

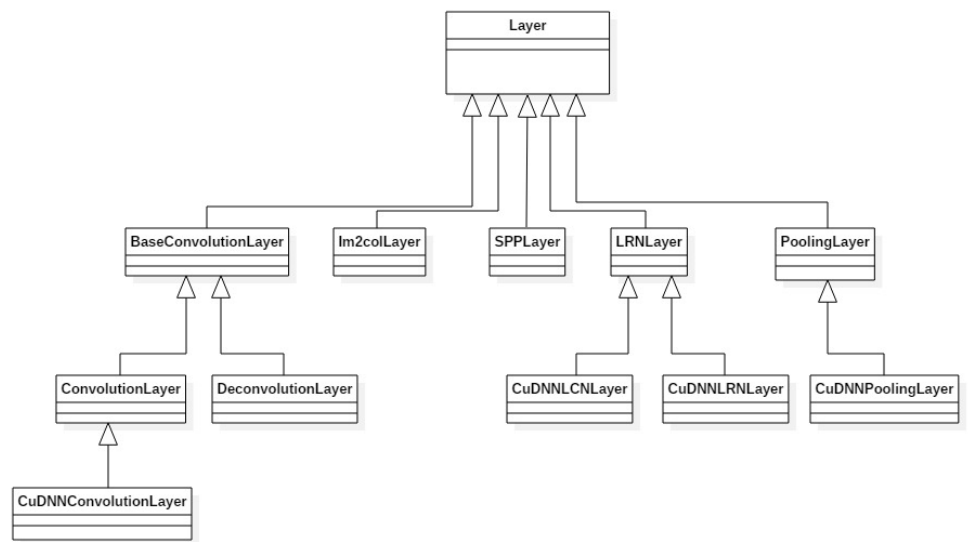
Caffe源码解析5: Conv_Layer

转载请注明出处，楼燚(yì)航的blog，<http://home.cnblogs.com/louyihang-loves-baiyan/>

Vision_layer里面主要是包括了一些关于一些视觉上的操作，比如卷积、反卷积、池化等等。这里的类跟data layer一样好很多种继承关系。主要包括了这几个类，其中CuDNN分别是CUDA版本，这里先不讨论，在这里先讨论ConvolutionLayer

- BaseConvolutionLayer
- ConvolutionLayer
- DeconvolutionLayer
- CuDNNConvolutionLayer
- Im2colLayer
- LRNLayer
- CuDNNLRNLayer
- CuDNNLCNLayer
- PoolingLayer
- CuDNNPoolingLayer
- SPPLayer

这里我画了一个类图，将关系梳理了一下：



BaseConvolutionLayer

其继承自Layer，是一个卷积以及反卷积操作的基类，首先我们来看BaseConvolutionLayer的LayerSetUp函数

```
void BaseConvolutionLayer<Dtype>::LayerSetUp(const vector<Blob<Dtype>*>& bottom,
const vector<Blob<Dtype>*>& top)
//首先这里主要是在配置卷积kernel 的size,padding,stride以及inputs
ConvolutionParameter conv_param = this->layer_param_.convolution_param();
force_nd_im2col_ = conv_param.force_nd_im2col();
```

公告

昵称：楼燚航的blog
园龄：1年5个月
粉丝：60
关注：1
+加关注

2016年4月						
日	一	二	三	四	五	六
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

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
更多链接

随笔档案

- 2016年3月 (2)
- 2016年2月 (2)
- 2016年1月 (7)
- 2015年12月 (2)
- 2015年11月 (5)
- 2015年10月 (5)
- 2015年9月 (1)
- 2015年8月 (2)
- 2015年7月 (1)
- 2015年6月 (1)
- 2015年5月 (1)
- 2015年4月 (3)
- 2014年12月 (1)

最新评论

1. Re:opencv 3.0 DPM Cascade 检测 (附带TBB和openMP加速)
好了，问题解决了，我用的opencv版本是2 412，所以导致那么慢，换成310就没有这问题了

```

channel_axis_ = bottom[0]->CanonicalAxisIndex(conv_param.axis());
const int first_spatial_axis = channel_axis_ + 1;
const int num_axes = bottom[0]->num_axes();
num_spatial_axes_ = num_axes - first_spatial_axis;
CHECK_GE(num_spatial_axes_, 0);
vector<int> bottom_dim_blob_shape(1, num_spatial_axes_ + 1);
vector<int> spatial_dim_blob_shape(1, std::max(num_spatial_axes_, 1));
// 设置kernel的dimensions
kernel_shape_.Reshape(spatial_dim_blob_shape);
int* kernel_shape_data = kernel_shape_.mutable_cpu_data();

```

接着是设置相应的stride dimensions，对于2D，设置在h和w方向上的stride，代码太长列出简要的

```

pad_.Reshape(spatial_dim_blob_shape);
int* pad_data = pad_.mutable_cpu_data();
pad_data[0] = conv_param.pad_h();
pad_data[1] = conv_param.pad_w();
.....一堆if else判断

```

对于kernel的pad也做相应设置

```

pad_.Reshape(spatial_dim_blob_shape);
int* pad_data = pad_.mutable_cpu_data();
pad_data[0] = conv_param.pad_h();
pad_data[1] = conv_param.pad_w();

```

接下来是对widhts 和bias左设置和填充，其中blob[0]里面存放的是filter weights,而blob[1]里面存放的是biases，当然biases是可选的，也可以没有

```

//设置相应的shape，并检查
vector<int> weight_shape(2);
weight_shape[0] = conv_out_channels_;
weight_shape[1] = conv_in_channels_ / group_;

bias_term_ = this->layer_param_.convolution_param().bias_term();
vector<int> bias_shape(bias_term_, num_output_);

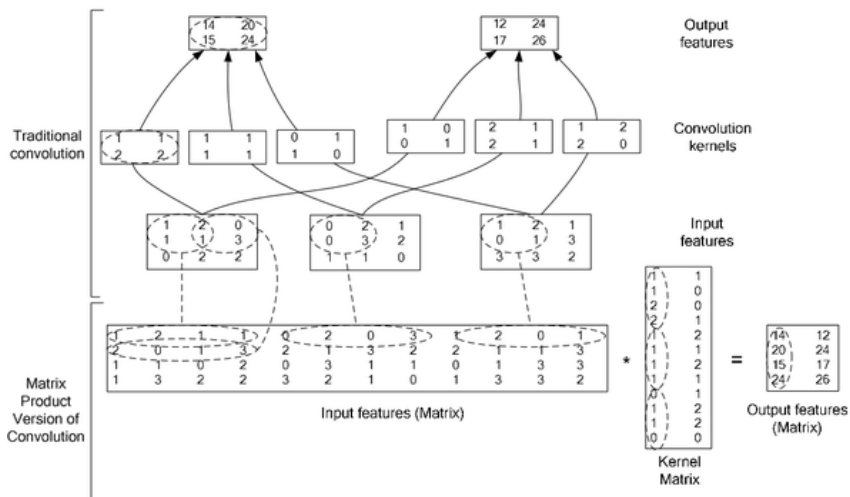
//填充权重
this->blobs_[0].reset(new Blob<Dtype>(weight_shape));
shared_ptr<Filler<Dtype> > weight_filler(GetFiller<Dtype>(
    this->layer_param_.convolution_param().weight_filler()));
weight_filler->Fill(this->blobs_[0].get());
//填充偏置项
if (bias_term_) {
    this->blobs_[1].reset(new Blob<Dtype>(bias_shape));
    shared_ptr<Filler<Dtype> > bias_filler(GetFiller<Dtype>(
        this->layer_param_.convolution_param().bias_filler()));
    bias_filler->Fill(this->blobs_[1].get());
}

```

ConvolutionLayer

ConvolutionLayer继承了BaseConvolutionLayer，主要作用就是将一副image做卷积操作，使用学到的filter的参数和biaes。同时在Caffe里面，卷积操作做了优化，变成了一个矩阵相乘的操作。其中有两个比较主要的函数是im2col以及col2im。

图中上半部分是一个传统卷积，下图是一个矩阵相乘的版本



下图是在一个卷积层中将卷积操作展开的具体操作过程，他里面按照卷积核的大小取数据然后展开，在同一张图

--gaosi123

2. Re:opencv 3.0 DPM Cascade 检测（附带TBB和openMP加速）

BTW, 我电脑是i7 4790k + 16GB内存，所以硬件设备应该不会是限制。不知道问题出在哪里

--gaosi123

3. Re:opencv 3.0 DPM Cascade 检测（附带TBB和openMP加速）

如果可以，欢迎留个email

--gaosi123

4. Re:opencv 3.0 DPM Cascade 检测（附带TBB和openMP加速）

你好，我也是直接把DPM代码拷贝到工程里，但是想你这样直接拷进去不会报错吗？我直接拷贝进去按照你的来，报错信息如下：
 Error 4 error C2039: 'dpm': is not a memb.....

--gaosi123

5. Re:Fast RCNN 训练自己数据集 (2修改数据读取接口)

@楼蒺航的blog楼主你好！我在EdgeBoxe s提取OP的时候也是直接用的默认参数，并且将坐标[x y w h]变成了左上右下的形式，但是发现检测车的时候效果并没有Selective Search好，.....

—JustJay

阅读排行榜

1. Fast RCNN 训练自己数据集 (2修改数据读取接口)(3946)
2. Fast RCNN 训练自己数据集 (1编译配置)(3166)
3. Fast RCNN 训练自己的数据集（3训练和检测）(3097)
4. RCNN (Regions with CNN) 目标物检测 Fast RCNN的基础(2096)
5. Hog SVM 车辆 行人检测(979)

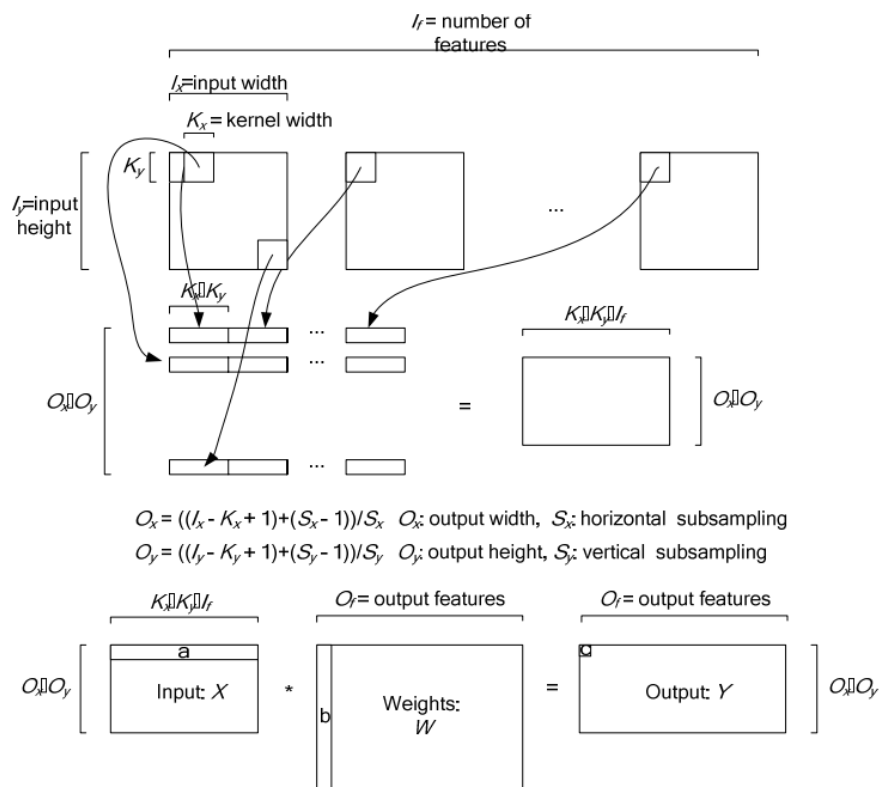
评论排行榜

1. Fast RCNN 训练自己数据集 (2修改数据读取接口)(22)
2. Fast RCNN 训练自己数据集 (1编译配置)(21)
3. Fast RCNN 训练自己的数据集（3训练和检测）(5)
4. opencv 3.0 DPM Cascade 检测（附带TBB和openMP加速）(4)
5. DPM检测模型 训练自己的数据集 读取接口修改(2)

推荐排行榜

1. Fast RCNN 训练自己数据集 (2修改数据读取接口)(5)
2. 车脸检测 Adaboost 检测过程(3)
3. Caffe 抽取CNN网络特征 Python(2)
4. DPM检测模型 训练自己的数据集 读取接口修改(2)
5. RCNN (Regions with CNN) 目标物检测 Fast RCNN的基础(2)

里的不同卷积核选取的逐行摆放，不同N的话，就在同一行后面继续拼接，不同个可以是多个通道，但是需要注意的是同一行里面每一段都应该对应的是原图中中一个位置的卷积窗口。



对于卷积层中的卷积操作，还有一个group的概念要说明一下，groups是代表filter 组的个数。引入group主要是为了选择性的连接卷基层的输入端和输出端的channels，否则参数会太多。每一个group 和 1/group的input 通道和 1/group 的output通道进行卷积操作。比如有4个input， 8个output，那么1-4属于第一组，5-8属于第二个group

ConvolutionLayer里面，主要重写了Forward cpu和Backward cpu

```
void ConvolutionLayer<Dtype>::Forward_cpu(const vector<Blob<Dtype>*>& bottom,
      const vector<Blob<Dtype>*>& top) {
    const Dtype* weight = this->blobs_[0]->cpu_data();
    for (int i = 0; i < bottom.size(); ++i) {
        const Dtype* bottom_data = bottom[i]->cpu_data();
        Dtype* top_data = top[i]->mutable_cpu_data();
        for (int n = 0; n < this->num_; ++n) {
            this->forward_cpu_gemm(bottom_data + n * this->bottom_dim_, weight,
                top_data + n * this->top_dim_);
            if (this->bias_term_) {
                const Dtype* bias = this->blobs_[1]->cpu_data();
                this->forward_cpu_bias(top_data + n * this->top_dim_, bias);
            }
        }
    }
}
```

可以看到其实这里他调用了`forward_cpu_gemm`，而这个函数内部又调用了`math_function`里面的`caffe_cpu_gemm`的通用矩阵相乘接口，GEMM的全称是General Matrix Matrix Multiply。其基本形式如下：

$$C = \alpha * op(A) * op(B) + \beta * C,$$

```
template <typename Dtype>
void ConvolutionLayer<Dtype>::Backward_cpu(const vector<Blob<Dtype>*>& top,
      const vector<bool>& propagate_down, const vector<Blob<Dtype>*>& bottom) {
    //反向传播梯度误差
    const Dtype* weight = this->blobs_[0]->cpu_data();
    Dtype* weight_diff = this->blobs_[0]->mutable_cpu_diff();
    for (int i = 0; i < top.size(); ++i) {
        const Dtype* top_diff = top[i]->cpu_diff();
        const Dtype* bottom_data = bottom[i]->cpu_data();
        Dtype* bottom_diff = bottom[i]->mutable_cpu_diff();

        //如果有bias项, 计算Bias导数
        if (this->bias_term_ && this->param_propagate_down_[1]) {
            Dtype* bias_diff = this->blobs_[1]->mutable_cpu_diff();
            for (int n = 0; n < this->num_; ++n) {
                this->backward_cpu_bias(bias_diff, top_diff + n * this->top_dim );
            }
        }
    }
}
```

```
    }  
}  
  
//计算weight  
if (this->param_propagate_down[0] || propagate_down[i]) {  
    for (int n = 0; n < this->num_; ++n) {  
        // 计算weights权重的梯度  
        if (this->param_propagate_down[0]) {  
            this->weight_cpu_gemm(bottom_data + n * this->bottom_dim_,  
                                   top_diff + n * this->top_dim_, weight_diff);  
        }  
        //计算bottom数据的梯度，下后传递  
        if (propagate_down[i]) {  
            this->backward_cpu_gemm(top_diff + n * this->top_dim_, weight,  
                                    bottom_diff + n * this->bottom_dim_);  
        }  
    }  
}  
}
```

[好文要顶](#)
[关注我](#)
[收藏该文](#)



 楼燚航的blog
关注 - 1
粉丝 - 60
[+加关注](#)

1 0

(请您对文章做出评价)

« 上一篇: [Caffe源码解析4: Data_layer](#)

» 下一篇: [Caffe源码解析6: Neuron_Layer](#)

posted @ 2016-01-23 23:30 楼燚航的blog 阅读(529) 评论(1) 编辑 收藏

评论列表

#1楼 2016-03-05 14:49 K_Augus

楼主博客写得不错！！
我想问一下，这篇博文的图是用什么画的啊？

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

【推荐】50万行VC++源码:大型组态工控、电力仿真CAD与GIS源码库



最新IT新闻:
LG确认: 开发Friends模块设备需要取得授权并协同开发
触角越来越广 华为能成为中国的三星吗?
微软认知服务: 人工智能的技术拼图
知己知彼, 百战不殆: 一篇文章看懂隐藏在阿尔法狗背后的深度学习
女性玩家崛起 研发女性游戏要注意什么
» 更多新闻...



最新知识库文章:
我是一个线程
为什么未来是全栈工程师的世界?
程序bug导致了天大的损失，要枪毙程序猿吗?

- [如何运维千台以上游戏云服务器](#)
- [架构漫谈（一）：什么是架构？](#)
- » [更多知识库文章...](#)