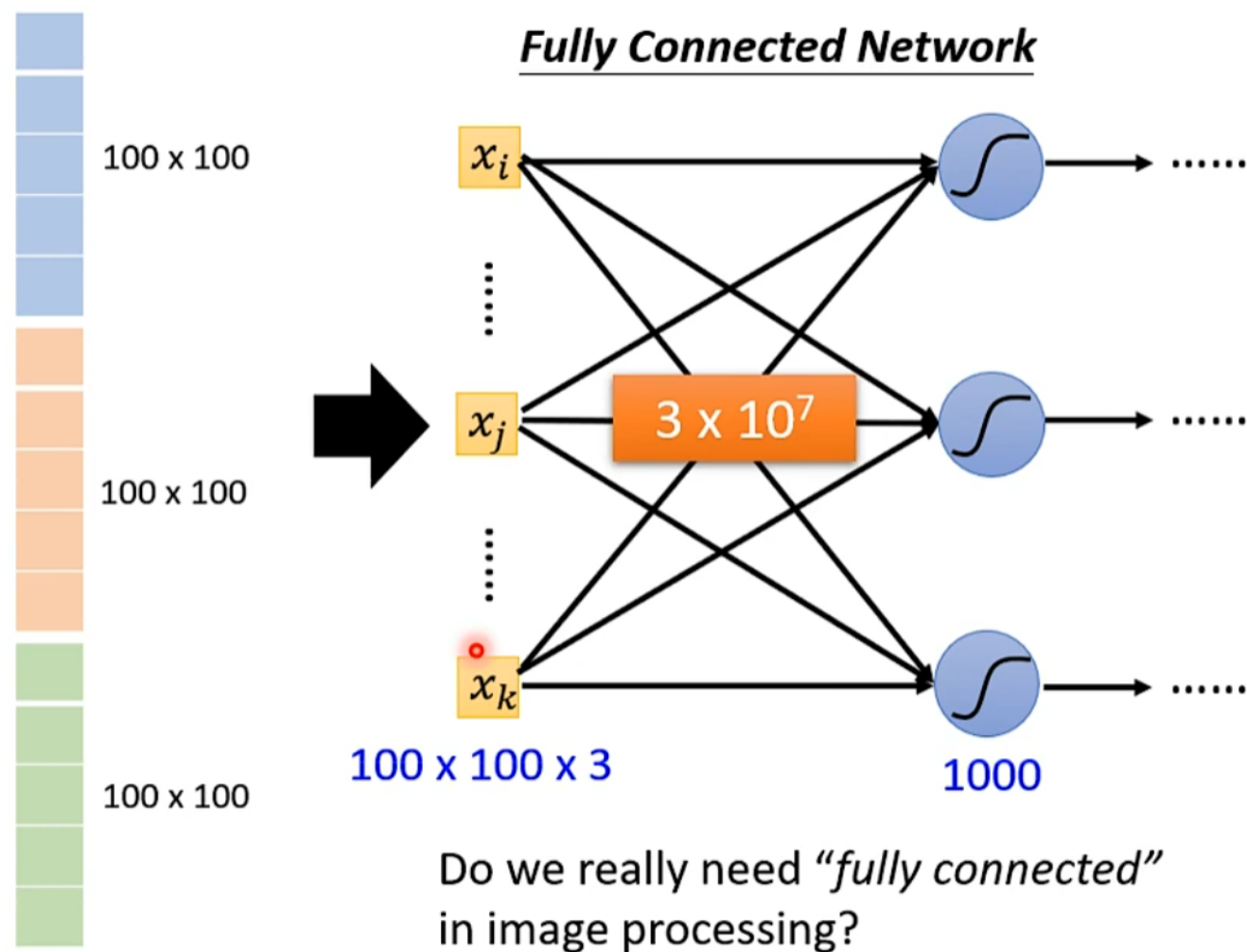


# CNN

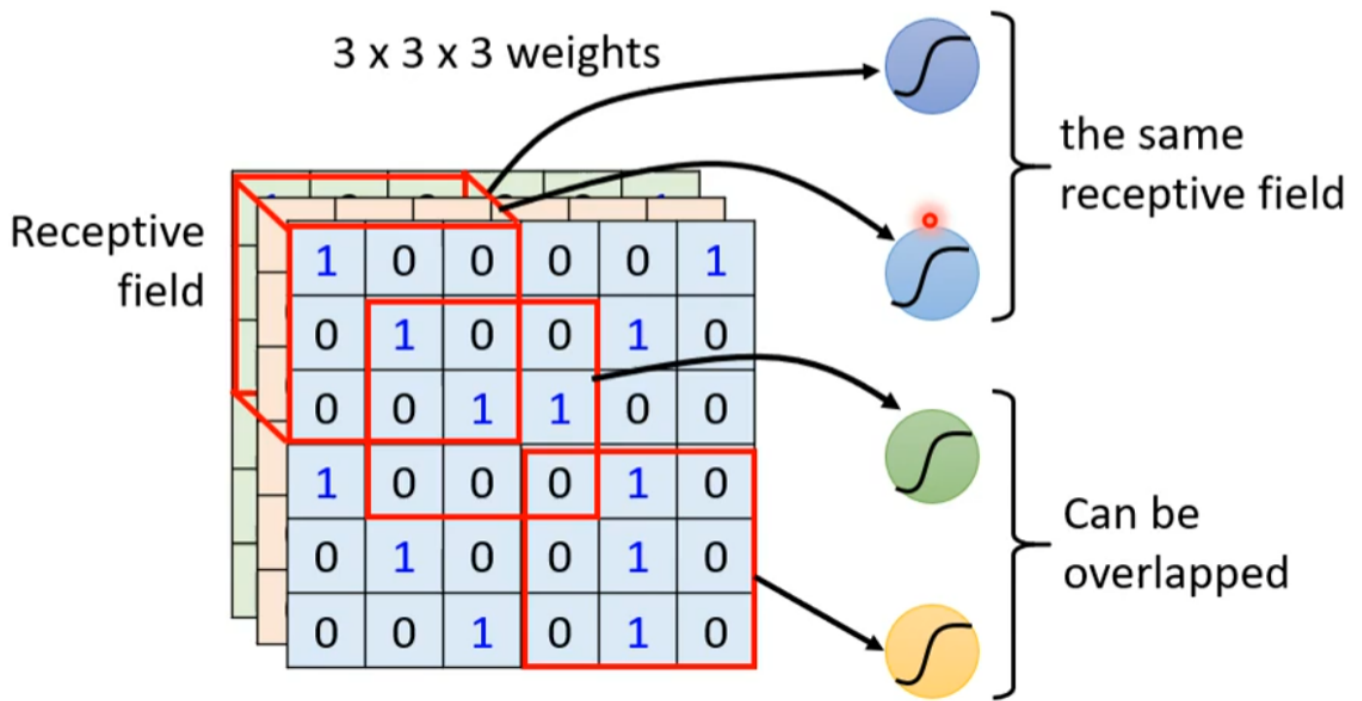
## 引入：图像分类

- (1) 目前对图像数据集的处理还是预处理统一像素大小
- (2) 3维tensor可以flatten和cat
- (3) 庞大的运算和参数量更容易过拟合



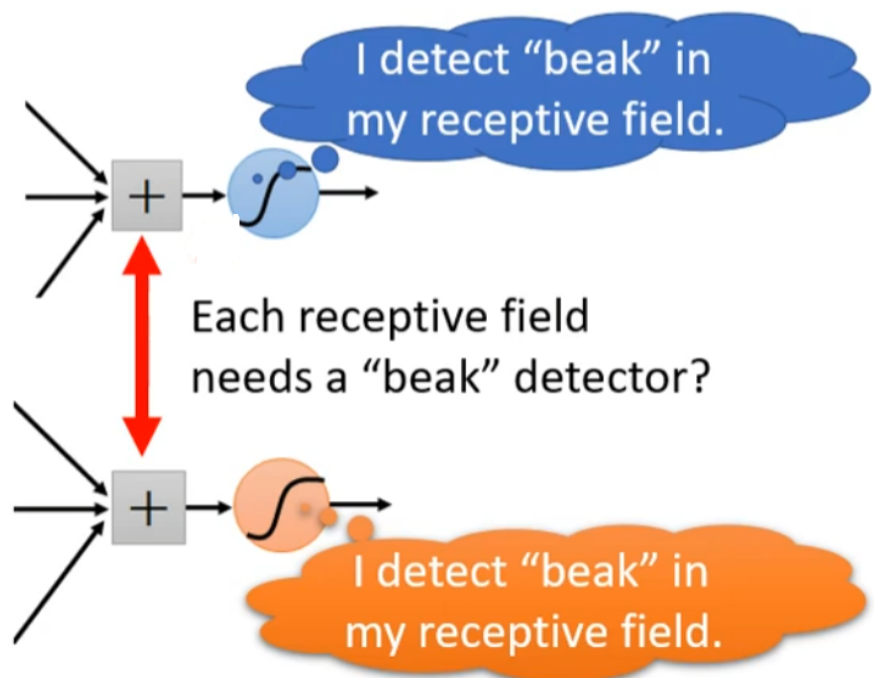
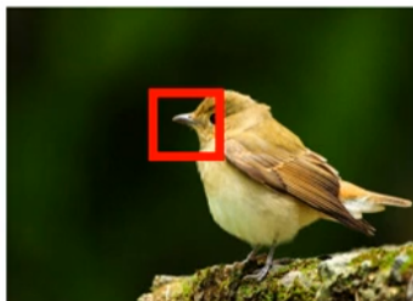
解决方法：

- 对于图像的辨别本身并不需要看全部像素，可以重点看一些模块（Receptive Field）。



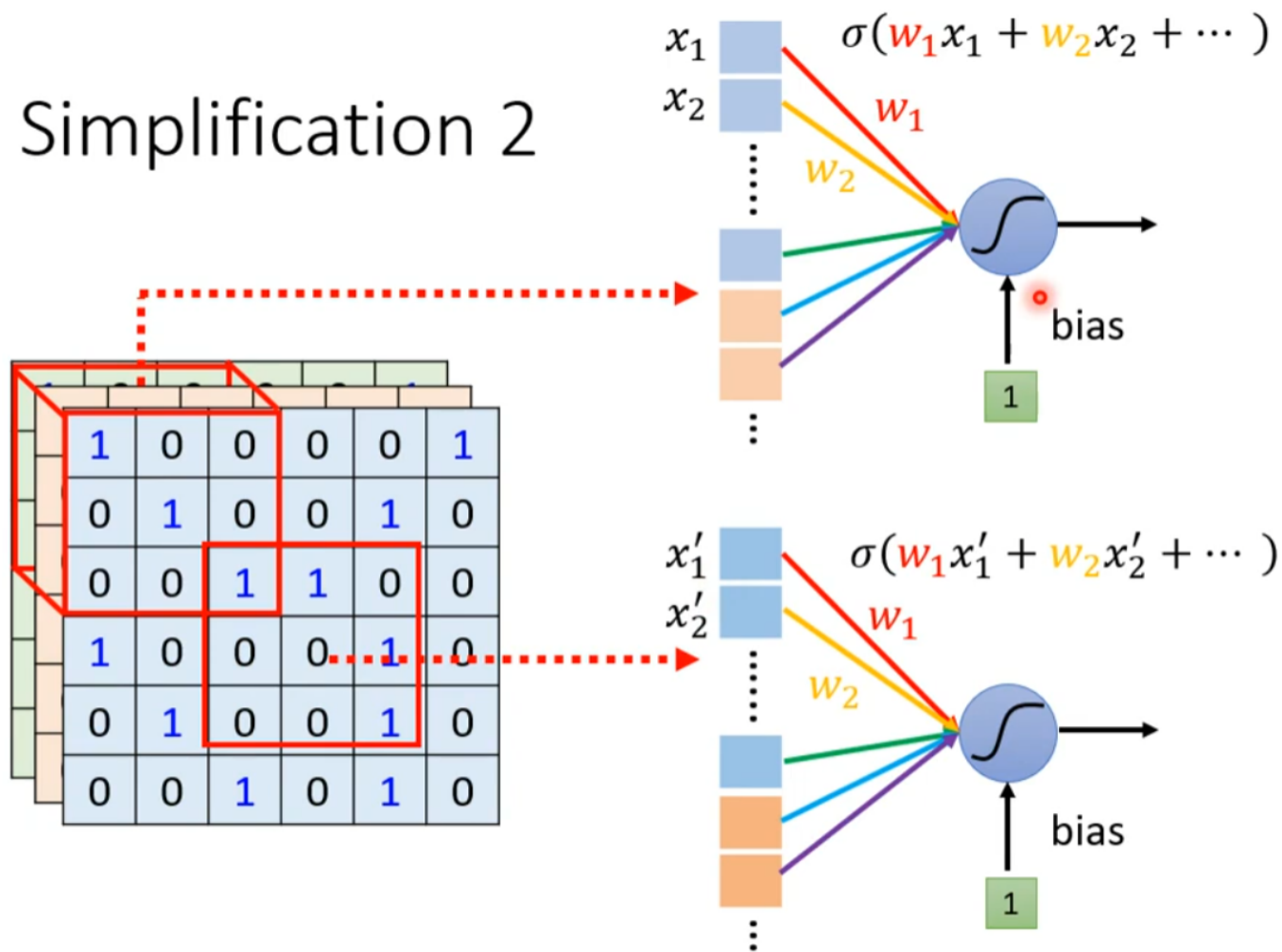
区域的形状? 边的大小? 边缘这么处理? -> padding

- 特征可能出现在不同的区域
- The same patterns appear in different regions.



是否需要对多个区域单独检测? -> 共享参数

## Simplification 2

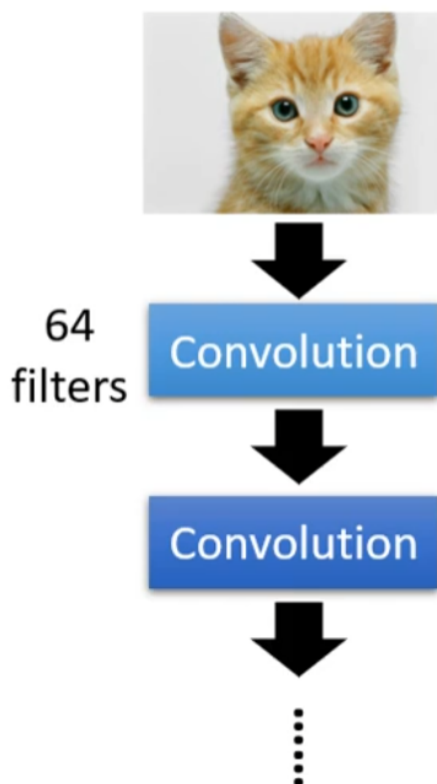


- subsampling (拿掉一些相邻的像素点) -> pooling

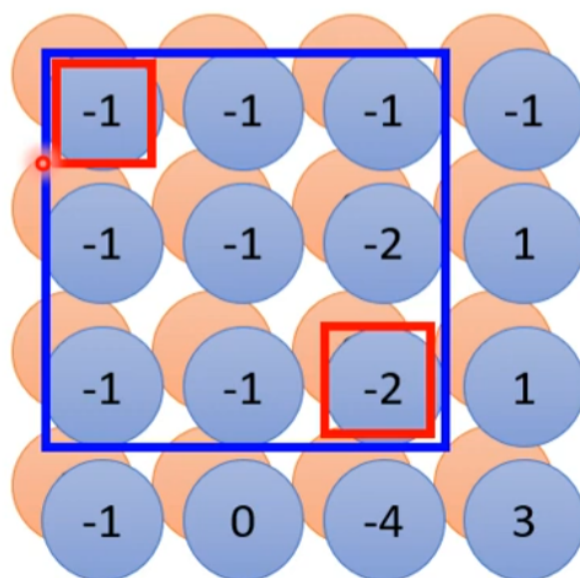
## Convolutional Layer卷积层运算原理

一个卷积核的数值都是未知的（未知参数），每个核对每一个3\*3进行张量乘法运算。多个卷积核扩大channel。

## Multiple Convolutional Layers



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

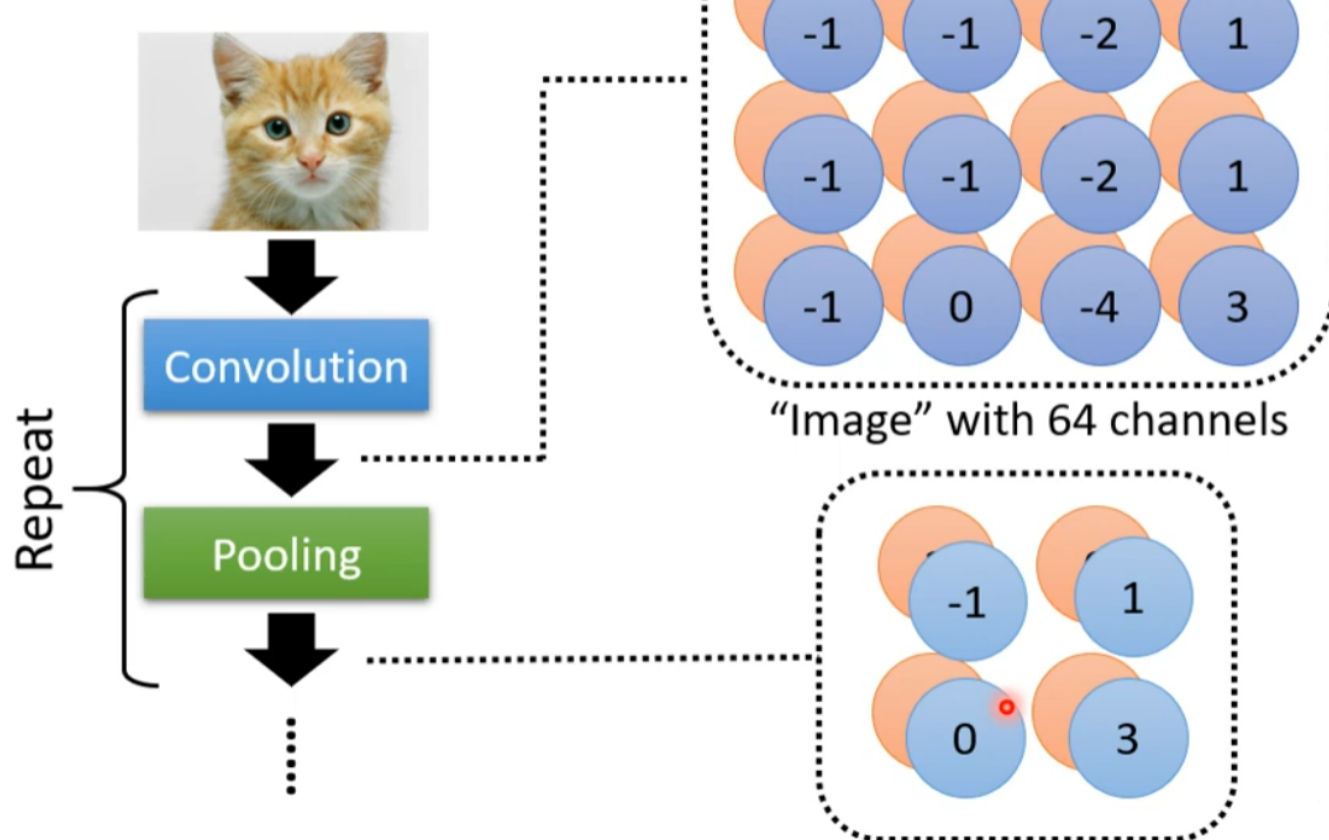


卷积层越深看的范围越大

## Max Pooling

在每个分块里选最大的像素值

## Convolutional Layers + Pooling



一般卷积层和池化层交替出现，不断选取有代表性的 -> 减少运算量；但针对细节的图像任务这种方法是不可取的，因此如果算力足够是可以不用做池化的

## 最后回归引入问题

最后flatten用全注意力softmax分类