Task 1: Formula : $y = \text{ReLu}\left(\sum_{c=1}^{C} x^c \cdot w^c + b\right)$

1) output h/w = 4x5 matrix ⇒ 4x5 = 20 positions per filter

output for y_1 :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

output for y_2 :

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Tensor = (3, 4, 5)

output for y_3 :

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2) for a patch $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, a kernel $\begin{bmatrix} w & x \\ y & z \end{bmatrix}$ and a bias β plus a ReLu (everything) we can see here :

if $\begin{bmatrix} \boxed{w} & x \\ y & \boxed{z} \end{bmatrix}$ $\boxed{}$ are 1's and the rest 0 and $\begin{bmatrix} \boxed{a} & b \\ c & \boxed{d} \end{bmatrix}$ are 1's then we can apply output = $\max(0, ((1\cdot1)+(1\cdot1)+\beta))$

3) output_shape = 2x3 | since no padding is allowed we skip last...

1)
$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

2)
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

3)
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

4) stride of 2 in a convolution: — Reduces $(H \times W)$ of feature map

roughly $\frac{H}{2} \times \frac{W}{2}$

- Helps to compress the representation
- Helps the network to focus on important features hence reduce computation.

- Max Pooling can be used for the same purpose (only compression)

5) Let L_{in} : length of the input

K : Kernel size

S : Stride

P : Padding

$$L_{out} = \left\lfloor \frac{L_{in} + 2P - K}{S} \right\rfloor + 1$$

Task 2:

1) Receptive field with $s = 1$ $= 1 + (k-1) \cdot L$

with $K = 3$ (kernel), $L = 3$ (3 Layers)

$RF = 1 + (3-1) \cdot 3 = 7$ which give $\boxed{7 \times 7}$

2) Using recursive formula :

$$j_i = J_{i-1} \cdot s_i$$

$$r_i = r_{i-1} + (k_i + 1) \cdot J_{i-1}$$

Let $r_0 = 1$, $J_0 = 1$:

$j_1 = 1 \cdot 2 = 2$; $r_1 = 1 + (4-1) \cdot 1 = 4$

$j_2 = 2 \cdot 2 = 4$; $r_2 = 4\ (4-1) \cdot 2 = 10$

$RF = \boxed{10 \times 10}$

3) • The deeper the CNN, the larger the receptive fields each layer builds on top of the previous one.

• If we use larger kernel size and bigger strides > 1 and Pooling can increase RF without increasing depth