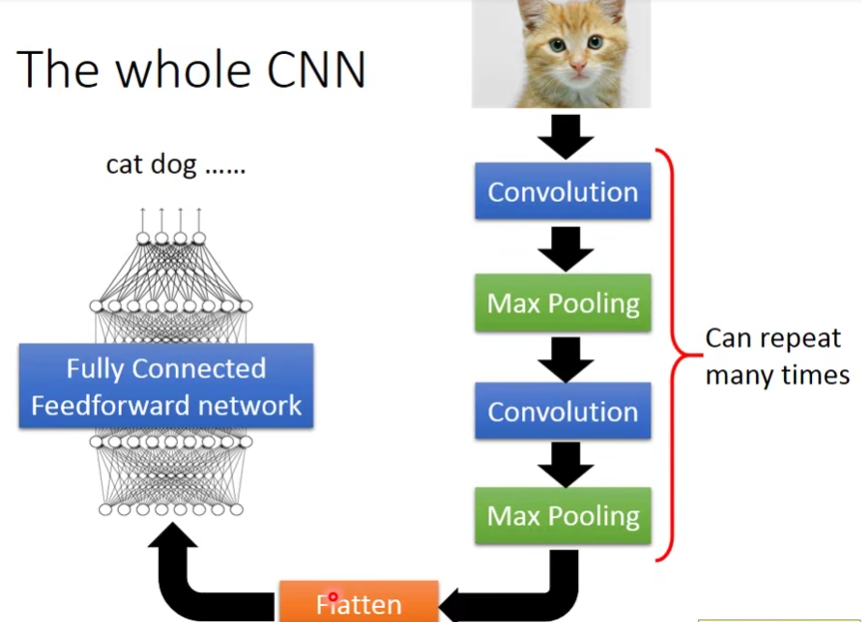
# 机器学习笔记

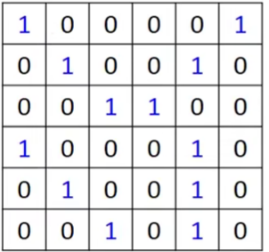
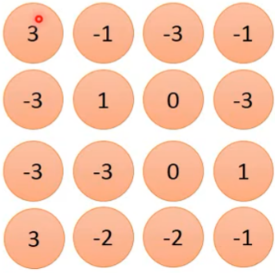
李宏毅教授机器学习(视频地址：https://www.bilibili.com/video/BV1JE411g7XF?p=3&vd\_source=0a27b8c0284e7cd3b72bc0c89478f2d7)

# 初学Convolutional Neural Network(卷积神经网络)

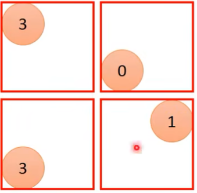
CNN常常用于影像处理，其大致步骤如图所示，



CNN-Convolution

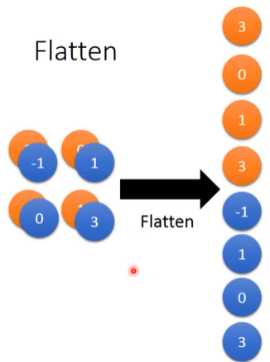
假设我们输入的图片是用矩阵来表示的，如右图，其中0/1分别代表白/黑色，在convolution layer里面，我们有一组filter，每个filter也是一个矩阵，如，然后我们将每个filter对原矩阵的一部分进行内积，然后挪动一个距离stride，这个距离取决于我们自己，如果stride=1，我们用filter 1对原矩阵经过convolution过程之后就得到一个4\*4的矩阵，对于其他的filter的操作以此类推，然后我们得到的若干个矩阵我们把他们称为feature map。如果图片是彩色的话我们就用红绿蓝三种矩阵来表示，就相当于一个立方体，而我们的filter也需要是一个立方体。

CNN-Max Pooling

经过convolution之后我们得到的输出还是比较大，所以我们需要对其进行采样，我们对每一部分取平均值或者最大值保留，使得image缩小，比如filter 1对应的矩阵就变成，所以做完一次convolution和一次max pooling之后就将原来的image缩小了，如上面原来6\*6的image就变成了2\*2的image，而新得到的image的深度取决于我们filter的数量，有多少个filter深度就是多少。

Convolution->Max pooling这个步骤可以循环多次去缩小image。

Flatten

Flatten做的事情就是将feature map拉直，如图所示，然后就可以放进fully connected feedforward network里面。

CNN是如何学习的？

假设我们有若干个filter，我们用来代表第k个filter，第i行j列的元素，然后我们对每个元素进行求和，用其表示第k个filter的作用大小，然后我们需要找一张image可以让第k个filter的作用最大化，，我们可以用gradient descent来完成这件事。

但是我们这样找出来的image可能完全不符合我们的要求，所以我们需要对其进行一些限制，令，其中是每个像素上对应的值。

这样的做法也是deep dream的精神，我们将filter的绝对值调大，也就是正值更大，负值更小，然后去做gradient descent。

CNN应用的范围是十分广泛的，之前很火的alpha go也有运用了CNN的技术。在自然语言识别中也经常运用到CNN。