # We can list operations
31.     #op.values() gives you a list of tensors it produces
32.     #op.name gives you the name
33.     #输入,输出结点也是operation,所以,我们可以得到operation的名字
34.     for op in graph.get_operations():
35.         print(op.name, op.values())
36.         # prefix/Placeholder/inputs_placeholder
37.         # ...
38.         # prefix/Accuracy/predictions
39.     #操作有:prefix/Placeholder/inputs_placeholder
40.     #操作有:prefix/Accuracy/predictions
41.     #为了预测,我们需要找到我们需要feed的tensor,那么就需要该tensor的名字
42.     #注意prefix/Placeholder/inputs_placeholder仅仅是操作的名字
    字,prefix/Placeholder/inputs_placeholder:0才是tensor的名字
43.     x = graph.get_tensor_by_name('prefix/Placeholder/inputs_placeholder:0')
44.     y = graph.get_tensor_by_name('prefix/Accuracy/predictions:0')
45.
46.     with tf.Session(graph=graph) as sess:
47.         y_out = sess.run(y, feed_dict={
48.             x: [[3, 5, 7, 4, 5, 1, 1, 1, 1, 1]] # < 45
49.         })
50.         print(y_out) # [[ 0.]] Yay!
51.         print ("finish")

```

说明：

- 1、在预测的过程中,当把freeze后的模型加载进来后,我们只需要定义好输入的
- 2、在这里要注意一下tensor_name和ops_name,

注意prefix/Placeholder/inputs_placeholder仅仅是操作的名字,prefix/Placeholder/inputs_placeholder:0才是tensor的名字

x = graph.get_tensor_by_name('prefix/Placeholder/inputs_placeholder:0')—

关闭





一居租房



3、要获取图中ops的名字和对应的tensor的名字,可用如下的代码:

```
[python]
01. # We can list operations
02. #op.values() gives you a list of tensors it produces
03. #op.name gives you the name
04. #输入,输出结点也是operation,所以,我们可以得到operation的名字
05. for op in graph.get_operations():
06.     print(op.name, op.values())
```

上面是使用了Saver()来保存模型,也可以使用sv = tf.train.Supervisor()来保存模型

```
[python]
01. #-*- coding:utf-8 -*-
02. import tensorflow as tf
03. import numpy as np
04.
05.
06. with tf.variable_scope('Placeholder'):
07.     inputs_placeholder = tf.placeholder(tf.float32, name=
[None, 10])
08.     labels_placeholder = tf.placeholder(tf.float32, name=
[None, 1])
09.
10. with tf.variable_scope('NN'):
11.     W1 = tf.get_variable('W1', shape=
[10, 1], initializer=tf.random_normal_initializer(stddev=1)
12.     b1 = tf.get_variable('b1', shape=[1], initializer=tf.c
13.     W2 = tf.get_variable('W2', shape=
[10, 1], initializer=tf.random_normal_initializer(stddev=1)
14.     b2 = tf.get_variable('b2', shape=[1], initializer=tf.c
15.
16.     a = tf.nn.relu(tf.matmul(inputs_placeholder, W1) + b1)
```

关闭





一居租房



```

17.         a2 = tf.nn.relu(tf.matmul(inputs_placeholder, W2) + b2)
18.
19.         y = tf.div(tf.add(a, a2), 2)
20.
21.     with tf.variable_scope('Loss'):
22.         loss = tf.reduce_sum(tf.square(y - labels_placeholder) / 2)
23.
24.     with tf.variable_scope('Accuracy'):
25.         predictions = tf.greater(y, 0.5, name="predictions")
26.         correct_predictions = tf.equal(predictions, tf.cast(labels_placeholder, tf.bool), name=
27.             accuracy = tf.reduce_mean(tf.cast(correct_predictions, tf.float32))
28.
29.
30.     adam = tf.train.AdamOptimizer(learning_rate=1e-3)
31.     train_op = adam.minimize(loss)
32.
33.     # generate_data
34.     inputs = np.random.choice(10, size=[10000, 10])
35.     labels = (np.sum(inputs, axis=1) > 45).reshape(-1, 1).astype(np.float32)
36.     print('inputs.shape:', inputs.shape)
37.     print('labels.shape:', labels.shape)
38.
39.
40.     test_inputs = np.random.choice(10, size=[100, 10])
41.     test_labels = (np.sum(test_inputs, axis=1) > 45).reshape(-1, 1).astype(np.float32)
42.     print('test_inputs.shape:', test_inputs.shape)
43.     print('test_labels.shape:', test_labels.shape)
44.
45.     batch_size = 32
46.     epochs = 10
47.
48.     batches = []
49.     print("%d items in batch of %d gives us %d full batches and" %
50.         len(inputs),
51.         batch_size,
52.         len(inputs) // batch_size,
53.         batch_size - len(inputs) // batch_size,
54.         len(inputs) - (len(inputs) // batch_size) * 32)
55. )

```

关闭





一居租房



```

56. for i in range(len(inputs) // batch_size):
57.     batch = [ inputs[batch_size*i:batch_size*i+batch_size], labels[batch_size*i:batch_size*i+batch_size] ]
58.     batches.append(list(batch))
59. if (i + 1) * batch_size < len(inputs):
60.     batch = [ inputs[batch_size*(i + 1):], labels[batch_size*(i + 1):] ]
61.     batches.append(list(batch))
62. print("Number of batches: %d" % len(batches))
63. print("Size of full batch: %d" % len(batches[0]))
64. print("Size if final batch: %d" % len(batches[-1]))
65.
66. global_count = 0
67.
68. #with tf.Session() as sess:
69. sv = tf.train.Supervisor()
70. with sv.managed_session() as sess:
71.     #sess.run(tf.initialize_all_variables())
72.     for i in range(epochs):
73.         for batch in batches:
74.             # print(batch[0].shape, batch[1].shape)
75.             train_loss , _ = sess.run([loss, train_op], feed_dict={
76.                 inputs_placeholder: batch[0],
77.                 labels_placeholder: batch[1]
78.             })
79.             # print('train_loss: %d' % train_loss)
80.
81.             if global_count % 100 == 0:
82.                 acc = sess.run(accuracy, feed_dict={
83.                     inputs_placeholder: test_inputs,
84.                     labels_placeholder: test_labels
85.                 })
86.                 print('accuracy: %f' % acc)
87.                 global_count += 1
88.
89.         acc = sess.run(accuracy, feed_dict={
90.             inputs_placeholder: test_inputs,
91.             labels_placeholder: test_labels
92.         })
93.         print("final accuracy: %f" % acc)
94.         #在session当中就要将模型进行保存

```

关闭





一居租房



```
95.         #saver = tf.train.Saver()
96.         #last_chkp = saver.save(sess, 'results/graph.chkp')
97.         sv.saver.save(sess, 'results/graph.chkp')
98.
99.     for op in tf.get_default_graph().get_operations():
100.         print(op.name)
```

注意：使用了sv = tf.train.Supervisor(),就不需要再初始化了,将sess.run(tf.initialize_all_variables())注释掉,否则会报错.

顶 踩
0 0

关闭

上一篇 tensorflow的设计理念

下一篇 错误Tensor is not an element of this graph tensorflow

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





一居租房



- 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
- 使用keras实现简单的前向全... 2017-02-03 阅读 107
- tensorflow CNN for mnist 2016-12-18 阅读 140
- Tensorflow实现MultiInput & ... 2017-02-08 阅读 58
- keras查看网络结构,复现别人.. 2017-02-07 阅读 47
- keras 实现CNN 进行手写字... 2017-02-03 阅读 149
- CNN的超参数 & 宽卷积和窄... 2017-01-01 阅读 243

五篇文章

广告

马云刘强东雷军，下一个是你 - 在中国赚钱，您必须知道一件事

中国第三次造富运动高潮迭起，3月底发生的一件大事，更将成为造富运动的关键转折



参考知识库



Python知识库

22340 关注 | 1443 收录

猜你在找

shell脚本精简教程

2016高薪Linux培训教程-shell脚本入门及正则表达式

开发LinuxShell脚本程序【一】

C# For Unity系列之基础篇

开发LinuxShell脚本程序（三）

信息检索之文档评分词项权重

建立在线社交网络中的关系权重

Python调用已训练好的caffe模型

利用caffe训练好的模型进行测试

caffe初探4对训练得到的模型进

关闭





一居租房



查看评论

暂无评论

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 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
coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Sc
Angular Cloud Foundry Redis Scala Django Bootstrap

关闭

[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

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved

