个人资料

lingerlanlan

访问： 219485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

caffe源码分析--data_layer.cpp

分类： 深度学习（deep learning） 2014-05-31 21:24 5375人阅读 评论(5) 收藏 举报

神经网络 caffe 深度学习 deep learning

dataLayer作为整个网络|

数据从leveldb中取。leve |过图片转换过来的。

网络建立的时候，

datalayer主要是负责设置一些参数，比如batchsize，channels，height，width等。

这次会通过读leveldb一个数据块来获取这些信息。

然后启动一个线程来预先从leveldb拉取一批数据，这些数据是图像数据和图像标签。

正向传播的时候，

datalayer就把预先拉取好数据拷贝到指定的cpu或者gpu的内存。

然后启动新线程再预先拉取数据，这些数据留到下一次正向传播使用。

```cpp
// Copyright 2013 Yangqing Jia

#include <stdint.h>
#include <leveldb/db.h>
#include <pthread.h>

#include <string>
#include <vector>

#include "caffe/layer.hpp"
#include "caffe/util/io.hpp"
#include "caffe/vision_layers.hpp"

using std::string;

namespace caffe {

template <typename Dtype>
void* DataLayerPrefetch(void* layer_pointer) {
  CHECK(layer_pointer);
  DataLayer<Dtype>* layer = reinterpret_cast<DataLayer<Dtype>*>(layer_pointer);
  CHECK(layer);
  Datum datum;
  CHECK(layer->prefetch_data_);
  Dtype* top_data = layer->prefetch_data_->mutable_cpu_data();//数据
  Dtype* top_label = layer->prefetch_label_->mutable_cpu_data();//标签
  const Dtype scale = layer->layer_param_.scale();
  const int batchsize = layer->layer_param_.batchsize();
  const int cropsize = layer->layer_param_.cropsize();
```

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，能说下能么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
wwdzhtxnjwcnmd: 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 (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 (5550)
caffe源码分析--data_lay (5374)
神经网络：caffe特征可初 (4679)
word2vec源码解析之wo (4510)
caffe源码分析--Blob类代 (4386)
deep learning实践经验总 (4225)

推荐文章

```
30.   const bool mirror = layer->layer_param_.mirror();
31.
32.   if (mirror && cropsize == 0) {//当前实现需要同时设置mirror和cropsize
33.     LOG(FATAL) << "Current implementation requires mirror and cropsize to be "
34.         << "set at the same time.";
35.   }
36.   // datum scales
37.   const int channels = layer->datum_channels_;
38.   const int height = layer->datum_height_;
39.   const int width = layer->datum_width_;
40.   const int size = layer->datum_size_;
41.   const Dtype* mean = layer->data_mean_.cpu_data();
42.   for (int itemid = 0; itemid < batchsize; ++itemid) {//每一批数据的数量是batchsize，一个循环拉取一
张？
43.     // get a blob
44.     CHECK(layer->iter_);
45.     CHECK(layer->iter_->Valid());
46.     datum.ParseFromString(layer->iter_->value().ToString());//利用迭代器拉取下一批数据
47.     const string& data = datum.data();
48.     if (cropsize) {//如果需要裁剪
49.       CHECK(data.size()) << "Image cropping only support uint8 data";
50.       int h_off, w_off;
51.       // We only do random crop when we do training.
52.       //只是在训练阶段做随机裁剪
53.       if (Caffe::phase() == Caffe::TRAIN) {
54.         // NOLINT_NEXT_LINE(runtime/threadsafe_fn)
55.         h_off = rand() % (height - cropsize);
56.         // NOLINT_NEXT_LINE(runtime/threadsafe_fn)
57.         w_off = rand() % (width - cropsize);
58.       } else {//测试阶段固定裁剪
59.         h_off = (height - cropsize) / 2;
60.         w_off = (width - cropsize) / 2;
61.       }
62.       // NOLINT_NEXT_LINE(runtime/threadsafe_fn)
63.       //怎么感觉下面两种情况的代码是一样的？
64.       if (mirror && rand() % 2) {
65.         // Copy mirrored version
66.         for (int c = 0; c < channels; ++c) {
67.           for (int h = 0; h < cropsize; ++h) {
68.             for (int w = 0; w < cropsize; ++w) {
69.               top_data[((itemid * channels + c) * cropsize + h) * cropsize
70.                        + cropsize - 1 - w] =
71.                   (static_cast<Dtype>(
72.                       (uint8_t)data[(c * height + h + h_off) * width
73.                                     + w + w_off])
74.                     - mean[(c * height + h + h_off) * width + w + w_off])
75.                   * scale;
76.             }
77.           }
78.         }
79.       } else {
80.         // Normal copy
81.         for (int c = 0; c < channels; ++c) {
82.           for (int h = 0; h < cropsize; ++h) {
83.             for (int w = 0; w < cropsize; ++w) {
84.               top_data[((itemid * channels + c) * cropsize + h) * cropsize + w]
85.                   = (static_cast<Dtype>(
86.                       (uint8_t)data[(c * height + h + h_off) * width
87.                                     + w + w_off])
88.                     - mean[(c * height + h + h_off) * width + w + w_off])
89.                   * scale;
90.             }
91.           }
92.         }
93.       }
94.     } else {//如果不需要裁剪
95.       // we will prefer to use data() first, and then try float_data()
96.       //我们优先考虑data()，然后float_data()
97.       if (data.size()) {
98.         for (int j = 0; j < size; ++j) {
99.           top_data[itemid * size + j] =
100.              (static_cast<Dtype>((uint8_t)data[j]) - mean[j]) * scale;
101.        }
102.      } else {
103.        for (int j = 0; j < size; ++j) {
104.          top_data[itemid * size + j] =
105.              (datum.float_data(j) - mean[j]) * scale;
106.        }
107.      }
```

```
    }

    top_label[itemid] = datum.label();
    // go to the next iter
    layer->iter_->Next();
    if (!layer->iter_->Valid()) {
      // We have reached the end. Restart from the first.
      DLOG(INFO) << "Restarting data prefetching from start.";
      layer->iter_->SeekToFirst();
    }
  }

  return reinterpret_cast<void*>(NULL);
}

template <typename Dtype>
DataLayer<Dtype>::~DataLayer<Dtype>() {
  // Finally, join the thread
  CHECK(!pthread_join(thread_, NULL)) << "Pthread joining failed.";
}

template <typename Dtype>
void DataLayer<Dtype>::SetUp(const vector<Blob<Dtype>*>& bottom,
      vector<Blob<Dtype>*>* top) {
  CHECK_EQ(bottom.size(), 0) << "Data Layer takes no input blobs.";
  CHECK_EQ(top->size(), 2) << "Data Layer takes two blobs as output.";
  // Initialize the leveldb
  leveldb::DB* db_temp;
  leveldb::Options options;
  options.create_if_missing = false;
  options.max_open_files = 100;
  LOG(INFO) << "Opening leveldb " << this->layer_param_.source();
  leveldb::Status status = leveldb::DB::Open(
      options, this->layer_param_.source(), &db_temp);
  CHECK(status.ok()) << "Failed to open leveldb "
      << this->layer_param_.source() << std::endl << status.ToString();
  db_.reset(db_temp);
  iter_.reset(db_->NewIterator(leveldb::ReadOptions()));//通过迭代器来操纵leveldb
  iter_->SeekToFirst();
  // Check if we would need to randomly skip a few data points
  //是否要随机跳过一些数据
  if (this->layer_param_.rand_skip()) {
    // NOLINT_NEXT_LINE(runtime/threadsafe_fn)
    unsigned int skip = rand() % this->layer_param_.rand_skip();
    LOG(INFO) << "Skipping first " << skip << " data points.";
    while (skip-- > 0) {//循环次数
      iter_->Next();
      if (!iter_->Valid()) {
        iter_->SeekToFirst();
      }
    }
  }
  // Read a data point, and use it to initialize the top blob.
  //读取一个数据点，用来初始化topblob。所谓初始化，只要是指reshape。
  //可以观察到下面iter_调用调用next。所以这次读取只是用来读取出来channels等参数的，不作处理。
  Datum datum;
  datum.ParseFromString(iter_->value().ToString());//利用迭代器读取第一个数据点
  // image图像数据
  int cropsize = this->layer_param_.cropsize();//裁剪大小
  if (cropsize > 0) {//需要裁剪
    (*top)[0]->Reshape(
        this->layer_param_.batchsize(), datum.channels(), cropsize, cropsize);
    prefetch_data_.reset(new Blob<Dtype>(
        this->layer_param_.batchsize(), datum.channels(), cropsize, cropsize));
  } else {//不需要裁剪
    (*top)[0]->Reshape(
        this->layer_param_.batchsize(), datum.channels(), datum.height(),
        datum.width());
    prefetch_data_.reset(new Blob<Dtype>(
        this->layer_param_.batchsize(), datum.channels(), datum.height(),
        datum.width()));
  }
  LOG(INFO) << "output data size: " << (*top)[0]->num() << ","
      << (*top)[0]->channels() << "," << (*top)[0]->height() << ","
      << (*top)[0]->width();
  // label标签数据
  (*top)[1]->Reshape(this->layer_param_.batchsize(), 1, 1, 1);
  prefetch_label_.reset(
      new Blob<Dtype>(this->layer_param_.batchsize(), 1, 1, 1));
```

```
  // datum size
  datum_channels_ = datum.channels();
  datum_height_ = datum.height();
  datum_width_ = datum.width();
  datum_size_ = datum.channels() * datum.height() * datum.width();
  CHECK_GT(datum_height_, cropsize);
  CHECK_GT(datum_width_, cropsize);
  // check if we want to have mean是否要减去均值
  if (this->layer_param_.has_meanfile()) {
    BlobProto blob_proto;
    LOG(INFO) << "Loading mean file from" << this->layer_param_.meanfile();
    ReadProtoFromBinaryFile(this->layer_param_.meanfile().c_str(), &blob_proto);
    data_mean_.FromProto(blob_proto);
    CHECK_EQ(data_mean_.num(), 1);
    CHECK_EQ(data_mean_.channels(), datum_channels_);
    CHECK_EQ(data_mean_.height(), datum_height_);
    CHECK_EQ(data_mean_.width(), datum_width_);
  } else {
    // Simply initialize an all-empty mean.
    data_mean_.Reshape(1, datum_channels_, datum_height_, datum_width_);
  }
  // Now, start the prefetch thread. Before calling prefetch, we make two
  // cpu_data calls so that the prefetch thread does not accidentally make
  // simultaneous cudaMalloc calls when the main thread is running. In some
  // GPUs this seems to cause failures if we do not so.
  prefetch_data_->mutable_cpu_data();
  prefetch_label_->mutable_cpu_data();
  data_mean_.cpu_data();
  DLOG(INFO) << "Initializing prefetch";
  CHECK(!pthread_create(&thread_, NULL, DataLayerPrefetch<Dtype>,
      reinterpret_cast<void*>(this))) << "Pthread execution failed.";
  DLOG(INFO) << "Prefetch initialized.";
}

template <typename Dtype>
void DataLayer<Dtype>::Forward_cpu(const vector<Blob<Dtype>*>& bottom,
      vector<Blob<Dtype>*>* top) {
  // First, join the thread 等待线程结束
  CHECK(!pthread_join(thread_, NULL)) << "Pthread joining failed.";
  // Copy the data拷贝数据到top，即该层的输出
  memcpy((*top)[0]->mutable_cpu_data(), prefetch_data_->cpu_data(),
      sizeof(Dtype) * prefetch_data_->count());
  memcpy((*top)[1]->mutable_cpu_data(), prefetch_label_->cpu_data(),
      sizeof(Dtype) * prefetch_label_->count());
  // Start a new prefetch thread启动新线程拉取下一批数据
  CHECK(!pthread_create(&thread_, NULL, DataLayerPrefetch<Dtype>,
      reinterpret_cast<void*>(this))) << "Pthread execution failed.";
}

// The backward operations are dummy - they do not carry any computation.
template <typename Dtype>
Dtype DataLayer<Dtype>::Backward_cpu(const vector<Blob<Dtype>*>& top,
      const bool propagate_down, vector<Blob<Dtype>*>* bottom) {
  return Dtype(0.);
}

INSTANTIATE_CLASS(DataLayer);

}  // namespace caffe
```

本文作者：linger

本文链接：http://blog.csdn.net/lingerlanlan/article/details/27348265

上一篇 全局内存性能测试
下一篇 使用squid架设自己的代理服务器

查看评论

4楼 沧海1梦 2015-07-08 10:08发表

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，还没找到修改输入的地方

3楼 进击的城管 2015-01-29 11:10发表

ou no```caffe更新了 我的心好痛。。。代码变了

2楼 houqiqi 2014-11-09 23:55发表

博主你好，我想问下你是用什么来查看caffe的源代码的，我用gedit感觉效率太低了，查看一个调用函数都要自己去猜这个函数可能在什么位置，大概在哪个文件下~~~能将您的配置说一下吗

Re: lingerlanlan 2014-11-13 00:34发表

回复houqiqi：我在Ubuntu装了cuda之后，会有一个叫nsight的ide，其实是Eclipse的扩展版本。

1楼 barbara2008 2014-07-15 10:24发表

top_data[((itemid * channels + c) * cropsize + h) * cropsize + cropsize - 1 - w]和84行的index不一样