

linger(心怀梦想，活在当下)

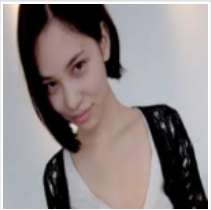
机器学习，深度学习，数据挖掘，推荐系统，分布式算法

目录视图

摘要视图

RSS 订阅

个人资料



lingerlanlan

访问：219484次

积分：3991

等级：BLOG 5

排名：第3738名

原创：156篇

转载：0篇

译文：2篇

评论：260条

文章搜索

文章分类

深度学习（deep learning） (28)

机器学习 (16)

cuda(GPU programming) (20)

文本挖掘 (5)

c/c++ (15)

dota外挂 (5)

hack programming (6)

web2.0 (5)

工具源码 (7)

语言学习 (22)

设计模式 (2)

读书笔记 (1)

翻译 (1)

足球大数据 (2)

大杂烩 (18)

Hadoop (12)

Spark (2)

sklearn (1)

文章存档

2015年08月 (1)

caffe源码分析--Blob类代码研究

分类：深度学习（deep learning）

2014-04-26 11:42

4386人阅读

评论(7)

收藏

举报

cuda

caffe

深度学习

神经网络

作者:linger

转自须注明转自：http://lingerlanlan/article/details/24379689

数据成员

shared_ptr<SyncedMemory>data_;//data数据，指向SyncedMemory的智能指针

shared_ptr<SyncedMemory>diff_;//表示“差”，用于更新data_

intnum_;

intchannels_;

intheight_;

intwidth_;

intcount_;

构造函数

Blob():num_(0),channels_(0),height_(0),width_(0),count_(0),data_(),diff_() {}

功能：简单的初始化

explicitBlob(constintnum,constintchannels,constintheight,constintwidth);

功能：调用Reshape函数，初始化数据成员

template<typenameDtype>

Blob<Dtype>::Blob(constintnum,constintchannels,constintheight,

constintwidth) {

Reshape(num,channels,height,width);

}

http://blog.csdn.net/lingerlanlan/article/details/24379689

1/8

2015年07月 (3)
2015年06月 (3)
2015年05月 (3)
2015年04月 (8)

展开

最新评论

总结一下用caffe跑图片数据的研
liangzhituzi: @zzq1989_:可能是那
两个文件路径的问题，可以看
train_prototxt里面的路径

deep learning实践经验总结
查志强: 问下，怎样判断“错误”的
标签？

神经网络：caffe特征可视化的代
fqss0436: 博主，您好，谢谢您
分享代码。在调试您的代码时，
程序中断于175行
caffe_test_net.For...

我所写的CNN框架 VS caffe
gzp95: 楼主，求问一下您写的代
码的速度和caffe的速度有多大的
差距。因为最近在实现word2vec
的cud...

总结一下用caffe跑图片数据的研
依然_范佩西11: 训练完的模型，
如是调用呢，能说一下测试单
张图像或者批量图像的流程么

Dota全图那些事儿
女主、女主:。。。单机理论效
果，实际不好用啊。。。。支
持一下~不错的

caffe源码修改：抽取任意一张图
wwdzhxjwcnmd: 想请教一下博
主，caffe网络中batch_size和
crop_size这两个参数的含义是什
么？哪一...

caffe源码分析--data_layer.cpp
沧海1梦: 请问caffe中如何修改输
入和裁剪尺寸，因为我的图像大
小是48的，想通过修改alexnet来
训练，还...

caffe卷积神经网络框架安装
yang123jx: 我也遇到
relu_layer.cu:29 check failed
error == cudaSuc...

caffe卷积神经网络框架安装
yang123jx: 我也遇到
relu_layer.cu:29 check failed
error == cudaSuc...

阅读排行

总结一下用caffe跑图片数据 (7192)
word2vector学习笔记 (6942)
caffe神经网络框架的辅助 (6147)
caffe源码修改：抽取任意 (5905)
caffe卷积神经网络框架安装 (5550)
caffe源码分析--data_layer (5374)
神经网络：caffe特征可视化 (4679)
word2vec源码解析之word (4510)
caffe源码分析--Blob类代码 (4386)
deep learning实践经验总结 (4225)

推荐文章

析构造函数

virtual ~Blob() {}

功能：啥都没做？

```
void Reshape(const int num, const int height,
              const int width, const int channels);
```

功能：初始化数据成员，智能指针指向SyncedMemory对象。此时SyncedMemory对象其实并没有为自己的“数据”申请内存，只是自己“数据”的大小（size）。

```
template<typename Dtype>
```

```
void Blob<Dtype>::Reshape(const int num, const int channels, const int height,
```

```
const int width) {
```

```
    CHECK_GE(num, 0);
```

```
    CHECK_GE(channels, 0);
```

```
    CHECK_GE(height, 0);
```

```
    CHECK_GE(width, 0);
```

```
    num_ = num;
```

```
    channels_ = channels;
```

```
    height_ = height;
```

```
    width_ = width;
```

```
    count_ = num * channels_ * height_ * width_;
```

```
    if (count_ > 0) {
```

```
        data_.reset(new SyncedMemory(count_ * sizeof(Dtype)));
```

```
        diff_.reset(new SyncedMemory(count_ * sizeof(Dtype)));
```

```
    } else {
```

```
        data_.reset(reinterpret_cast<SyncedMemory*>(NULL));
```

```
        diff_.reset(reinterpret_cast<SyncedMemory*>(NULL));
```

```
    }
```

```
}
```

成员访问函数

功能：就是返回一些成员变量

```
inline int num() const { return num_; }

inline int channels() const { return channels_; }

inline int height() const { return height_; }

inline int width() const { return width_; }

inline int count() const { return count_; }

inline int offset(const int n, const int c = 0, const int h = 0, const int w = 0) const {
    return ((n * channels_ + c) * height_ + h) * width_ + w;
}

// 计算偏移量，因为数据在内存是一维数组形式的，所以需要计算偏移量来访问
}
```

“数据”指针返回函数

功能：其实这些函数就是调用SyncedMemory的函数，来返回数据的指针

```
const Dtype* cpu_data() const;

const Dtype* gpu_data() const;

const Dtype* cpu_diff() const;

const Dtype* gpu_diff() const;

Dtype* mutable_cpu_data();

Dtype* mutable_gpu_data();

Dtype* mutable_cpu_diff();

Dtype* mutable_gpu_diff();

inline Dtype data_at(const int n, const int c, const int h,
                    const int w) const {
    // 从cpu访问数据data
    return *(cpu_data() + offset(n, c, h, w));
}

inline Dtype diff_at(const int n, const int c, const int h,
                    const int w) const {
    // 从cpu访问数据diff
    return *(cpu_diff() + offset(n, c, h, w));
}
```

函数 void Update()

功能：更新data_的数据，就是减去diff_的数据。

```
template<typename Dtype>
void Blob<Dtype>::Update() {
    //We will perform update based on where the data is located.
    switch(data_>head()) {
        case SyncedMemory::HEAD_AT_CPU:
            //perform computation on CPU
            caffe_axpy<Dtype>(count_, Dtype(-1),
                reinterpret_cast<const Dtype*>(diff_>cpu_data()),
                reinterpret_cast<Dtype*>(data_>mutable_cpu_data()));
            //在math_functions.cpp可以找到该函数的实现，其实这函数也是封装了mkl的函数。这里调用
            //是为了实现了两个向量的减法。

            break;

        case SyncedMemory::HEAD_AT_GPU:
        case SyncedMemory::SYNCED:
            //perform computation on GPU
            caffe_gpu_axpy<Dtype>(count_, Dtype(-1),
                reinterpret_cast<const Dtype*>(diff_>gpu_data()),
                reinterpret_cast<Dtype*>(data_>mutable_gpu_data()));
            //在math_functions.cpp可以找到该函数的实现，其实这函数也是封装了cublas的函数。这
            //里调用是为了实现了两个向量的减法。

            break;

        default:
            LOG(FATAL) << "Syncedmemnot initialized.";
    }
}
```

函数void CopyFrom(const Blob<Dtype>&source, bool copy_diff = false, bool reshape = false);

功能：从source拷贝数据。copy_diff作为标志来区分是拷贝data还是拷贝diff。

```
template<typename Dtype>
void Blob<Dtype>::CopyFrom(const Blob&source, bool copy_diff, bool reshape) {
    if(num_!= source.num() || channels_!= source.channels() ||
```

```

height_!= source.height() || width_!= source.width()) {

    if(reshape) {

        Reshape(source.num(), source.channels(), source.height(), source.width());

    }else{

        LOG(FATAL)<<"Trying to copy blobs of different sizes.";

    }

}

switch(Caffe::mode()) {

case Caffe::GPU:

    if(copy_diff) {

        CUDA_CHECK(cudaMemcpy(diff_>mutable_gpu_data(), source.gpu_diff(),
            sizeof(Dtype)*count_, cudaMemcpyDeviceToDevice));

    }else{

        CUDA_CHECK(cudaMemcpy(data_>mutable_gpu_data(), source.gpu_data(),
            sizeof(Dtype)*count_, cudaMemcpyDeviceToDevice));

    }

    break;

case Caffe::CPU:

    if(copy_diff) {

        memcpy(diff_>mutable_cpu_data(), source.cpu_diff(),
            sizeof(Dtype)*count_);

    }else{

        memcpy(data_>mutable_cpu_data(), source.cpu_data(),
            sizeof(Dtype)*count_);

    }

    break;

default:

    LOG(FATAL)<<"Unknown caffe mode.";

}

}

```

函数 `void FromProto(const BlobProto&proto);`

功能：从proto读数据进来，其实就是反序列化

```
template<typename Dtype>
```

```
void Blob<Dtype>::FromProto(const BlobProto&proto) {  
  
  Reshape(proto.num(), proto.channels(), proto.height(), proto.width());  
  
  //copy data  
  
  Dtype*data_vec = mutable_cpu_data();  
  
  for(int i = 0; i < count_++;i) {  
  
    data_vec[i]=proto.data(i);  
  
  }  
  
  if(proto.diff_size() > 0) {  
  
    Dtype*diff_vec = mutable_cpu_diff();  
  
    for(int i = 0; i < count_++;i) {  
  
      diff_vec[i]=proto.diff(i);  
  
    }  
  
  }  
  
}
```

函数void ToProto(BlobProto*proto, bool write_diff = false) const;

功能: 序列化到proto保存

```
template<typename Dtype>  
  
void Blob<Dtype>::ToProto(BlobProto*proto, bool write_diff) const {  
  
  proto->set_num(num_);  
  
  proto->set_channels(channels_);  
  
  proto->set_height(height_);  
  
  proto->set_width(width_);  
  
  proto->clear_data();  
  
  proto->clear_diff();  
  
  const Dtype*data_vec = cpu_data();  
  
  for(int i = 0; i < count_++;i) {  
  
    proto->add_data(data_vec[i]);  
  
  }  
  
  if(write_diff) {  
  
    const Dtype*diff_vec = cpu_diff();  
  
    for(int i = 0; i < count_++;i) {  
  
      proto->add_diff(diff_vec[i]);  
  
    }  
  
  }  
  
}
```

}

版权声明：本文为博主原创文章，未经博主允许不得转载。

上一篇 给cuda核函数传递二维数组的一种方法
下一篇 浮点运算误差

顶 0 踩 0

主题推荐 color 代码 源码 class 数据

猜你在找

- 韦东山嵌入式Linux第一期视频
- (帆软)地产行业商业智能解决方案
- 3D游戏开发基础
- C语言及程序设计初步
- Java之路
- 机器学习Machine Learning&深度学习Deep Learning资
- 百来篇关于机器学习和深度学习的资料
- 百来篇关于机器学习和深度学习的资料
- 机器学习Machine Learning&深度学习Deep Learning资
- 机器学习Machine Learning&深度学习Deep Learning资

准备好了么？跳吧！ 更多职位尽在 CSDN JOB

数据分析工程师	我要跳槽	高级商业数据分析师	我要跳槽
腾讯科技（深圳）有限公司	20-40K/月	上海点我吧信息技术有限公司	20-40K/月
数据分析师---SQL	我要跳槽	数据挖掘 / 数据分析工程师	我要跳槽
欧唯特信息服务有限公司	6-9K/月	上海智子信息科技有限公司	8-16K/月

查看评论

3楼 diang2 2015-04-07 10:01发表

十分感谢博主分享,想问问如CHECK_GE,CHEK_LE,CHEK,LOG之类的是宏定义吗,它们定义在哪个文件下?一直没搞懂它们是做什么用的.
再次感谢博主,博文很有帮助,谢谢~~

2楼 zhangvvok 2014-12-19 11:39发表

博主能不能写个宏观一点的caffe源码分析，总体框架框架还是不了解。

Re: lingerlanlan 2015-01-01 15:52发表

回复zhangvvok： 没时间研究caffe了

Re: whjxnyzh 2015-02-05 16:51发表

回复lingerlanlan： 楼主有时间写个整体架构吧。。实在是看晕菜了

1楼 司令 2014-07-04 22:23发表

博主好！
我查了caffe_gpu_axpy这个函数里面封装了cublasSaxpy，但是这个函数在cublas中的说明是 y = alpha * x + y 就是说caffe_gpu_axpy其实是两个向量的加法，这么说caffe计算出来的diff是负值以实现相减？

Re: lingerlanlan 2014-07-08 16:05发表

回复司令： 请注意有个实参是： Dtype(-1)。

Re: 司令2014-07-11 01:20发表

C

回复lingerlanlan：啊哈，之前没有注意到啊，多谢提醒！

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