

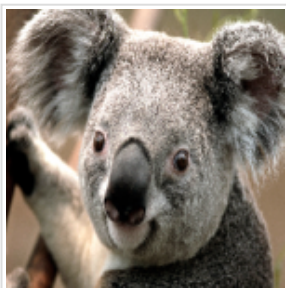
lujiandong1的专栏

目录视图

摘要视图

RSS 订阅

个人资料



BYR_jiandong



访问：333848次

积分：5551

等级：BLOG > 6

排名：第4998名

原创：227篇 转载：42篇

译文：0篇 评论：58条

文章搜索

异步赠书：9月重磅新书升级，本本经典 程序员9月书讯 每周荐书：ES6、虚拟现实、物联网（评论送书）

tensorflow将训练好的模型freeze,即将权重固化到图里面,并使用该模型预测

标签：tensorflow

2016-11-28 21:54

5833人阅读

评论(1)

分类：tensorflow调研 (39)

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.freeze的含义就是将该模型的图结构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,

一、首先运行脚本tiny_model.py

```
[python]
01.  #-*- coding:utf-8 -*-
02.  import tensorflow as tf
03.  import numpy as np
```

关闭

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](#) (21)

[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)

[Python 基础](#) (25)

```

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.
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) / 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.

```

关闭

[SVM \(1\)](#)[推荐系统 \(2\)](#)[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 \(8\)](#)[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 \(40\)](#)

文章存档

[2017年08月 \(2\)](#)[2017年06月 \(2\)](#)[2017年05月 \(1\)](#)[2017年04月 \(1\)](#)[2017年03月 \(5\)](#)[展开](#)

阅读排行

[安装scikit-learn , win7 64位 \(15235\)](#)[SVM的两个参数 C 和 gamma \(11626\)](#)[pandas使用get_dummies \(9479\)](#)[import sys sys.path.append \(8087\)](#)[Python 列表的清空 \(7961\)](#)[tensorflow conv2d的padding \(7387\)](#)[pandas contact 之后, 一 \(6689\)](#)

```

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], 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 of 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]

```

[关闭](#)

tensorflow中dropout的用 (6070)

python脚本的两种调用方 (6041)

tensorflow载入数据的三 (5996)

评论排行

tensorflow读取数据到队 (10)

tesnsorflow 使用LSTM进 (5)

machine learning week6 (4)

连续特征离散化达到更好 (4)

error LNK2019: 无法解析 (3)

tensorflow中关于队列使 (3)

tensorflow载入数据的三 (3)

xgboost cross_validation (2)

tensorflow中dropout的用 (2)

XGBoost Stopping to Av (2)

推荐文章

* CSDN新版博客feed流内测用户征集令

* Android检查更新下载安装

* 动手打造史上最简单的Recycleview 侧滑菜单

* TCP网络通讯如何解决分包粘包问题

* SDCC 2017之大数据技术实战线上峰会

* 快速集成一个视频直播功能

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 - lujiandong1的专栏 - CSDN博客

```

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,才能得到这个执行结果,使用saver.save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-00001
-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

```

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：

关闭

最新评论

操作系统的信号量 进程互斥 同步
ASC_haha: 太棒了!

tensorflow中关于队列使用的实验
ouweiqi: 差评, lz的代码自己都没有测试嘛q.enqueue_many((,))明显的错误

python脚本的两种调用方式
chen_csdn_long: Python有什么速成的方法吗。

Python程序的执行过程 解释型语
liurong07:

tesnorflow实现N个epoch训练数
mylibrary1: 貌似这么做epoch的初衷并没有达到: 在一个epoch遍历所有的样本, 不同epoch之间重新shuf...

FM算法能够学习到原始特征的er
sooner高: 哥们你好像说错了, 建议看看博客
<http://blog.csdn.net/google1989010>

tensorflow载入数据的三种方式
翻水水的工作日志: 不要误人子弟好吗?? 好多错误

keras 对于大数据的训练,无法一
chasingdream119: 您好, 请问最大进程数workers应该怎么设置呢, 我在本地设置超过1就报错: ValueError: ...

tensorflow载入数据的三种方式
qq_22541367: 非常非常好的文章, 谢谢博主, 受教了!!!

tensorflow conv2d的padding解释
csdn0006: 感谢

- 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 = "/".join(input_checkpoint.split('/')[:-1])
15.     output_graph = absolute_model_folder + "/frozen_model.pb"
16.
17.     # Before exporting our graph, we need to precise what we can output nodes
18.     # this variables is plural, because you can have multiple output nodes
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来,或者哪些结点可以丢弃
22.     #所以,output_node_names必须根据不同的网络进行修改
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_devices)
30.
```

关闭

```
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 to the filesystem
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_graph_def.node))
52.
53.
54. if __name__ == '__main__':
55.     parser = argparse.ArgumentParser()
56.     parser.add_argument("--model_folder", type=str, help="Model folder to export")
57.     args = parser.parse_args()
58.
59.     freeze_graph(args.model_folder)
```

关闭

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("--
frozen_model_filename", default="results/frozen_model.pb", type=str, help="Frozen model f:
26.      args = parser.parse_args()
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
```

关闭

```

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feeds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]] # < 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tensor和目标tensor即可
- 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')`一定要使用tensor的名字

- 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

```

关闭


```

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='inputs_placeholder', sh
[None, 10])
08.     labels_placeholder = tf.placeholder(tf.float32, name='labels_placeholder', sh
[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) / 2)

```

关闭

```
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="correct_predictions")
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 %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], 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))
```

关闭

```
62. print("Number of batches: %d" % len(batches))
63. print("Size of full batch: %d" % len(batches[0]))
64. print("Size of 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
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)
```

关闭

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

3

踩

0

[上一篇](#) tensorflow的设计理念[下一篇](#) 错误Tensor is not an element of this graph tensorflow

关闭

相关文章推荐

- Tensorflow保存模型，恢复模型，使用训练好的模...
- 【TensorFlow】Caffe模型转tensorflow模型并使用...
- Presto的服务治理与架构在京东的实践与应用--王...
- Retrofit 从入门封装到源码解析
- 如何让TensorFlow模型运行提速36.8%
- TensorFlow教程03：针对机器学习初学者的MNIS...
- 深入掌握Kubernetes应用实践--王渊命
- 自然语言处理工具Word2Vec
- Tensorflow保存模型，恢复模型，使用训练好的模...
- TensorFlow学习（二）：变量常量类型

- Python基础知识汇总
- TensorFlow 模型保存/载入的两种方法
- Android核心技术详解
- tensorflow从已经训练好的模型中，恢复(指定)权重...
- tensorflow将训练好的模型freeze,即将权重固化到...
- tensorflow将训练好的模型

查看评论

1楼 [RossYoung](#) 2017-06-09 15:05发表



博主写的真好，网上查了好久，都是简单的保存某个变量，没写怎么把图和相关变量对应重新加载的，非常感谢

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

[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

400-660-0108 | 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 © 1999-2017, CSDN.NET, All Rights Reserved



关闭