CNN-2

Represent an image = C x m x N
↳ channel.

28 x 28

C x m x N

C = 1

C x m x N

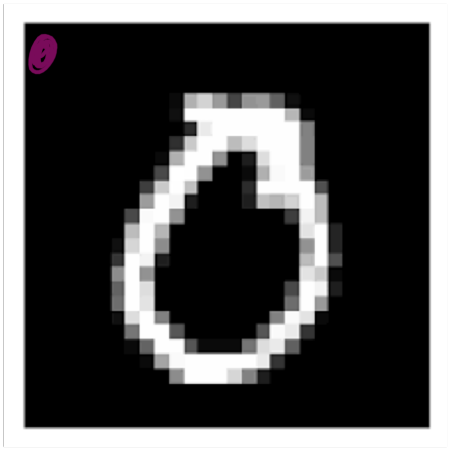
C = 3

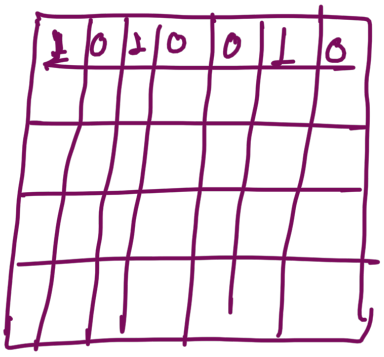
C x m x N

C = 1

C x m x N

= 3



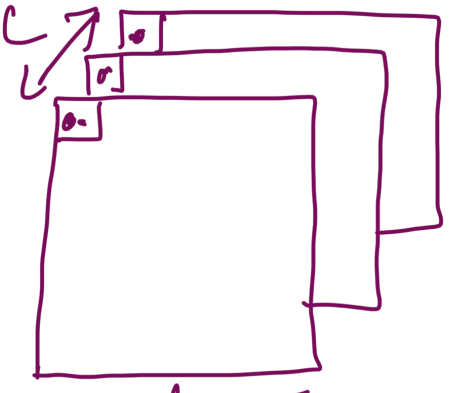


C=1

28*28

C=3

RGB



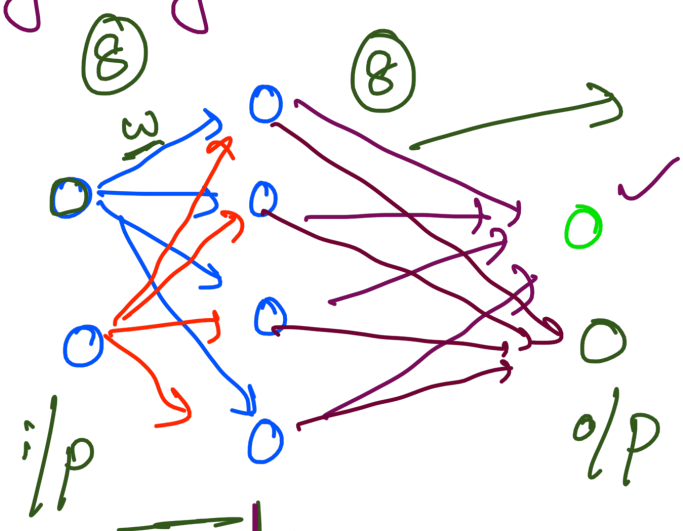
Tensor

Grey Scale vs Color Image (RGB)

Digital Images can be represented as Tensor

Convolutional Neural Network

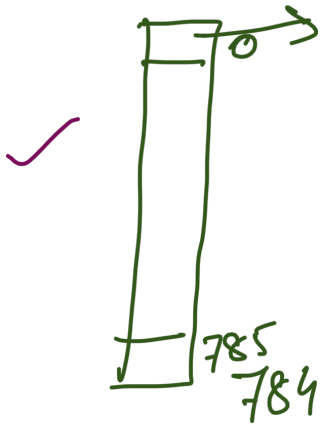
Initially Fully Connected

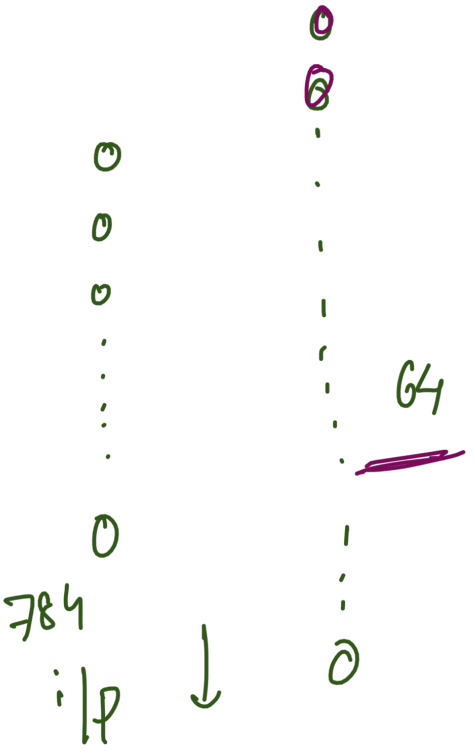


Hidden layers (4 neurons)

# of weights = 0 4 4+4 = 8
0 4

Image: 28x28





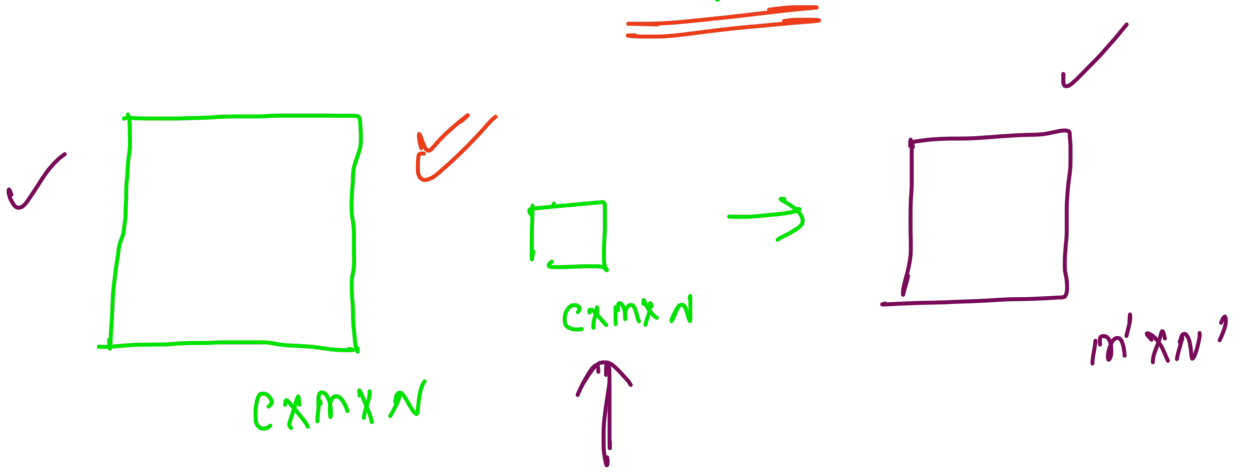
weights = 784 x 64
= 50,176

≈ 50K

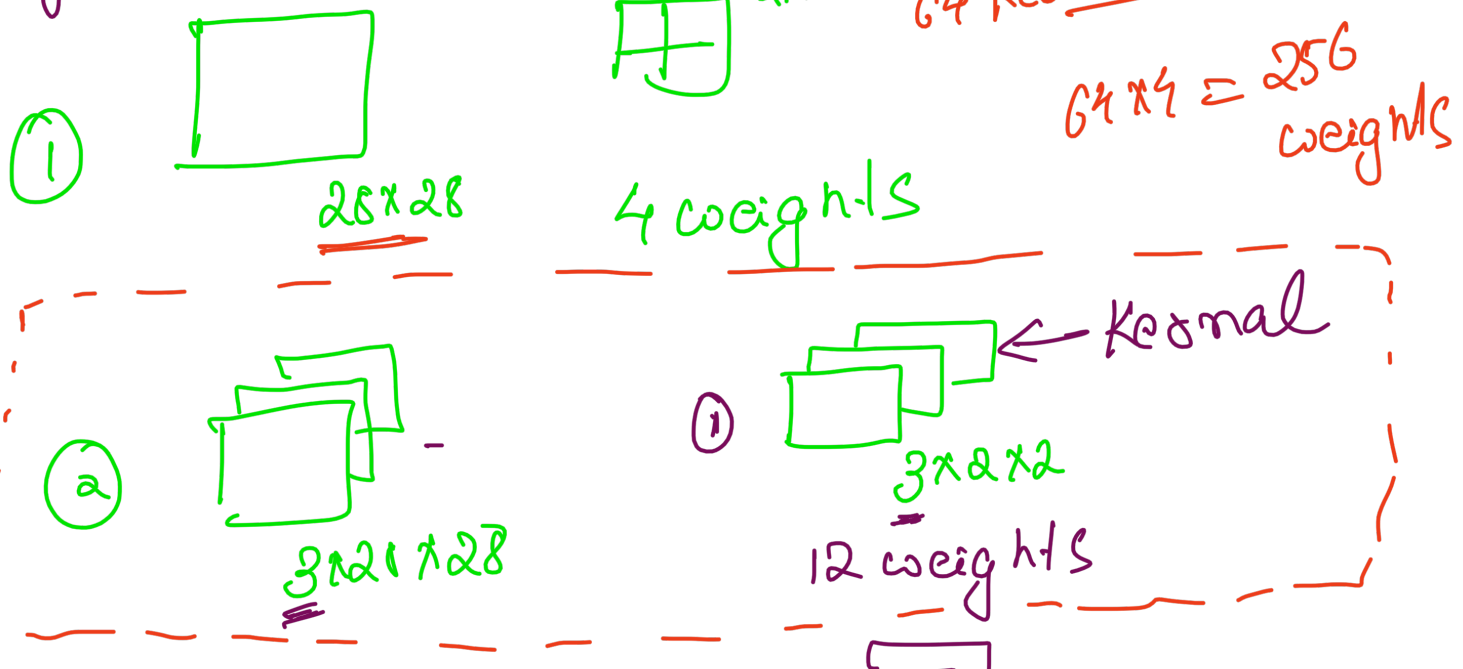
weights that need to be learned is growing exponentially → Disadvantage

↓

CNN



Eg:



Memory

Computation

↓

CNN efficient ✓

64 x 12 = 768 weights

3 x m x N

RGB

8 Kernals

3 x w x h

8 x w x h

C = 1

stride = 1 p = 0

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

i/p 1 x 5 x 5

weights

1	1
1	1

1 x 2 x 2

2	2	2	2

1 channel

Size ?

1 x 1 + 0 x 1 + 0 x 1 + 1 x 1

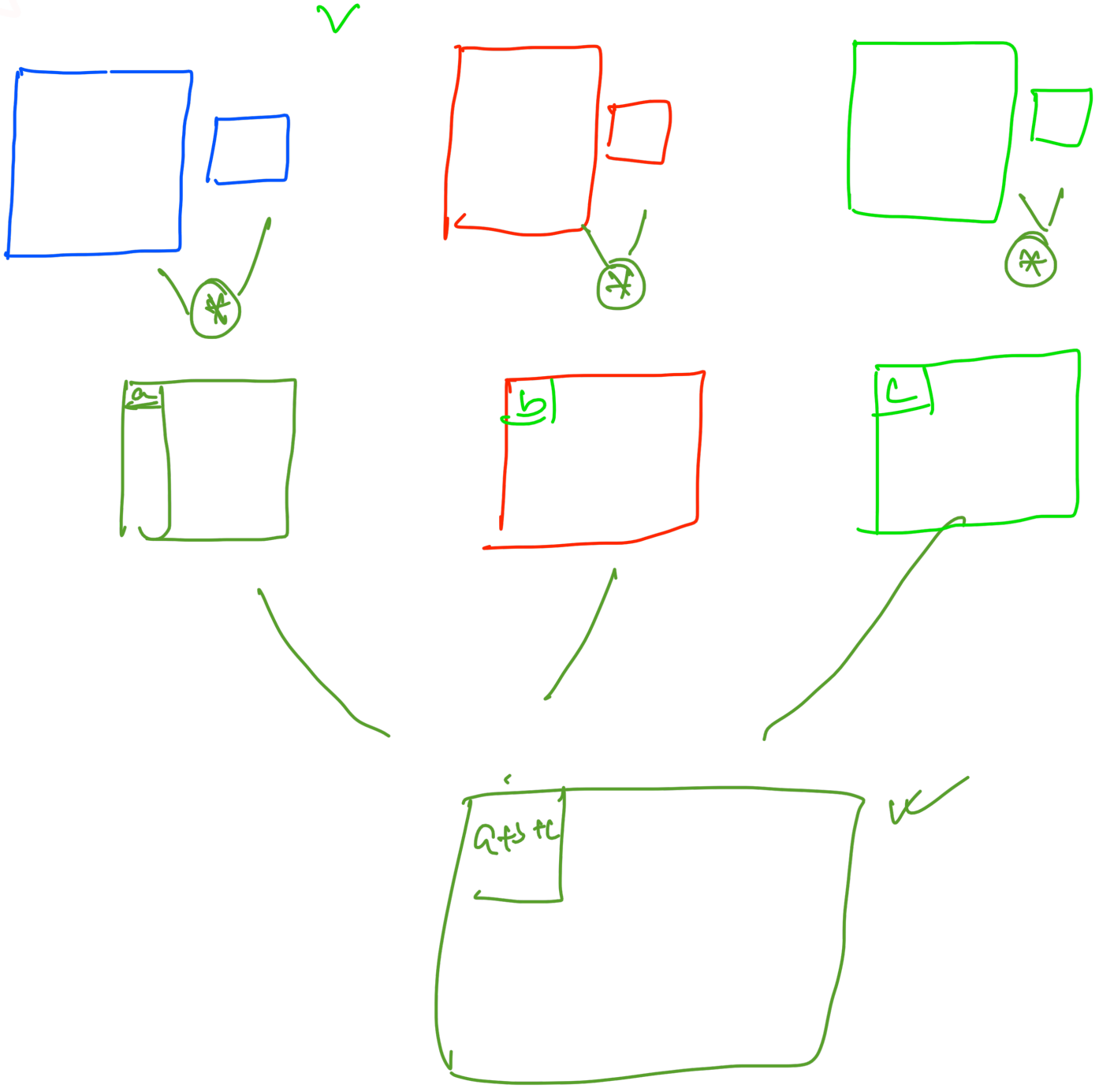
= 2

3 x 5 x 5

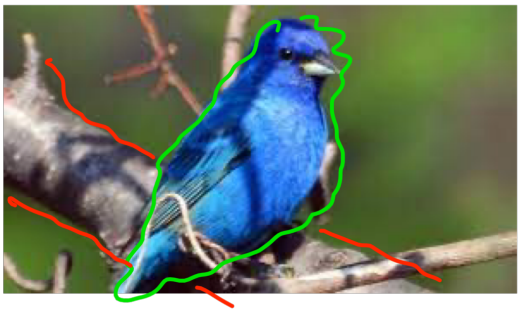
Kernal

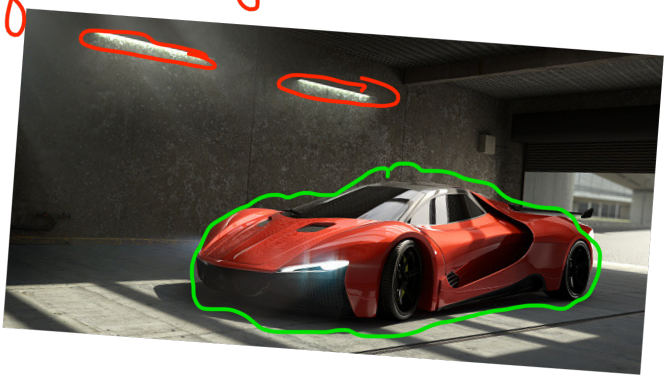
3 x 2 x 2

3 channel


a
b
c

Different Approaches of an algorithm





1. Classification:
   Presence or absence =

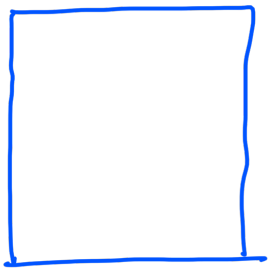
2. Segmentation: ✓

CIFAR-10

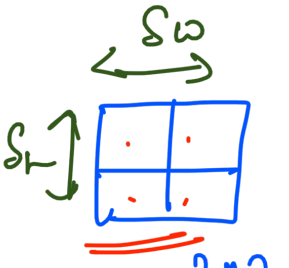
- - - - - - - - -

Stride Padding Polling & Activation Function.

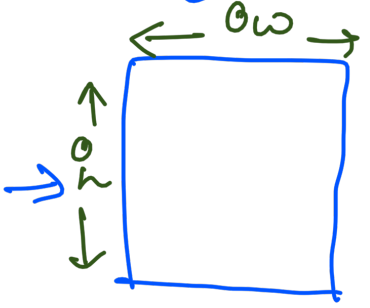
Formula (generalized) output size: Stride = 1, P = 0



5 x 5



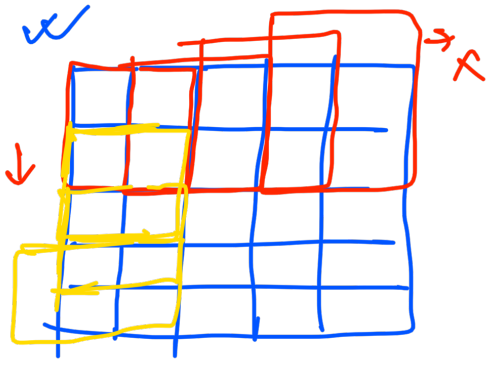
2 x 2
ω x h



4 x 4

o/p.

m x N



Stride = 1

$$O_\omega = \frac{m-\omega}{S_\omega} + 1$$

$$O_h = \frac{m-h}{S_h} + 1$$

Eg: $\frac{5-2}{1} + 1$ ①

$= 3+1$

$= 4$

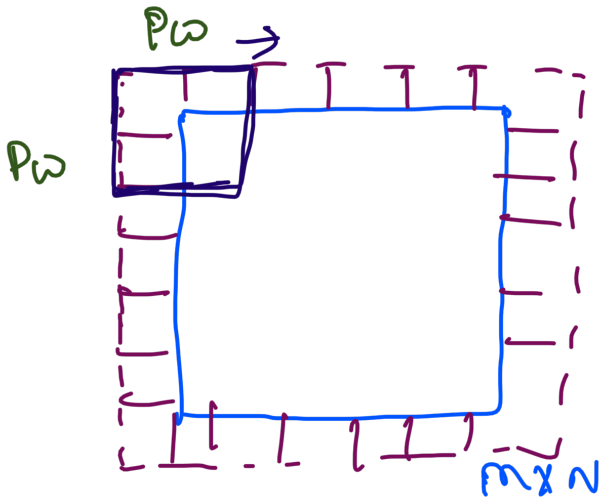
② $\frac{10-2}{1} + 1 = 9$ ✓

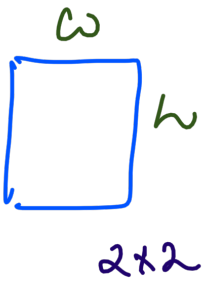
m x N = 10 x 10

ω x h = 2 x 2 → 9 x 9

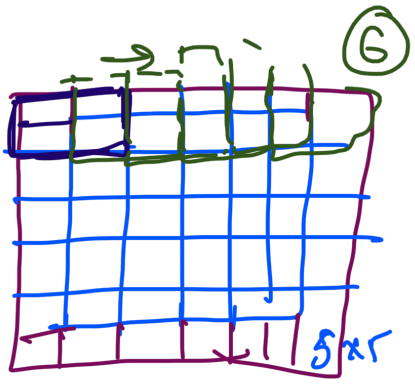
S = 1

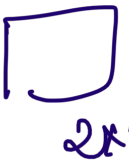
Padding $P=1$ $S=1$

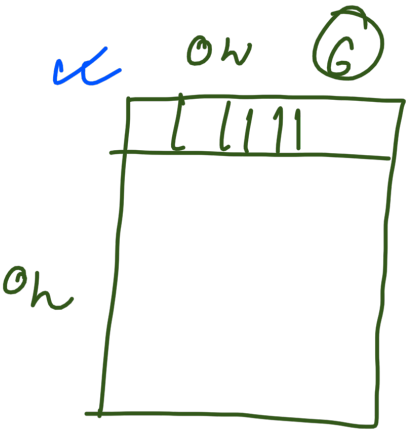




Eg:







$m \times n = 5 \times 5$

Kernal $= 2 \times 2$ $\quad S=1$ $\quad P=1$

$$O_w = O_h = \frac{5-2+2(1)}{1} + 1$$

$$= 5 - 2 + 2 + 1$$

$$= 6$$

$$O_w = \frac{m - w + 2P_w}{S_w} + 1$$

$$O_h = \frac{m - w + 2P_h}{S_h} + 1$$

Stride $= 2$

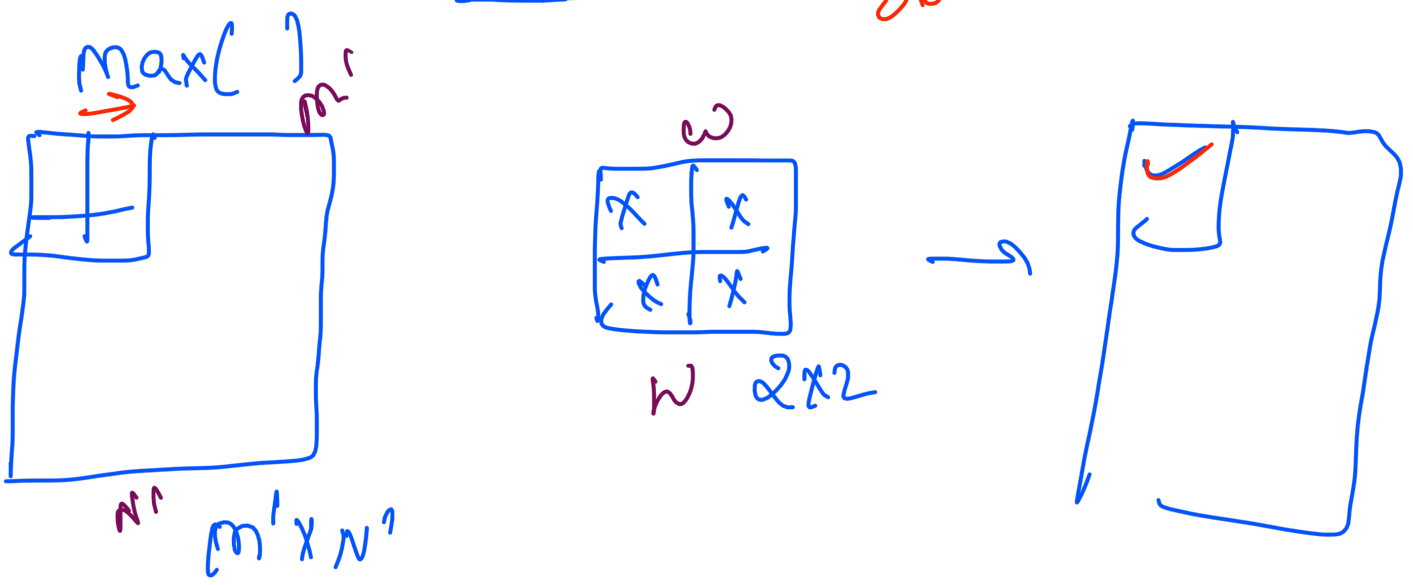
$$O_w = O_h = \left\lfloor \frac{5-2+2(1)}{2} \right\rfloor + 1 = \left\lfloor \frac{5}{2} \right\rfloor + 1$$

$$= 2 + 1$$

$$= 3$$

Max Pool

Stride = 2



Maximum values from those blocks of numbers → MaxPools

— No weights are learned

eg: m x n
5 x 5
s = 1
k = 2 x 2

output of maxpooling:

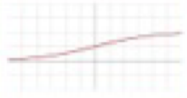
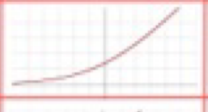
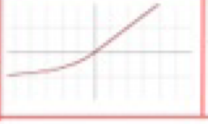
$$\frac{m-w}{s}+1$$

$$\frac{5-2}{1}+1$$

$$= 3+1$$

$$= 4$$

Activation Functions

ACTIVATION FUNCTION	PLOT	EQUATION	DERIVATIVE	RANGE
Linear		$f(x) = x$	$f'(x) = 1$	$(-\infty, \infty)$
Binary Step		$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{if } x \neq 0 \\ \text{undefined} & \text{if } x = 0 \end{cases}$	$\{0, 1\}$
Sigmoid		$f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$	$(0, 1)$
Hyperbolic Tangent(tanh)		$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$	$f'(x) = 1 - f(x)^2$	$(-1, 1)$
Rectified Linear Unit(ReLU)		$f(x) = \begin{cases} 0 & \text{if } x<0 \\ x & \text{if } x\geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ \text{undefined} & \text{if } x = 0 \end{cases}$	$[0, \infty)$
Softplus		$f(x) = \ln(1 + e^{x})$	$f'(x) = \frac{1}{1+e^{-x}}$	$(0, 1)$
Leaky ReLU		$f(x) = \begin{cases} 0.01x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0.01 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	$(-1, 1)$
Exponential Linear Unit(ELU)		$f(x) = \begin{cases} \alpha(e^{x} - 1) & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$	$f'(x) = \begin{cases} \alpha e^{x} & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ 1 & \text{if } x = 0 \text{ and } \alpha = 1 \end{cases}$	$[0, \infty)$