

tensorflow1.4 c++编译以及API使用

原创

2018年01月15日 14:20:28

标签：tensorflow1.4 / c / ubuntu16.04 / python

281



0

摘要：最近在研究如何使用tensorflow c++ API调用tensorflow python环境下训练得到的网络模型文件。参考了很多博客文档，一路上踩了很多坑，现将自己的方法步骤记录下来，希望能够帮到有需要的人！（本文默认读者对python环境下tensorflow的使用已经比较熟悉了）

方法简要梳理如下：

1. 安装bazel，然后用bazel编译tensorflow源码，产生我们需要的库文件。
2. 在python环境使用tensorflow训练一个深度神经网络，本文以mnist为例。将训练好的模型和参数冻结在pb文件中。
3. 在C++环境下，调用pb文件，对图片进行预测。最终结果如下图所示，程序成功识别到图片中的数字为1，且概率为0.95。



zwx1995zwx

原创

3

粉丝

2

喜欢

1

评论

0

等级：博客 1

访问量：319

积分：34

排名：192万+

希腊房价

以上内容由 百度 为你推荐



他的最新文章

更多

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

```
zwx@zwx-All-Series: ~/try_c++/tensorflow_mnist/build
2018-01-20 14:25:23.283084: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
296] Adding visible gpu device 0
2018-01-20 14:25:23.283197: I tensorflow/core/common_runtime/gpu/gpu_device.cc:9
83] Creating TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 wit
h 75 MB memory) -> physical GPU (device: 0, name: GeForce GTX 970, pci bus id: 0
000:01:00.0, compute capability: 5.2)
Tensor<type: float shape: [1,28,28,1] values: [[[0.992156923][0.0823529437][0]]]
...>
Output tensor si
Tensor<type: float shape: [1,10] values: [0.000222602568 0.951202273 0.00674965]
...>
Class 0 prob:0.000222603,
Class 1 prob:0.951202,
Class 2 prob:0.00674965,
Class 3 prob:0.00721129,
Class 4 prob:0.0018,
Class 5 prob:0.00155,
Class 6 prob:0.0119467,
Class 7 prob:0.00218654,
Class 8 prob:0.00189468,
Class 9 prob:0.00553808,
Final class id: 1
Final class prob: 0.951202
zwx@zwx-All-Series:~/try_c++/tensorflow_mnist/build$
```

具体程序参考项目：

<https://github.com/zhangcliff/tensorflow-c-mnist.git>

1.安装bazel

[plain]

1. `echo "deb [arch=amd64] http://storage.googleapis.com/bazel-apt stable jdk1.8" | sudo te
e /etc/apt/sources.list.d/bazel.list`

基于opencv和zbar的二维码定位与解码

文章分类

tensorflow	1篇
opencv	1篇
风格转移	1篇

文章存档

2018年3月	1篇
2018年1月	2篇

他的热门文章

tensorflow1.4 c++编译以及API使用

📖 269

基于opencv和zbar的二维码定位与解码

📖 19

图像任意指定区域的神经风格转移（论文
解读以及tensorflow实现）

📖 18

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

```
[plain]
```

```
1. sudo apt-get update
```

```
[plain]
```

```
1. sudo apt-get install bazel
```

2.tensorflow的下



本博文使用的tensorflow版本为1.4，其他版本的c++编译可能会有一些不一样。



```
[plain]
```

```
1. git clone https://github.com/tensorflow/tensorflow.git
```



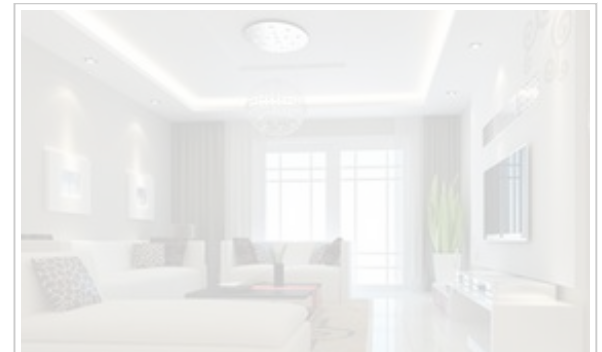
3.tensorflow的c++编译。

3.1 进入tensorflow文件夹中，首先进行项目配置。

```
[plain]
```

```
1. ./configure
```

下面我贴出在我的机器上各选项的选择：值得注意的是，如果我们要使用cuda和cudnn的话，一定要搞清楚自己机器上使用的cuda和cudnn的版本（尤其是cudnn），例如我使用的是cuda8.0和cudnn6.0.21。



九十平米装修



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗣 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```
zwx@zwx-Lenovo-IdeaPad-Y410P:~/tensorflow$ ./configure
You have bazel 0.9.0 installed.
Please specify the location of python. [Default is /home/zwx/anaconda2/bin/python]:

Found possible Python library paths:
  /home/zwx/anaconda2/lib/python2.7/site-packages
  /home/zwx/tensorflow
Please input the desired Python library path to use. Default is [/home/zwx/anaconda2/lib/python2.7/site-packages]

Do you wish to build TensorFlow with jemalloc as malloc support? [Y/n]: n
No jemalloc as malloc support will be enabled for TensorFlow.

Do you wish to build TensorFlow with Google Cloud Platform support? [Y/n]: n
No Google Cloud Platform support will be enabled for TensorFlow.

Do you wish to build TensorFlow with Hadoop File System support? [Y/n]: n
No Hadoop File System support will be enabled for TensorFlow.

Do you wish to build TensorFlow with Amazon S3 File System support? [Y/n]: n
```

<http://blog.csdn.net/zwx1995zwx>

```
zwx@zwx-Lenovo-IdeaPad-Y410P: ~/tensorflow
Do you wish to build TensorFlow with Amazon S3 File System support? [Y/n]: n
No Amazon S3 File System support will be enabled for TensorFlow.

Do you wish to build TensorFlow with XLA JIT support? [y/N]: n
No XLA JIT support will be enabled for TensorFlow.

Do you wish to build TensorFlow with GDR support? [y/N]: n
No GDR support will be enabled for TensorFlow.

Do you wish to build TensorFlow with VERBS support? [y/N]: n
No VERBS support will be enabled for TensorFlow.

Do you wish to build TensorFlow with OpenCL SYCL support? [y/N]: n
No OpenCL SYCL support will be enabled for TensorFlow.

Do you wish to build TensorFlow with CUDA support? [y/N]: y
CUDA support will be enabled for TensorFlow.

Please specify the CUDA SDK version you want to use, e.g. 7.0. [Leave empty to default to CUDA 9.0]: 8.0

Please specify the location where CUDA 8.0 toolkit is installed. Refer to README.md for more details. [Default is /usr/local/cuda]:
```

```
zwx@zwx-Lenovo-IdeaPad-Y410P: ~/tensorflow

Please specify the location where CUDA 8.0 toolkit is installed. Refer to README
.md for more details. [Default is /usr/local/cuda]:

Please specify the cuDNN version you want to use. [Leave empty to default to cuD
NN 7.0]: 6.0.21

Please specify the location where cuDNN 6.0.21 library is installed. Refer to RE
ADME.md for more details. [Default is /usr/local/cuda]:

Please specify a list of comma-separated Cuda compute capabilities you want to b
uild with.
You can find the compute capability of your device at: https://developer.nvidia.
com/cuda-gpus.
Please note that each additional compute capability significantly increases your
build time and binary size. [Default is: 3.0]

Do you want to use clang as CUDA compiler? [y/N]: n
nvcc will be used as CUDA compiler.
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

```
zwx@zwx-Lenovo-IdeaPad-Y410P: ~/tensorflow
Do you want to use clang as CUDA compiler? [y/N]: n
nvcc will be used as CUDA compiler.

Please specify which gcc should be used by nvcc as the host compiler. [Default is /usr/bin/gcc]:

Do you wish to build TensorFlow with MPI support? [y/N]: n
No MPI support will be enabled for TensorFlow.

Please specify optimization flags to use during compilation when bazel option "--config=opt" is specified [Default is -march=native]:

Would you like to interactively configure ./WORKSPACE for Android builds? [y/N]: n
Not configuring ./WORKSPACE for Android builds.

Preconfigured Bazel build configs. You can use any of the below by adding "--config=<>" to your build command. See tools/bazel.rc for more details.
--config=mkl          # Build with MKL support.
--config=monolithic   # Config for mostly static monolithic build.
Configuration finished
zwx@zwx-Lenovo-IdeaPad-Y410P:~/tensorflow$
```

3.2 使用bazel命令进行编译。编译的时间比较长，我在i3-4150cpu上编译了一个小时左右的时间。

[plain]

1. `bazel build --config=opt --config=cuda //tensorflow:libtensorflow_cc.so`

如果没有显卡则使用如下命令进行编译

[plain]

1. `bazel build --config=opt //tensorflow:libtensorflow_cc.so`

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

编译完成后，在bazel-bin/tensorflow中会生成两个我们需要的库文件：libtensorflow_cc.so 和 libtensorflow_framework.so。

在后面我们用C++调用tensorflow时需要链接这两个库文件。

4. 使用tensorflow + api调用图模型（.pb文件）。

tensorflow 编译好之后，我们使用tensorflow c++ api调用一个已经冻结的图模型（.pb文件）

具体程序参考项目：



<https://github.com/zcliff/tensorflow-c-mnist.git>



4.1 在python环境下生成一个图模型（.pb文件）

对于tensorflow，在Python环境下的使用是最方便的，tensorflow的python api也是最多最全面的。因此我们在python环境下，训练了一个深度神经网络模型，并将模型和参数都冻结在一个pb文件中。为后面使用C++ API调用这个pb文件做好准备。我们以经典的mnist为例。

数据处理与模型的训练，这里就不多说了（默认读者对python环境下tensorflow的使用已经比较熟悉）。这里要说的是pb文件的生成，使用一下代码：

[python]

```
1. from tensorflow.python.framework.graph_util import convert_variables_to_constants
2. graph = convert_variables_to_constants(sess, sess.graph_def, ["softmax"])
3. tf.train.write_graph(graph, 'models', 'model.pb', as_text=False)
```

其中，convert_variables_to_constants()函数将参数变量冻结在图模型中，其中第三个参数为网络输出tensor的名字(name)。因为我的网络输出是这样定义的：y_conv = tf.nn.softmax(logits,name='sof

write_graph()函数生成.pb文件，第二个参数为生成pb文件的文件夹，第三个参数为pb文件的名字。

将上面三行代码加入到你的模型训练的python脚本中，最后便可以得到我们需要的pb文件。

4.2 c++环境下调用pb文件。

第一步，加载模型



```
[cpp]
1. Session* se
2. Status status = NewSession(SessionOptions(), &session); //创建新会话Session
3.
4.
5. string model_path="model.pb";
6. GraphDef graphdef; //Graph Definition for current model
7.
8.
9. Status status_load = ReadBinaryProto(Env::Default(), model_path, &graphdef); //从pb文件中读取图模型;
10. if (!status_load.ok()) {
11.     std::cout << "ERROR: Loading model failed..." << model_path << std::endl;
12.     std::cout << status_load.ToString() << "\n";
13.     return -1;
14. }
15. Status status_create = session->Create(graphdef); //将模型导入会话Session中;
16. if (!status_create.ok()) {
17.     std::cout << "ERROR: Creating graph in session failed..." << status_create.ToString() << std::endl;
18.     return -1;
19. }
20. cout << "Session successfully created."<< endl;
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

[cpp]

```

1. Status ReadTensorFromImageFile(const string& file_name, const int input_height,
2.                                const int input_width, const float input_mean,
3.                                const float input_std,
4.                                std::vector<Tensor>* out_tensors)

```

[cpp]

```

1. static Status ReadEntireFile(tensorflow::Env* env, const string& filename,
2.                               Tensor* output)

```



ReadEntireFile()函数读取文件内容，并将其赋值给其第三个输入参数 Tensor* output，但是这个tensor并不能直接输入：我们加载的模型，需要经过一定的预处理。

在ReadTensorFromImageFile()函数，建立一个会话session，在会话中对读取到的tensor进行预处理，例如，对tensor解码(DecodeJpeg)，resize，归一化等等。

第三步，运行模型

[cpp]

```

1. const Tensor& resized_tensor = resized_tensors[0];
2. vector<tensorflow::Tensor> outputs;
3. string output_node = "softmax";
4. Status status_run = session->Run({{"inputs", resized_tensor}}, {output_node}, {}, &outputs);

```

resized_tensors的类型是std::vector<Tensor>，是一个vector容器。其在程序中，是ReadTensorFromImageFile()函数的最后一个输入参数，对读取到的图片tensor进行预处理后便保存在这个容器中。模型预测时使用的函数为session->Run({{"inputs", resized_tensor}}, {output_node}, {}, &outputs)。

值得注意的是，"inputs"是图模型输入tensor的名字(name)，变量output_node保存的是图模型输出te

如果你有一个pb文件，可是不知道它的输入输出tensor的名字，我们可以在python环境中使用API加载这个模型，然后将模型中的所有operation打印出来，第一项便是输入tensor，最后一项便是输出tensor。

[python]

```
1. print(sess.graph.get_operations())
2. print(sess.graph.get_operations()[0])
3. print(sess.graph.get_operations()[1])
```

0

第四步，从模型输出tensor中获得各类别的概率。

[cpp]

```
1. Tensor t = ts[0]; // Fetch the first tensor
2. auto tmap = tensor<float, 2>(); // Tensor Shape: [batch_size, target_class_num]
3. int output_dim = t.shape().dim_size(1); // Get the target_class_num from 1st dimension

4.
5. // Argmax: Get Final Prediction Label and Probability
6. int output_class_id = -1;
7. double output_prob = 0.0;
8. for (int j = 0; j < output_dim; j++)
9. {
10.     std::cout << "Class " << j << " prob:" << tmap(0, j) << ", " << std::endl;
11.     if (tmap(0, j) >= output_prob) {
12.         output_class_id = j;
13.         output_prob = tmap(0, j);
14.     }
15. }
16. std::cout << "Final class id: " << output_class_id << std::endl;
17. std::cout << "Final class prob: " << output_prob << std::endl;
```

4.3 使用cmake进行编译

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

nsorflow_cc.so , libtensorflow_framework.so) 放入其中。调用pb文件进行预测的C++文件，取名为tf.cpp，放在tensorflow_mnist目录下。文件结构如下图所示。

下面给出我的CMakeLists.txt的文件内容

```
[plain]
1. cmake_minimum_required (VERSION 2.8.8)
2. project (tf_example)
3. set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -g -std=c++11 -W")
4.
5. link_directories(./lib)
6. include_directories(
7.     /home/zwx/tensorflow/tensorflow
8.     /home/zwx/tensorflow/bazel-genfiles
9.     /home/zwx/tensorflow/tensorflow/contrib/makefile/downloads/nsync/public
10.    /usr/local/include/eigen3
11.    /home/zwx/tensorflow/bazel-bin/tensorflow
12.    /home/zwx/tensorflow/tensorflow/contrib/makefile/gen/protobuf/include
13. )
14. add_executable(tf_test tf.cpp)
15. target_link_libraries(tf_test tensorflow_cc tensorflow_framework)
```

最后进入build文件夹，对该工程进行编译：

```
[plain]
1. cd build
2. cmake ..
3. make
```

不过，在make 这一步很大几率会报错，我将我碰到的几个问题和解决方法写在这里，仅供参考。

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

```
zwx@zwx-Lenovo-IdeaPad-Y410P: ~/try/tensorflow_mnist/build
[ 50%] Building CXX object CMakeFiles/tf_test.dir/tf.cpp.o
In file included from /home/zwx/tensorflow/tensorflow/core/framework/variant.h:26:0,
                 from /home/zwx/tensorflow/tensorflow/core/framework/allocator.h:26,
                 from /home/zwx/tensorflow/tensorflow/core/framework/tensor.h:20,
                 from /home/zwx/tensorflow/tensorflow/cc/framework/ops.h:21,
                 from /home/zwx/tensorflow/tensorflow/cc/ops/const_op.h:19,
                 from /home/zwx/try/tensorflow_mnist/tf.cpp:9:
/home/zwx/tensorflow/bazel-genfiles/tensorflow/core/framework/tensor.pb.h:12:2:
error: #error This file was generated by a newer version of protoc which is
#error This file was generated by a newer version of protoc which is
^
/home/zwx/tensorflow/bazel-genfiles/tensorflow/core/framework/tensor.pb.h:13:2:
error: #error incompatible with your Protocol Buffer headers. Please update
#error incompatible with your Protocol Buffer headers. Please update
^
/home/zwx/tensorflow/bazel-genfiles/tensorflow/core/framework/tensor.pb.h:14:2:
error: #error your headers.
#error your headers.
^
In file included from /home/zwx/tensorflow/tensorflow/core/framework/variant.h:26:0,
```

解决方法：安装正确版本的protobuf。在ubuntu16.04，tensorflow1.4下，应该安装protobuf-3.4.0。

问题二：nsync_cv.h文件缺失

```
zwx@zwx-Lenovo-IdeaPad-Y410P: ~/try/tensorflow_mnist/build
zwx@zwx-Lenovo-IdeaPad-Y410P:~/try/tensorflow_mnist/build$ make
[ 50%] Building CXX object CMakeFiles/tf_test.dir/tf.cpp.o
In file included from /home/zwx/tensorflow/tensorflow/core/platform/mutex.h:31:0
,
      from /home/zwx/tensorflow/tensorflow/core/framework/variant.h:3
1,
      from /home/zwx/tensorflow/tensorflow/core/framework/allocator.h
:26,
      from /home/zwx/tensorflow/tensorflow/core/framework/tensor.h:20
,
      from /home/zwx/tensorflow/tensorflow/cc/framework/ops.h:21,
      from /home/zwx/tensorflow/tensorflow/cc/ops/const_op.h:19,
      from /home/zwx/try/tensorflow_mnist/tf.cpp:9:
/home/zwx/tensorflow/tensorflow/core/platform/default/mutex.h:25:22: fatal error
: nsync_cv.h: 没有那个文件或目录
compilation terminated.
CMakeFiles/tf_test.dir/build.make:62: recipe for target 'CMakeFiles/tf_test.dir/
tf.cpp.o' failed
make[2]: *** [CMakeFiles/tf_test.dir/tf.cpp.o] Error 1
CMakeFiles/Makefile2:67: recipe for target 'CMakeFiles/tf_test.dir/all' failed
make[1]: *** [CMakeFiles/tf_test.dir/all] Error 2
Makefile:83: recipe for target 'all' failed
make: *** [all] Error 2
zwx@zwx-Lenovo-IdeaPad-Y410P:~/try/tensorflow_mnist/build$
```

正常情况下，该文件应该在路径tensorflow/tensorflow/contrib/makefile/downloads/nsync/public里。如果出现这个问题，很可能是tensorflow/tensorflow/contrib/makefile/下没有downloads文件夹，可能是编译的时候网络不好，没有下载这个文件夹。

解决方法：进入 tensorflow/tensorflow/contrib/makefile/ 文件夹下，找到脚步。然后回到tensorflow 文件夹下，执行该脚本。

[plain]

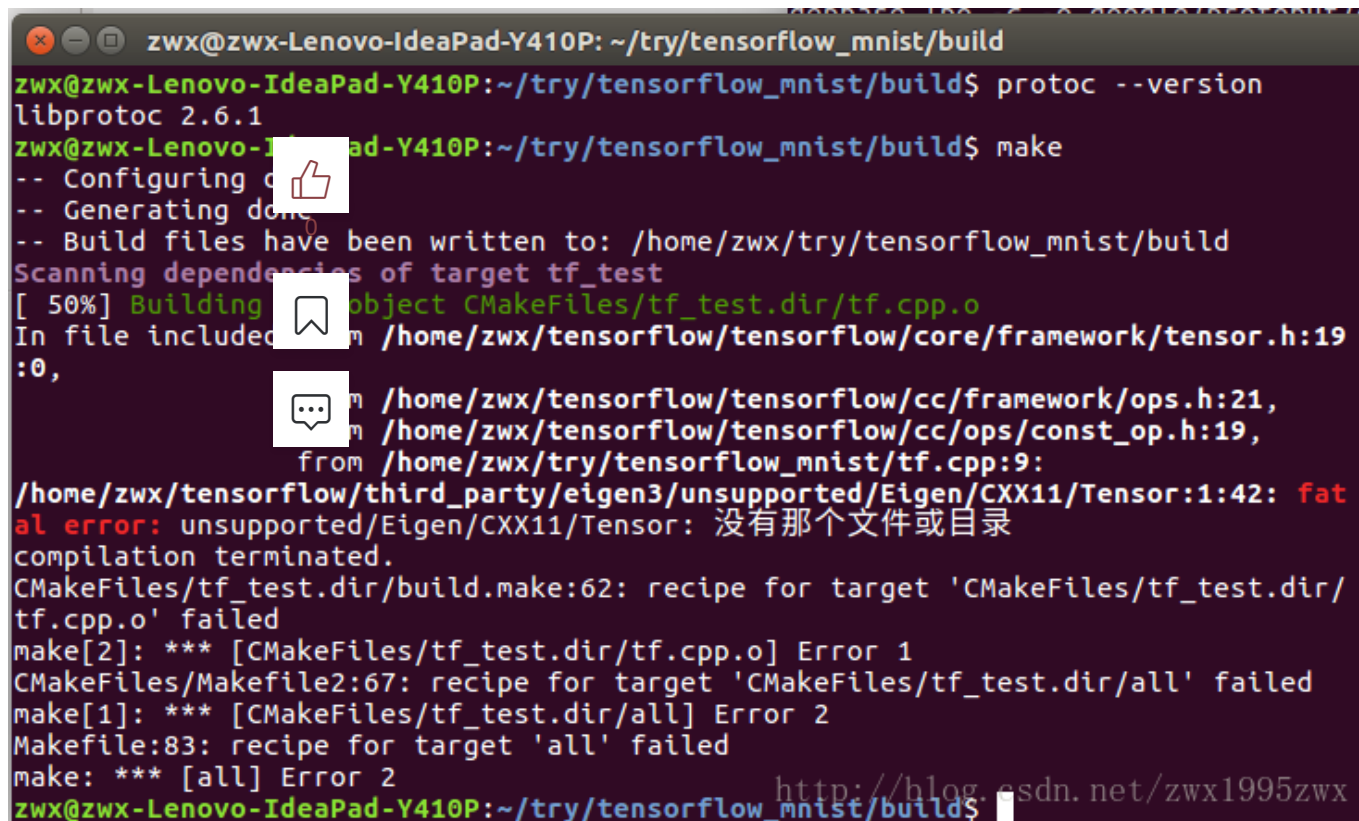
1. ./tensorflow/contrib/makefile/download_dependencies.sh

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

问题三：



```
zwx@zwx-Lenovo-IdeaPad-Y410P: ~/try/tensorflow_mnist/build
zwx@zwx-Lenovo-IdeaPad-Y410P:~/try/tensorflow_mnist/build$ protoc --version
libprotoc 2.6.1
zwx@zwx-Lenovo-IdeaPad-Y410P:~/try/tensorflow_mnist/build$ make
-- Configuring done
-- Generating done
-- Build files have been written to: /home/zwx/try/tensorflow_mnist/build
Scanning dependencies of target tf_test
[ 50%] Building CXX object CMakeFiles/tf_test.dir/tf.cpp.o
In file included from /home/zwx/tensorflow/tensorflow/core/framework/tensor.h:19:0,
                 from /home/zwx/tensorflow/tensorflow/cc/framework/ops.h:21,
                 from /home/zwx/tensorflow/tensorflow/cc/ops/const_op.h:19,
                 from /home/zwx/try/tensorflow_mnist/tf.cpp:9:
/home/zwx/tensorflow/third_party/eigen3/unsupported/Eigen/CXX11/Tensor:1:42: fatal error: unsupported/Eigen/CXX11/Tensor: 没有那个文件或目录
compilation terminated.
CMakeFiles/tf_test.dir/build.make:62: recipe for target 'CMakeFiles/tf_test.dir/tf.cpp.o' failed
make[2]: *** [CMakeFiles/tf_test.dir/tf.cpp.o] Error 1
CMakeFiles/Makefile2:67: recipe for target 'CMakeFiles/tf_test.dir/all' failed
make[1]: *** [CMakeFiles/tf_test.dir/all] Error 2
Makefile:83: recipe for target 'all' failed
make: *** [all] Error 2
zwx@zwx-Lenovo-IdeaPad-Y410P:~/try/tensorflow_mnist/build$
```

解决方法：这个问题的造成原因和问题二是一样的，查看下载好的downloads文件夹，发现其中有一个文件夹为eigen，进入eigen文件夹执行以下命令。

- ```
[plain]
1. mkdir build
2. cd build
3. cmake
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册



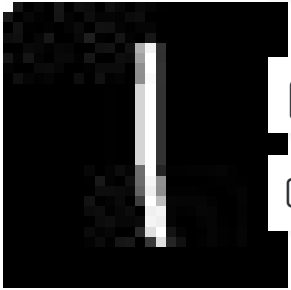
安装完毕后，在usr/local/include目录下会出现eigen3文件夹。

#### 4.4 运行可执行程序

make成功后，在build目录下会出现一个可执行文件tf\_test。将一张28\*28的数字图片也放在build路径下，文件名为digi



0



[plain]

```
1. ./tf_test digit.jpg
```

结果如下图所示：

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

```
zwx@zwx-All-Series: ~/try_c++/tensorflow_mnist/build
2018-01-20 14:25:23.283084: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
296] Adding visible gpu device 0
2018-01-20 14:25:23.283197: I tensorflow/core/common_runtime/gpu/gpu_device.cc:9
83] Creating TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 wit
h 75 MB memory) -> physical GPU (device: 0, name: GeForce GTX 970, pci bus id: 0
000:01:00.0, compute capability: 5.2)
Tensor<type: float shape: [1,28,28,1] values: [[[0.992156923][0.0823529437][0]]]
...>
Output tensor si
Tensor<type: float shape: [1,10] values: [0.000222602568 0.951202273 0.00674965]
...>
Class 0 prob:0.000222603,
Class 1 prob:0.951202273,
Class 2 prob:0.00674965,
Class 3 prob:0.00721129,
Class 4 prob:0.0018,
Class 5 prob:0.00155,
Class 6 prob:0.0119467,
Class 7 prob:0.00218654,
Class 8 prob:0.00189468,
Class 9 prob:0.00553808,
Final class id: 1
Final class prob: 0.951202
zwx@zwx-All-Series:~/try_c++/tensorflow_mnist/build$
```

从中我们可以看到该c++程序识别到digit.jpg图片为数字一，为1的概率为0.95。

总结：这篇博文主要介绍了如何从源码编译tensorflow c++ API，并且使用c++ API调用一个在python环境下已经训练好并冻结参数的模型文件（.pb文件），最终生成一个可执行文件tf\_test。通过运行该文件，我们成功识别了手写体数字。

具体脚本参考项目：

[https://github.com/zhangcliff/tensorflow\\_c\\_mnist.git](https://github.com/zhangcliff/tensorflow_c_mnist.git)

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册



0



加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

## Windows下编译tensorflow-gpu教程



DawnRanger

2017年08月31日 21:58

📖 3166

官方文档：1. 在windows下安装tensorflow 2. 在windows下编译tensorflow(cmake)首先是系统配置：OS：Windows Server 2012 R2(相...



## tensorflow c++编译



fireflychh

2017年10月23日 17:55

📖 1114

1. 安装bazel，可以在🗨️官网上查找到具体安装方法 2. git clone 一份tensorflow的源码 上官网看，（需要clone相对稳定的版本如r1.3,否则可能...



## 一个数学公式教你秒懂天下英语

老司机教你一个数学公式秒懂天下英语

## Ubuntu1604 下编译并使用tensorflow c++库

----- 安装tensorflow c++库 ----- 1. 安装bazel 具体怎么安装可以在bazel官网看看 有直接的...



lovekkss

2017年08月11日 19:04

📖 1224

## Tensorflow C++ 编译和调用图模型



rockingdingo

2017年07月19日 23:00

📖 15412

最近在研究如何打通tensorflow线下python脚本训练建模，线上生产环境用C++代码直接调用预先训练好的模型完成

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

[登录](#)[注册](#)

## tensorflow源码编译



a446712385

2018年01月24日 09:45

31

1. 下载bazel安装文件，最新版本 <https://github.com/bazelbuild/bazel/releases> 安装成功运行报错：/lib64/libc.so.6: ve  
rsi...

## 程序员不会英语怎么行？

老司机教你一个数学公



懂天下英语

0



## 从源码编译TensorFlow CPU版本



guxi123

2017年07月11日 16:47

3323

1 安装Python3 的依赖sit-get install python3-numpy python3-dev python3-pip python3-wheel 2 从源码安装Baz  
el ...

## 源代码编译 tensorflow 1.3.0



u014234504

2018年01月10日 10:37

107

1. My os environment Centos 7.4 2. Install bazel and protobuf bazel version 0.5...

## 编写TensorFlow文档



u010859707

2017年06月14日 19:55

573

我们欢迎来自社区的Tensorflow文档的贡献。本文档介绍了如何为该文档做出贡献。特别地，本文档解释了以下内容：文件所在位置。如何进行一致的编辑。在提交文档之前如何构建和测试您的文档更改。您...

## 如何用C++实现自己的Tensorflow



dev\_csdn

2017年11月10日 15:59

13884

原文：How To Write Your Own Tensorflow in C++ 作者：Ray Zhang 翻译：无阻我飞扬 摘要：TensorFlow是由谷歌  
基于DistBelief

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

## Tensorflow C++学习(二)



jmh1996

2017年06月14日 10:24

8556

前言本节主要介绍（一）中的代码，了解如何使用C++创建图和

tensor，并使用它们进行计算。代码// tensorflow/cc/example/example.cc#include "tensorf...

## 如何实现Visual Studio C++调用TensorFlow代码

对机器学习感兴趣的VS 们，在Windows上搭建好TensorFlow开发环境后，应该会迫不及待地想要编写一个C++程序调用TensorFlow代码。只要能调用成功，你就可以将机器学习融入到你的...



ShadowN1ght

20



9月11日 15:45



4428

## 开局只有5名囚犯,从荒岛到建立帝国！

突然火爆了80后朋友圈,都在耕地建家园,打造自己的帝国时代！

百度广告

## c++调用tensorflow教程



qq\_34484472

2017年08月02日 21:33

11779

目前深度学习越来越火，学习、使用tensorflow的相关工作者也越来越多。但是目前绝大部分的python都是拥有着丰富的python的API，而c++的API不够完善。这就导致绝大多数是使用tenso...

## 将TensorFlow的网络导出为单个文件



EncodeTS

2017年01月14日 20:03

3915

有时候，我们需要将TensorFlow的模型导出为单个文件（同时包含模型架构定义与权重），方便在其他地方使用（如在c++中部署网络）。利用`tf.train.write\_graph()`默认情况下只导...

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

Check out the related post: Loading TensorFlow graphs via host languages (using the C API). The cur...



mydear\_11000 2016年08月09日 08:36 2645

## 从零开始使用tensorflow (1) ——安装



juanjuan1314 2016年08月23日 19:16 4253

1. Python和jdk1.8之产...安装好了。 2. Pip install <https://storage.googleapis.com/tensorflow/mac/...>



0

## Tensorflow C++编译和调用图模型



c2a2o2 2017年09月25日 16:36 1536

简介 最近在研究如何打... orflow线下 python 的脚本训练建模, 利用freeze\_graph工具输出.pb图文件, 之后再线上生产环境用C++代码直接... 预先训练好的模型完成预测...



## 还为Unity3D 代码被反编译而头疼？

无需手动加密Assembly.DLL代码，自动编译mono，防止反编译



## tensorflow c++接口编译



junxiacaocao 2018年01月25日 17:11 80

源码编译安装部分省，c++接口编译参考：<http://www.deeplnlp.org/blog/tensorflow-cpp-build-for-production/> ...

## 【转】tensorflow学习使用路线



Callon\_H 2017年03月04日 20:37 1663

版权声明：本文为博主hjimce的原创文章，原文地址为<http://blog.csdn.net/hjimce/article/details/51899683>。...

## 【TensorFlow代码笔记】c\_ani.h



s\_sunnyy 2017年04月17日 15:44 1931

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册



## 如何编译运行tensorflow的demo



moyimoyi123

2017年08月06日 14:02



1308

1.安装编译工具bazel，具体可以参照官方教程。 <https://docs.bazel.build/versions/master/install-ubuntu.html> 2. 配置t  
ensor...



0



加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册