

SSD feature map 选择解析



陈继科 (/u/4b16c099ba4b) [+ 关注](#)

2016.11.03 19:26 字数 288 阅读 1685 评论 11 喜欢 2

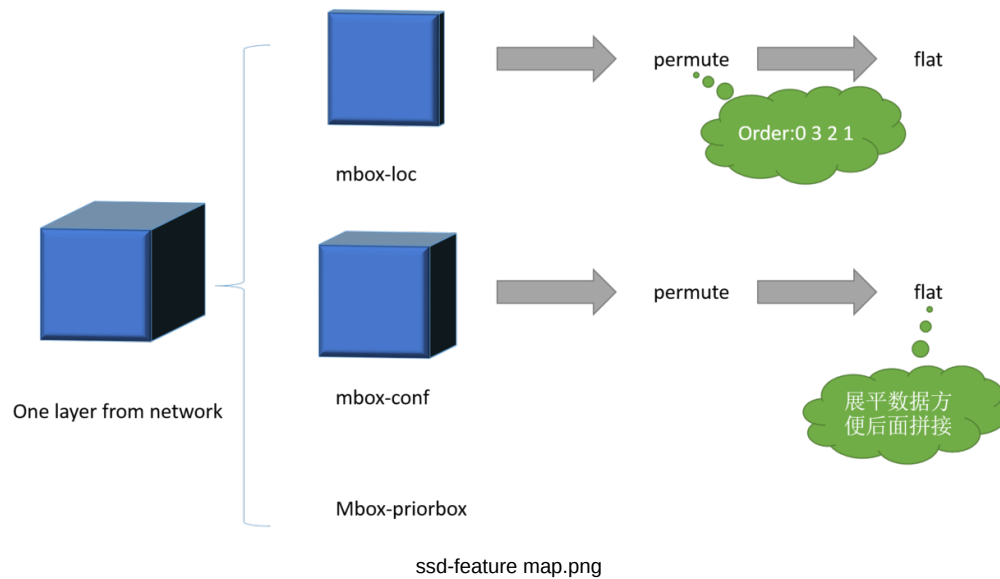
(/u/4b16c099ba4b)

针对不同的物体大小 (Object Scales)，传统的方法将图像转化成不同的大小，分别处理然后把结果综合。

这里ssd从不同的卷积层利用featuremap，可以达到同样的效果

生成预测的方法

如下图所示：



最左侧是选取的神经网络中的一个“图像”层

每个层做3个处理：

- (1) 生成loc预测，厚度4 x box
- (2) 生成类别预测，厚度21(类别) x box
- (3) 生成priorbox，这里面有个box大小范围、宽长比(2 3)等等

```
prior_box_param {
  min_size: 276.0
  max_size: 330.0
  aspect_ratio: 2
  aspect_ratio: 3
  flip: true
  clip: true
  variance: 0.1
  variance: 0.1
  variance: 0.2
  variance: 0.2
}
```

ssd实例说明

(1) 基本网络

Layer name	"图像"规格
input	3x300x300
conv1_1	64x300x300
conv1_2	64x300x300
pool_1	64x150x150
conv2_1	128x150x150
conv2_2	128x150x150
pool_2	128x75x75
conv3_1	256x75x75
conv3_2	256x75x75
conv3_3	256x75x75
pool_3	256x38x38
conv4_1	512x38x38
conv4_2	512x38x38
conv4_3	512x38x38
pool_4	512x19x19
conv5_1	512x19x19
conv5_2	512x19x19
conv5_3	512x19x19
-----	VGG昏割线
fc6 (convolution kernel dilation)	1024x19x19
fc7	1024x19x19
conv6_1	256x19x19

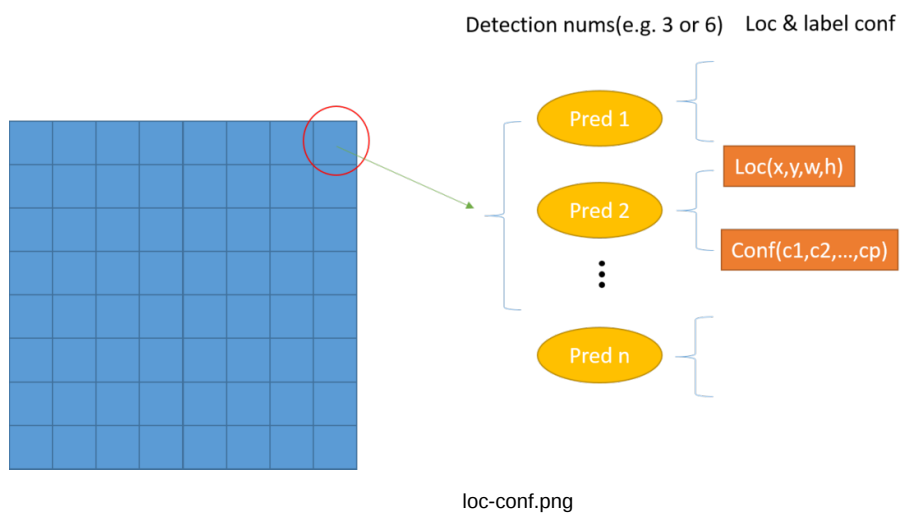


Layer name	"图像"规格
conv6_2	512x10x10
conv7_1	128x10x10
conv7_2(10-3+1*2)/2+1	256x5x5
conv8_1	128x5x5
conv8_2	256x3x3
pool6	25611

选取提取特征的层

Layer name	"图像"规格	特征生成	特征说明
conv4_3	512x38x38	mbox-loc conv	38x38x12(=3x4)
		mbox-conf conv	38x38x63(=3x21)
		prior-box	box min:30
fc7	1024x19x19	mbox-loc conv	19x19x24(=6x4)
		mbox-conf conv	19x19x126(=6x21)
		prior-box	box min:60 max:114
conv6_2	512x10x10	mbox-loc conv	10x10x24(=6x4)
		mbox-conf conv	10x10x126(=6x21)
		prior-box	box min:114 max:168
conv7_2	256x5x5	mbox-loc conv	5x5x24(=6x4)
		mbox-conf conv	5x5x126(=6x21)
		prior-box	box min:168 max:222
conv8_2	256x3x3	mbox-loc conv	3x3x24(=6x4)
		mbox-conf conv	3x3x126(=6x21)
		prior-box	box min:222 max:276
pool_6	256x1x1	mbox-loc conv	1x1x24(=6x4)
		mbox-conf conv	1x1x126(=6x21)
		prior-box	box min:276 max:330





所以对一张图一共提供：

$38 \times 38 \times 3 + (19 \times 19 + 10 \times 10 + 5 \times 5 + 3 \times 3 + 1 \times 1) \times 6 = 7308$ 个detection

每个detection包括4个值表示位置和21个值表示每个类的概率

为了实现ssd，原生的caffe是不行的

要定义新层：

Normalize

Permute

MultiBoxLoss等

一篇定义新层的方法如下所示：

<http://blog.csdn.net/kuaitoukid/article/details/41865803> (<http://blog.csdn.net/kuaitoukid/article/details/41865803>)

设计feature map

已知一个神经网络，选特定层，再后面加：



```

layer {
  name: "conv6_2_mbox_loc"
  type: "Convolution"
  bottom: "conv6_2"
  top: "conv6_2_mbox_loc"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 24////////////////////////////////////这样fe
    pad: 1////////////////////////////////////
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv6_2_mbox_loc_perm"
  type: "Permute"
  bottom: "conv6_2_mbox_loc"
  top: "conv6_2_mbox_loc_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
  name: "conv6_2_mbox_loc_flat"
  type: "Flatten"
  bottom: "conv6_2_mbox_loc_perm"
  top: "conv6_2_mbox_loc_flat"
  flatten_param {
    axis: 1
  }
}
layer {
  name: "conv6_2_mbox_conf"
  type: "Convolution"
  bottom: "conv6_2"
  top: "conv6_2_mbox_conf"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 126////////////////////////////////////

```



```
    pad: 1//////////////////////////////////////////这样fe
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv6_2_mbox_conf_perm"
  type: "Permute"
  bottom: "conv6_2_mbox_conf"
  top: "conv6_2_mbox_conf_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
  name: "conv6_2_mbox_conf_flat"
  type: "Flatten"
  bottom: "conv6_2_mbox_conf_perm"
  top: "conv6_2_mbox_conf_flat"
  flatten_param {
    axis: 1
  }
}
layer {
  name: "conv6_2_mbox_priorbox"
  type: "PriorBox"
  bottom: "conv6_2"
  bottom: "data"
  top: "conv6_2_mbox_priorbox"
  prior_box_param {
    min_size: 114.0 ////////////////////////////////////////////
    max_size: 168.0
    aspect_ratio: 2
    aspect_ratio: 3
    flip: true
    clip: true
    variance: 0.1
    variance: 0.1
    variance: 0.2
    variance: 0.2
  }
}
```





陈继科 (/u/4b16c099ba4b)

写了 27353 字，被 20 人关注，获得了 17 个喜欢 (/u/4b16c099ba4b)

+ 关注

复旦大学微电子本科生

如果觉得我的文章对您有用，请随意赞赏。您的支持将鼓励我继续创作！

赞赏支持

喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button)

2







更多分享

(http://cwb.assets.jianshu.io
/notes
/images
/6534032
/weibo
/image_867f9af05a7c.jpg)




登录 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-com)

11条评论

只看作者

按喜欢排序 按时间正序 按时间倒序



WendyCui (/u/70309b24b6d6)

2楼 · 2017.03.14 11:56

(/u/70309b24b6d6)

请教一下，从conv3_3的75*75是如何经过pool_3得到38*38的呢？网络结构中这一步的pool仍然是kernel size=2，stride=2，pad=0的呀

赞

回复

- 陈继科 (/u/4b16c099ba4b)：@WendyCui (/users/70309b24b6d6) caffe中pad 0是通过offset的方式进行的，即并不是矩阵加0变大，而是如果pad0则边缘用来pool的数据少于kernel的大小，所以即使除不尽也没关系，只是它拿来做pool的数据少了，具体可以看下caffe源码

2017.03.14 12:11 回复
- WendyCui (/u/70309b24b6d6)：那没有除尽余下的部分就按原来的保留了是嘛？

2017.03.14 14:40 回复
- 陈继科 (/u/4b16c099ba4b)：@WendyCui (/users/70309b24b6d6) 没有除尽的部分取Max or Ave作为结果，若是剩余1x1就相当于保留

2017.03.14 15:36 回复

添加新评论 | 还有2条评论，展开查看





小酒窝_a753 (/u/86333112903c)

3楼 · 2017.05.08 15:30

(/u/86333112903c)

你好，请问PriorBox层中box的最大最小size是如何计算的，而且它还有一个参数step，请问你知道它表示什么吗？

👍 赞 💬 回复

陈继科 (/u/4b16c099ba4b)：@小酒窝_a753 (/users/86333112903c) feature map从前往后检测的box从小到达，有一点的经验成分在里面，同时你也可以参考两层3*3结合对感受面积的变化（9-25）

2017.05.08 15:46 💬 回复

陈继科 (/u/4b16c099ba4b)：@小酒窝_a753 (/users/86333112903c) Priorbox里好像没有step呀

2017.05.08 15:48 💬 回复

小酒窝_a753 (/u/86333112903c)：@陈继科 (/users/4b16c099ba4b) gitub上开源的代码运行ssd_pascal.py文件生成的网络文件train.prototxt中就有，和ssd_pascal.py文件中的step相对应。这个step和min_size，max_size的计算问题我已经弄懂啦，谢谢你的回复 😊

2017.05.10 22:18 💬 回复

✍️ 添加新评论



喻茸sophie (/u/4224a1c10d55)

4楼 · 2017.07.17 16:15

(/u/4224a1c10d55)

最近在看ssd，发现居然是你写的，哈哈，厉害

👍 赞 💬 回复

