


# 卷积神经网络中用1\*1 卷积有什么作用或者好处呢？

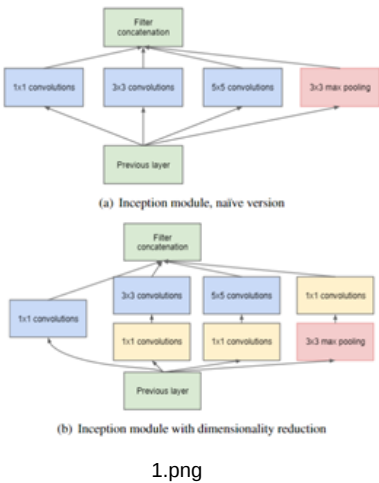


jiandanjinxin (/u/8ad7903302b3) + 关注  
2017.08.10 17:11\* 字数 1376 阅读 29 评论 0 喜欢 3  
(/u/8ad7903302b3)

1X1卷积核最开始是在颜水成论文 [1312.4400] Network In Network (<https://arxiv.org/abs/1312.4400>) 中提出的，后来被[GoogLeNet 1409.4842] Going Deeper with Convolutions (<https://arxiv.org/abs/1409.4842>)的Inception结构继续应用了。能够使用更小channel的前提就是sparse比较多 不然1\*1效果也不会很明显

## 1.实现跨通道的交互和信息整合

1×1的卷积层（可能）引起人们的重视是在NIN的结构中，论文中林敏师兄的想法是利用MLP代替传统的线性卷积核，从而提高网络的表达能力。文中同时利用了跨通道pooling的角度解释，认为文中提出的MLP其实等价于在传统卷积核后面接cccp层，从而实现多个feature map的线性组合，实现跨通道的信息整合。而cccp层是等价于1×1卷积的，因此细看NIN的caffe实现，就是在每个传统卷积层后面接了两个cccp层（其实就是接了两个1×1的卷积层）



## 2.进行卷积核通道数的降维和升维

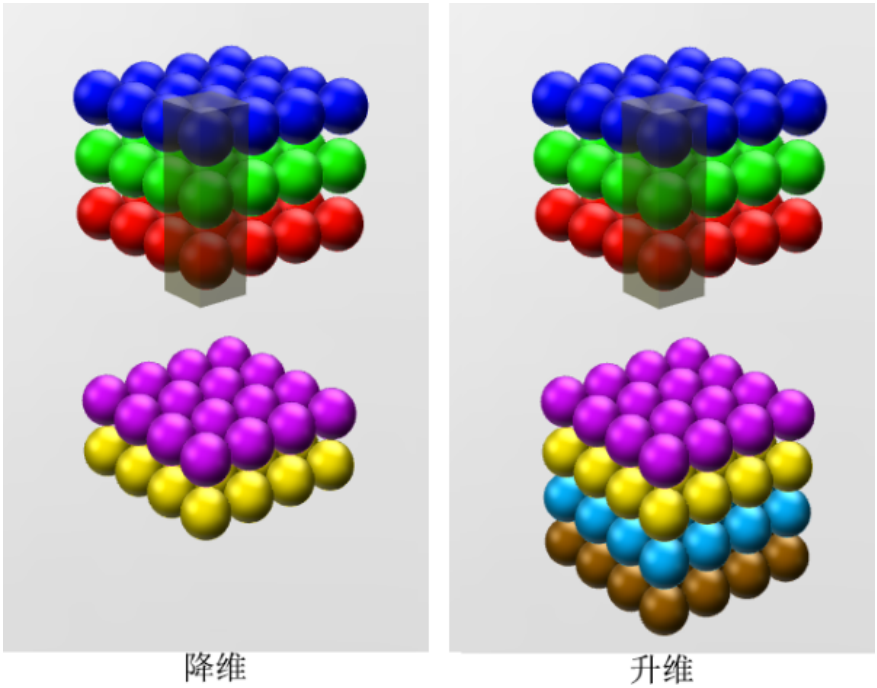
由于3X3卷积或者5X5卷积在几百个filter的卷积层上做卷积操作时相当耗时，所以1X1卷积在3X3卷积或者5X5卷积计算之前先降低维度。那么，1X1卷积的主要作用有以下几点：

- 1、降维（ dimension reductionality ）。比如，一张500 X500且厚度depth为100 的图片在20个filter上做1X1的卷积，那么结果的大小为500X500X20。
- 2、加入非线性。卷积层之后经过激励层，1X1的卷积在前一层的学习表示上添加了非线性激励（ non-linear activation ），提升网络的表达能力；

What is Depth of a convolutional neural network? (<https://stackoverflow.com/questions/32294261/what-is-depth-of-a-convolutional-neural-network>)

如果卷积的输出输入都是一个平面，那么 1 X1卷积核并没有什么意义，它是完全不考虑像素与周边其他像素关系。但卷积的输出输入是长方体，所以 1 X1卷积实际上对每个像素点，在不同的channels上进行线性组合（信息整合），且保留原有平面结构，调控depth,从而完成升维或降维的功能。

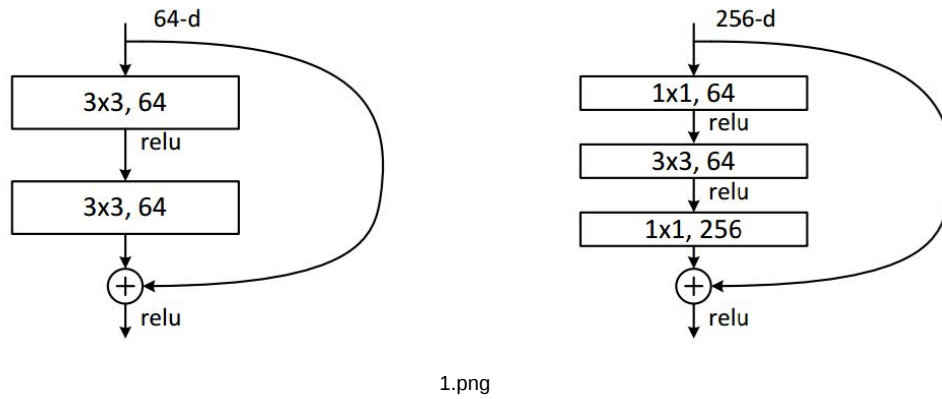
下图所示，如果选择 2 个filters的 1 X1卷积层，那么数据就从原来的depth3降到了2。若用 4 个 filters,则起到了升维的作用。



1.png

MSRA的ResNet同样也利用了1×1卷积，并且是在3×3卷积层的前后都使用了，不仅进行了降维，还进行了升维，使得卷积层的输入和输出的通道数都减小，参数数量进一步减少，如下图的结构。





## Simple Answer

Most simplistic explanation would be that 1x1 convolution leads to dimensionality reduction. For example, an image of 200 x 200 with 50 features on convolution with 20 filters of 1x1 would result in size of 200 x 200 x 20. But then again, is this the best way to do dimensionality reduction in the convolutional neural network? What about the efficacy vs efficiency?

One by One [ 1 x 1 ] Convolution - counter-intuitively useful (<http://iamaaditya.github.io/2016/03/one-by-one-convolution/>)

## Complex Answer

### Feature transformation

Although 1x1 convolution is a 'feature pooling' technique, there is more to it than just sum pooling of features across various channels/feature-maps of a given layer. 1x1 convolution acts like coordinate-dependent transformation in the filter space (<https://plus.google.com/118431607943208545663/posts/2y7nmBuh2ar>) (<https://plus.google.com/118431607943208545663/posts/2y7nmBuh2ar%5D>). It is important to note here that this transformation is strictly linear, but in most of application of 1x1 convolution, it is succeeded by a non-linear activation layer like ReLU. This transformation is learned through the (stochastic) gradient descent. But an important distinction is that it suffers with less over-fitting due to smaller kernel size (1x1).

**3.可以在保持feature map 尺寸不变（即不损失分辨率）的前提下大幅增加非线性特性，把网络做得很deep**

### Deeper Network

One by One convolution was first introduced in this paper titled Network in Network (<http://arxiv.org/abs/1412.0056>)



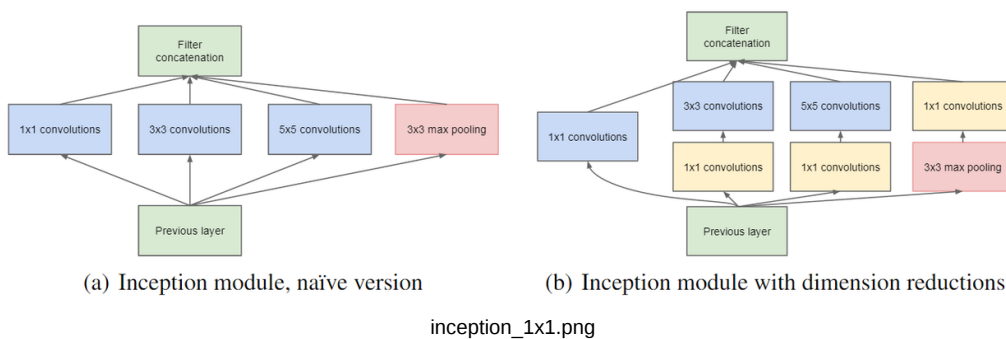
ps://arxiv.org/abs/1312.4400). In this paper, the author's goal was to generate a deeper network without simply stacking more layers. It replaces few filters with a smaller perceptron layer with mixture of 1x1 and 3x3 convolutions. In a way, it can be seen as "going wide" instead of "deep", but it should be noted that in machine learning terminology, 'going wide' is often meant as adding more data to the training. Combination of 1x1 (x F) convolution is mathematically equivalent to a multi-layer perceptron[https://www.reddit.com/r/MachineLearning/comments/3oln72/1x1\_convolutions\_why\_use\_them/cvyxood/] (https://www.reddit.com/r/MachineLearning/comments/3oln72/1x1\_convolutions\_why\_use\_them/cvyxood/%5D)

## Inception Module

In GoogLeNet architecture, 1x1 convolution is used for two purposes

To make network deep by adding an "inception module" like Network in Network paper,  
To reduce the dimensions inside this "inception module".  
To add more non-linearity by having ReLU immediately after every 1x1 convolution.

Here is the screenshot from the paper, which elucidates above points :



It can be seen from the image on the right, that 1x1 convolutions (in yellow), are specially used before 3x3 and 5x5 convolution to reduce the dimensions. It should be noted that at a two step convolution operation can always be combined into one, but in this case and in most other deep learning networks, convolutions are followed by non-linear activation and hence convolutions are no longer linear operators and cannot be combined.

In designing such a network, it is important to note that initial convolution kernel should be of size larger than 1x1 to have a receptive field capable of capturing locally spatial information. According to the NIN paper, 1x1 convolution is equivalent to cross-channel parametric pooling layer. From the paper - "This cascaded cross channel parametric pooling structure allows complex and learnable interactions of cross channel information".

Cross channel information learning (cascaded 1x1 convolution) is biologically inspired because human visual cortex have receptive fields (kernels) tuned to different orientation . For e.g



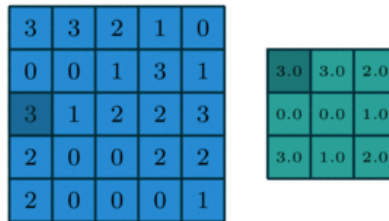
RotBundleFiltersListPlot3D.gif

RotBundleFiltersListPlot3D.gif

Different orientation tuned receptive field profiles in the human visual cortex Source (<http://bmia.bmt.tue.nl/education/courses/fev/course/notebooks/Convolution.html>)

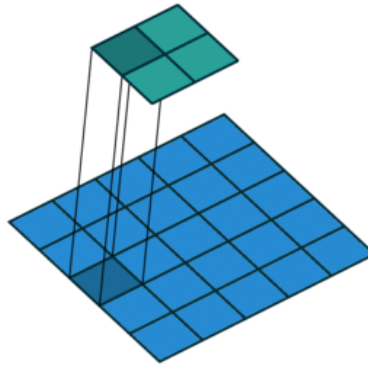
## More Uses

1x1 Convolution can be combined with Max pooling



numerical\_max\_pooling.gif

1x1 Convolution with higher strides leads to even more reduction in data by decreasing resolution, while losing very little non-spatially correlated information.



no\_padding\_strides.gif

Replace fully connected layers with 1x1 convolutions as Yann LeCun believes they are the same



-In Convolutional Nets, there is no such thing as “fully-connected layers”. There are only convolution layers with 1x1 convolution kernels and a full connection table.– Yann LeCun

Convolution gif images generated using this wonderful code ([https://github.com/vdumoulin/conv\\_arithmetic](https://github.com/vdumoulin/conv_arithmetic)), more images on 1x1 convolutions and 3x3 convolutions can be found here ([http://gpgpu.cs-i.brandeis.edu/convolution\\_images/](http://gpgpu.cs-i.brandeis.edu/convolution_images/))

参考文献

<https://www.zhihu.com/question/56024942> (<https://www.zhihu.com/question/56024942>)

📖 日记本 (/nb/3059498)

举报文章 © 著作权归作者所有



jiandanjinxin (/u/8ad7903302b3)

写了 89021 字，被 54 人关注，获得了 69 个喜欢 (/u/8ad7903302b3)

+ 关注

敏于行而慎于言,讷于言而敏于行 <http://blog.csdn.net/jiandanjinxin> <https://github.com/jiandanjinxin>

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) | 3



更多分享

(<http://cwb.assets.jianshu.io/notes/images>  
<http://cwb.assets.jianshu.io/weibo>  
[/image\\_f91530985bb9.jpg](http://cwb.assets.jianshu.io/image_f91530985bb9.jpg))



登录后发表评论 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-comment) 15636374

评论



智慧如你，不想发表一点想法 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-nocomments-text)咩~

被以下专题收入，发现更多相似内容



深度学习 (/c/c4e9f3b2259a?utm\_source=desktop&utm\_medium=notes-included-collection)

