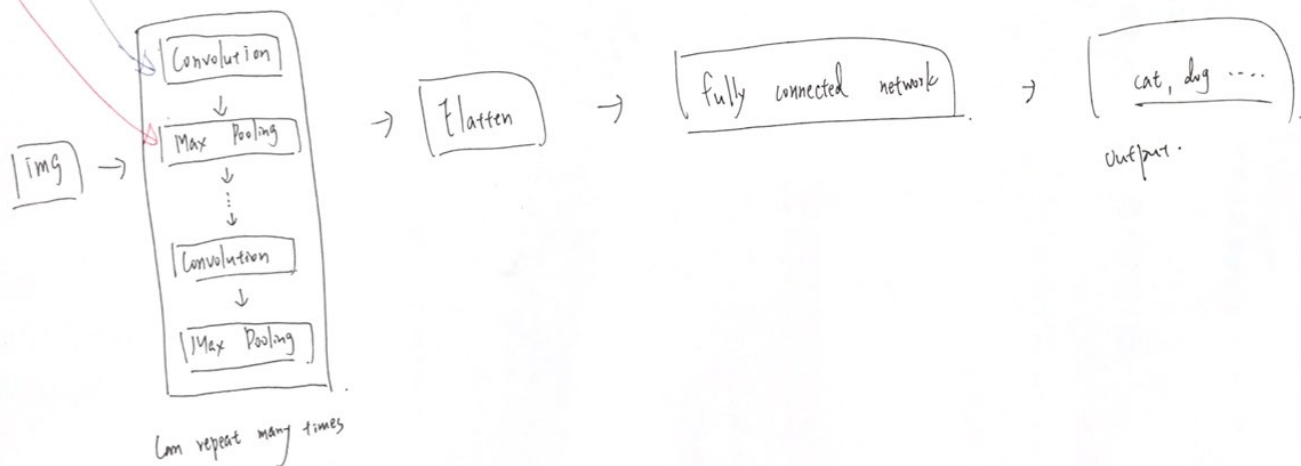


一. pattern 只佔 img 一部分, 所以 neuron 偵測一小部分即可, 不需看整張 img.
每個 neuron 連接到小區域, 不需連接到整張圖.

二. 同樣的 pattern 出現在 img 的不同區域, 使用同樣的 ~~參數~~ 參數, ~~即可~~ detector 出來. ^{即可 detection.}

⇒ 要求兩組 neuron 使用同一組參數.

三. 對 img 做 subsampling (抽奇數行、偶數列的 pixel 拿掉) 成原 1/2 大小, 將 img 變小以減少參數.



Convolution

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

6x6 img (黑白)

Filter 1 与 img 做内积.
stride = 1, 使测度移动一格.
得 4x4 Matrix

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Filter 1 要检测的 pattern
位于 img 的左上方与左下方
∴ 使用 Filter 1 即可

-1	-1	-1
-1	1	-1
-1	-1	1

Filter 1 Matrix

Filter 2 Matrix

...

* Each filter detects a small pattern (3x3)
These are the network parameters to be learned.

Filter 2 与 img 做内积
stride = 1

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

得第二 4x4 的 matrix

* Matrix 合起来
称为 Feature Map

几个 Filter 得几个 img
ex. 100 个 filter → 100 张 img

CNN 自动处理
不同 scale 的问题
→ 须从 CNN 前接另 network,
将 img 的某些位置展新, 缩放

Colorful img

img
input 3x3x6

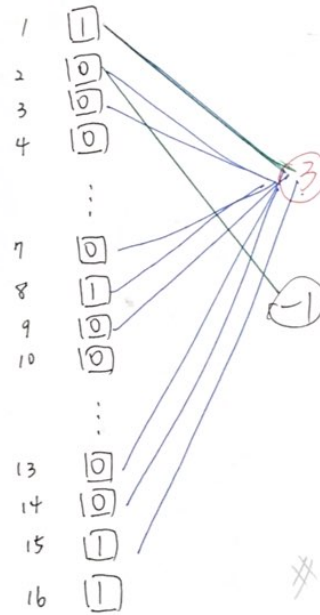
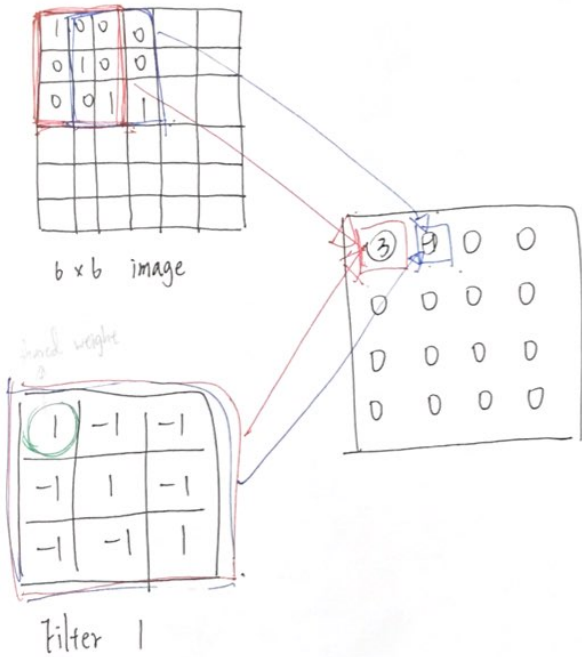
Filter
input 3x3x3

Convolution

→ Filter 的 9 个值 与 img 对应的 9 个值 做内积.

∴ 每一个 Filter 含 ~~不同颜色所代表~~ 的 channel
不同.

∴ 将 channel (每一个 RGB 颜色的 matrix) 合在一起算.



neuron

原本应连接 36 个 input

→ ∴ detect 一个 pattern
不需看整张图

∴ Only connect to 9
input.

→ 用较少参数

* 原本每个 neuron 有各自独立
的 weight.

Conv → 减少每个 neuron 连接的
input.

这些 neuron 共用同 weight

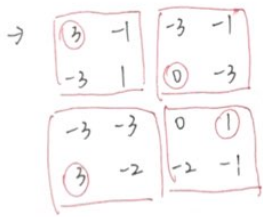
③ — pixel 1. weight

11

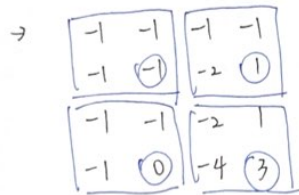
① — pixel 2. weight

Max Pooling 將 feature map 降維。

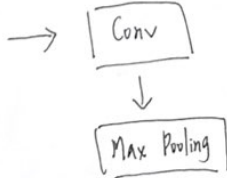
Filter 1 的 img Conv



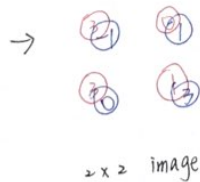
Filter 2



將原來 4 个 value 合成 1 个 value。
以縮小 image，使參數減少



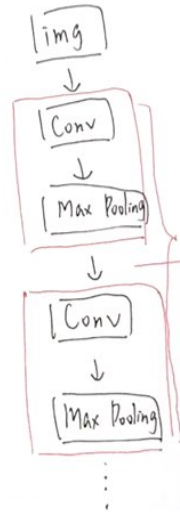
New image
but smaller



* 每个 pixel 的深度
以 Filter 数量決定

ex. 50 个 Filter → 50 維

例子 25 个 Filter → 25 維



A new img

Filter 25 个
feature map 25 个

↓
output 25 个

Filter 25 个

feature map ~~25 个~~

output 25 个