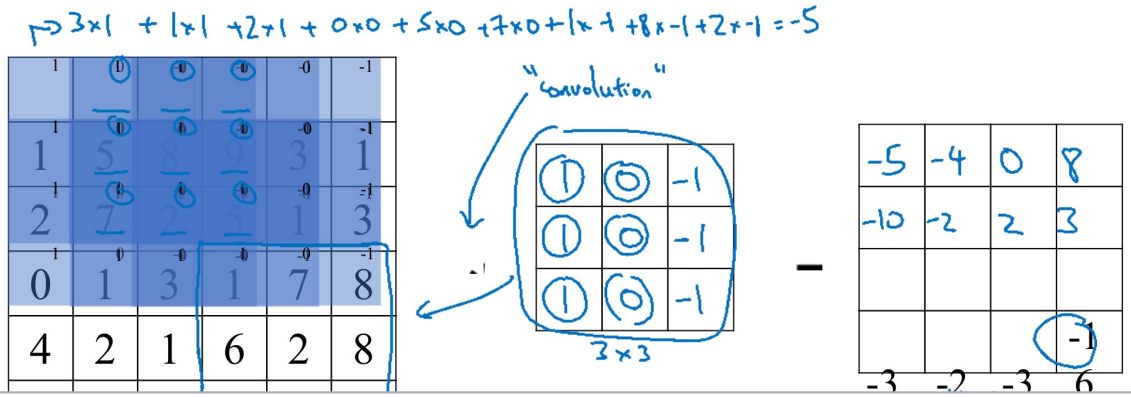
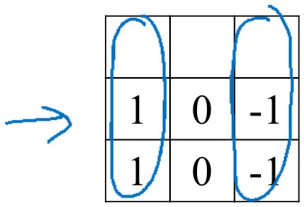
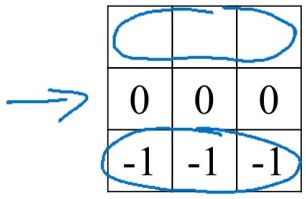
Vertical edge detection



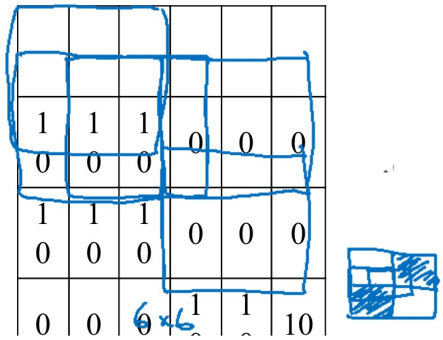
Vertical and Horizontal Edge Detection

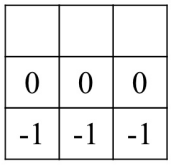


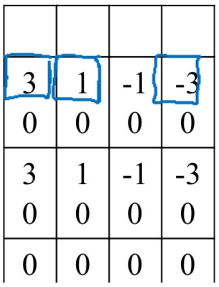
Vertical



Horizontal

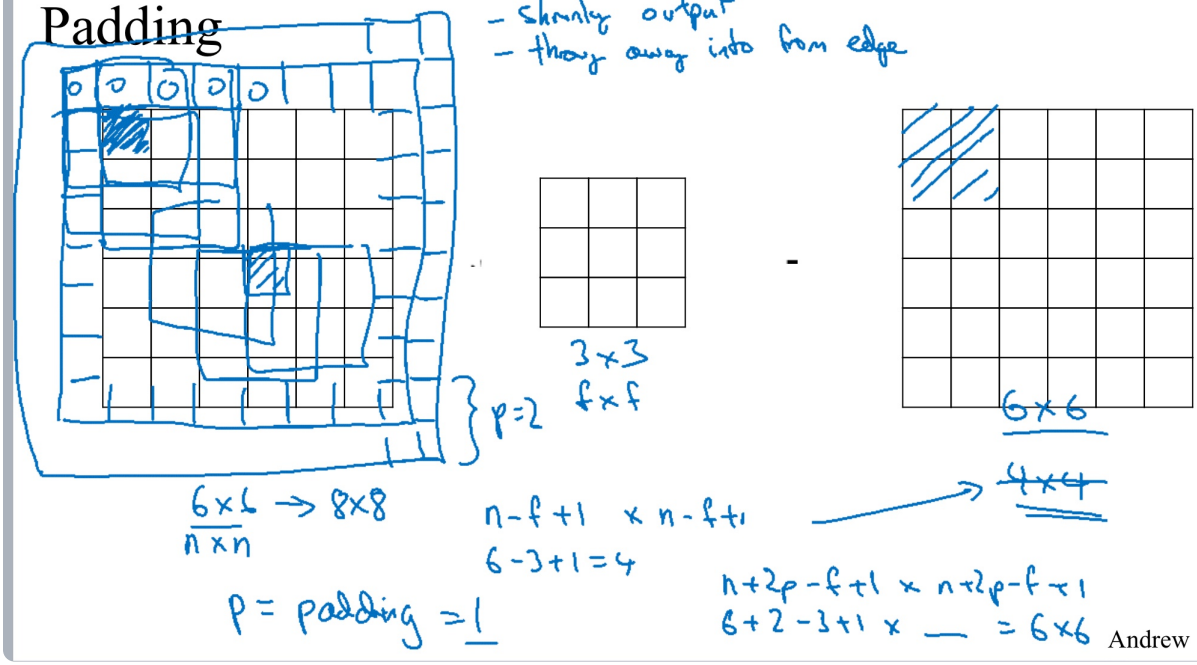






Andrew

Padding



Andrew

→ $n+2p-f+1$ ⇒ o/p

Valid and Same convolutions

↳ no padding

"Valid": $n \times n \quad * \quad f \times f \quad \rightarrow \quad n-f+1 \times n-f+1$

$6 \times 6 \quad * \quad 3 \times 3 \quad \rightarrow \quad 4 \times 4$

"Same": Pad so that output size is the same as the input size.

$n+2p-f+1 \times n+2p-f+1$

$n+2p-f+1 = n \Rightarrow p = \frac{f-1}{2}$

$3 \times 3 \quad p = \frac{3-1}{2} = 1$ | $5 \times 5 \quad f=5 \quad p=2$

f is usually odd

1×1
3×3
5×5
7×7

Andrew

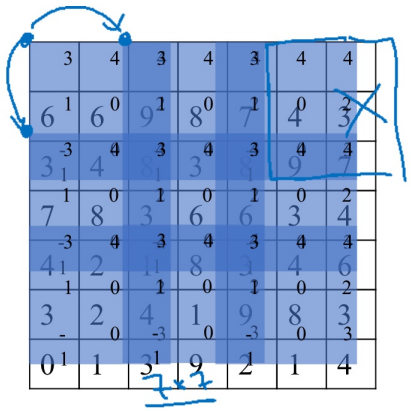
$p = \frac{f-1}{2}$

300×300×3

- - - - - - - - - 100×100×3

90000

Strided convolution



1	0	2
-1	0	3

3×3

stride = 2

$\lfloor z \rfloor = \text{floor}(z)$

91	100	83
69	91	127
44	72	74

3×3

$n \times n$ * $f \times f$

padding p stride s

$s = 2$

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

$\frac{7+0-3}{2} + 1 = \frac{4}{2} + 1 = 3$

Andrew

Summary of convolutions

$n \times n$ image $f \times f$ filter

padding p stride s

300 × 300 × 3

5×5 × 100

300 − 5

295 + 1

296

Output size:

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

Andrew

$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$

15×15×8 $p=2$

15+2+2

$\frac{63-7}{1}+1$ 56

$p = \frac{f-1}{2}$ $\frac{7-1}{2} \Rightarrow 3$

63+6−7

15×15×8

15+4

63×63×16

7×7

$s \to 2$

$p \to 0$

$\frac{63+0-7}{2}+1$

$\frac{56}{2}$ 29+

Technical note on cross-correlation vs. convolution

Convolution in math textbook:

6	6	9	8	7	4
3	4	8	3	8	9
7	8	3	6	6	3
4	2	1	8	3	4
3	2	4	1	9	8

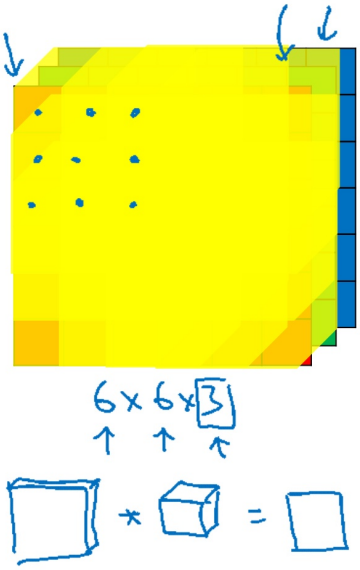
1	0	2
-1	9	7

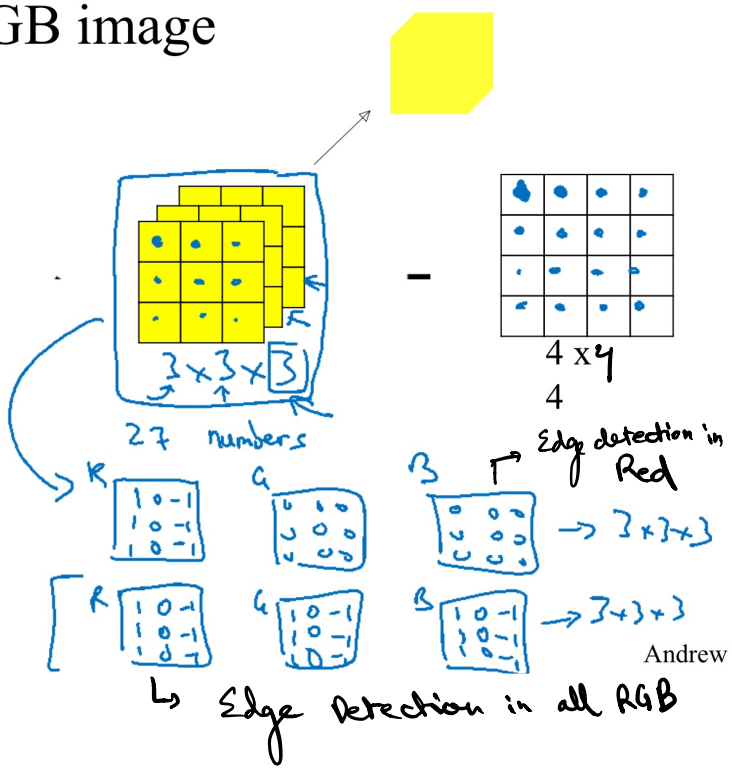
7	2	5
9	0	4
-1	1	3

$(A * B) * C = A * (B * C)$

Andrew

Convolutions on RGB image

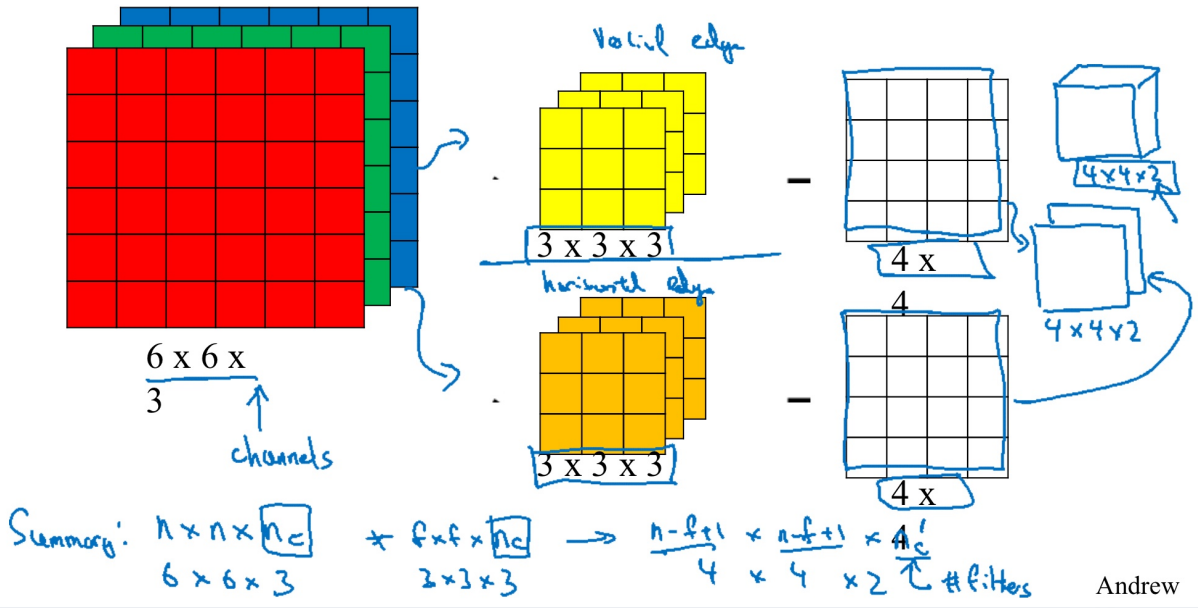




Andrew

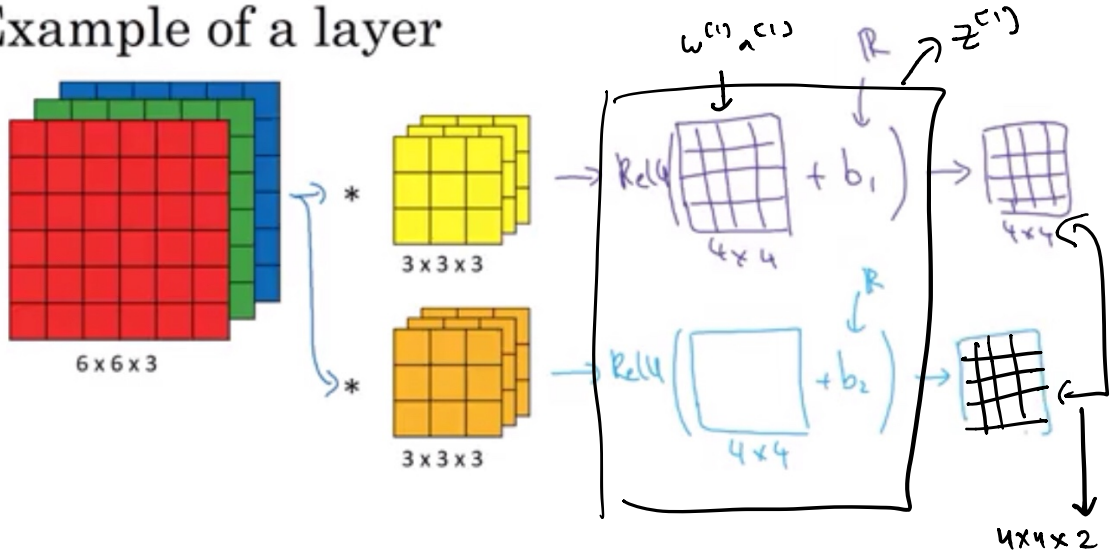
↳ Edge detection in all RGB

Multiple filters



Andrew

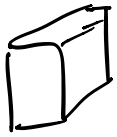
Example of a layer

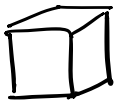


Andrew Ng

No. of parameters in 1 layer

10 filters, 3x3x3 in one layer of neural network





$3 \times 3 \times 3 = 27 + \text{bias} = 28$ ... $\times 10 \Rightarrow 280$ parameters

$f^{[l]}$ → filter size

$p^{[l]}$ → padding

$s^{[l]}$ → stride

$n_c^{[l]}$ → number of filters

Input: $n_H^{[l-1]} \times n_w^{[l-1]} \times n_c^{[l-1]}$

Output: $n_H^{[l]} \times n_w^{[l]} \times n_c^{[l]}$

$$n_w^{[l]} = \left\lfloor \frac{n^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor$$

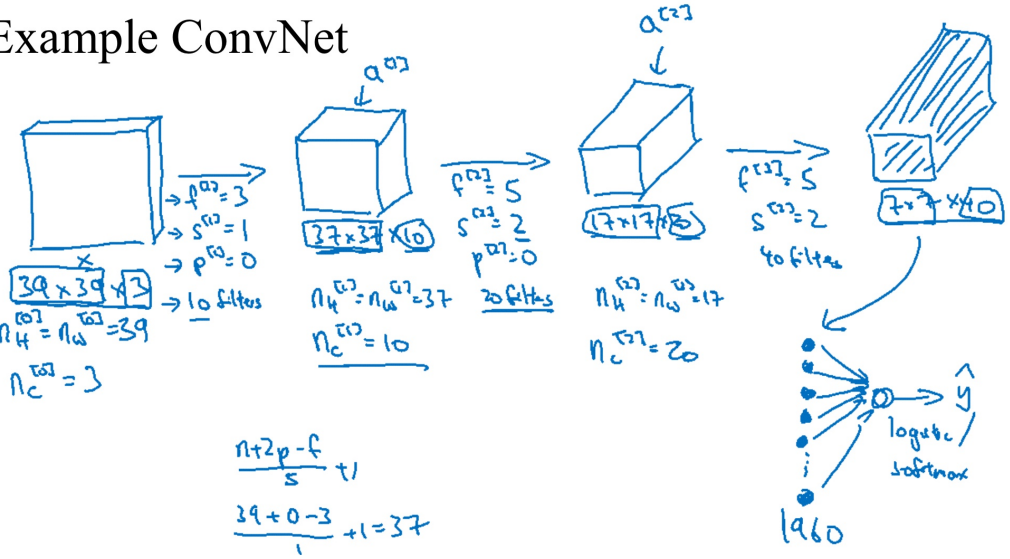
Each filter's $f^{[l]} \times f^{[l]} \times n_c^{[l-1]}$

Activation $a^{[l]} = n_H^{[l]} \times n_w^{[l]} \times n_c^{[l]}$

$A^{[l]} = m \times n_H^{[l]} \times n_w^{[l]} \times n_c^{[l]}$

Weights = $f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]}$

Example ConvNet



$\frac{n+2p-f}{s} + 1$

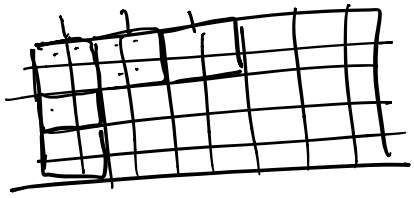
$\frac{39+0-3}{1} + 1 = 37$

Andrew

3 Pooling

→ No parameters to learn during backprop.

32x32X16 — Max pool → → 16X16



f = 2

s = 2

63x63X16

n_f = 32

7X7