Space & Compute Complexity:

$y = (wx + b)$

→ Space required to store the model's weights.

+

→ Space required to store the activations of 1 forward pass.

→ #operations.

- (a) ⇒ addition → 1 (wx+b)
- (b) ⇒ 1 assignment → wx+b into y.
- (c) ⇒ 1 multiplication → wx.

MAC

⇒

SC ⇒ How much memory is required to do "1" forward of an input?

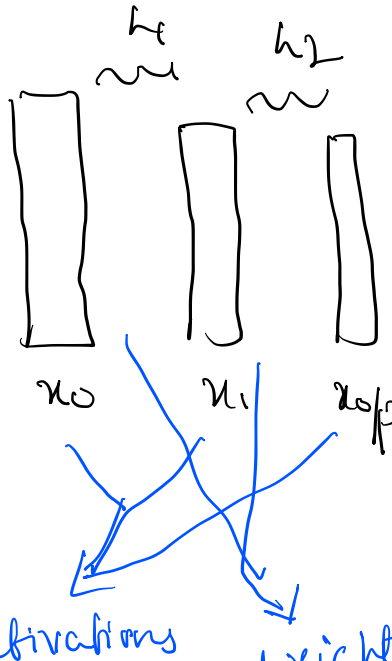
minimum (bs=1)

→ Space for parameters + Space for activations

Space $(L_1 + L_2)$ + Space $(x_0 + x_1 + x_{op})$

↓ #parameters

↓ #activations. activations.



params x datatype

act. x datatype.

- generally ⇒ FP32
- Half pt. precision ⇒ FP16.
- Quantized (8bit) ⇒ 8bit.

Space complexity: is affected by.

- # parameters.
- input size / input activation size.
  - (x_0)
  - batch size (usually checked for 1)
- datatype.

Examples (Lenets)

Input → 6C5W0P1S → (MP) 2W2S → 16C5W0P1S → (MP) 2W2S → 400(FC)120 → 120(FC)84 → 84(FC)10

(x_0) (w_1, b_1) z_1 z_2 (w_2, b_2) z_3 z_4 (w_3, b_3) z_5 (w_4, b_4) z_6 (w_5, b_5) z_7

Consider: i/p ⇒ x_0 ∈ $\mathbb{R}^{1\times32\times32}$

↳ represented as x_0 (writing as x_0 for simplicity)

Activations

Size (x_0) = $1\times32\times32$

= 1,024

Size (z_0) = $6 \times 28 \times 28$

$= \frac{32-5+2(0)}{1} + 1$

$= 28$

Size (z_1) = 4,704.

Size (z_2) = $\in \mathbb{R}^{6\times14\times14}$ = 1,176.

Size (z_3) = 1600

Size (z_4) = 400

Size (z_5) = 120

Size (z_6) = 84

Size (z_7) = 10

Total act = 8,094.

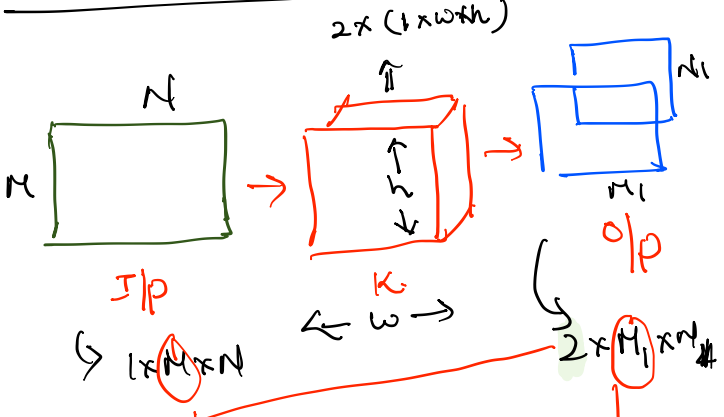
↳ Size = (8,094) × datatype

usually = (FP32)

Parameters

Size (w_1, b_1) =

derivation & cnn formulae.



$N_1 = \left(\frac{N - w + 2P_w}{S_w} + 1\right)$

$N_1 = \left(\frac{N - h + 2P_h}{S_h} + 1\right)$

where is it coming from?

⇓

Kernels in 'k'

⇒ # output channels = # independent kernels in k.

↳ Size of each such kernel?

$$\left[(\#\text{ o/p channels}) \times (\#\text{ i/p channels}) \times w \times h \right]$$

$(6 \times 1 \times 5 \times 5)$

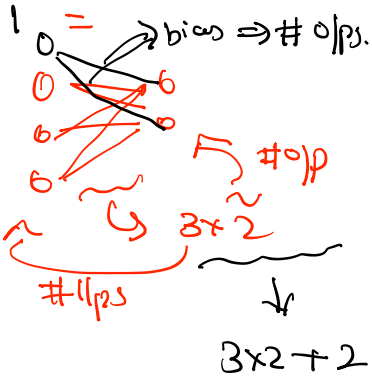
$Size(w_1) = 150.$

$Size(b_1) = \#\text{ o/p channels}$
$= 6$

$Size(w_1, b_1) = 150 + 6$
$= 156$

$Size(w_2, b_2) = 16 \times 6 \times 5 \times 5 + 16$
$= 2416$

$48,120 = Size(w_3, b_3) = 400 \times 120 + 120$
$\Rightarrow (\text{i/p} \times \text{o/p} + \text{o/p})$



$Size(w_4, b_4) = 120 \times 84 + 84$
$= 10,164$

$Size(w_5, b_5) = 84 \times 10 + 10$
$= 850$

$Total\ params = 156 + 2416 + 48,120 + 10,164 + 850$
$= 61,706$

$Size = (\quad) \times datatype$

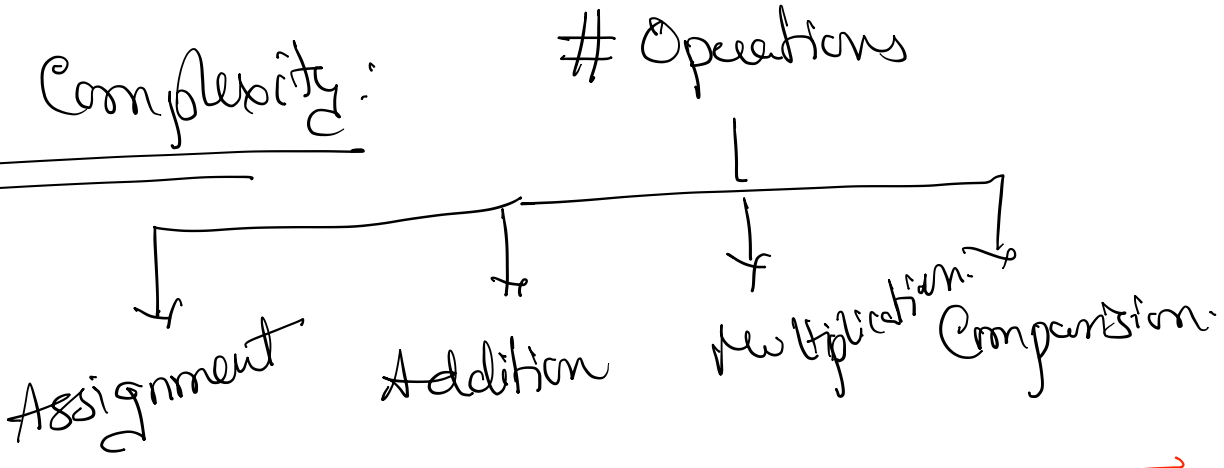
usually FP32.

Total space: $(\text{Total}_{act} + \text{Total}_{params}) \times \text{datatype}$.

$= (8,094 + 61,706) \times (dt)$

$\approx$ 69k $\times (dt)$

Compute Complexity:



---

for a perceptron:

$y = Wx + b$.

- y: o/p activation (1x1)
- Wx: (1x1)
- $+b$: (1x1)
- Wx: 1 multiplication. (M)
- $+b$: 1 addition. (A)
- $y =$: 1 assignment. (implicit)
- x: i/p activation (1x1)

1 o/p activation

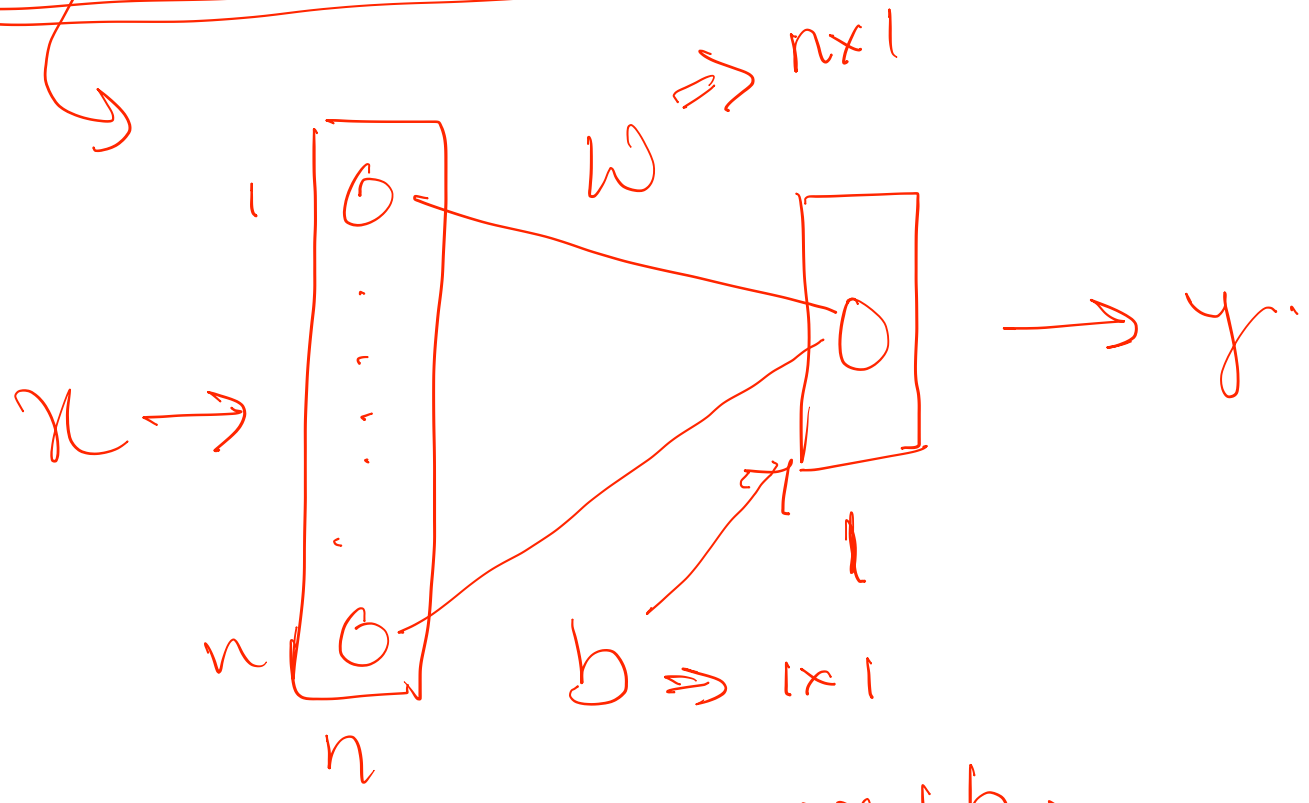
non-pipelined architecture ⟸ (1M + 1A) ⇒ 2 operations

⇓

(no - parallel processing.)

↳ per o/p activation

↳ depends on size of input ⇒ (1)



$y = wx + b$.

↳ 1 x 1

$wx \Rightarrow \sum_{i=1}^{n} w_i x_i$

$\hookrightarrow$ $n \rightarrow$ multiplications

$(n-1) \rightarrow$ additions.

$\pm b \Rightarrow 1 \rightarrow$ addition

total operations $\Rightarrow (nM + nA)$

$= 2n$ operations.

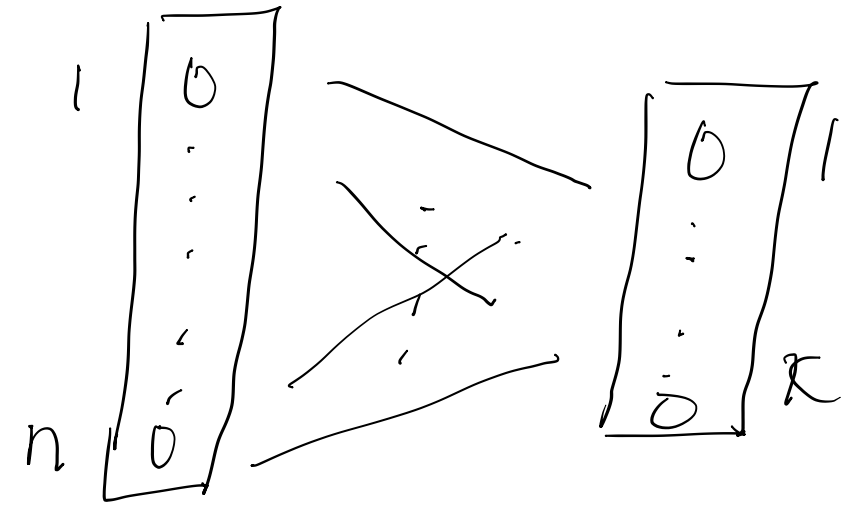
for 1 opnode $\Rightarrow$ $2n$ ops.

$(M + A)$

Size of input.

(non pipelined)

General NN



$$\#\text{ operations} = K(2n)$$

- K → # op nodes
- (M+A) ← 2n
- # ip nodes

non-pipelined ⇒ M & A happen sequentially.

pipelined ⇒ parallel compute.
↳ (M & A happen parallely)

(FMA block is used)

(pipelined)

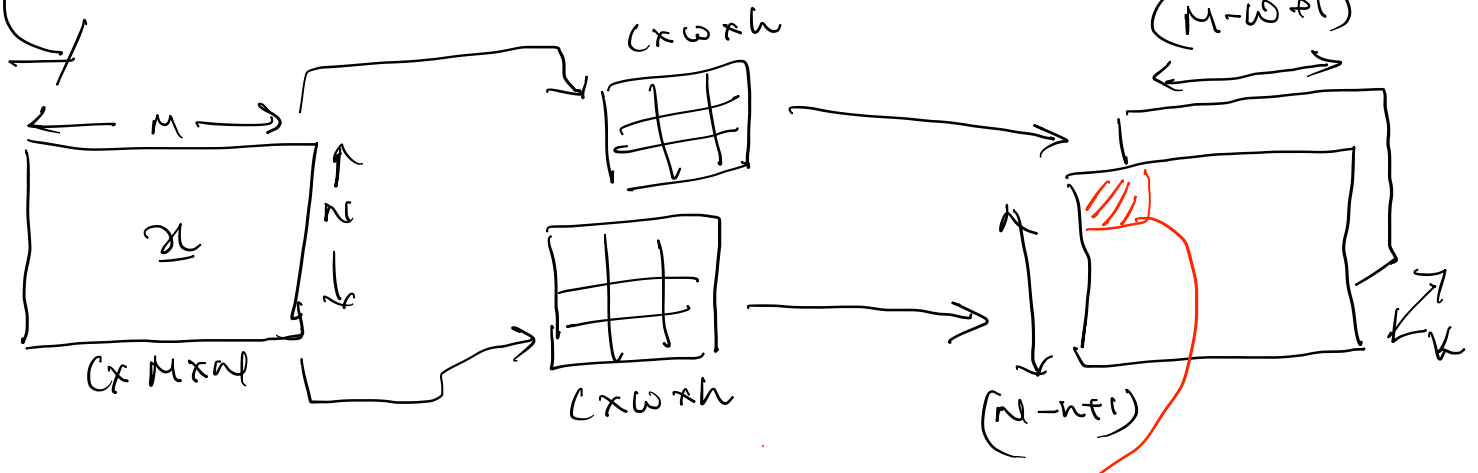
$$\#\text{ ops} = K(n+1)$$

Compute Complexity of CNNs:

Consider:

- An Input, $\underline{x} \in \mathbb{R}^{C \times M \times N}$.
- A conv. kernel, $\underline{w} \in \mathbb{R}^{K \times C \times w \times h}$
- o/p activation, $\underline{z} \in \mathbb{R}^{K \times (M-w+1) \times (N-h+1)}$

$(S=1, P=0)$



for 1 o/p node $\Rightarrow$ $z_{k,x,y} = w_{k,0} + \sum_{i=1}^{w} \sum_{j=1}^{h} w_{k,i,j} \, x_{(x+i),(y+j)}$

($w_{k,0}$: bias)

$\Downarrow$

2chw operations $\Leftarrow$ 1 A + chw Ms + (chw − 1) As

↳ # o/p nodes = $K \times (M-w+1)(N-h+1)$ $(\because S=0, P=1)$

$K \times \left(\frac{M-w+2P_w}{S_w}+1\right)\left(\frac{N-h+2P_h}{S_h}+1\right)$ ← else

Total # operations = $\left[K \times \left(\frac{M-W+2P_W}{S_W}+1\right)\left(\frac{N-h+2P_h}{S_h}+1\right)\right][2chw]$

(Cross-pipelined)

$\left[K \times \left(\frac{M-W+2P_W}{S_W}+1\right)\left(\frac{N-h+2P_h}{S_h}+1\right)\right]$ → # o/p nodes

$[2chw]$ → #ops per o/p node

Total ops (pipelined) = $\left[K \times \left(\frac{M-W+2P_W}{S_W}+1\right)\left(\frac{N-h+2P_h}{S_h}+1\right)\right](chw+1)$

(IM & IA in parallel using block FMA)

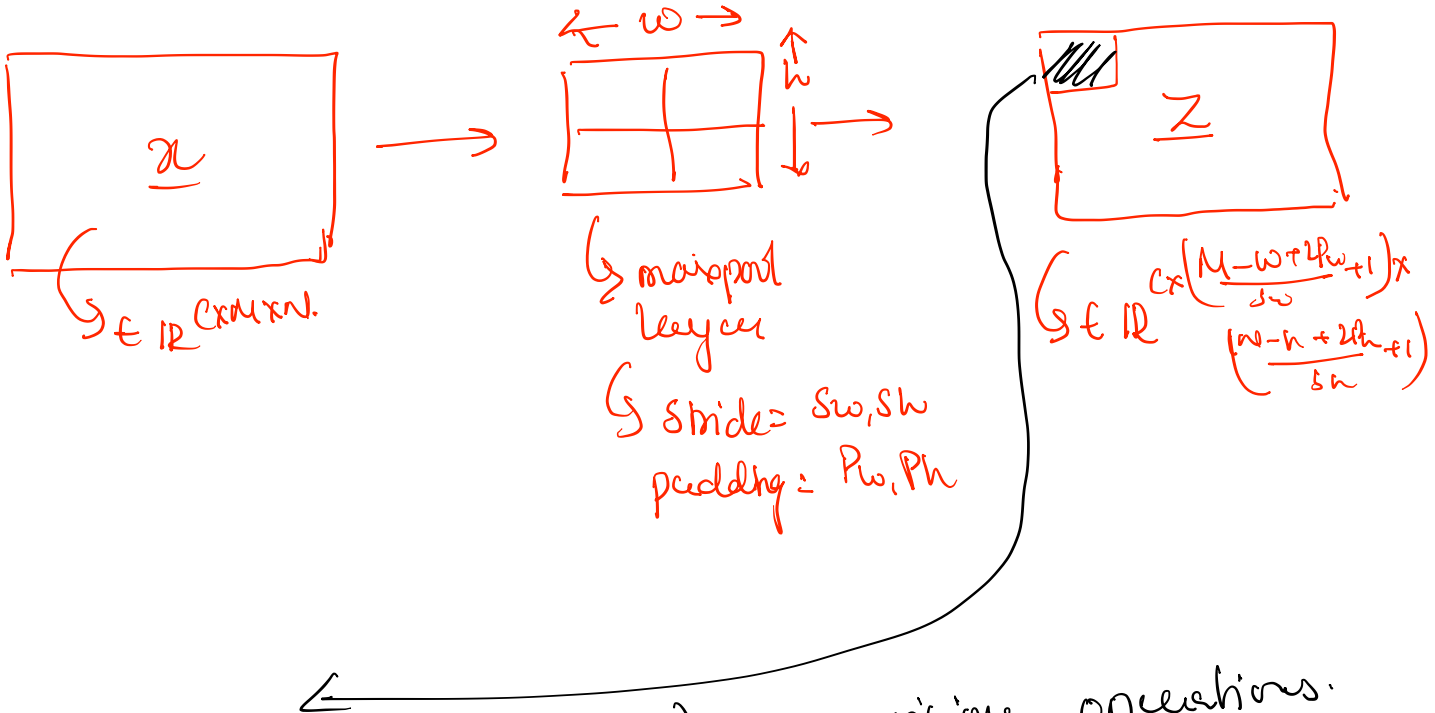
$\left[K \times \left(\frac{M-W+2P_W}{S_W}+1\right)\left(\frac{N-h+2P_h}{S_h}+1\right)\right]$ → # o/p nodes

$(chw+1)$ → #ops per o/p node

Fused Multiply and Add H/w Accelerator.

↳ supports parallel addition & multiplication operations.

Compute Complexity of Maxpool layer:



(hw − 1) comparision operations.

per 1 o/p node:

Total ops ⇒ $\left[C \times \left(\frac{M-w+P_w}{S_w}+1\right)\left(\frac{N-h+P_h}{S_h}+1\right)\right]\left[hw-1\right]$

$\left[C \times \left(\frac{M-w+P_w}{S_w}+1\right)\left(\frac{N-h+P_h}{S_h}+1\right)\right]$ ↓ # o/p nodes

$[hw-1]$ ↓ # ops per node.

· Speed :- # operations (Floating point operations per second)

(in order of the GFLOPs) ← FLOPs (based on processor's clock)

Latency: Time taken to process an input (image).

$$\alpha = \frac{\#\text{ Total operations}}{\text{Speed}} \quad (\text{units} = \text{seconds})$$

Throughput: # Images processed / unit time.

$$T = \frac{\text{Speed}}{\#\text{ Total Operations}} \quad (\text{units} \Rightarrow \underset{\uparrow}{\text{updates}}/\text{sec})$$

images

Assuming all images are processed sequentially.

Next Week => Final Week: (Tuesday – Saturday).

⇓

Advanced Concepts.

Tuesday → Hyperparameters. (> 20-25).

Wednesday → Hyperparameter optimization techniques.

- manual search. (input and working each Hp)
- Grid Search.
- BO
- GA (briefly)

→ Tutorial on Saturday

Thursday → Choosing best models.

Deployment. ∈ { → performance based. → based on compute.

Topic of your choice: FRIDAY.