We have $\underline{\underline{X}} = (x_1, x_2, x_3, x_4, x_5, x_6, \ldots, x_n)$ as input data of size $1 \times n$

$\underline{\underline{K}} = (\underbrace{K_1, K_2,}\, K_3)$ as kernel of size 1×3

$s = 1$
$n = n$
$f = 3$

We will convolve K on X to get

(x_2, x_3, x_4)

$$y_1 = K_1 x_1 + K_2 x_2 + K_3 x_3$$
$$y_2 = K_1 x_2 + K_2 x_3 + K_3 x_4$$
$$y_3 = K_1 x_3 + K_2 x_4 + K_3 x_5$$
$$y_4 = K_1 x_4 + K_2 x_5 + K_3 x_6$$
$$\vdots$$
$$y_{n-2} = K_1 x_{n-2} + K_2 x_{n-1} + K_3 x_n.$$

$$y^T = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-2} \end{pmatrix}$$

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

$\frac{n-3}{1} + 1$

$\underline{\underline{n-2}}$

We can apply the same convolution again on y as well.

$\underline{\underline{X}} \xrightarrow{K} Y \xrightarrow{K} \underline{\underline{Z}}$

$$z_1 = K_1 y_1 + K_2 y_2 + K_3 y_3$$
$$= K_1 (K_1 x_1 + K_2 x_2 + K_3 x_3) + K_2 (K_1 x_2 + K_2 x_3 + K_3 x_4) + K_3 (K_1 x_3 + K_2 x_4 + K_3 x_5)$$

$$= K_1^2 x_1 + (K_1 K_2 + K_2 K_1) x_2 + (K_1 K_3 + K_2^2 + K_3 K_1) x_3 + (K_2 K_3 + K_2 K_3) x_4 + K_3^2 x_5$$

$$\boxed{z_1 = K_1^2 x_1 + 2K_2 K_1 x_2 + (K_2^2 + 2K_1 K_3) x_3 + 2K_2 K_3 x_4 + K_3^2 x_5}$$

Similarly, we can calculate $z_2, z_3, z_4, \ldots z_{n-2}$

with
$$\boxed{z_m = K_1^2 x_m + 2K_1 K_2 x_{m+1} + (K_2^2 + 2K_1 K_3) x_{m+2} + 2K_2 K_3 x_{m+3} + K_3^2 x_{m+4}}$$

$X \rightarrow Y \rightarrow Z$

$(K_1^2 \quad 2K_1 K_2 \quad K_2^2 + 2K_1 K_3 \quad 2K_2 K_3 \quad K_3^2)$

$X \rightarrow Y \rightarrow Z$

$X \rightarrow Z$

$$\underline{K} = \underbrace{\left(k_1^2,\ 2k_1k_2,\ k_2^2 + 2k_1k_3,\ 2k_2k_3,\ k_3^2\right)}_{1\times 5}$$

Question

(1)

Consider a 1x3 convolution (as per Deep Learning terminology) kernel $\overbrace{o = [o, o, o]}^{}$ = [-1 2 -1]. Output is generated by stacking 4 such operators on the 1-D input data (i.e., the operator is applied four times on the input). What is the width of the equivalent operator O that will give the same output, when applied only once on the input data?

1X3

$X \rightarrow Y \rightarrow\ \ \rightarrow\ \ \rightarrow F$

We have an input data X as below

$$X = \underbrace{(x_1, x_2, \ldots\ldots x_n)}$$

We have kernel $\underline{K = (-1, 2, -1)}$. We have to apply it 4 times.

Let's try to visually understand the operations.

The 4 convolutions will look like below -

$$X \xrightarrow[\text{I}]{} Y \xrightarrow[\text{II}]{} Z \xrightarrow[\text{III}]{} U \xrightarrow[\text{IV}]{} V$$

$\left[X \xrightarrow{k} V\right.$

$V = (v_1, v_2, \ldots\ \ )$

$U = (u_1, u_2, u_3 \ldots)$

Let us pick v_1 from V.

$$\underline{v_1} = \underbrace{k_1u_1 + k_2u_2 + k_3u_3}$$

$\left[\text{For calculating } v_1,\ \text{we needed } z_1, z_2, \cdots z_5\right]$

Here $u_1 = \underline{k_1z_1} + k_2z_2 + k_3z_3$

$u_2 = k_1z_2 + k_2z_3 + k_3z_4$

$Z = (z_1, z_2, z_3, z_4, z_5, \ldots\ldots)$

$$u_2 = k_1 z_2 + k_2 z_3 + k_3 z_4$$
$$u_3 = k_1 z_3 + k_2 z_4 + k_3 z_5$$

$$Z = (z_1, z_2, z_3, z_4, z_5, \ldots)$$

Also
$$z_1 = k_1 y_1 + k_2 y_2 + k_3 y_3$$
$$z_2 = k_1 y_2 + k_2 y_3 + k_3 y_4$$
$$z_3 = k_1 y_3 + k_2 y_4 + k_3 y_5$$
$$z_4 = k_1 y_4 + k_2 y_5 + k_3 y_6$$
$$z_5 = k_1 y_5 + k_2 y_6 + k_3 y_7$$

[For calculating v_1, we needed values $y_1, y_2, \ldots y_7$]

$$Y = (y_1, y_2, y_3, y_4, \ldots y_7, y_8, \ldots)$$

Here,
$$y_1 = k_1 x_1 + k_2 x_2 + k_3 x_3$$
$$y_2 = k_1 x_2 + k_2 x_3 + k_4 x_4$$
$$y_3 = k_1 x_3 + k_2 x_4 + k_4 x_5$$

$$X = (x_1, x_2, x_3, \ldots)$$

$x_1 \cdots \quad x_9$.

$$y_4 = k_1 x_4 + k_2 x_5 + k_3 x_6$$

$$y_5 = k_1 x_5 + k_2 x_6 + k_3 x_7$$

$$y_6 = k_1 x_6 + k_2 x_7 + k_3 x_8$$

$$y_7 = k_1 x_7 + k_2 x_8 + k_3 x_9$$

For calculating V_1, we will need $(x_1, x_2, \cdots\ x_9)$

Hence we will need an operator of width 9 that will be applied on X to give V. i.e. $X \xrightarrow{K'} V$ (where K' is 1×9)

A. Consider a 1x3 convolution (as per Deep Learning terminology) operator o =[o_1, o_2, o_3]= [-1 2 -1]. Output is generated by stacking two such operators on the 1-D input data (i.e., the operator is applied twice on the input).

a) What is the size of the equivalent operator O that will give the same output, when applied only once on the input data?

Height of O = 1 Width of O = 2*3-1 = 5

b) Show the elements O(i,j) of equivalent operator O.

O can be obtained by cross-correlation of o with o.

O = [1 -4 6 -4 1]

$$O(1,j)=\sum_{k=1}^{j} o_k o_{4-k}$$

$X \xrightarrow{k} Y \xrightarrow{k} Z$

$X \xrightarrow{k'} Z$

We have input data as $\underline{X} = (x_1, x_2, \cdots x_m)$

$K = (-1, 2, -1)$

$Y = (\underbrace{y_1, y_2, y_3,} \cdots)$

we apply the kernel K on X twice to get

$$X \xrightarrow{K} Y \xrightarrow{K} Z$$

for $z_1 \in \underline{Z}$ we have $z_1 = \underbrace{k_1 y_1 + k_2 y_2 + k_3 y_3}$ [We need y_1, y_2, y_3 for z_1]

Now
$$\left.\begin{aligned} y_1 &= k_1 x_1 + k_2 x_2 + k_3 x_3 \\ y_2 &= k_1 x_2 + k_2 x_3 + k_3 x_4 \\ y_3 &= k_1 x_3 + k_2 x_4 + k_3 x_5 \end{aligned}\right\}$$ [We need $x_1, x_2, \cdots x_5$ for z_1]

Hence we must have 1 vector K' of order 1×5 as a kernel to reach Z from X.

$z_1 = k_1 y_1 + k_2 y_2 + k_3 y_3$

Here $k_1 y_1 + k_2 y_2 + k_3 y_3$

$$= k_1(\underbrace{k_1 x_1 + k_2 x_2 + k_3 x_3}) + k_2[k_1 x_2 + k_2 x_3 + k_3 x_4] + k_3[\underbrace{k_1 x_3 + k_2 x_4 + k_3 x_5}]$$

$$= k_1^2 x_1 + 2k_1 k_2 x_2 + (2k_1 k_3 + k_2^2) x_3 + 2k_2 k_3 x_4 + k_3^2 x_5$$

$$= \underbrace{(x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5)} \begin{pmatrix} k_1^2 \\ 2k_1 k_2 \\ k_2^2 + 2k_1 k_3 \\ 2k_2 k_3 \\ k_3^2 \end{pmatrix}$$

$K = [-1, 2, -1]$ $\underline{1 \times 3}$

$X \longrightarrow Z$

$X \longrightarrow Z$

We get $K' = \begin{pmatrix} (-1)^2 \\ 2(-1)(2) \\ 2^2 + 2(-1)(-1) \\ 2(2)(-1) \\ (-1)^2 \end{pmatrix}^T = \begin{pmatrix} 1 \\ -4 \\ 6 \\ -4 \\ 1 \end{pmatrix}^T$ as required kernel of order 1×5

Question

$\underbrace{A}_{n \times n} \xrightarrow{K} B_{(n-1)\times(n-1)} \xrightarrow{K} \underbrace{C}_{(n-2)\times(n-2)}$

Consider a 2 X 2 Convolution (as per Deep Learning Terminology) operator O

-1	-1
1	1

$= K$

$A_{n \times n}$ $\left\lfloor \frac{n-1-2}{1} + 1 \right\rfloor = n-1$

$n - 1 - 2 + 1 = n - 2$

$A \xrightarrow{V} C$

Output is generated by stacking 2 such operators on the 2-D input data (i.e. the operator is applied twice on the input)

What is the size of the equivalent Operator “V” when applied only once on input data?

$n = 3$

$A_{3\times3} \rightarrow B_{2\times2} \rightarrow C_{1\times1}$

Show the elements of equivalent operator V

We have $K = \begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} k_1 & k_2 \\ k_3 & k_4 \end{pmatrix}$ as a kernel of 2×2 size.

Let us consider a 3×3 input matrix A as below.

$$\underline{A} = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{pmatrix}$$

$a_1 k_1 + a_2 k_2 + a_4 k_3 + a_5 k_4$

We apply the kernel K on A twice to get

$$A \xrightarrow{K} B \xrightarrow{K} C$$

We have B of size $\left\lfloor \frac{n-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n-f}{s} + 1 \right\rfloor = \left\lfloor \frac{3-2}{1} + 1 \right\rfloor \times \left\lfloor \frac{3-2}{1} + 1 \right\rfloor$

$= 2 \times 2$

We get $\underline{B} = \begin{pmatrix} p & q \\ r & s \end{pmatrix}$ on applying K on B we get

$C = (c)$

$C = (c)$ where $c = k_1 p + k_2 q + k_3 r + k_4 s$

[i.e, C is a matrix of size 1×1]

$K = \begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}$

$A = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{pmatrix}$

$p = \underline{-a_1 - a_2 + a_5} + a_4$

$q = \underline{-a_2 - a_3 + a_5 + a_6} = a_2 k_1 + a_3 k_2 + a_5 k_3 + a_6 k_4$

$B = \begin{pmatrix} p & q \\ r & s \end{pmatrix}$

$r = \underline{-a_4 - a_5 + a_7 + a_8}$

$s = -a_5 - a_6 + a_8 + a_9$

$\underline{\underline{\begin{pmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{pmatrix}}}$

Now $c = -p - q + r + s$

$= -(-a_1 - a_2 + a_5 + a_4) - (-a_2 - a_3 + a_5 + a_6) +$

$(-a_5 - a_6 + a_8 + a_9)$

$= -(-a_1 - a_2 + a_5 + a_4) - (\ldots$

$(-a_4 - a_5 + a_7 + a_8) + (-a_5 - a_6 + a_8 + a_9)$

$$\boxed{c = \underline{\underline{a_1}} + 2\underline{\underline{a_2}} \cancel{-2a_5} - 2\underline{\underline{a_4}} + \underline{\underline{a_3}} - 4\underline{\underline{a_5}} - 2\underline{\underline{a_6}} + \underline{\underline{a_7}} + 2\underline{\underline{a_8}} + \underline{\underline{a_9}}}$$

We can use a kernel

$$\underline{\underline{K'}} = \begin{pmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{pmatrix}$$ directly on A to get c attained above.

This means we apply a 3X3 kernel on 3X3 matrix to get result c.

The required 3X3 kernel is K'.