

法律声明

本课件包括演示文稿、示例、代码、题库、视频和声音等内容，深度之眼和讲师拥有完全知识产权；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或者机构不得盗版、复制、仿造其中的创意和内容，我们保留一切通过法律手段追究违反者的权利。

课程详情请咨询

- 微信公众号：深度之眼
- 客服微信号：deepshare0920



公众号



微信

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文



deepshare.net

深度之眼

卷积层

导师：余老师

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文

目录

1/ 1d/2d/3d卷积

2/ 卷积-`nn.Conv2d()`

3/ 转置卷积-`nn.ConvTranspose`

1d/2d/3d Convolution

Dimension of Convolution

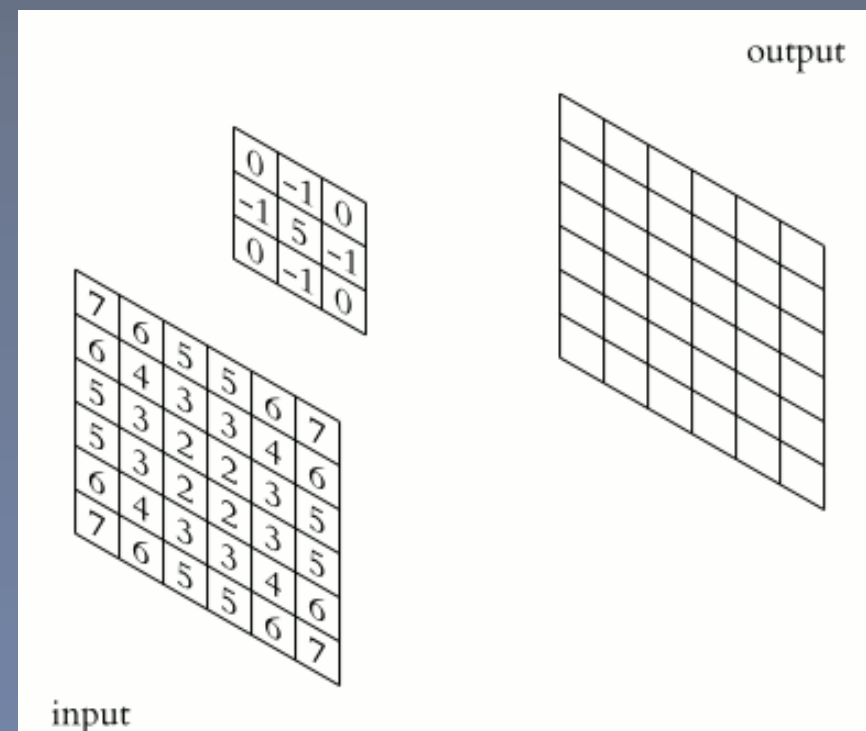
1d/2d/3d Convolution

Dimension of Convolution

卷积运算：卷积核在输入信号（图像）上滑动，相应位置上进行**乘加**

卷积核：又称为滤波器，过滤器，可认为是某种模式，某种特征。

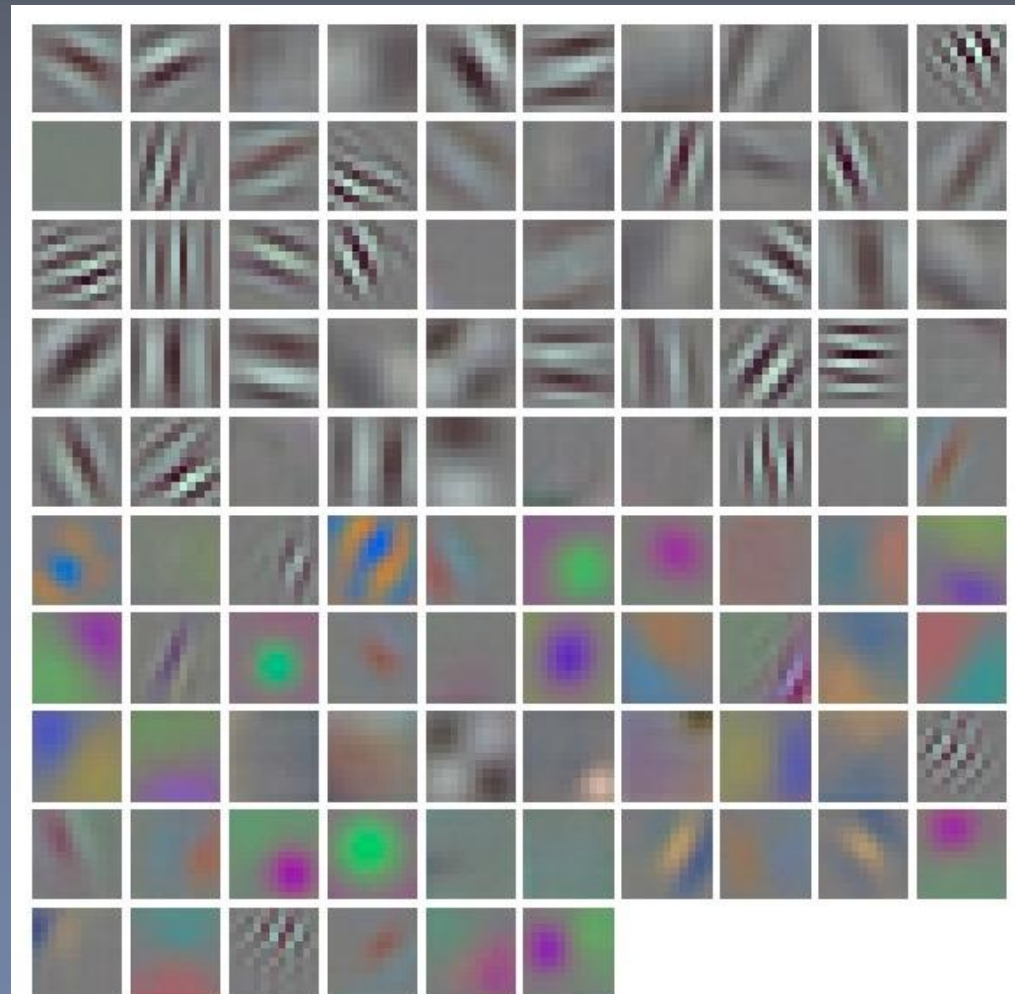
卷积过程类似于用一个模版去图像上寻找与它相似的区域，与卷积核模式越相似，激活值越高，从而实现特征提取



1d/2d/3d Convolution

Dimension of Convolution

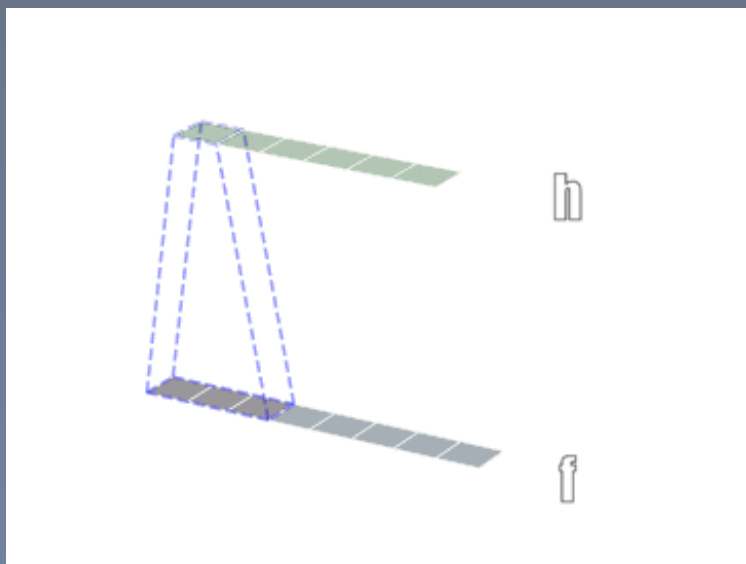
AlexNet卷积核可视化，发现卷积核学习到的是**边缘，条纹，色彩**这一些细节模式



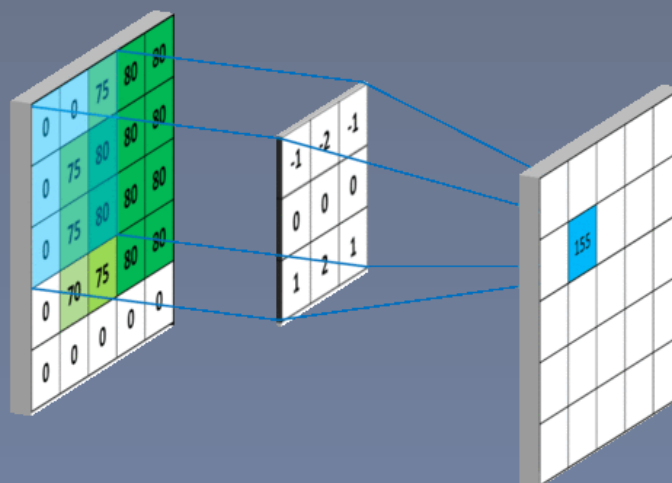
1d/2d/3d Convolution

Dimension of Convolution

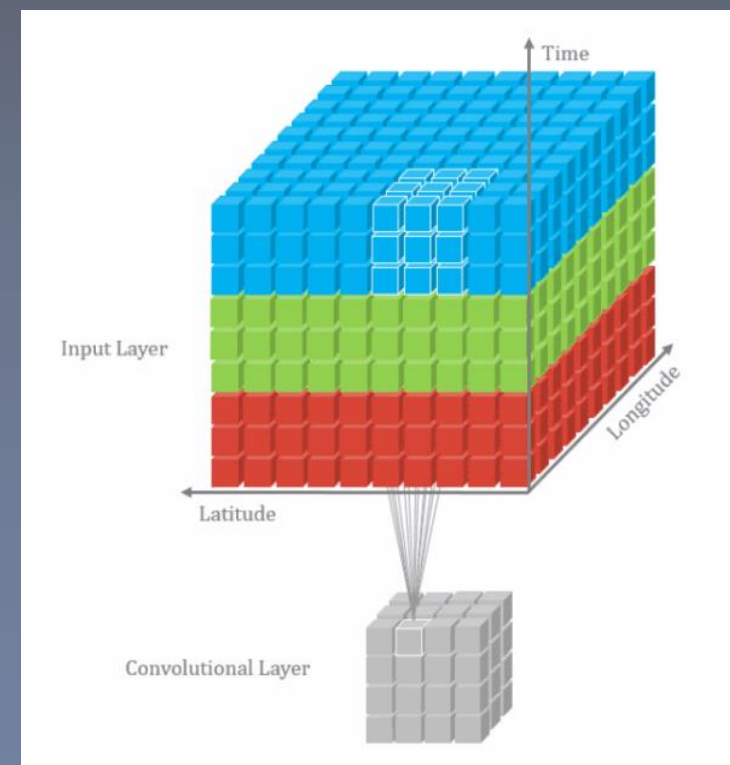
卷积维度：一般情况下，卷积核在几个维度上滑动，就是几维卷积



1维卷积



2维卷积



3维卷积

nn.Conv2d

nn.Conv2d

nn.Conv2d

nn.Conv2d



deepshare.net

深度之眼

nn.Conv2d

功能：对多个二维信号进行二维卷积

主要参数：

- **in_channels**：输入通道数
- **out_channels**：输出通道数，等价于卷积核个数
- **kernel_size**：卷积核尺寸
- **stride**：步长
- **padding**：填充个数

```
nn.Conv2d(in_channels,  
          out_channels,  
          kernel_size,  
          stride=1,  
          padding=0,  
          dilation=1,  
          groups=1,  
          bias=True,  
          padding_mode='zeros')
```

- **dilation**：空洞卷积大小
- **groups**：分组卷积设置
- **bias**：偏置

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文

nn.Conv2d

nn.Conv2d



deepshare.net

深度之眼

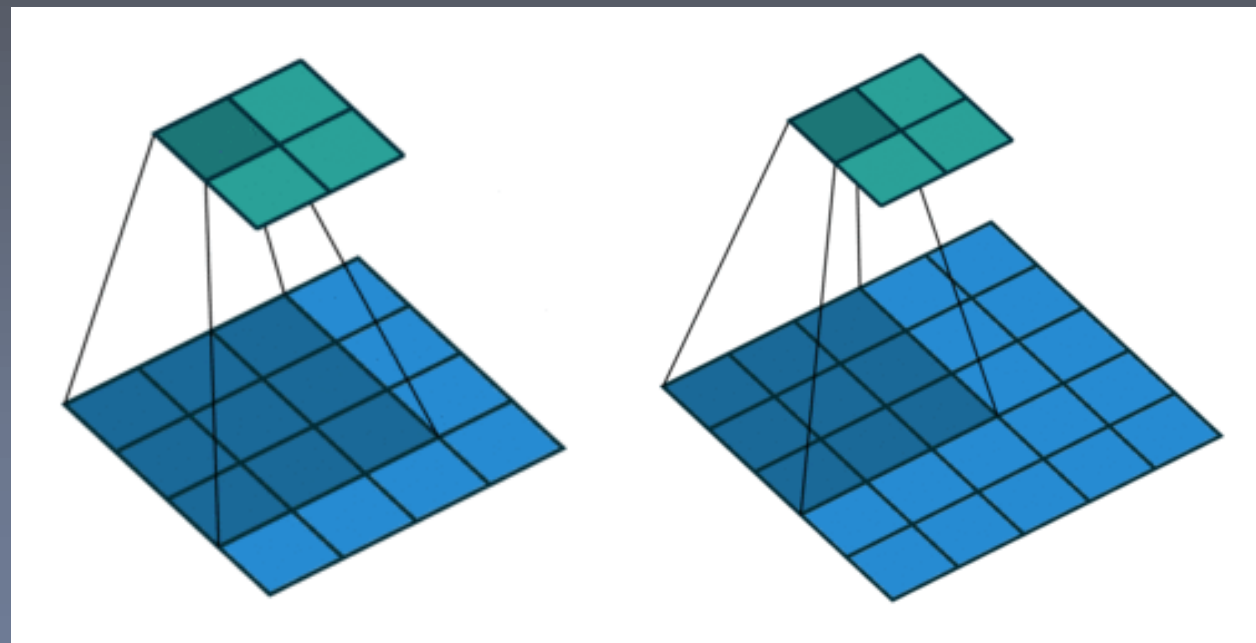
nn.Conv2d

功能：对多个二维信号进行二维卷积

主要参数：

- **in_channels**：输入通道数
- **out_channels**：输出通道数，等价于卷积核个数
- **kernel_size**：卷积核尺寸
- **stride**：步长
- **padding**：填充个数

- **dilation**：空洞卷积大小
- **groups**：分组卷积设置
- **bias**：偏置



关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文

nn.Conv2d

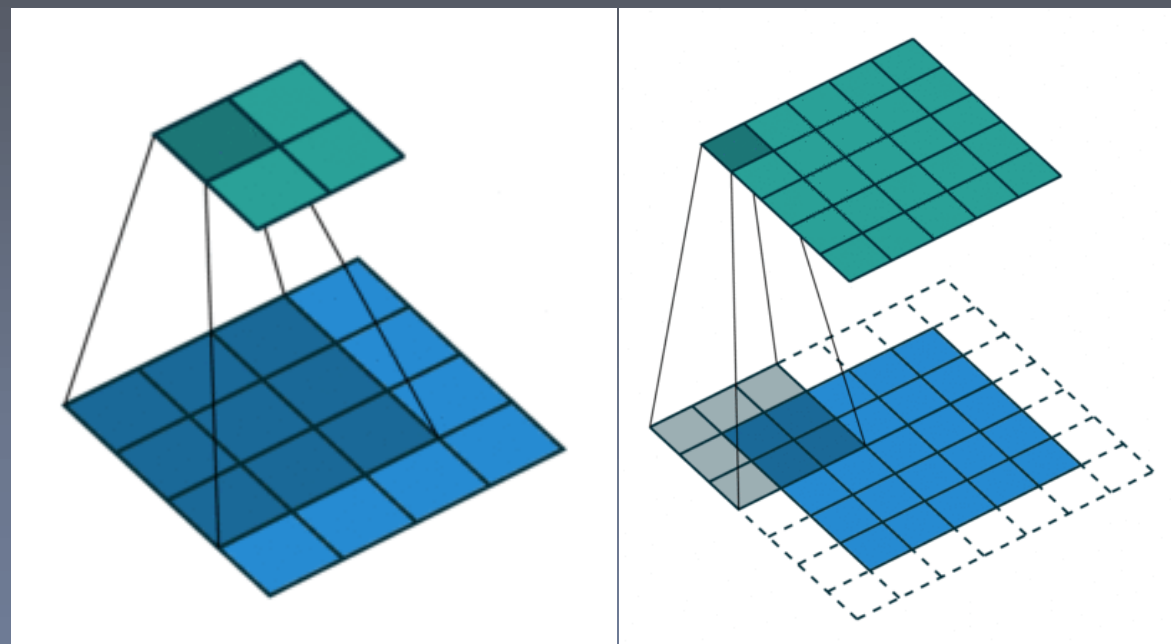
nn.Conv2d

nn.Conv2d

功能：对多个二维信号进行二维卷积

主要参数：

- **in_channels**：输入通道数
- **out_channels**：输出通道数，等价于卷积核个数
- **kernel_size**：卷积核尺寸
- **stride**：步长
- **padding**：填充个数



- **dilation**：空洞卷积大小
- **groups**：分组卷积设置
- **bias**：偏置

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文

nn.Conv2d

nn.Conv2d



deepshare.net

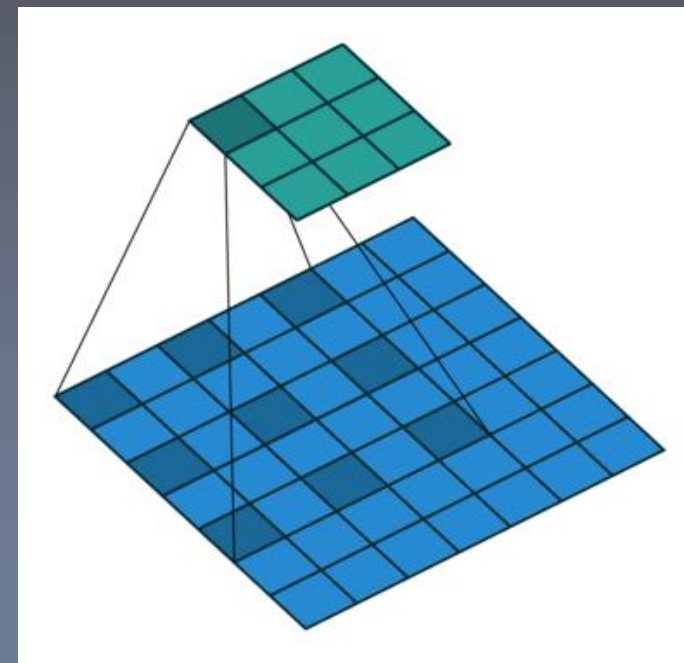
深度之眼

nn.Conv2d

功能：对多个二维信号进行二维卷积

主要参数：

- **in_channels**：输入通道数
- **out_channels**：输出通道数，等价于卷积核个数
- **kernel_size**：卷积核尺寸
- **dilation**：空洞卷积大小
- **stride**：步长
- **groups**：分组卷积设置
- **padding**：填充个数
- **bias**：偏置



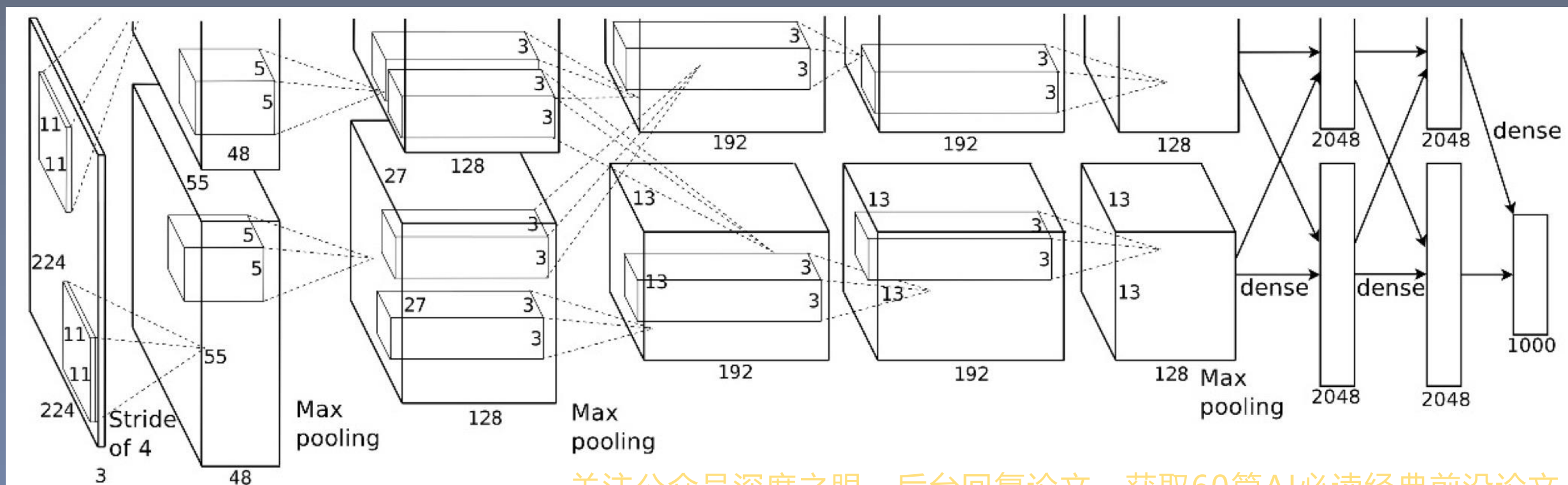
关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文

nn.Conv2d

nn.Conv2d

nn.Conv2d

功能：对多个二维信号进行二维卷积



关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文

nn.Conv2d

nn.Conv2d



deepshare.net

深度之眼

nn.Conv2d

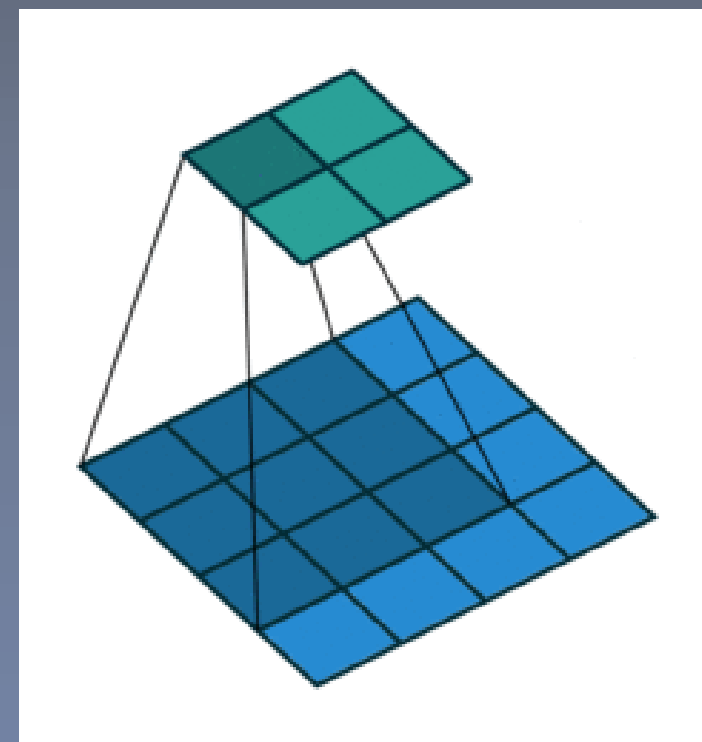
功能：对多个二维平面信号进行二维卷积

尺寸计算：

简化版： $out_{size} = \frac{In_{size} - kernel_{size}}{stride} + 1$

完整版：

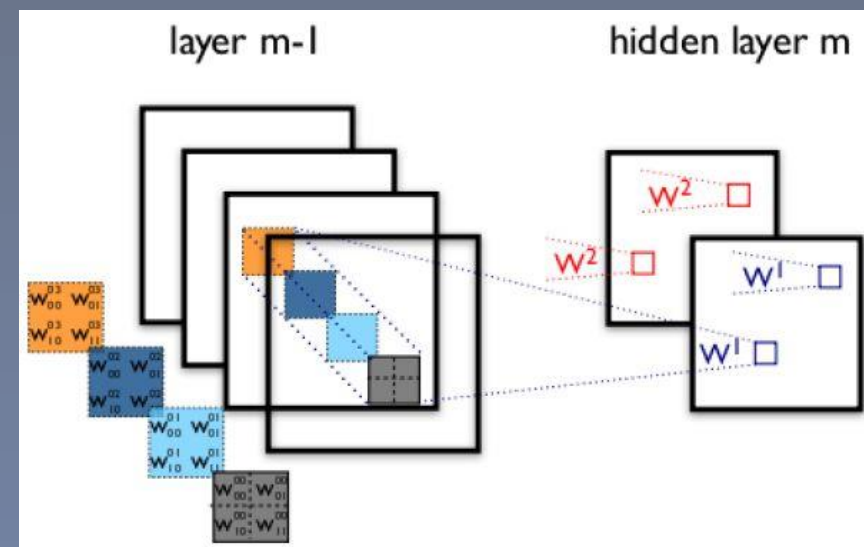
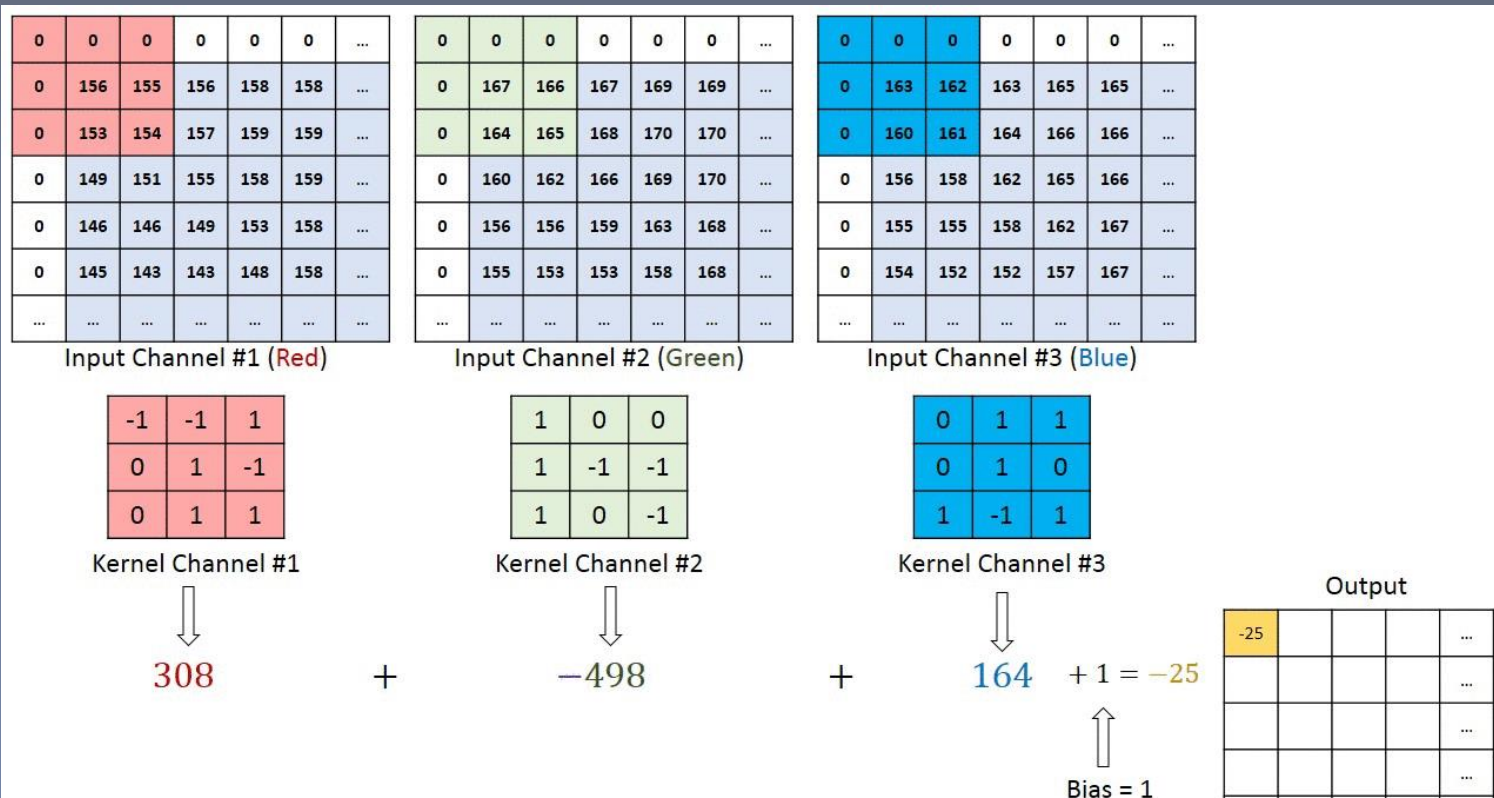
$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times padding[0] - dilation[0] \times (kernel_size[0] - 1) - 1}{stride[0]} + 1 \right\rfloor$$



1d/2d/3d Convolution

Dimension of Convolution

卷积维度：一般情况下，卷积核在几个维度上滑动，就是几维卷积



转置卷积

Transpose Convolution

转置卷积

Transpose Convolution

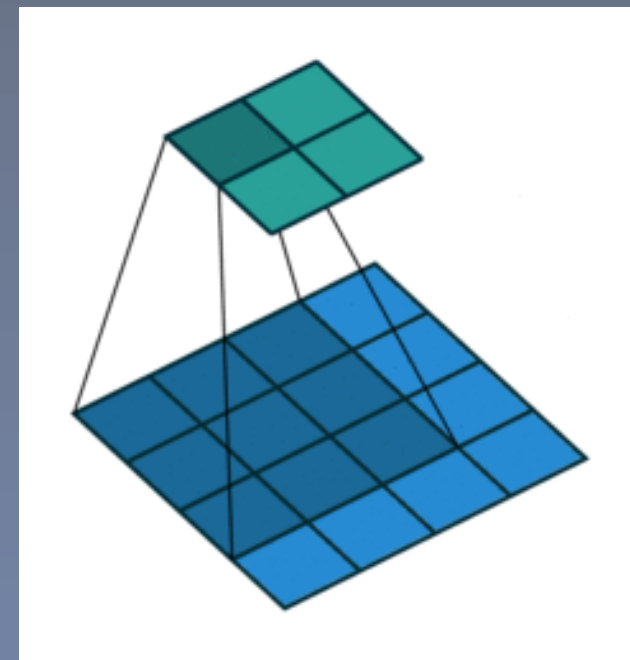
转置卷积又称为反卷积(~~反卷积~~ Deconvolution)和部分跨越卷积(Fractionally-strided Convolution)，用于对图像进行上采样(UpSample)

为什么称为转置卷积？

假设图像尺寸为 4×4 ，卷积核为 3×3 ，padding=0，stride=1

正常卷积：

图像： $I_{16 \times 1}$ 卷积核： $K_{4 \times 16}$ 输出： $O_{4 \times 1} = K_{4 \times 16} * I_{16 \times 1}$



转置卷积

Transpose Convolution

为什么称为转置卷积？

正常卷积：

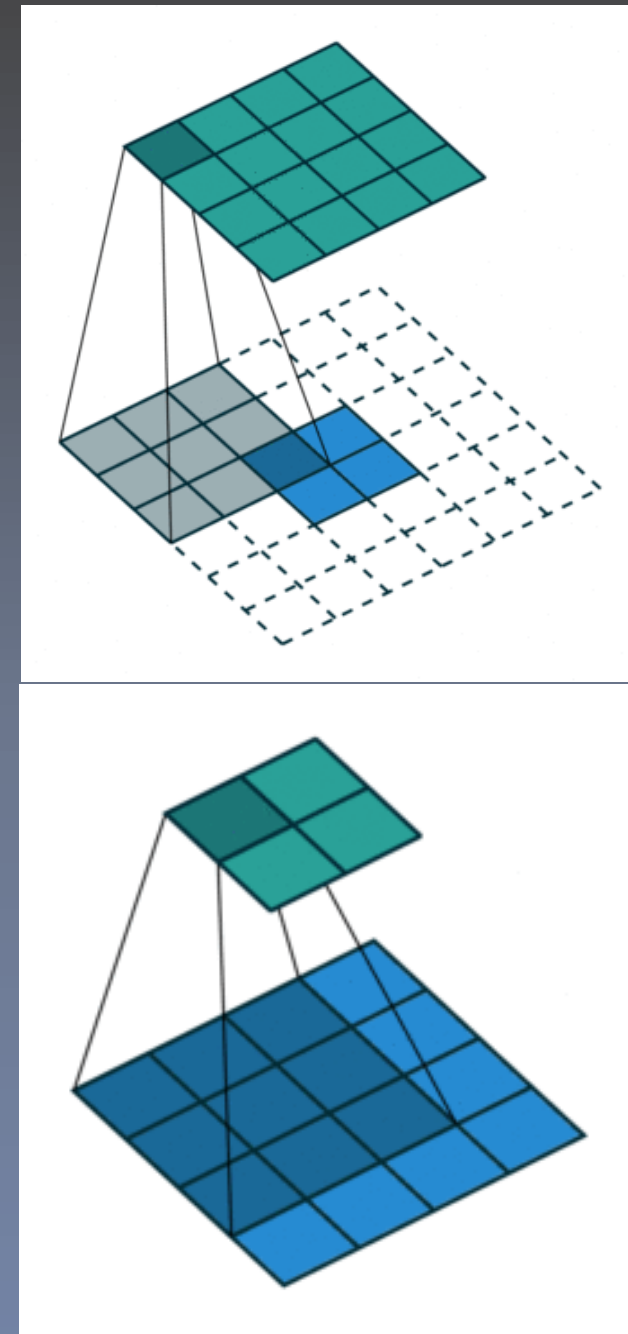
假设图像尺寸为 4×4 ，卷积核为 3×3 ，padding=0，stride=1

图像： $I_{16 \times 1}$ 卷积核： $K_{4 \times 16}$ 输出： $O_{4 \times 1} = K_{4 \times 16} * I_{16 \times 1}$

转置卷积：

假设图像尺寸为 2×2 ，卷积核为 3×3 ，padding=0，stride=1

图像： $I_{4 \times 1}$ 卷积核： $K_{16 \times 4}$ 输出： $O_{16 \times 1} = K_{16 \times 4} * I_{4 \times 1}$



转置卷积

Transpose Convolution

`nn.ConvTranspose2d`

功能：转置卷积实现上采样

主要参数：

- **in_channels**：输入通道数
- **out_channels**：输出通道数
- **kernel_size**：卷积核尺寸
- **stride**：步长
- **padding**：填充个数



deepshare.net

深度之眼

```
nn.ConvTranspose2d(in_channels,
                    out_channels,
                    kernel_size,
                    stride=1,
                    padding=0,
                    output_padding=0,
                    groups=1,
                    bias=True,
                    dilation=1,
                    padding_mode='zeros')
```

- **dilation**：空洞卷积大小
- **groups**：分组卷积设置
- **bias**：偏置

转置卷积

Transpose Convolution

`nn.ConvTranspose2d`

功能：转置卷积实现上采样

尺寸计算：

简化版： $out_{size} = (in_{size} - 1) * stride + kernel_{size}$ $out_{size} = \frac{In_{size} - kernel_{size}}{stride} + 1$

完整版：

$$H_{out} = (H_{in} - 1) \times stride[0] - 2 \times padding[0] + dilation[0] \times (kernel_size[0] - 1) + output_padding[0] + 1$$

棋盘效应：推荐文章《Deconvolution and Checkerboard Artifacts》
关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文

—— 结 语 ——

在这次课程中，学习了nn模块中卷积层

在下次课程中，我们将会学习

nn中其他常用网络层



关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文



deepshare.net

深度之眼

联系我们：

电话：18001992849

邮箱：service@deepshare.net

QQ：2677693114



公众号



客服微信

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典前沿论文