

# BryantLJ学习的地方

~ 道阻且长，行则将至 ~

[目录视图](#)[摘要视图](#)[RSS 订阅](#)

## 个人资料



遍地流金

[关注](#)[发私信](#)

访问：61803次

积分：787

等级：

排名：千里之外

原创：23篇

转载：1篇

译文：0篇

评论：10条

异步赠书：Kotlin领衔10本好书  
客编程（评论送书）

SDCC 2017之区块链技术实战线上峰会

程序员8月书讯

每周荐书：Ja

## Caffe学习（十）protobuf及caffe.proto解析

2016-12-30 17:01

1927人阅读

评论(0)

收藏

分类：

Caffe学习（9）

版权声明：本文为博主原创文章，未经博主允许不得转载。

[目录\(?\)](#)

[\[+\]](#)

## 使用protobuf的原由

一个好的软件框架应该要有明确的输入和输出，对于CNN网络而言，其主要有两部分组成：网络具体结构和网络的具体优化算法及参数。对于框架的使用者而言，用户只需输入两个描述文件即可得到对该网

## 文章搜索

## 文章分类

- 机器学习 (4)
- Caffe学习 (10)
- 深度学习 (1)
- linux-C编程 (5)
- 生活技能必备 (0)
- python (2)
- github (1)
- tensorflow (1)

## 文章存档

- 2017年03月 (3)
- 2016年12月 (1)
- 2016年11月 (2)
- 2016年10月 (1)
- 2016年08月 (17)

## 阅读排行

- Caffe学习（四）数据层及参数... (15096)
- Caffe学习（一）Caffe+ubuntu1... (10603)
- Caffe学习（六）损失层及其参... (8548)
- 深度学习（一）深度学习的发... (6232)

络的优化结果，这无疑是非常方便的。

caffe框架选择使用谷歌的开源protobuf工具对这两部分进行描述，解析和存储，这一部分为caffe的实现节省了大量的代码。

如前面讲述的目标检测demo，py-faster-rcnn，其主要分为训练和测试两个过程，两个过程的核心文件都是prototxt格式的文本文件。

如训练过程

输入：

（1）slover.prototxt。描述网络训练时的各种参数文件，如训练的策略，学习率的变化率，率等参数

（2）train.prototxt。描述训练网络的网络结构文件。

（3）test.prototxt。描述测试网络的网络结构文件。

输出：

VGG16.caffemodel：保存的训练好的网络参数文件。

## protobuf的使用流程

protobuf工具主要是数据序列化存储和解析。在实际使用的时候主要是作为一个代码自动生成工具来使用，通过生成对所定义的数据结构的标准读写代码，用户可以通过标准的读写接口从文件中进行数据的读取，解析和存储。

[关闭](#)

机器学习（三）Loss函数优化...	(2321)
Caffe学习（十）protobuf及caff...	(1927)
Caffe学习（七）激活函数	(1807)
python-numpy常见ndarray操作	(1758)
Caffe学习（八）运行py-faster-...	(1569)
Caffe学习（二）Mnist手写数字..	(1548)

#### 评论排行

Caffe学习（九）使用Nsight Ec...	(5)
Caffe学习（六）损失层及其参...	(4)
Caffe学习（一）Caffe+ubuntu1...	(1)
机器学习（三）Loss函数优化...	(0)
机器学习（二）广义线性模型...	(0)
Caffe学习（五）视觉层及其设...	(0)
Caffe学习（四）数据层及参数...	(0)
Caffe学习（三）Caffe模型的结...	(0)
tensorflow学习（一）tensorflo...	(0)
Caffe学习（二）Mnist手写数字..	(0)

#### 推荐文章

- \* CSDN日报20170828——《4个方法快速打造你的阅读清单》
- \* 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之区块链技术实战线上峰会
- \* 四大线程池详解

目前proto支持C++，python，java等语言，这里主要演示caffe中使用的C++调用。

主要使用过程为：

- （1）编写XXX.proto文件。该文件里主要定义了各种数据结构及对应的数据类型，如int，string等。
- （2）使用protoc对XXX.proto文件进行编译，生成对应的数据结构文件的读取和写入程序，程序接口都是标准化的。生成的文件一般名为XXX.pb.cc和XXX.pb.h。
- （3）在新程序中使用XXX.pb.c和XXX.pb.h提供的代码。

## 简易caffe.proto编写解析示例

为了后面更加清楚的理解protobuf工具，这里一个简单的caffe.proto为例进行solver.prototxt和train\_nnetotxt的解析

### caffe.proto文件编写：

```
1 syntax = "proto2";
2
3 package caffe; //c++ namespace
4
5 message NetParameter {
6   optional string name = 1; // consider giving the network a name
7   repeated LayerParameter layer = 2; // ID 100 so layers are printed last.
8 }
9
10 message SolverParameter {
11   optional string train_net = 1;
```

[关闭](#)

## 最新评论

Caffe学习 (一) Caffe+ubuntu14.04+Cuda8...

柚子Betty : 你好! 我装完Ubuntu14.04系统后, 也是显示卡顿, 我的是双显卡, Intel 620和NVIDIA...

Caffe学习 (九) 使用Nsight Eclipse调试py...

guotong1988 : mark

Caffe学习 (九) 使用Nsight Eclipse调试py...

遍地流金 : @hawkjk: 你能正常的attach吗? 正常attach需要以sudo方式启动nsight的, 不然...

Caffe学习 (九) 使用Nsight Eclipse调试py...

hawkjk : 你好楼主, 我按照你的步骤已经运行到了联合调试的第二步, 即已经启动第二个调试配置并且找到了Python...

Caffe学习 (九) 使用Nsight Eclipse调试py...

hawkjk : 你好楼主, 我按照你的步骤已经运行到了联合调试的第二步, 即已经启动第二个调试配置并且找到了Python...

Caffe学习 (九) 使用Nsight Eclipse调试py...

hawkjk : 你好楼主, 我按照你的步骤已经运行到了联合调试的第二步, 即已经启动第二个调试配置并且找到了Python...

```

12 optional float base_lr = 2;
13 optional string lr_policy = 3;
14 optional NetParameter net_param = 4;
15 }
16
17 message ParamSpec {
18   optional string name = 1;
19   optional float lr_mult = 3 [default = 1.0];
20   optional float decay_mult = 4 [default = 1.0];
21 }
22 // LayerParameter next available layer-specific ID: 147 (last added: recurrent_param`
23 message LayerParameter {
24   optional string name = 1; // the layer name
25   optional string type = 2; // the layer type
26   repeated string bottom = 3; // the name of each bottom blob
27   repeated string top = 4; // the name of each top blob
28   repeated ParamSpec param = 6;
29   // Layer type-specific parameters.
30   optional ConvolutionParameter convolution_param = 106;
31   optional PythonParameter python_param = 130;
32 }
33
34 message ConvolutionParameter {
35   optional uint32 num_output = 1; // The number of outputs for the layer
36
37   // Pad, kernel size, and stride are all given as a single value for equal
38   // dimensions in all spatial dimensions, or once per spatial dimension.
39   repeated uint32 pad = 3; // The padding size; defaults to 0
40   repeated uint32 kernel_size = 4; // The kernel size
41   repeated uint32 stride = 6; // The stride; defaults to 1
42 }
43
44 message PythonParameter {
45   optional string module = 1;
46   optional string layer = 2;

```

关闭

```
47 // This value is set to the attribute `param_str` of the `PythonLayer` object
48 // in Python before calling the `setup()` method. This could be a number,
49 // string, dictionary in Python dict format, JSON, etc. You may parse this
50 // string in `setup` method and use it in `forward` and `backward`.
51 optional string param_str = 3 [default = ""];
52 }
```

## 编译生成caffe.pb.cc与caffe.pb.h文件

```
1 protoc caffe.proto --cpp_out=.//在当前目录生成cpp文件及头文件
```

## 编写测试文件main.cpp

```
1 #include <fcntl.h>
2 #include <unistd.h>
3
4 #include <iostream>
5 #include <string>
6 #include <google/protobuf/io/coded_stream.h>
7 #include <google/protobuf/io/zero_copy_stream_impl.h>
8 #include <google/protobuf/text_format.h>
9
10 #include "caffe.pb.h"
11 using namespace caffe;
12 using namespace std;
13
14 using google::protobuf::io::FileInputStream;
15 using google::protobuf::Message;
16 bool ReadProtoFromTextFile(const char* filename, Message* proto) {
17     int fd = open(filename, O_RDONLY);
18     FileInputStream* input = new FileInputStream(fd);
```

关闭

```
19  bool success = google::protobuf::TextFormat::Parse(input, proto);
20  delete input;
21  close(fd);
22  return success;
23  }
24
25
26  int main()
27  {
28      SolverParameter SGD;
29
30      if(!ReadProtoFromTextFile("solver.prototxt", &SGD))
31      {
32          cout<<"error opening file"<<endl;
33          return -1;
34      }
35
36      cout<<"hello,world"<<endl;
37      cout<<SGD.train_net()<<endl;
38      cout<<SGD.base_lr()<<endl;
39      cout<<SGD.lr_policy()<<endl;
40
41      NetParameter VGG16;
42      if(!ReadProtoFromTextFile("train.prototxt", &VGG16))
43      {
44          cout<<"error opening file"<<endl;
45          return -1;
46      }
47      cout<<VGG16.name()<<endl;
48      return 0;
49  }
```

关闭

## 编写solver与train网络描述文件

solver.prototxt内容

```
1 train_net: "/home/bryant/cuda-test/train.prototxt"
2 base_lr: 0.001
3 lr_policy: "step"
```

train.prototxt内容：

```
1 name: "VGG_ILSVRC_16_layers"
2 layer {
3   name: 'input-data'
4   type: 'Python'
5   top: 'data'
6   top: 'im_info'
7   top: 'gt_boxes'
8   python_param {
9     module: 'roi_data_layer.layer'
10    layer: 'RoIDataLayer'
11    param_str: "'num_classes': 2"
12  }
13 }
14
15 layer {
16   name: "conv1_1"
17   type: "Convolution"
18   bottom: "data"
19   top: "conv1_1"
20   param {
21     lr_mult: 0
22     decay_mult: 0
23   }
24   param {
25     lr_mult: 0
26     decay_mult: 0
```

关闭

```
27     }  
28     convolution_param {  
29         num_output: 64  
30         pad: 1  
31         kernel_size: 3  
32     }  
33 }
```

## 编译链接，生成main

```
1 g++ caffe.pb.cc main.cpp -o main -lprotobuf
```

## 运行结果

```
1 bryant@bryant:~/cuda-test/src$ ./main  
2 hello,world  
3 /home/bryant/cuda-test/train.prototxt  
4 0.001  
5 step  
6 VGG_ILSVRC_16_layers  
7 bryant@bryant:~/cuda-test/src$
```

顶

2

踩

0

- [上一篇](#) [github新建项目入门](#)



- [下一篇](#)    [机器学习（四）从信息论交叉熵的角度看softmax/逻辑回归损失](#)

#### 相关文章推荐

- VGG-16 prototxt
- 30天系统掌握机器学习--唐宇迪
- 如何进行finetune
- 【免费】XGBoost模型原理及其表现--卿来云
- DeepID Generation I
- 全能项目经理训练视频
- VGG 16 prototxt
- Python网络爬虫快速入门指导
- Faster-RCNN + OHEM + vgg16 + prototxt文件及相关..
- Android实战基础知识
- caffe+python 使用训练好的VGG16模型 对 单张图片...
- 机器学习需要的掌握的数学知识汇总
- VGG-16 prototxt
- 使用Caffe尝试DeepID
- Caffe源码中caffe.proto文件分析
- caffe 有关prototxt文件的设置解读

#### 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

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](mailto: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