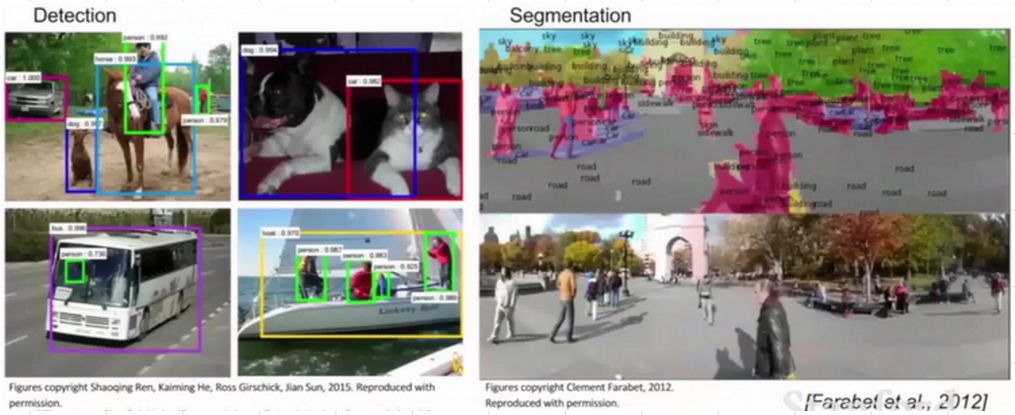
L5卷积神经网络：卷积和优化

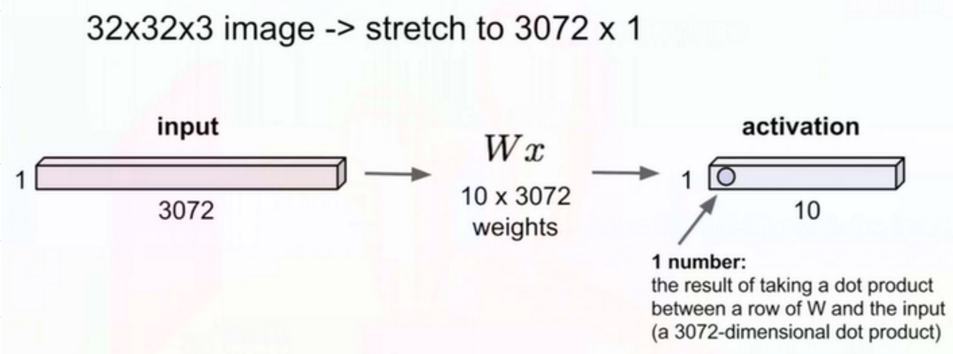
2020年4月10日　15:15

1. 目前神经网络的应用
   a. convnet：

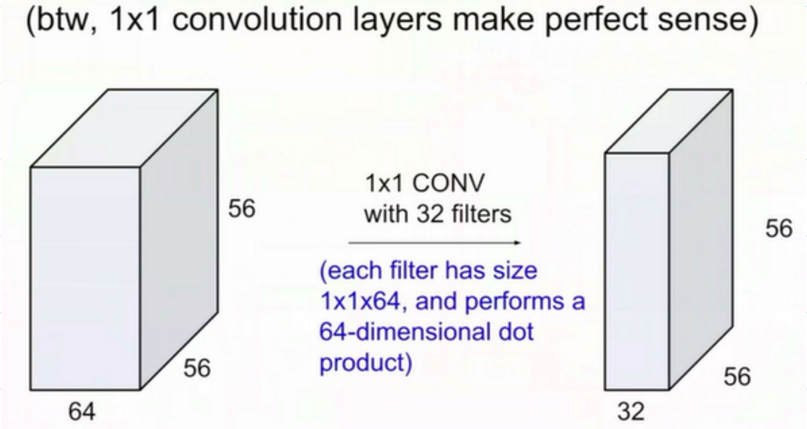


      i. 识别图片中物体，绘制边界框，标记轮廓像素，人脸识别，视频分类。姿势识别，增强学习任务，医学图像解释诊断，星系分类，路标识别，图像描述

2. 工作原理：
   a. 全连接层



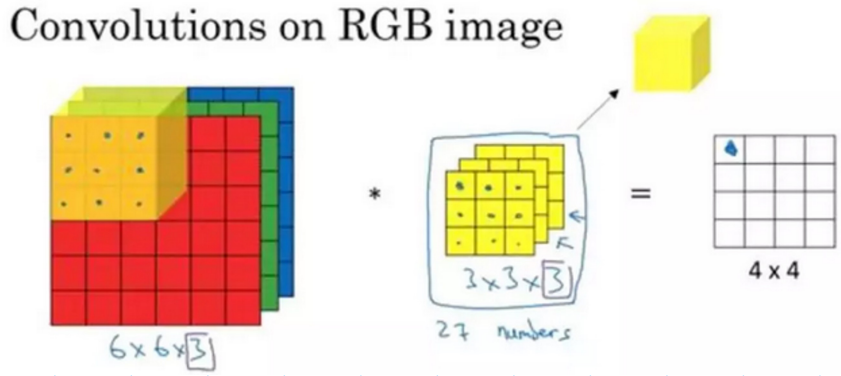
   b. 卷积层（不会破坏图片原有的空间结构，如图片数据不用展开为一行，仍是三维结构）
      i. 卷积核常见大小：3x3，5x5，7x7



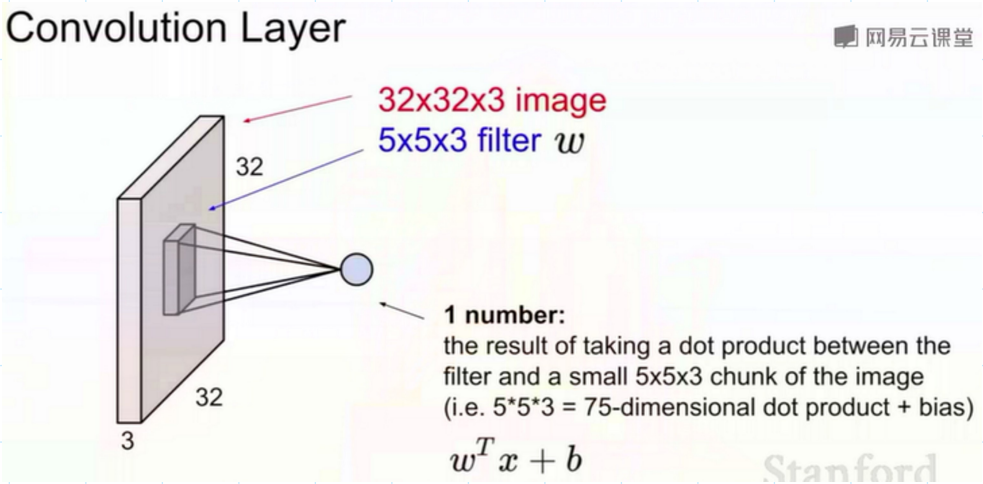
      ii. 卷积核（滤波器）使用方法：使用卷积核滑动进行空间定位后，计算每一定位的点积（卷积核与选中的图像块）

此处点积计算时需要对数据进行展开：

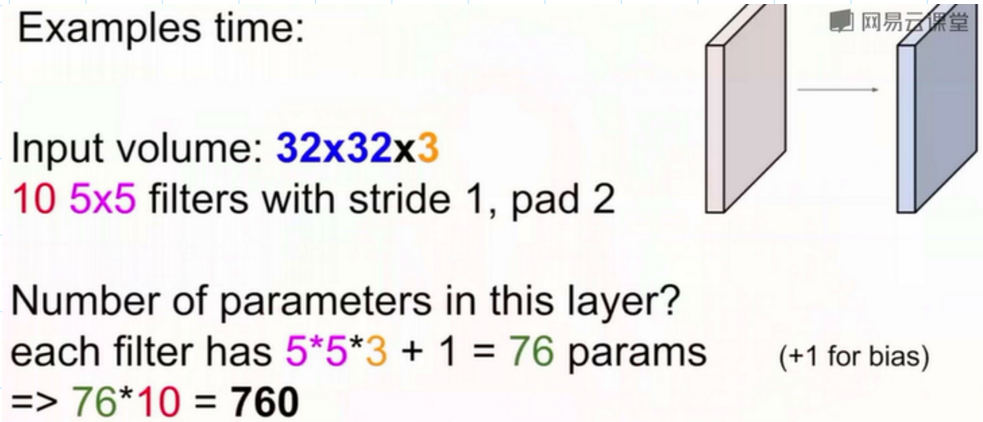
那么究竟如何理解三通道的卷积过程？单通道的卷积操作我们已经知道，就是直接对感受野与滤波器进行元素相乘求和，那三通道呢？我们可以将 3*3*3 的滤波器想象为一个三维的立方体，为了计算立方体滤波器在输入图像上的卷积操作，我们首先将这个三维的滤波器放到左上角，让三维滤波器的 27 个数依次乘以红绿蓝三个通道中的像素数据，即滤波器的前 9 个数乘以红色通道中的数据，中间 9 个数乘以绿色通道中的数据，最后 9 个数乘以蓝色通道中的数据。将这些数据加总起来，就得到输出像素的第一个元素值。如下图所示：



iii. 卷积核通道数（空间深度）与输入图像应该一致（图中的T不等同于转置，表示展开为长向量）



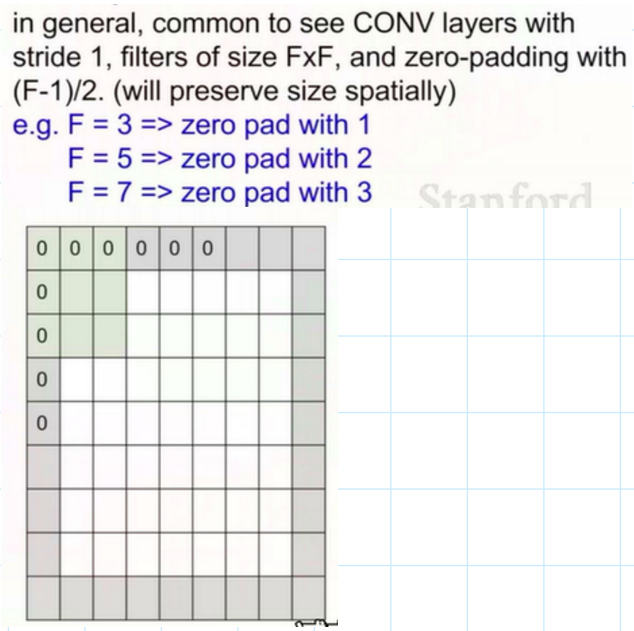
iv.



注意偏差项bias

v. 零填补：可以用0对输入值进行填补后再使用卷积核，从而使输入值与输出值大小相同（全尺寸输出以避免丢失过多边角信息）

不同卷积核、不同步长进行零填充所用宽度不同



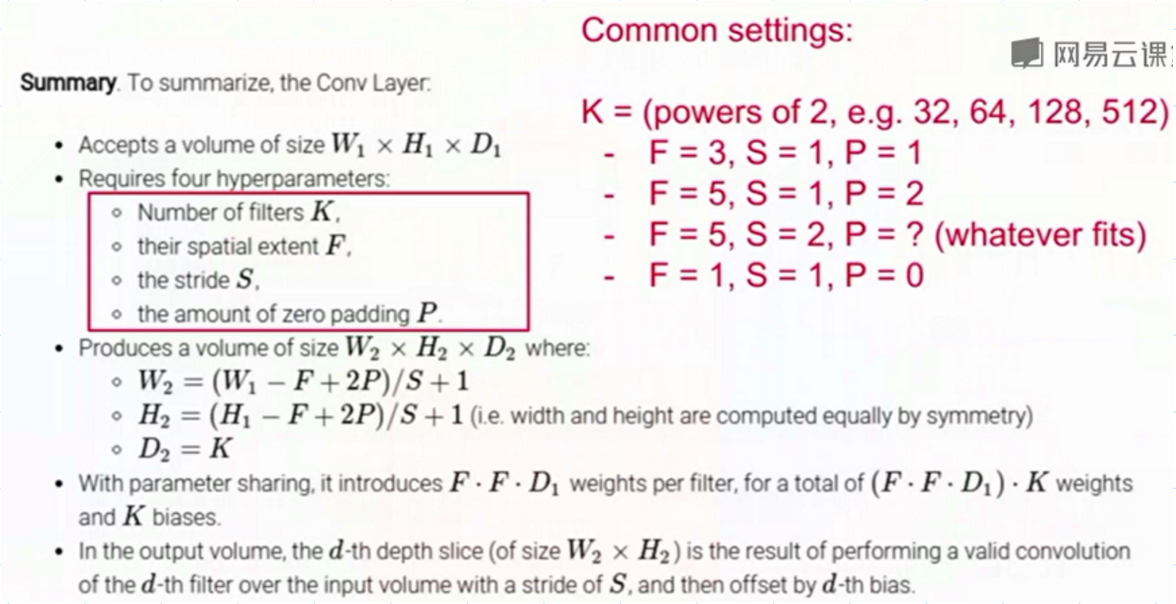
vi. 卷积核滑动方式：从左上角开始，按选定步长（会影响输出/激活映射的数据结构）滑动，遍历所有像素

1) 步长选择应使卷积核和图像能够拟合

2) 横纵步长可不同但不常用

vii. 可使用多个卷积核对一个输入进行卷积得到多个激活映射，n个卷积核对应n个激活映射并整合为一个输出（output size： （N-F)/stride+1）

viii.



c. 卷积神经网络的构成

i. 一个神经网络一般由一系列的卷积层堆叠而成，其中穿插使用一些激活函数（CONV，ReLU，池化层），其中每一层的输出又可作为下一层的输入，对最后输出使用全连接层输出各类权重。

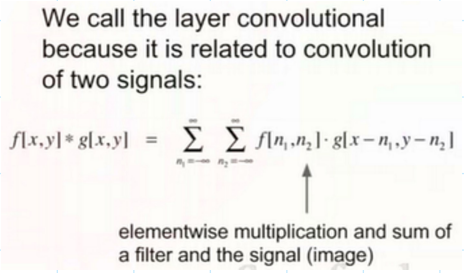
ii. 完成前面几层的卷积核的学习后一般会得到一些低阶的图像特征：边缘……

中间 复杂 边角、斑点

iii. 训练完成后的卷积层的可视化输出展示了什么样的输入可以使激活函数在该神

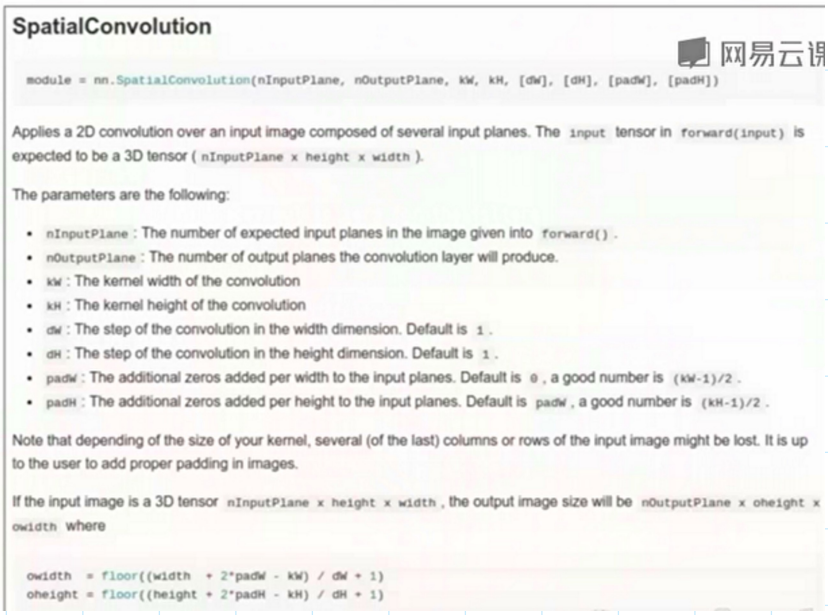
经元的输出最大化

d. “卷积”的由来：



e. 框架举例：

i.



SpatialConvolution

网易云课

```
module = nn.SpatialConvolution(nInputPlane, nOutputPlane, kW, kH, [dW], [dH], [padW], [padH])
```

Applies a 2D convolution over an input image composed of several input planes. The `input` tensor in `forward(input)` is expected to be a 3D tensor (`nInputPlane x height x width`).

The parameters are the following:

- `nInputPlane` : The number of expected input planes in the image given into `forward()` .
- `nOutputPlane` : The number of output planes the convolution layer will produce.
- `kW` : The kernel width of the convolution
- `kH` : The kernel height of the convolution
- `dW` : The step of the convolution in the width dimension. Default is `1` .
- `dH` : The step of the convolution in the height dimension. Default is `1` .
- `padW` : The additional zeros added per width to the input planes. Default is `0` , a good number is `(kW-1)/2` .
- `padH` : The additional zeros added per height to the input planes. Default is `padW` , a good number is `(kH-1)/2` .

Note that depending of the size of your kernel, several (of the last) columns or rows of the input image might be lost. It is up to the user to add proper padding in images.

If the input image is a 3D tensor `nInputPlane x height x width` , the output image size will be `nOutputPlane x oheight x owidth` where

```
owidth  = floor((width  + 2*padW - kW) / dW + 1)
oheight = floor((height + 2*padH - kH) / dH + 1)
```

f. 池化层：降采样处理

i. 最大池化法：

1) 解释a

a) 设定池化处理的区域大小（or滤波器的大小）

b) 设定步长使滤波器滑动后不会重叠（通常这么做）

c) 滤波器输出所在区域的最大值

2) 解释b

a) 将要处理的数据划分为各个相同大小的区域

b) 取各区域中的最大值

ii. 均值池化

iii. tips：

Accepts a volume of size $W_1 \times H_1 \times D_1$

Requires three hyperparameters:

- their spatial extent F,
- the stride S,

Produces a volume of size $W_2 \times H_2 \times D_2$ where:

- $W_2 = (W_1 - F)/S + 1$
- $H_2 = (H_1 - F)/S + 1$
- $D_2 = D_1$

Introduces zero parameters since it computes a fixed function of the input

Note that it is not common to use zero-padding for Pooling layers

F = 2, S = 2

F = 3, S = 2

iv. 可以只进行步长滑动代替池化或只进行池化不滑动