

# hansjorn

## Caffe模型读取

caffe模型最终保存使用过的protobuf形式，将一个已经训练好的caffe模型读取出来，可以参考如下：

1，包含的头文件：

```
1  #include <google/protobuf/io/coded_stream.h>
2  #include <google/protobuf/io/zero_copy_stream_impl.h>
3  #include <google/protobuf/text_format.h>
4
5  #include "caffe/proto/caffe.pb.cc"    //在caffe src/caffe、proto里面，是编译后自动
```

2，读取网络Message：

```
1  bool loadCaffeNet(const std::string& model_list, Message* proto){ //
2
3      using google::protobuf::io::FileInputStream;
4      using google::protobuf::io::ZeroCopyInputStream;
5      using google::protobuf::io::CodedInputStream;
6
7      //Message * proto;
8      std::vector<std::string> model_names;
9      boost::split(model_names, model_list, boost::is_any_of(",") );
10     bool success = false;
11     for (int i = 0; i < model_names.size(); ++i) {
12         std::cout<< "Finetuning from " << model_names[i];
13         const char* filename = model_names[i].c_str();
14         int fd = open(filename, O_RDONLY);
15         if ( fd < 0 ){
16             std::cout << "File not found: " << fd;
```

## 导航

[博客园](#)
[首页](#)
[新随笔](#)
[联系](#)
[订阅 XML](#)
[管理](#)

## 公告

昵称：[Sunite.Liu-CV&&ML](#)

园龄：[4年2个月](#)

粉丝：[2](#)

关注：[2](#)

[+加关注](#)

<	2017年9月						>
日	一	二	三	四	五	六	
27	28	29	30	31	1	2	
3	4	5	6	<a href="#">7</a>	<a href="#">8</a>	9	
10	11	12	13	<a href="#">14</a>	<a href="#">15</a>	16	
<a href="#">17</a>	18	19	20	21	22	23	
24	25	26	27	28	29	30	
1	2	3	4	5	6	7	

## 统计

随笔 - 58

文章 - 1

评论 - 0

引用 - 0

## 搜索

```

17         return -1;
18     }
19
20     ZeroCopyInputStream* raw_input = new FileInputStream(fd);
21     CodedInputStream* coded_input = new CodedInputStream(raw_input);
22     coded_input->SetTotalBytesLimit(INT_MAX, 536870912);
23
24     success = proto->ParseFromCodedStream(coded_input);
25
26     delete coded_input;
27     delete raw_input;
28     close(fd);
29
30     return success;
31 }
32 return success;
33 }

```

3, 参考caffe/proto/caffe.pb.cc 文件, 获取对应的参数

例如读取文件后:

```

1     std::string trained_filename = "lenet_iter_10000.caffemodel";
2
3     caffe::NetParameter net_protobuf;
4
5     if(loadCaffeNet(trained_filename, &net_protobuf)){
6         std::cout<<"load net param success"<<std::endl;
7     }else{
8         std::cout<<"load net param failed"<<std::endl;
9     }

```

获取网络层数:

```

1     int num_source_layers = net_protobuf.layer_size();

```

 谷歌搜索

## 常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

## 我的标签

[DeepLearning\(16\)](#)
[开发语言学习\(11\)](#)
[工具使用\(10\)](#)
[android相关\(9\)](#)
[计算机视觉相关\(5\)](#)
[linux相关\(4\)](#)
[工具使用 vim\(2\)](#)
[视觉跟踪\(2\)](#)
[机器学习日报\(1\)](#)
[工具使用 gdb\(1\)](#)
[更多](#)

## 随笔档案

[2017年9月 \(4\)](#)
[2017年8月 \(5\)](#)
[2017年6月 \(2\)](#)
[2017年3月 \(1\)](#)
[2017年1月 \(2\)](#)
[2016年12月 \(1\)](#)
[2016年11月 \(3\)](#)
[2016年9月 \(1\)](#)
[2016年8月 \(1\)](#)
[2016年7月 \(3\)](#)
[2016年6月 \(1\)](#)
[2016年5月 \(4\)](#)
[2016年4月 \(5\)](#)
[2016年3月 \(1\)](#)

```

1  for(int i=0; i<num_source_layers; ++i){
2      caffe::LayerParameter layer_param = net_protobuf.layer(i);
3      std::cout << layer_param.name() << std::endl;
4      std::cout << layer_param.type() << std::endl;
5
6      int blobsize = layer_param.blobs_size();
7      std::cout << "blobs_size: " << blobsize << std::endl;
8      for(int j=0; j<blobsize; j++){
9          int dataSize = layer_param.blobs(j).data_size();
10
11         if(j==0){
12             std::cout << "  weight data_size: " << dataSize << std::endl;
13             int ind_weight = dataSize;
14             weight = (float*)malloc(ind_weight*sizeof(float));
15             for(int index=0; index<dataSize; index++){
16                 weight[index] = layer_param.blobs(j).data(index);
17             }
18
19             std::cout<<" Convolution->:"<<std::endl;
20             std::cout<<" layer_param.blobs weight_n " << layer_param.blobs(0).shape().dim(0)
21             std::cout<<" layer_param.blobs weight_c " << layer_param.blobs(0).shape().dim(1)
22             std::cout<<" layer_param.blobs weight_h " << layer_param.blobs(0).shape().dim(2)
23             std::cout<<" layer_param.blobs weight_w " << layer_param.blobs(0).shape().dim(3)
24
25             }
26             else if(j==1){
27                 std::cout << "  bias data_size: " << dataSize << std::endl;
28                 int ind_bias = dataSize;
29                 bias = (float*)malloc(ind_bias*sizeof(float));
30                 for(int index=0; index<dataSize; index++){
31                     bias[index] = layer_param.blobs(j).data(index);
32                 }
33             }
34         }
35     }
36 }

```

[2016年2月 \(2\)](#)[2016年1月 \(3\)](#)[2015年12月 \(2\)](#)[2015年11月 \(3\)](#)[2015年10月 \(4\)](#)[2015年9月 \(1\)](#)[2015年8月 \(7\)](#)[2015年7月 \(1\)](#)[2015年3月 \(1\)](#)

## 阅读排行榜

1. DL4J (DeepLearning for java)(2212)
2. Batch Normalization 学习笔记(1263)
3. RNN and LSTM saliency Prediction Scene Label(1181)
4. Caffe模型读取(772)
5. CMakeList相关(485)

## 推荐排行榜

1. Batch Normalization 学习笔记(1)

Powered by:

[博客园](#)

Copyright © Sunite.Liu-CV&amp;&amp;ML

```
8         }  
9  
10        }  
11  
12    }  
13 }
```

以上仅仅是部分代码，需要注意调试！

其中caffe.pb.cc 和caffe.pb.hpp 文件是基于caffe.proto文件生成的。执行过程为：`protoc caffe.proto --cpp_out=.`；将caffe.proto文件，基于目前protobuf的版本生成对应的版本的.cc 和 .hpp文件。

好文要顶

关注我

收藏该文



Sunite.Liu-CV&&ML

关注 - 2

粉丝 - 2

+加关注

0

0

« 上一篇：[AsyncTask机制学习](#)

» 下一篇：[NDK，在JNI层使用AssetManager读取文件](#)

posted on 2015-09-17 13:02 [Sunite.Liu-CV&&ML](#) 阅读(772) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

( 评论功能已被禁用 )

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】腾讯云上实验室 1小时搭建人工智能应用

【推荐】可嵌入您系统的“在线Excel”！SpreadJS 纯前端表格控件

【推荐】阿里云“全民云计算”优惠升级



美团云  
MEITUAN OPEN SERVICES

GPU  
云主机 5折起

M60 原价1660/月 现价830/月 P40 原价3400/月 现价2100/月

查看详情

#### 最新IT新闻:

- Google将Fuchsia系统的内核重命名为Zircon
  - 微博推出“霸王授权协议”背后，是对优质创作者的抢夺
  - 酷炫！耐克推出能与iPhone互动的NBA球衣
  - 人类用了5400年，才做出了《猩球崛起3》里的大猩猩
  - 凡客的荣辱兴衰，是一场雷军也救不回来的“中年危机”
- » 更多新闻...



JIGUANG | 极光

极光统计 多维洞察用户  
增长运营指标

#### 最新知识库文章:

- Google 及其云智慧
  - 做到这一点，你也可以成为优秀的程序员
  - 写给立志做码农的大学生
  - 架构腐化之谜
  - 学会思考，而不只是编程
- » 更多知识库文章...