Number of People Aware of a Secret

ex-> N = 6, Delay = 2, forget = 4

(1) -> A

(2) -> A

(3) -> A B

(4) -> A B C

A -> Bhul gya ab

(5) -> B C D

(6) -> B C D
E F -> B Active

B -> Bhul gya

i -> ith Day

i, cur

forget 0 | ~~1,1~~ | 3,1 | 4,1 | 5,1 | 6,2 | <int, int>

delay 0 | ~~1,1~~ | ~~3,1~~ | 4,1 | 5,1 | 6,2 |

<int, int>

naye din -> Kitne new logo ko just secret pata chla

Day -> 3 4 5 6

cur = 0 1 0 2

Active spreaders

Day -> 3 4 5 6

ans = ~~1~~ 2 ~~3~~ 2 ~~3~~ 5

forgot delay

Soln -> (1) -> Active spreaders kam karna
(know People kam krdo)

② -> Make new active spreaders

if (curr > 0)
ans += curr

③ -> Spread the secret

forgot 0 | ~~1~~ | 3,1 | 4,1 | 5,1 | 6,2

delay 0 | ~~1~~ | ~~3,1~~ | ~~4,1~~ | 5,1 | 6,2

① -> forgot => <①, 1>
1 + 4 ≤ 1 X p.first + forgot ≤ i

delay => <1, 1>
1 + 2 ≤ 1 X

curr = 0
ans = 1

② -> Day = 2

forgot => <①, 1>
1 + 4 ≤ 2 X

delay => <1, 1>
1 + 2 ≤ 2 X

curr = 0
ans = 1

③ -> Day = 3

forgot => <1, 1>
1 + 4 ≤ 3 -> X

Delay => <1, 0> -> also pop it
1 + 2 ≤ 3 -> ✓

curr = curr + p.second

curr = 0 + 1 = 1

ans = curr
ans = 2

-> Push in forgot & delay queue
of i^{th} Day, kitne log add hue

(4) => Day = 4

forgot : 1, 1
1 + 4 ≤ 4 ✗

delay : 3, 1
3 + 2 ≤ 4 ✗

curr = 1
-> So, spread (Also add in both queues)
ans = curr => ans = 3

(5) -> Day = 5

forgot : 1, 1
1 + 4 ≤ 5 ✓ (pop)

ans -= p.second
curr = curr - p.second

ans = 2
curr = 0

forgot : 3, 1
3 + 4 ≤ 5 ✗

delay δ : 3, 1

3 + 2 ≤ 5 ✓

curr = curr + p.second = 6 + 1 = 7

ans += curr = 2 + 1 = 3

(6) => Day = 6

forgot => 3, 1

3 + 4 ≤ 6 ✗

delay => 4, 1 -> pop it

4 + 2 ≤ 6 ✓

curr = curr + p.second = 1 + 1 = 2

ans += curr = 3 + 2 = 5

Add in both queue

M-2 -> DP

(a) -> With Two vector

forget ->

0	0	0	0	1	0	0
0	1	2	3	4	5	6

delay ->

0	0	1	0	0	0	0

curr = 0
ans = 1

n = 6, delay = 2, forget = 4

i = 0

curr = curr + delay[0] = 0
= 0 + 0

curr = curr - forget[0]
= 0 - 0 = 0

ans = ans + curr = 1
ans = ans - forget[0] (Kitna log jo khud gaye)
= 1 - 0 = 1

update delay

0 + 2 < 6 -> ✓
delay[2] += curr => 0 + 0 = 0

forget 0 + 4 < 6 -> ✓
forget[2] += curr = 0

What happen on one layer

25/04/24

I/p -> $n_H^{[l-1]} \times n_w^{[l-1]} \times n_c^{[l-1]}$

o/p -> $n_H^{[l]} \times n_w^{[l]} \times n_c^{[l]}$ at l^{th} layer

filter size -> $f^{[l]} \times f^{[l]} \times n_c^{[l-1]}$

Activations -> $a^{[l]} : n_H^{[l]} \times n_w^{[l]} \times n_c^{[l]}$

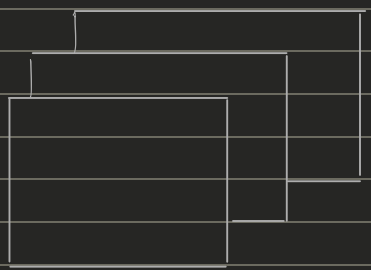
Weights -> $f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]}$

$n_c^{[l]}$ -> No of filters

we want to detect variety of edges. That's why we more no of edges.

Bias -> $n_c^{[l]}$ -> when we Broadcast it $(1, 1, 1, n_c^{[l]})$

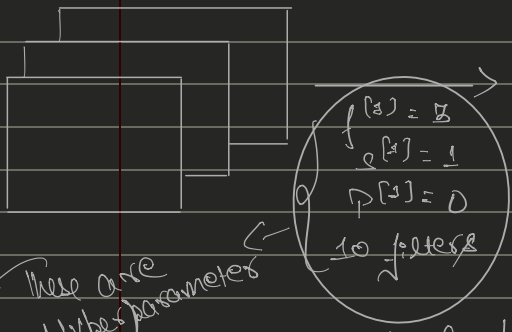
CNN ->

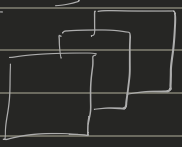


$39 \times 39 \times 3$

$n_H^{[0]} = 3$, $n_w^{[0]} = 39$, $n_c^{[0]} = 3$

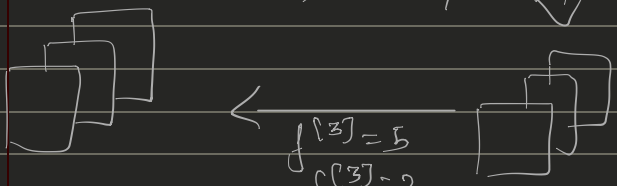
$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \Rightarrow \left\lfloor \frac{39+0-3}{1} + 1 \right\rfloor$$





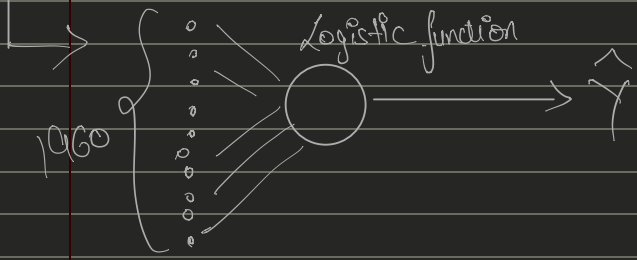
$37 \times 37 \times 10$

$f^{[2]} = 5$
$s^{[2]} = 2$
$p^{[2]} = 0$
20 filters



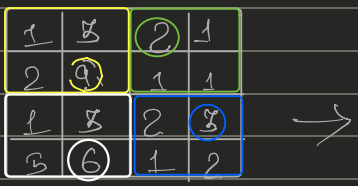
$17 \times 17 \times 20$

$7 \times 7 \times 40 = 1960$



Pooling Layer -> (POOL)

Max Pooling



$s = 2$
$f = 2$

$$\left\lfloor \frac{n-f}{s} + 1 \right\rfloor \Rightarrow \left\lfloor \frac{4-2}{2} + 1 \right\rfloor = \lfloor 1+1 \rfloor = \lfloor 2 \rfloor = 2$$

Avg Pooling ->

$S = 2$

$f = 2$

1	3	2	1
2	9	1	1
1	4	2	5
3	6	1	2

→

3.75	1.25
4	2

1	3	2	1	3
2	9			5
1				2
8				0
5	6		2	9

→

9	9	5
9	9	5
8	6	9

Note -> In pooling layer, there is no parameter to learn.