

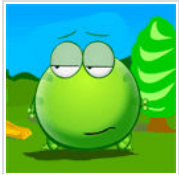
网络资源是无限的

目录视图

摘要视图

RSS 订阅

个人资料



fengbingchun



访问：2252591次
积分：25003
等级：**BLOG > 7**
排名：第202名

原创：341篇 转载：144篇
译文：0篇 评论：1434条

文章分类

- Android (9)
- ActiveX (18)
- Bar Code (16)
- Caffe (20)
- C# (5)
- Climg (4)
- Contour Detection (9)
- CxImage (6)
- Code::Blocks (3)
- Cloud Computing (1)
- C/C++ (82)
- CUDA (10)
- CMake (3)
- Design Patterns (25)
- Database/Dataset (4)
- Deep Learning (9)
- Eclipse (3)
- Emgu CV (1)
- Eigen (1)
- FFmpeg (1)
- Feature Extraction (1)
- FreeType (1)
- Face (8)
- GPU (3)
- Git (3)
- GCC (1)
- GDAL (5)

CSDN学院招募微信小程序讲师啦 程序员简历优化指南！ 【观点】移动原生App开发 PK HTML 5开发 云端应用征文大赛，秀绝招，赢无人机！

tiny-cnn开源库的使用(MNIST)

2016-01-24 15:09 9482人阅读 评论(93) 收藏 举报

本文档已收录于： 深度学习知识库

分类： Caffe (19) Neural Network (12) Deep Learning (8)

版权声明：本文为博主原创文章，未经博主允许不得转载。

tiny-cnn是一个基于CNN的开源库，它的License是BSD 3-Clause。作者也一直在维护更新，对进一步掌握CNN很有帮助，因此下面介绍下tiny-cnn在windows7 64bit vs2013的编译及使用。

- 从<https://github.com/nyanp/tiny-cnn>下载源码：

\$ git clone <https://github.com/nyanp/tiny-cnn.git> 版本号为77d80a8，更新日期2016.01.22
- 源文件中已经包含了vs2013工程，vc/tiny-cnn.sln，默认是win32的，examples/main.cpp需要OpenCV的支持，这里新建一个x64的控制台工程tiny-cnn；
- 仿照源工程，将相应.h文件加入到新控制台工程中，新加一个test_tiny-cnn.cpp文件；
- 将examples/mnist中test.cpp和train.cpp文件中的代码复制到test_tiny-cnn.cpp文件中；

```
[cpp]
01. #include <iostream>
02. #include <string>
03. #include <vector>
04. #include <algorithm>
05. #include <tiny_cnn/tiny_cnn.h>
06. #include <opencv2/opencv.hpp>
07.
08. using namespace tiny_cnn;
09. using namespace tiny_cnn::activation;
10.
11. // rescale output to 0-100
12. template <typename Activation>
13. double rescale(double x)
14. {
15.     Activation a;
16.     return 100.0 * (x - a.scale().first) / (a.scale().second - a.scale().first);
17. }
18.
19. void construct_net(network<mse, adagrad>& nn);
20. void train_lenet(std::string data_dir_path);
21. // convert tiny_cnn::image to cv::Mat and resize
22. cv::Mat image2mat(image<>& img);
23. void convert_image(const std::string& imagefilename, double minv, double maxv, int w, int
24. void recognize(const std::string& dictionary, const std::string& filename, int target);
25.
26. int main()
27. {
28.     //train
29.     std::string data_path = "D:/Download/MNIST";
```

[HTML](#) (3)
[Image Recognition](#) (8)
[Image Processing](#) (18)
[Image Registration](#) (13)
[ImageMagick](#) (3)
[Java](#) (5)
[Linux](#) (20)
[Log](#) (2)
[Makefile](#) (2)
[Mathematical Knowledge](#) (6)
[Multi-thread](#) (4)
[Matlab](#) (33)
[MFC](#) (8)
[MinGW](#) (3)
[Mac](#) (1)
[Neural Network](#) (13)
[OCR](#) (9)
[Office](#) (2)
[OpenCL](#) (2)
[OpenSSL](#) (7)
[OpenCV](#) (86)
[OpenGL](#) (2)
[OpenGL ES](#) (3)
[OpenMP](#) (3)
[Photoshop](#) (1)
[Python](#) (4)
[Qt](#) (1)
[SIMD](#) (14)
[Software Development](#) (4)
[System architecture](#) (2)
[Skia](#) (1)
[SVN](#) (1)
[Software Testing](#) (4)
[Shell](#) (2)
[Socket](#) (3)
[Target Detection](#) (2)
[Target Tracking](#) (2)
[VC6](#) (6)
[VS2008](#) (16)
[VS2010](#) (4)
[VS2013](#) (3)
[vigra](#) (2)
[VLC](#) (5)
[VLFeat](#) (1)
[wxWidgets](#) (1)
[Watermark](#) (4)
[Windows7](#) (6)
[Windows Core Programming](#) (9)
[XML](#) (2)

Free Codes

[pudn](#)
[freecode](#)
[Peter's Functions](#)
[CodeProject](#)
[SourceCodeOnline](#)
[Computer Vision Source Code](#)
[Codesoso](#)
[Digital Watermarking](#)
[SourceForge](#)
[HackChina](#)
[oschina](#)

```
30.     train_lenet(data_path);
31.
32.     //test
33.     std::string model_path = "D:/Download/MNIST/LeNet-weights";
34.     std::string image_path = "D:/Download/MNIST/";
35.     int target[10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
36.
37.     for (int i = 0; i < 10; i++) {
38.         char ch[15];
39.         sprintf(ch, "%d", i);
40.         std::string str;
41.         str = std::string(ch);
42.         str += ".png";
43.         str = image_path + str;
44.
45.         recognize(model_path, str, target[i]);
46.     }
47.
48.     std::cout << "ok!" << std::endl;
49.     return 0;
50. }
51.
52. void train_lenet(std::string data_dir_path) {
53.     // specify loss-function and learning strategy
54.     network<mse, adagrad> nn;
55.
56.     construct_net(nn);
57.
58.     std::cout << "load models..." << std::endl;
59.
60.     // load MNIST dataset
61.     std::vector<label_t> train_labels, test_labels;
62.     std::vector<vec_t> train_images, test_images;
63.
64.     parse_mnist_labels(data_dir_path + "/train-labels.idx1-ubyte",
65.         &train_labels);
66.     parse_mnist_images(data_dir_path + "/train-images.idx3-ubyte",
67.         &train_images, -1.0, 1.0, 2, 2);
68.     parse_mnist_labels(data_dir_path + "/t10k-labels.idx1-ubyte",
69.         &test_labels);
70.     parse_mnist_images(data_dir_path + "/t10k-images.idx3-ubyte",
71.         &test_images, -1.0, 1.0, 2, 2);
72.
73.     std::cout << "start training" << std::endl;
74.
75.     progress_display disp(train_images.size());
76.     timer t;
77.     int minibatch_size = 10;
78.     int num_epochs = 30;
79.
80.     nn.optimizer().alpha *= std::sqrt(minibatch_size);
81.
82.     // create callback
83.     auto on_enumerate_epoch = [&]() {
84.         std::cout << t.elapsed() << "s elapsed." << std::endl;
85.         tiny_cnn::result res = nn.test(test_images, test_labels);
86.         std::cout << res.num_success << "/" << res.num_total << std::endl;
87.
88.         disp.restart(train_images.size());
89.         t.restart();
90.     };
91.
92.     auto on_enumerate_minibatch = [&]() {
93.         disp += minibatch_size;
94.     };
95.
96.     // training
97.     nn.train(train_images, train_labels, minibatch_size, num_epochs,
98.         on_enumerate_minibatch, on_enumerate_epoch);
99.
100.    std::cout << "end training." << std::endl;
101.
102.    // test and show results
103.    nn.test(test_images, test_labels).print_detail(std::cout);
104.
105.    // save networks
106.    std::ofstream ofs("D:/Download/MNIST/LeNet-weights");
107.    ofs << nn;
108. }
```

[关闭](#)

libsvm
joys99
CodeForge
cvchina
tesseract-ocr
sift
TiRG
imgSeek
OpenSURF

Friendly Link

OpenCL
Python
poesia-filter
TortoiseSVN
imgSeek
Notepad
Beyond Compare
CMake
VIGRA
CodeGuru
vchome
aforgenet
CVLAB
Doxygen
Coursera
OpenMP

Technical Forum

Matlab China
OpenCV China
The CImg Library
Open Computer Vision Library
CImage
ImageMagick
ImageMagick China
OpenCV_China
Subversion China

```
109.
110. void construct_net(network<mse, adagrad> nn) {
111.     // connection table [Y.Lecun, 1998 Table.1]
112. #define O true
113. #define X false
114.     static const bool tbl[] = {
115.         0, X, X, X, 0, 0, 0, X, X, 0, 0, 0, 0, X, 0, 0,
116.         0, 0, X, X, X, 0, 0, 0, X, X, 0, 0, 0, 0, X, 0,
117.         0, 0, 0, X, X, X, 0, 0, 0, X, X, 0, X, 0, 0, 0,
118.         X, 0, 0, 0, X, X, 0, 0, 0, 0, X, X, 0, X, 0, 0,
119.         X, X, 0, 0, 0, X, X, 0, 0, 0, 0, X, 0, 0, X, 0,
120.         X, X, X, 0, 0, 0, X, X, 0, 0, 0, 0, X, 0, 0, 0
121.     };
122. #undef O
123. #undef X
124.
125.     // construct nets
126.     nn << convolutional_layer<tan_h>(32, 32, 5, 1, 6) // C1, 1@32x32-in, 6@28x28-out
127.         << average_pooling_layer<tan_h>(28, 28, 6, 2) // S2, 6@28x28-in, 6@14x14-out
128.         << convolutional_layer<tan_h>(14, 14, 5, 6, 16,
129.         connection_table(tbl, 6, 16)) // C3, 6@14x14-in, 16@10x10-in
130.         << average_pooling_layer<tan_h>(10, 10, 16, 2) // S4, 16@10x10-in, 16@5x5-out
131.         << convolutional_layer<tan_h>(5, 5, 5, 16, 120) // C5, 16@5x5-in, 120@10x10-out
132.         << fully_connected_layer<tan_h>(120, 10); // F6, 120-in, 10-out
133. }
134.
135. void recognize(const std::string& dictionary, const std::string& filename, int target) {
136.     network<mse, adagrad> nn;
137.
138.     construct_net(nn);
139.
140.     // load nets
141.     std::ifstream ifs(dictionary.c_str());
142.     ifs >> nn;
143.
144.     // convert imagefile to vec_t
145.     vec_t data;
146.     convert_image(filename, -1.0, 1.0, 32, 32, data);
147.
148.     // recognize
149.     auto res = nn.predict(data);
150.     std::vector<std::pair<double, int> > scores;
151.
152.     // sort & print top-3
153.     for (int i = 0; i < 10; i++)
154.         scores.emplace_back(rescale<tan_h>(res[i]), i);
155.
156.     std::sort(scores.begin(), scores.end(), std::greater<std::pair<double, int>>());
157.
158.     for (int i = 0; i < 3; i++)
159.         std::cout << scores[i].second << ", " << scores[i].first << std::endl;
160.
161.     std::cout << "the actual digit is: " << scores[0].second << ", correct digit is: " << target << std::endl;
162.
163.     // visualize outputs of each layer
164.     //for (size_t i = 0; i < nn.depth(); i++) {
165.     //    auto out_img = nn[i]->output_to_image();
166.     //    cv::imshow("layer:" + std::to_string(i), image2mat(out_img));
167.     //}
168.     //// visualize filter shape of first convolutional layer
169.     //auto weight = nn.at<convolutional_layer<tan_h>>(0).weight_to_image();
170.     //cv::imshow("weights:", image2mat(weight));
171.
172.     //cv::waitKey(0);
173. }
174.
175. // convert tiny_cnn::image to cv::Mat and resize
176. cv::Mat image2mat(image<>& img) {
177.     cv::Mat ori(img.height(), img.width(), CV_8U, &img.at(0, 0));
178.     cv::Mat resized;
179.     cv::resize(ori, resized, cv::Size(), 3, 3, cv::INTER_AREA);
180.     return resized;
181. }
182.
183. void convert_image(const std::string& imagefilename,
184.     double minv,
185.     double maxv,
186.     int w,
187.     int h,
```

关闭

Technical Blog

邹宇华

深之JohnChen

HUNNISH

周伟明

superdott

carson2005

OpenHero

Netman(Linux)

wqvbjhc

yang_xian521

gnuhipc

gnuhipc

千里8848

CVART

tornadomeet

gotosuc

onezeros

hellogv

abcjennifer

czy_sparrow

评论排行

Windows7 32位机上, O (120)

tiny-cnn开源库的使用(MI (93)

Ubuntu 14.04 64位机上2 (89)

tesseract-ocr3.02字符识 (63)

Windows7上使用VS201 (47)

tesseract-ocr (42)

图像配准算法 (41)

Windows 7 64位机上Op (36)

OpenCV中resize函数五 (34)

小波矩特征提取matlab代 (30)

最新评论

Tesseract-OCR 3.04在Windows
fengbingchun: @iliked: 没有密码, 那个commit只是提示是从哪个commit fork过来的, 无需管那个

Tesseract-OCR 3.04在Windows
iliked: 问一下, 你第一句中的commit的那个密码, 怎么用啊
卷积神经网络(CNN)的简单实现(
fengbingchun: @hugl950123: 是需要opencv的支持, 你在本地opencv的环境配好了吗, 配好了就应该没...

卷积神经网络(CNN)的简单实现(
hugl950123: @fengbingchun: 博主请问一下, test_CNN_predict()函数是不是需要open...

卷积神经网络(CNN)的简单实现(
hugl950123: @fengbingchun: 博主请问一下, test_CNN_predict()函数是不是需要open...

卷积神经网络(CNN)的简单实现(
hugl950123: @fengbingchun: 谢谢, 能够成功运行了现在

卷积神经网络(CNN)的简单实现(
fengbingchun: @hugl950123: NN中一共有四个工程, 它们之间没有任何关系, 都是独立的, 如果要运行这篇文章的...

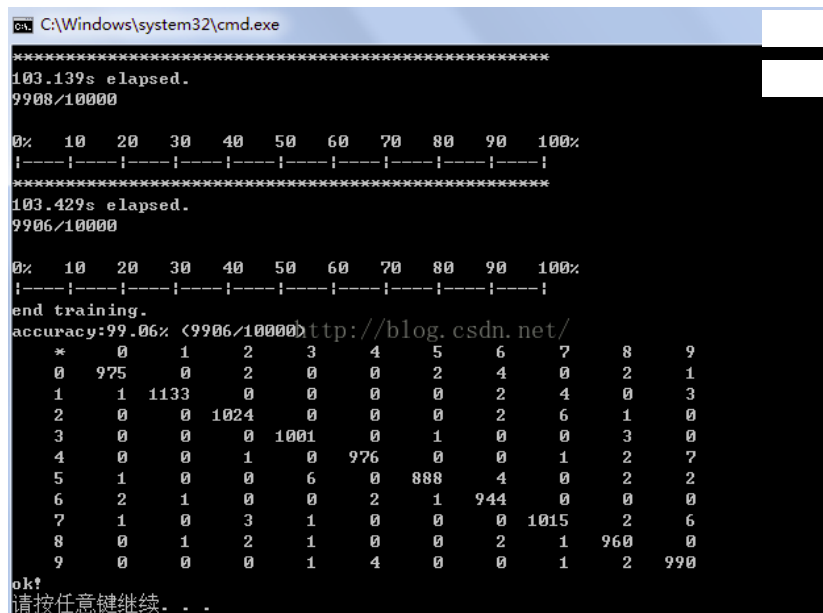
```
188.     vec_t& data) {
189.     auto img = cv::imread(imagefilename, cv::IMREAD_GRAYSCALE);
190.     if (img.data == nullptr) return; // cannot open, or it's not an image
191.
192.     cv::Mat<uint8_t> resized;
193.     cv::resize(img, resized, cv::Size(w, h));
194.
195.     // mnist dataset is "white on black", so negate required
196.     std::transform(resized.begin(), resized.end(), std::back_inserter(data),
197.         [=](uint8_t c) { return (255 - c) * (maxv - minv) / 255.0 + minv; });
198. }
```

5. 编译时会提示几个错误, 解决方法是:

(1)、error C4996, 解决方法: 将宏_SCL_SECURE_NO_WARNINGS添加到属性的预处理器定义中;

(2)、调用for_函数时, error C2668, 对重载函数的调用不明教, 解决方法: 将for_中的第三个参数强制转化为size_t类型;

6. 运行程序, train时, 运行结果如下图所示:



```
C:\Windows\system32\cmd.exe
*****
103.139s elapsed.
9908/10000

0% 10 20 30 40 50 60 70 80 90 100%
|---|---|---|---|---|---|---|---|---|---|
*****
103.429s elapsed.
9906/10000

0% 10 20 30 40 50 60 70 80 90 100%
|---|---|---|---|---|---|---|---|---|---|
end training.
accuracy:99.06% <9906/10000>http://blog.csdn.net/
* 0 1 2 3 4 5 6 7 8 9
0 975 0 2 0 0 2 4 0 2 1
1 1 1133 0 0 0 2 4 0 3
2 0 0 1024 0 0 2 6 1 0
3 0 0 0 1001 0 1 0 0 3 0
4 0 0 1 0 976 0 0 1 2 7
5 1 0 0 6 0 888 4 0 2 2
6 2 1 0 2 1 944 0 0 0
7 1 0 3 1 0 0 0 1015 2 6
8 0 1 2 1 0 0 2 1 960 0
9 0 0 0 1 4 0 0 1 2 990
ok!
请按任意键继续. . .
```

7. 对生成的model进行测试, 通过画图工具, 每个数字生成一张图像, 共10幅, 如下图:

0|1|2|3|4|5|6|7|8|9

通过导入train时生成的model, 对这10张图像进行识别, 识别结果如下图, 其中6和9被误识为5和1:

关闭

卷积神经网络(CNN)的简单实现(hugl950123: @fengbingchun:下的是新的, 我在CNN.cpp文件中每个函数都设置了断点, 还是没有变化=...

卷积神经网络(CNN)的简单实现(fengbingchun: @hugl950123:你用的是GitHub上最新的吗? 既然能编译过, 在Debug下设断点, 应该很快...

卷积神经网络(CNN)的简单实现(hugl950123: 博主, 请问我按照您的代码成功编译后执行结果窗口一闪而过, 并且里面什么内容也没有, 应该如何解决, 能不能...

- 阅读排行
- C#中OpenFileDialog的假

(47141)
- tesseract-ocr3.02字符识

(34575)
- 举例说明使用MATLAB C

(25987)
- OpenCV中resize函数五

(24317)
- 利用cvMinAreaRect2求

(24277)
- Windows 7 64位机上搭

(22586)
- opencv 检测直线、线

(20776)
- OpenCV运动检测跟踪(b

(20475)
- 图像配准算法

(19237)
- 有效的rtsp流媒体测试

(19143)

- 文章存档
- 2017年01月

(18)
- 2016年12月

(11)
- 2016年11月

(8)
- 2016年10月

(7)
- 2016年09月

(16)
- 展开



```
C:\Windows\system32\cmd.exe
0.23.0815
2.21.7515
3.20.4771
the actual digit is: 0, correct digit is: 0
1.71.2905
0.15.8632
5.14.5082
the actual digit is: 1, correct digit is: 1
2.86.5188
1.10.6617
7.8.54841
the actual digit is: 2, correct digit is: 2
3.76.7061
5.57.1777
7.45.5496
the actual digit is: 3, correct digit is: 3
4.45.647
7.25.5746
8.22.6369
the actual digit is: 4, correct digit is: 4
5.59.6614
4.39.8109
6.25.2819
the actual digit is: 5, correct digit is: 5
5.40.55
6.29.9669
1.18.1139
the actual digit is: 5, correct digit is: 6
7.74.7376
2.19.4558
8.7.43752
the actual digit is: 7, correct digit is: 7
8.53.9304
2.47.4621
7.30.0298
the actual digit is: 8, correct digit is: 8
1.98.5186
3.9.09157
0.6.80202
the actual digit is: 1, correct digit is: 9
ok!
请按任意键继续. . .
```

GitHub : <https://github.com/fengbingchun/NN>

顶

3

踩

0

上一篇 卷积神经网络(CNN)基础介绍

下一篇 tiny-cnn执行过程分析(MNIST)

我的同类文章

Caffe (19)	Neural Network (12)	Deep Learning (8)
• Caffe中Layer注册机制	2017-01-10 阅读 87	• windows7下解决caffe check... 2017-01-09 阅读 121
• cifar数据集介绍及到图像转...	2016-12-10 阅读 245	• 深度学习开源库tiny-dnn的使... 2016-12-04 阅读 465
• 卷积神经网络(CNN)代码实...	2016-12-03 阅读 2188	• 一步一步指引你在WINDOWS... 2016-03-20 阅读 1301
• Windows7 64bit VS2013 Ca...	2016-03-26 阅读 1550	• 卷积神经网络(CNN)的简单... 2016-03-06 阅读 7518
• tiny-cnn执行过程分析(MNIST)	2016-01-31 阅读 3557	• 卷积神经网络(CNN)基础介绍 2016-01-16 阅读 11361
更多文章		

layer

什么是学习综述 兼职工作

代码解析

开源网店系统 texture

数据分析师

数学之美 ndf

猜你在找

VC++Windows多线程实战图片编辑器
VC++DLL动态链接库编程
C++ 单元测试 (GoogleTest)
VC++游戏开发基础系列从入门到精通
Windows Server 2012 DHCP Server 管理

Tiny_cnn用自己的数据训练和测试
C++卷积神经网络实例tiny_cnn代码详解
C++卷积神经网络实例tiny_cnn代码详解
C++卷积神经网络实例tiny_cnn代码详解
C++卷积神经网络实例tiny_cnn代码详解

小苹果 海南三亚房产

短信接口

网络安全创业

代码解析

人工智能下载

什么是学习综述

查看评论

26楼 [开心地球仪](#) 2017-01-13 23:32发表



caffe训练好的库，tiny-dnn能直接使用吗？

Re: [fengbingchun](#) 2017-01-14 16:51发表



回复开心地球仪：文档中说是可以，不过没有亲自试过，你可以参考examples/caffe_converter这个

25楼 [qq_34337176](#) 2016-11-07 21:53发表



博主，您好，我刚接触这块东西，可否解释一下如何添加相应的.h文件，我添加不进去，拜托您了，非常感谢！

Re: [fengbingchun](#) 2016-11-08 08:21发表



回复qq_34337176：你可以参考下<https://github.com/fengbingchun/NN>，选中项目下的头文件夹，点击右键->添加->现有项即可。

24楼 [vicdd](#) 2016-10-18 21:53发表



你好。请问，scores这个数组里存放的是什么？scores[0.2]是什么意思？谢谢

Re: [fengbingchun](#) 2016-10-19 08:25发表



回复vicdd：存放数字的识别结果和置信度值

Re: [vicdd](#) 2016-10-19 11:51发表



回复fengbingchun：你好。能解释一下，`[=(uint8_t c) { return (255 - c) * (maxv - minv) / 255.0 + minv; }` resize不是已经改变尺寸了吗，为什么还要处理函数指针c呢？谢谢

Re: [fengbingchun](#) 2016-10-19 13:41发表



回复vicdd：将其值的范围从0~255变换到-1~1之间

23楼 [lincjsun](#) 2016-09-18 11:54发表



您好!请问我训练到一半了,已经生成了LeNet-weights文件,但由于电脑断电终止了,如何接着这个文件继续训练呢?谢谢!

Re: [fengbingchun](#) 2016-09-18 17:11发表



回复lincjsun：需要修改源码，将其生成的文件赋值给nn，不在调用init_weight()函数

Re: [lincjsun](#) 2016-09-19 13:32发表



回复fengbingchun：谢谢楼主了

22楼 [u012507022](#) 2016-07-03 20:13发表



博主，你好。tiny-cnn是不是只能处理灰度图像？

Re: [fengbingchun](#) 2016-07-04 08:38发表



回复u012507022：不是，tiny-cnn更新很频繁，你可以更新到最新看看。

关闭

21楼 zhushuo1992 2016-06-22 16:27发表



博主您好，我编译您的demo还有tiny_cnn里面给的vc12工程都不行，在debug+win32的时候，编译会出现以下错误：
d:\tiny-cnn-master\tiny-cnn-master\tiny_cnn\nodes.h(61): error C2899: 不能在模板声明之外使用类型名称
1>d:\tiny-cnn-master\tiny-cnn-master\tiny_cnn\nodes.h(62): error C2899: 不能在模板声明之外使用类型名称
1>d:\tiny-cnn-master\tiny-cnn-master\tiny_cnn\layers\dropout_layer.h(58): error C2610:
"tiny_cnn::dropout_layer::dropout_layer(tiny_cnn::dropout_layer &&)": 不是可默认为的特殊成员函数
1>d:\tiny-cnn-master\tiny-cnn-master\tiny_cnn\layers\dropout_layer.h(61): error C2610: "tiny_cnn::dropout_layer
&tiny_cnn::dropout_layer::operator =(tiny_cnn::dropout_layer &&)": 不是可默认为的特殊成员函数
希望博主指正！谢谢~

Re: u010076613 2016-07-14 15:46发表



回复zhushuo1992：我也报了这些错，请问你现在解决了吗？

Re: fengbingchun 2016-06-22 20:32发表



回复zhushuo1992：clone后你是不是作了修改，我的工程文件里并没有nodes.h这个文件。

Re: zhushuo1992 2016-06-22 16:31发表



回复zhushuo1992：编译平台是win764位+vs2013+opencv249

Re: fengbingchun 2016-06-22 20:31发表



回复zhushuo1992：默认只有x64的工程配置。clone下来后，在工程属性里修改opencv的路径成你本机的实际路径。然后直接编译即可。

20楼 张赛晖2 2016-06-09 21:42发表



博主，你知道如何使用lrm layer 吗？

Re: fengbingchun 2016-06-10 10:17发表



回复张赛晖2：不清楚。你可以参考下：<http://blog.csdn.net/kkk584520/article/details/42032013>

19楼 huxin3420 2016-05-20 14:06发表



博主，我用你的方法训练了以后，识别的时候用时特别久，大概识别一个需要将近1秒，然后用性能分析看了些，你写的recognise函数里面的ifs >> nn;这句用了98.5%的时间，请问博主知道大概的情况吗？

Re: fengbingchun 2016-05-20 15:47发表



回复huxin3420：你完全可以把ifs >> nn;放到外层，在开始前只需加载一次就可以了，完全没必要每次都重复加载model。后面我会调整下。

Re: huxin3420 2016-05-22 09:15发表



回复fengbingchun：多谢博主提醒，我已经改进了，现在识别一个0.1秒不到了

Re: fengbingchun 2016-05-22 10:49发表



回复huxin3420：赞！

18楼 Duke__Lee 2016-05-15 10:08发表



囧.....蒙蔽，有空的话可以在QQ 上面指导一下么.....547160794.多谢谢楼主

Re: fengbingchun 2016-05-15 13:06发表



回复Duke__Lee：不好意思，一般不上QQ，这方便的资料网上很：

关闭

17楼 Duke__Lee 2016-05-15 01:20发表



不好意思，您这个最后一步测试是怎么测试的啊.....小白.....感觉蒙蔽了。什么就导入train的module啊

Re: Duke__Lee 2016-05-29 13:40发表



回复Duke__Lee：您好，您是不是用的另外一个test.cpp啊？我在github上面看到了，但是filename一直不对，您调用测试图片的代码是怎么写的啊？图片格式是jpg吗？谢谢打扰了

Re: fengbingchun 2016-05-29 18:20发表



回复Duke__Lee：filename就是图像的绝对路径，什么格式图像都可以，因为它会用opencv的imread函数进行读取。

Re: fengbingchun 2016-05-15 09:30发表



回复Duke__Lee：保存最终train时生成的model即LeNet-weights，然后在test时，将生成的此model再读进去就可以直接predict了

16楼 [游客] 2016-05-11 15:45发表



博主，我将最后的全连接从输出改成了120-41，我自己做的训练集的标签是0-40一共四十一个字母的识别，但是每一次100%迭代完之后，准确率一直不变，一直都是1519/2000，不知道是怎么回事。还望博主指点。

Re: fengbingchun 2016-05-11 20:55发表



回复：难道是过拟合吗？你训练集数据多吗，覆盖率高吗？影响的因素很多。

15楼 止的不是战 2016-05-09 21:19发表



博主你好，用tiny_cnn可不可以直接训练jpg、png等图片呀？需不需要进行格式转换呀？

Re: fengbingchun 2016-05-10 08:15发表



回复止的不是战：是可以直接训练图像的。代码需要调整。

Re: 止的不是战 2016-05-10 16:07发表



回复fengbingchun：https://github.com/nyanp/tiny-cnn/wiki/Data-Format这里有只提到如何转换格式。博主有没有关于直接训练图片的的参考样例呢？

Re: fengbingchun 2016-05-10 21:17发表



回复止的不是战：还没有试过，只弄过简单的MNIST。你可以看看Caffe相关内容应该有介绍

14楼 张骞晖2 2016-05-04 20:00发表



我用这个直接训练 人脸，用于人脸识别率很低。给点指教。

Re: Duke__Lee 2016-05-14 13:17发表



回复张骞晖2：额，就是不知道怎么才能直接用图片进行训练啊。有没有博文推荐阿。谢谢

Re: Duke__Lee 2016-05-08 08:57发表



回复张骞晖2：你好，我们老师要用这个来识别麻将，问一下你怎么设置的人脸的图像的格式啊？麻将图片根本没有数据集，能否指教一二？我的qq：547160794.谢谢

Re: 张骞晖2 2016-05-13 14:40发表



回复Duke__Lee：人脸图像格式不存在问题，一把都可以。麻将数据其实完全自己进行拍摄。

Re: fengbingchun 2016-05-05 08:26发表



回复张骞晖2：没用它测试过人脸识别，你用caffe看看效果如何

13楼 张骞晖2 2016-05-03 22:09发表



解决了。

Re: fengbingchun 2016-05-04 08:21发表



回复张骞晖2：什么原因导致的，如何解决的？

Re: 张骞晖2 2016-05-04 08:59发表



回复fengbingchun：我图片资源没加载。我还遇到其他问题，不过是配置的问题。vs配置tbb opencv很繁琐。

Re: fengbingchun 2016-05-04 09:47发表



回复张骞晖2：没弄过intel TBB，有时间也搞搞 ^_^

关闭

Re: 张骞晖2 2016-05-04 09:53发表



回复fengbingchun：有没有微信什么的，加个好友，有些问题讨论下。

Re: fengbingchun 2016-05-04 13:30发表



回复张骞晖2：不好意思，平时一般不上 :-)

12楼 张骞晖2 2016-05-03 16:15发表



楼主遇到过

input dimension mismatch! 错误吗

Re: fengbingchun 2016-05-03 19:11发表



回复张骞晖2：没遇到过，设个断点跟下看看

11楼 张骞晖2 2016-05-02 20:03发表



我加了个tbb。x64 速度提升了好快

Re: Duke__Lee 2016-05-15 00:30发表



回复张骞晖2：能问一下你是怎么训练人脸的图像的么.....数据格式不是mnist的，要转过去？直接I
要怎么训练啊.....求指教。谢谢

Re: fengbingchun 2016-05-03 08:15发表



回复张骞晖2：赞！

10楼 止的不是战 2016-05-02 16:48发表



用Release模式，确实提速了不少。不过得system("pause");一下，不然运行完，还没来得及查看结果，就自行退出了。

这里有个问题请问下博主，train完会生成model的文件存储下来吗？为何都找不到呢？../MNIST/LeNet-weights中的LeNet-weights是文件夹吧？

Re: fengbingchun 2016-05-02 17:16发表



回复止的不是战：不会自动退出呀。如果直接从github下载的话，生成的model的名字就是LeNet-weights

Re: 止的不是战 2016-05-03 15:20发表



回复fengbingchun：谢谢博主，删掉了自己建的同名文件夹，看见保存的model了。

9楼 张骞晖2 2016-05-01 21:42发表



我打开的是项目中的vc12，不过训练 时间超过了半天。

Re: Duke__Lee 2016-05-30 13:34发表



回复张骞晖2：知道了谢谢。另外大家如果有问题的话，我是图片后缀出问题.....

Re: Duke__Lee 2016-05-30 13:19发表



回复张骞晖2：你好，这个中断异常怎么解决的啊？求解.....

Re: yueryuanmeng 2016-05-02 15:15发表



回复张骞晖2：改为release模式下运行，会快很多

Re: fengbingchun 2016-05-01 22:35发表



回复张骞晖2：怎么会那么长时间呢，你可以开启并行再试试

Re: 张骞晖2 2016-05-02 18:18发表



回复fengbingchun：为啥进行好几次 100%的train呢。我很困惑？第一次结果已经可以了，不知道怎么停止。

Re: fengbingchun 2016-05-03 08:14发表

关闭



回复张睿晖2：那个100%是迭代一次的进度，不是准确率

8楼 [yueryuanmeng](#) 2016-05-01 16:47发表



请问系统运行到最后在layer.h头文件处中断: throw nn_error("input dimension mismatch!" + detail_info); 不知道如何解决？

Re: [Duke__Lee](#) 2016-05-30 13:18发表



回复yueryuanmeng：你好，我也遇到了同样的问题，能问问怎么解决的吗

Re: [fengbingchun](#) 2016-05-01 18:14发表



回复yueryuanmeng：是在train时还是在test时出现的问题？是直接运行<https://github.com/fengbingchun/NN>时报的错误吗？

Re: [yueryuanmeng](#) 2016-05-02 15:13发表



回复fengbingchun：我就直接修改的这篇博客中的test_tiny-cnn.cpp，不过这个问题已经解决，现在已经可以实现对单幅图像进行测试。但是好像程序没办法进行循环的对您所手写的0-9数字进行预测。预测完某单幅图像后系统就停止往下跑了，不知道怎么回事？

Re: [fengbingchun](#) 2016-05-02 17:10发表



回复yueryuanmeng：test只是连续10次的一个for循环，不应该只跑一次呀，你i看，应该比较好定位。

Re: [yueryuanmeng](#) 2016-05-02 19:31发表



回复fengbingchun：看了一下，当预测完第一幅图像数字并显示结果后，就立马提示“xxx.exe已触发一个中断”，因为这个原因导致无法循环的。但是不知道为什么突然触发中断。

Re: [fengbingchun](#) 2016-05-03 08:25发表



回复yueryuanmeng：没遇到过你说的这种情况。debug下设个断点看看能具体定位吗

7楼 [u012507022](#) 2016-04-28 14:08发表



问一个傻瓜式的问题：VS13怎么建一个x64的控制台工程，我这里只显示Win32控制台应用程序

Re: [fengbingchun](#) 2016-04-28 14:57发表



回复u012507022：建完win32后，在解决方案平台上调整就可以了。

6楼 [huxin3420](#) 2016-04-26 20:57发表



博主，我用VS2013+OpenCV2.4.11尝试运行了一下你的DEMO，编译链接都没问题，但是一执行，就会出现弹出对话框：“应用程序无法正常启动（0xc000007b）”，不知博主是否遇到过这个问题。。。还望指点，谢谢。

Re: [fengbingchun](#) 2016-04-27 08:20发表



回复huxin3420：没遇到过，你注释掉main中的函数体，只剩一个return，看是否能够运行，你是否把工程属性中的与opencv相关的配置替换成了你安装的opencv

Re: [huxin3420](#) 2016-05-03 20:18发表



回复fengbingchun：博主，Tiny-CNN比如想改成52个大小写字母的识别，是否只需要在输出层把输出改成52，再用相应数据集训练就行了？

Re: [fengbingchun](#) 2016-05-03 21:19发表



回复huxin3420：没有那么简单，可以先了解下cnn基础：<http://blog.csdn.net/fengbingchun/article/details/50529500>

5楼 [如此_可好](#) 2016-04-07 23:04发表



博主，你训练cifar-10数据库了吗？那个怎么识别啊？预测函数总是出现错误。

Re: [fengbingchun](#) 2016-04-08 08:17发表



回复如此_可好：还没有试，有时间搞搞

4楼 [qq_32328215](#) 2016-04-07 12:34发表

hey、你好，我试验了这个demo，输出在测试过程中的每层的输出图像，就是你在代码中注释的：
// visualize outputs of each layer

关闭



回复fengbingchun：谢谢！Release模式下确实很快

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

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

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