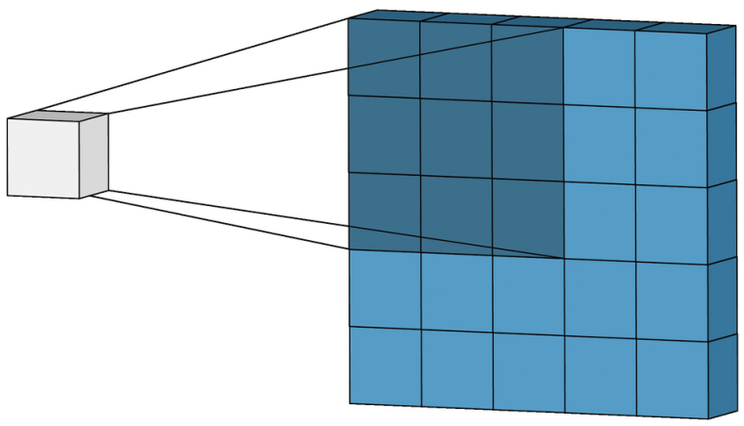
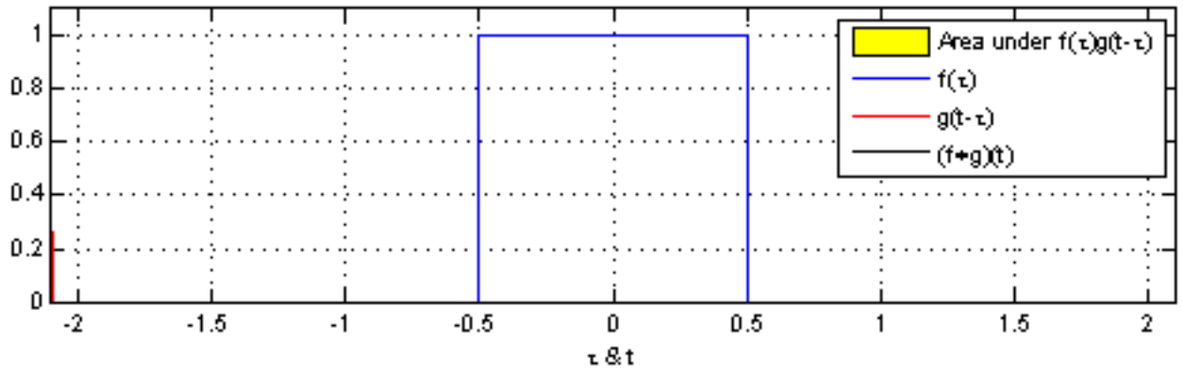
CONVOLUTION: 1D AND 2D

Math Tools Lab #5
October 11th, 2019



CONVOLUTION RECAP

A mathematical operation on two functions, f and g, that produces a third function expressing how the shape of one is modified by the other. Functionally, this means sliding one function on top of the other, multiplying and then adding.

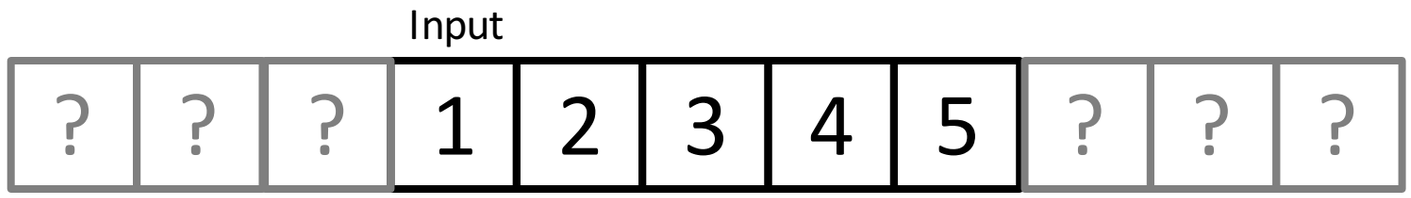


More formally, it is the integral of the product of two functions after one has been reversed and shifted:

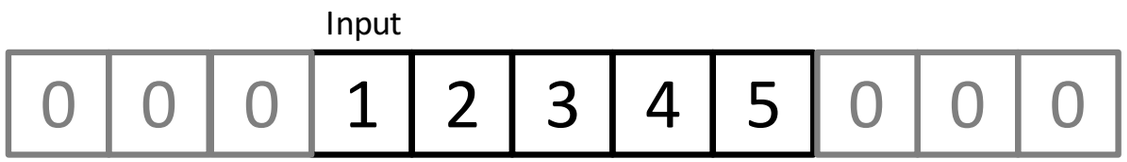
$$(f * g)(t) = \int_{-\infty}^{\infty} f(x)g(t-x)dx$$

PADDING

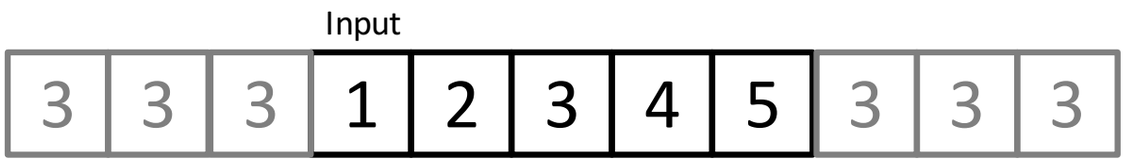
Let's take an arbitrary 1D signal. We don't know the values before we started or after we stopped recording...how can we pad our signal?



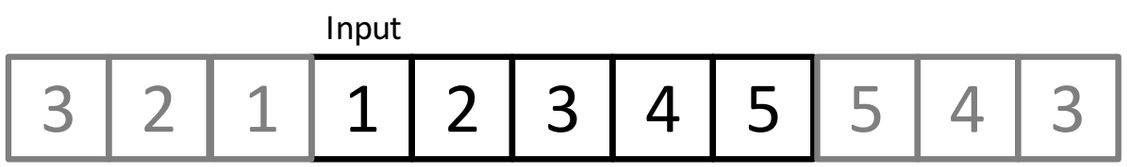
Zeros



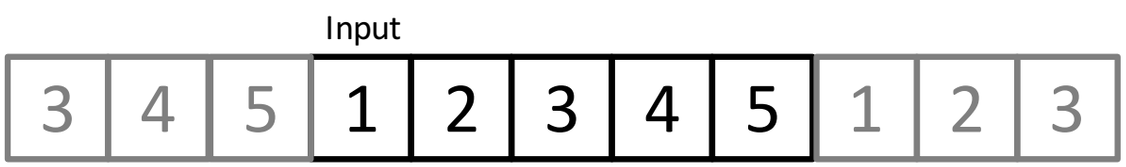
Constant (i.e. mean)



Reflectance

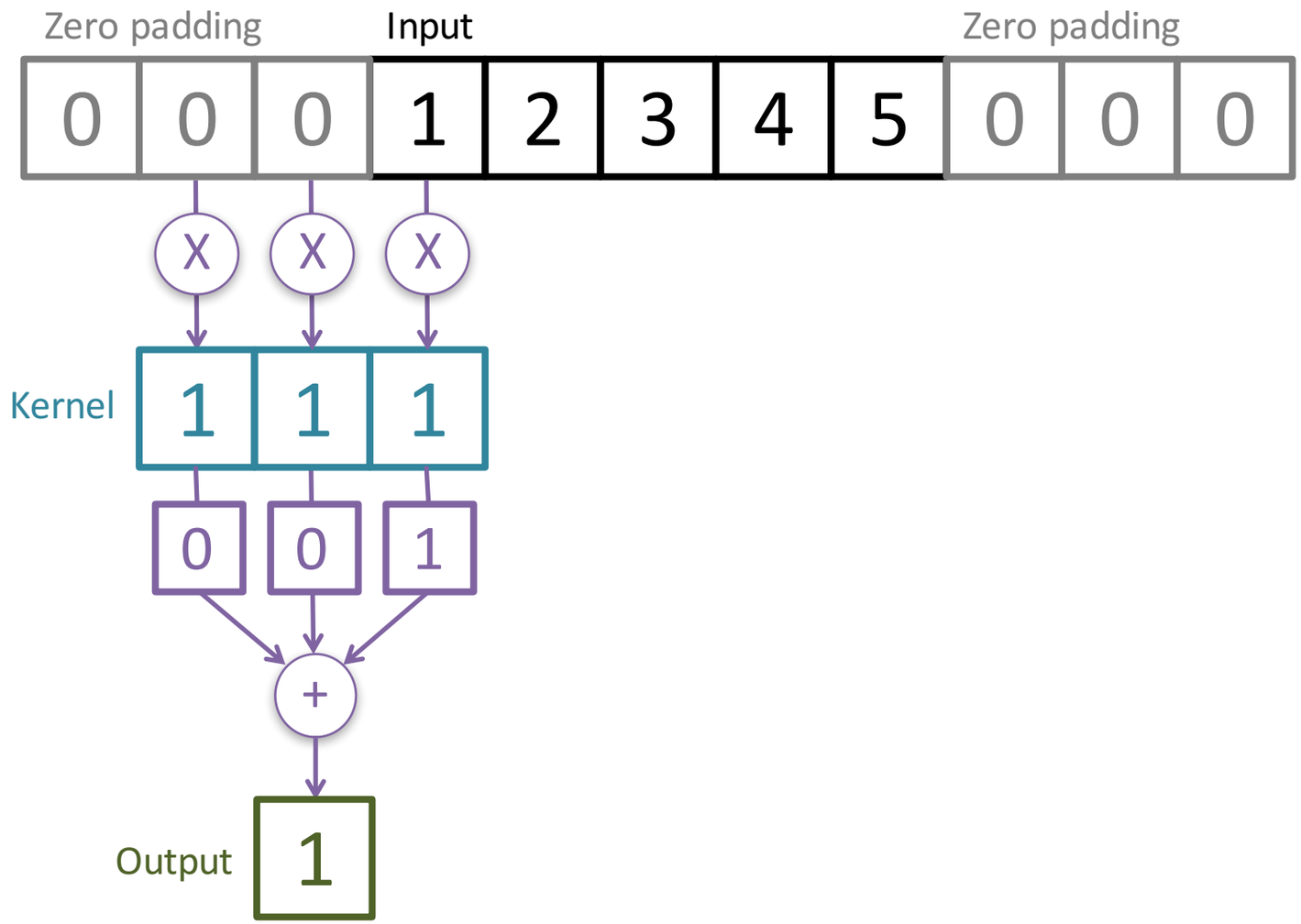


Circular (MATLAB default)

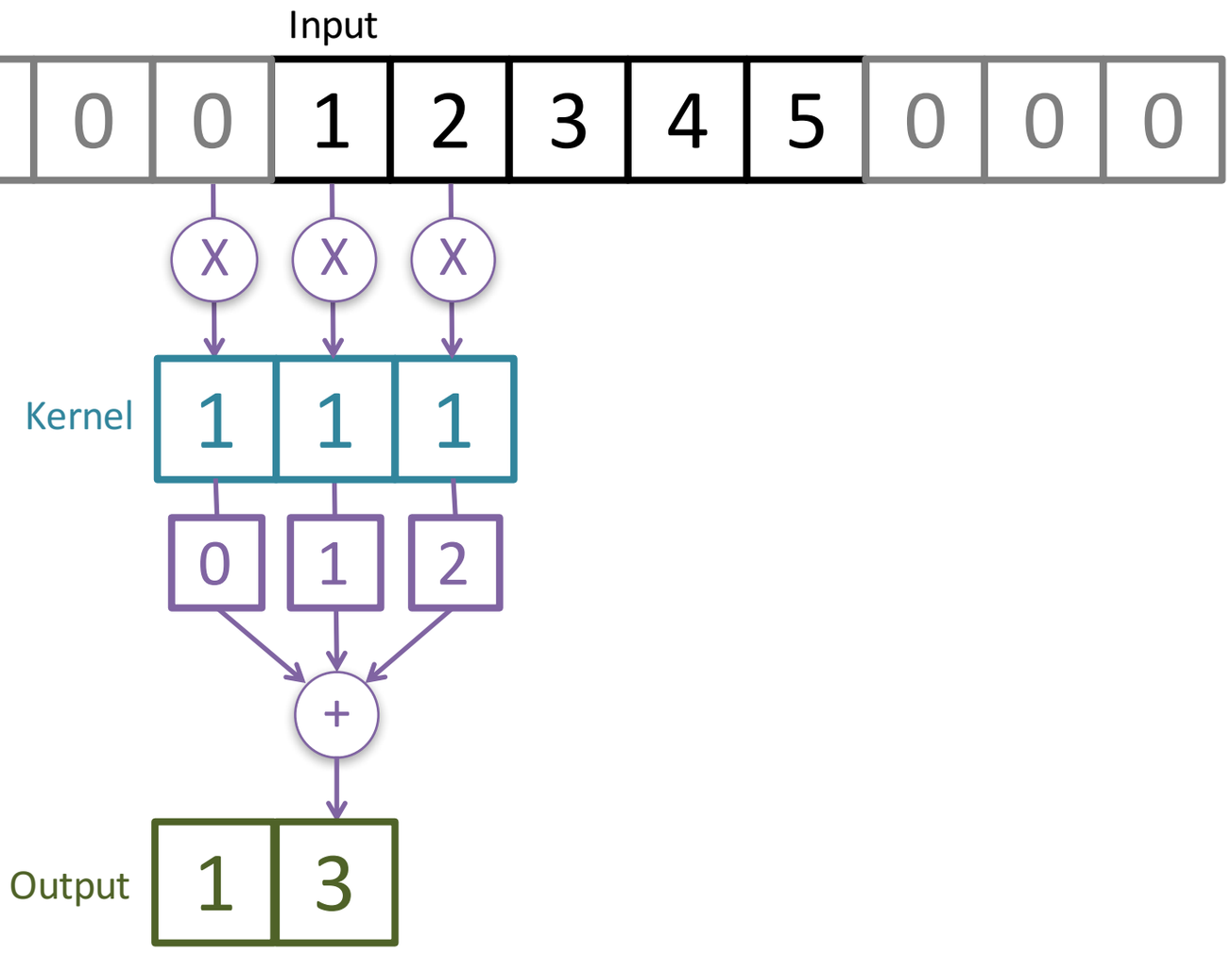


BOUNDARY HANDLING: "FULL"

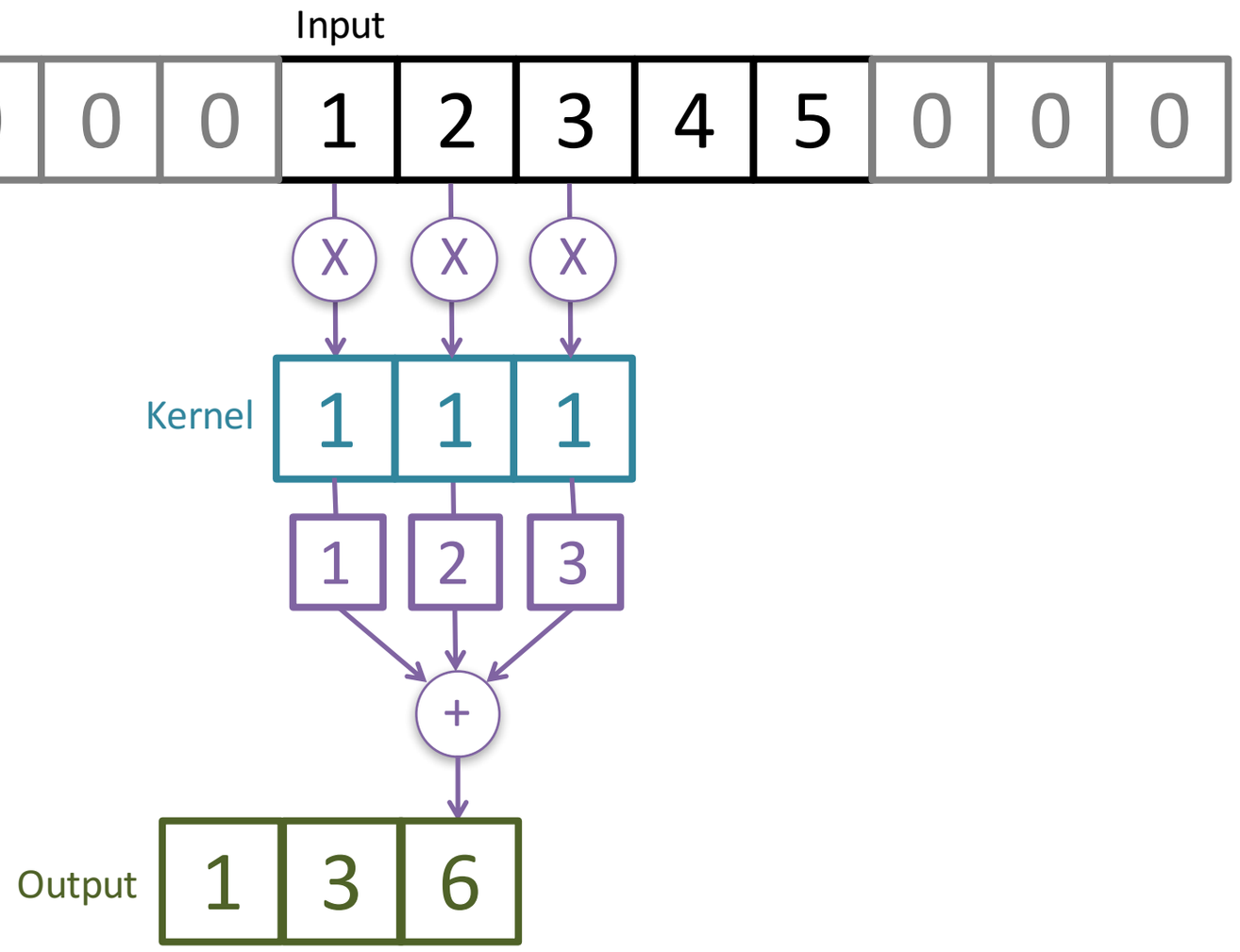
Makes use padding, rightmost part of kernel starts at input



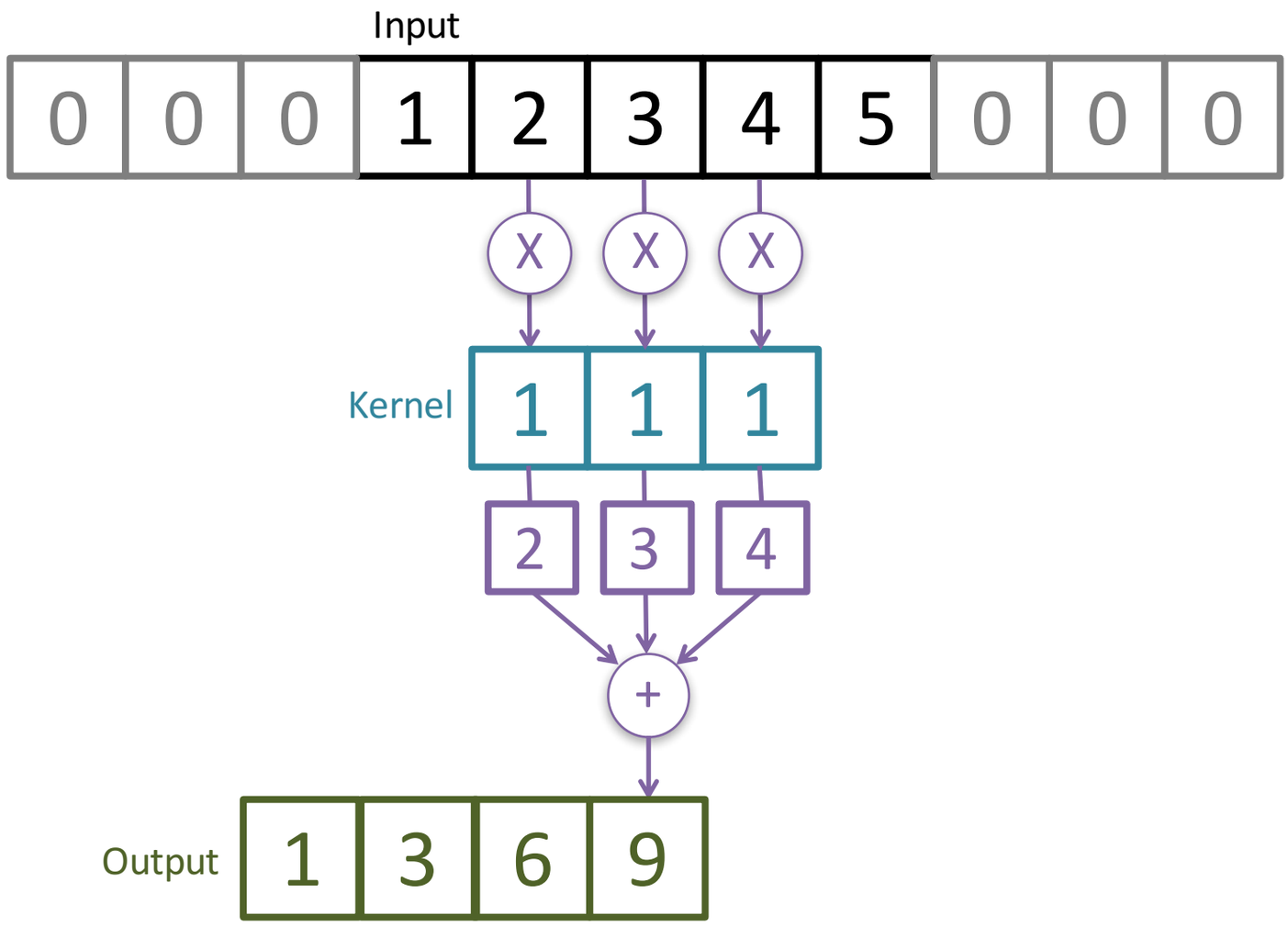
BOUNDARY HANDLING: "FULL"



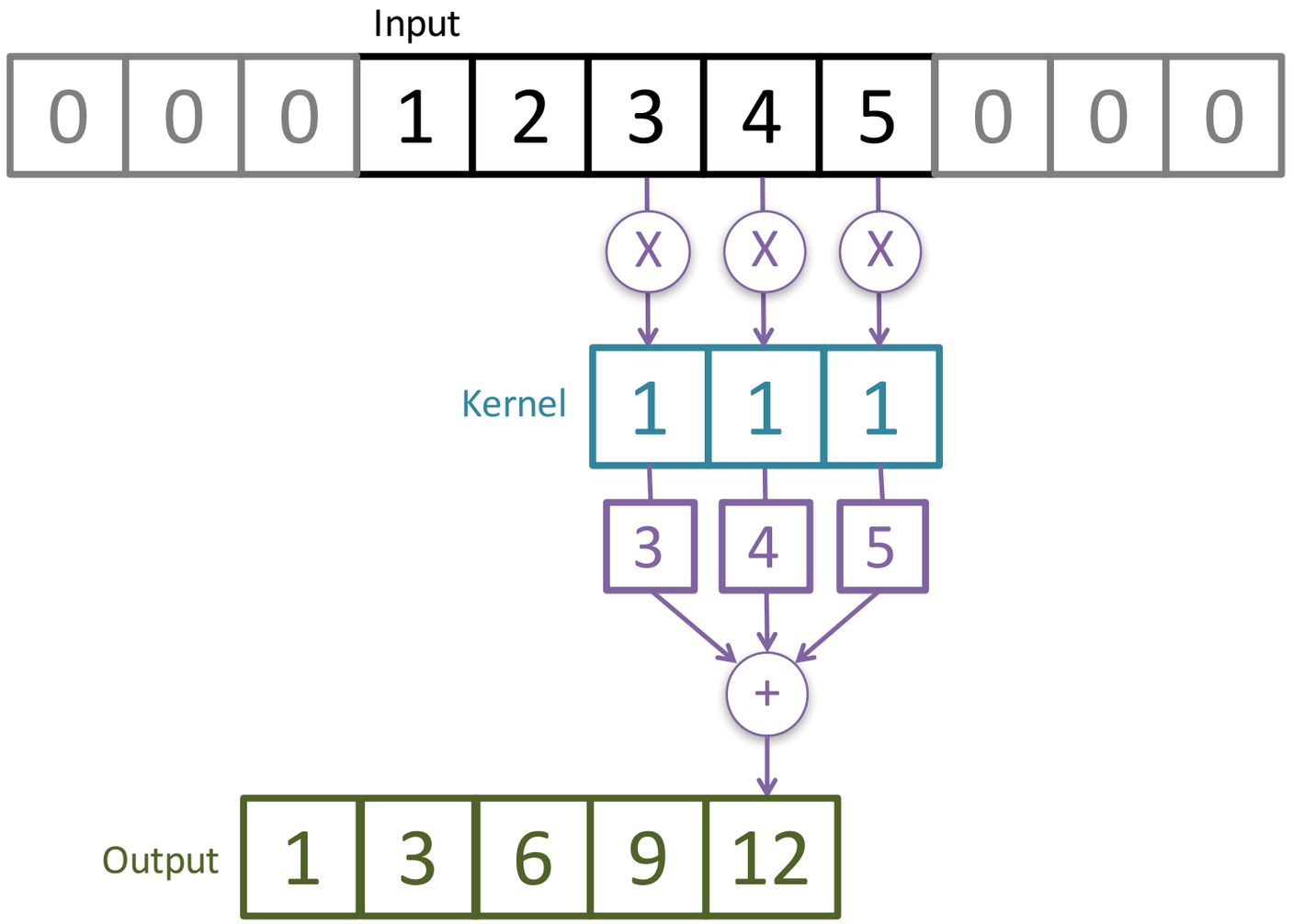
BOUNDARY HANDLING: "FULL"



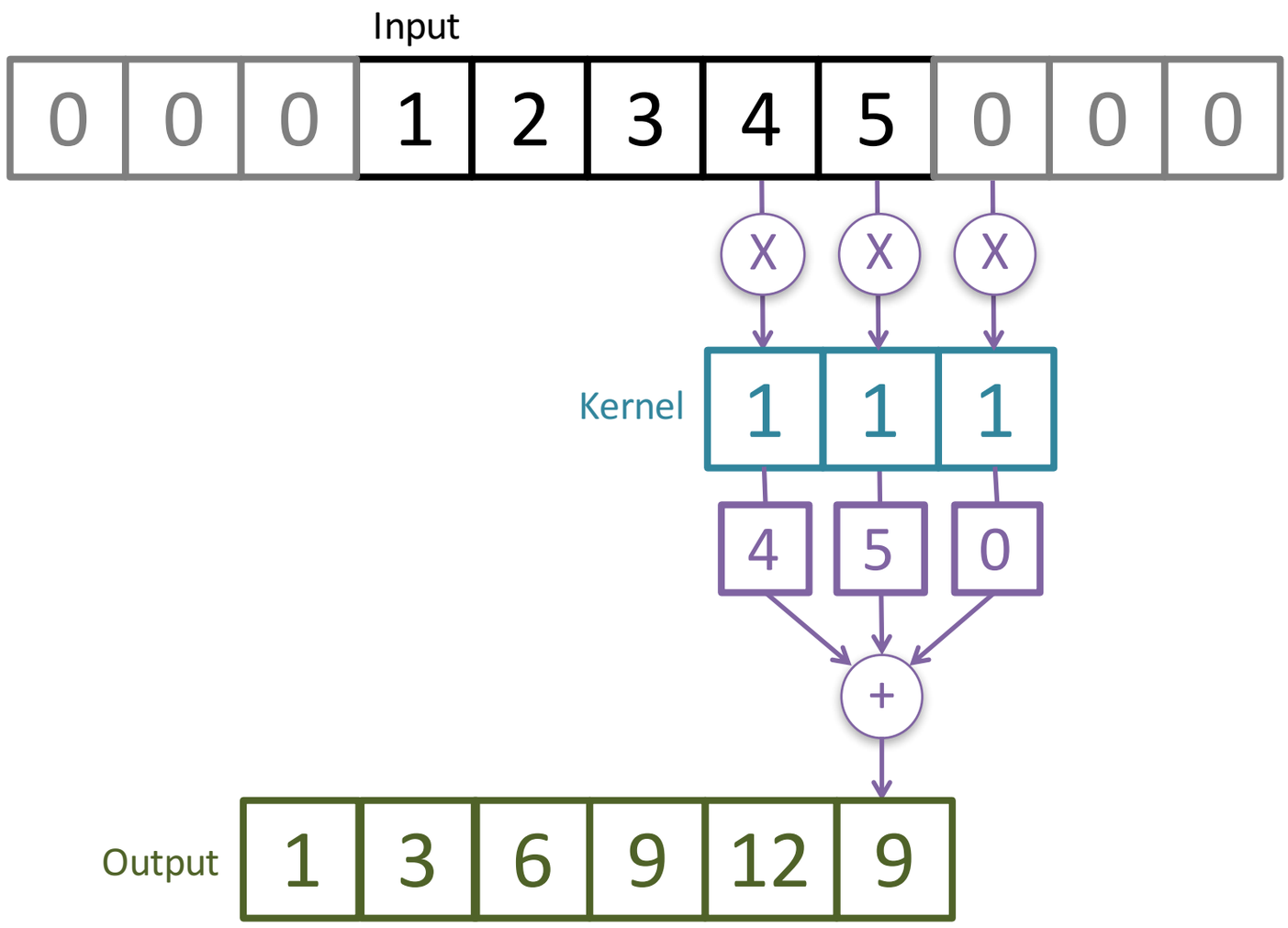
BOUNDARY HANDLING: "FULL"



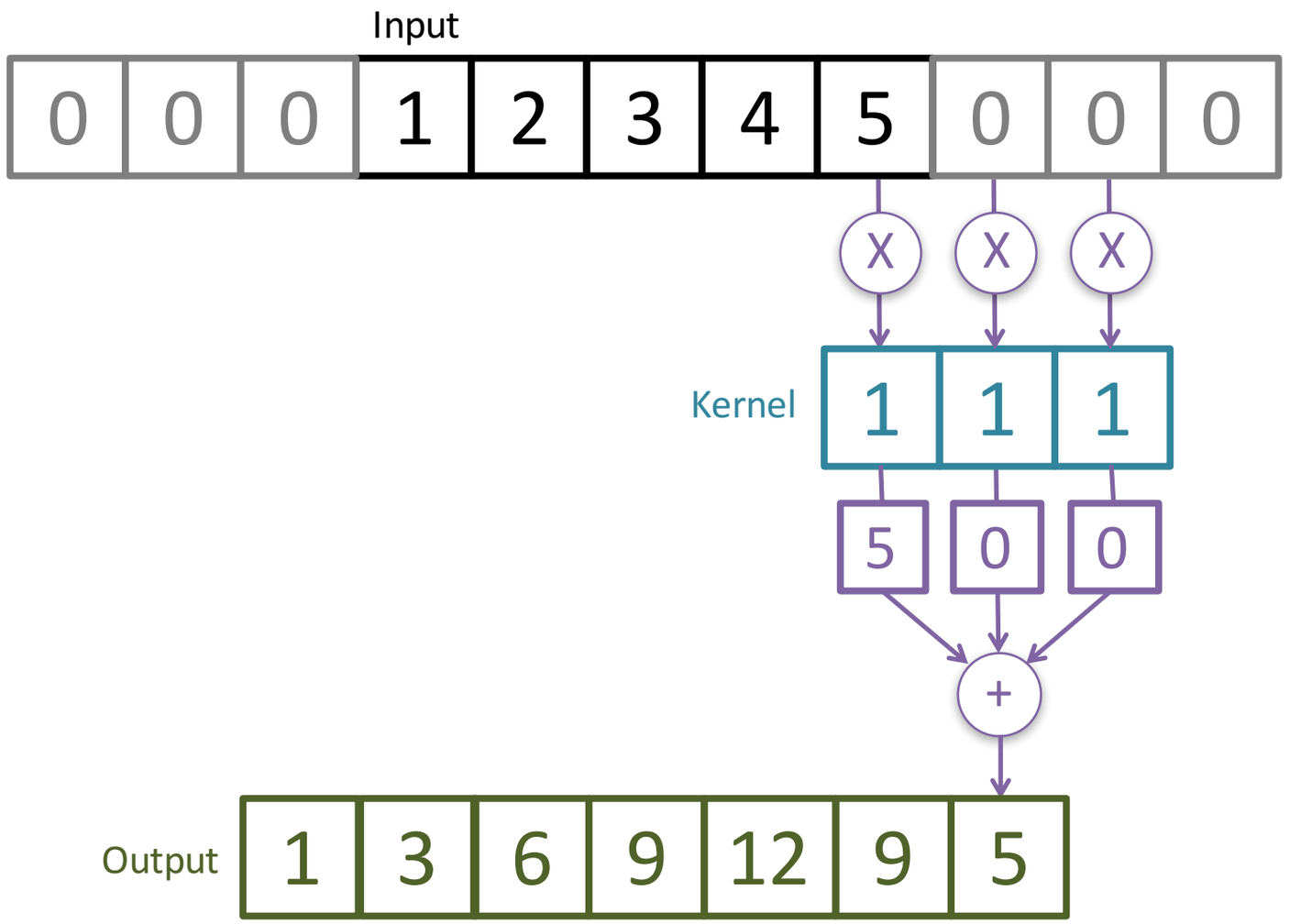
BOUNDARY HANDLING: "FULL"



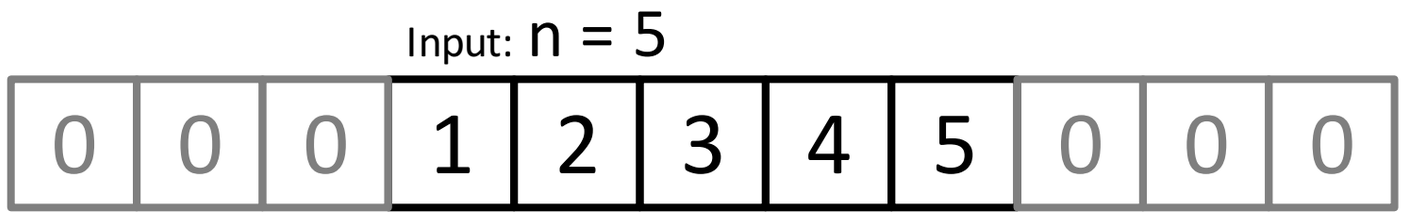
BOUNDARY HANDLING: "FULL"

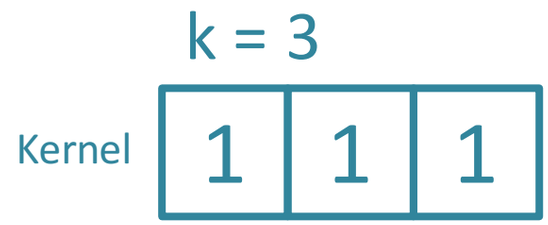


BOUNDARY HANDLING: "FULL"



BOUNDARY HANDLING: "FULL"



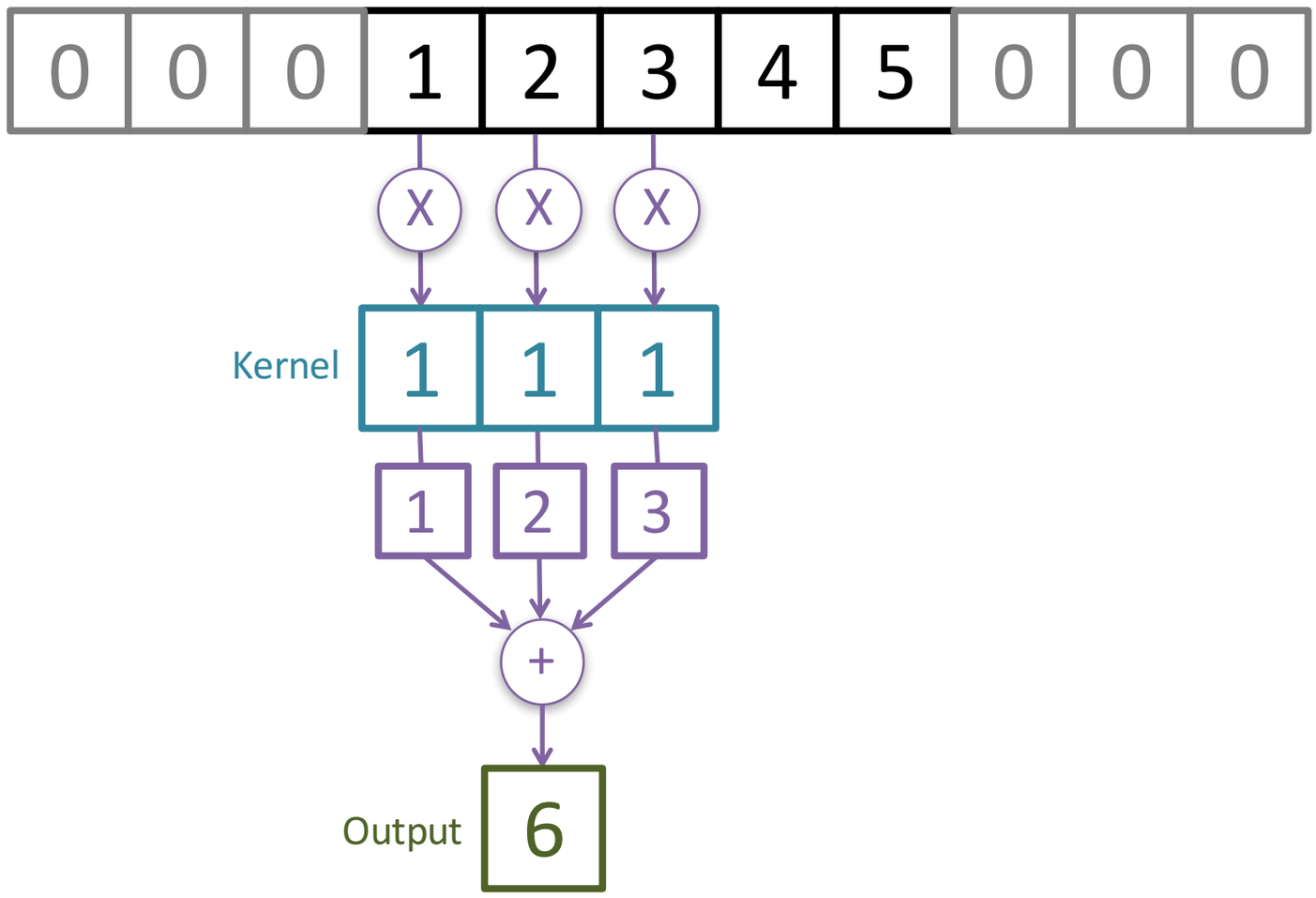


Output is longer than input

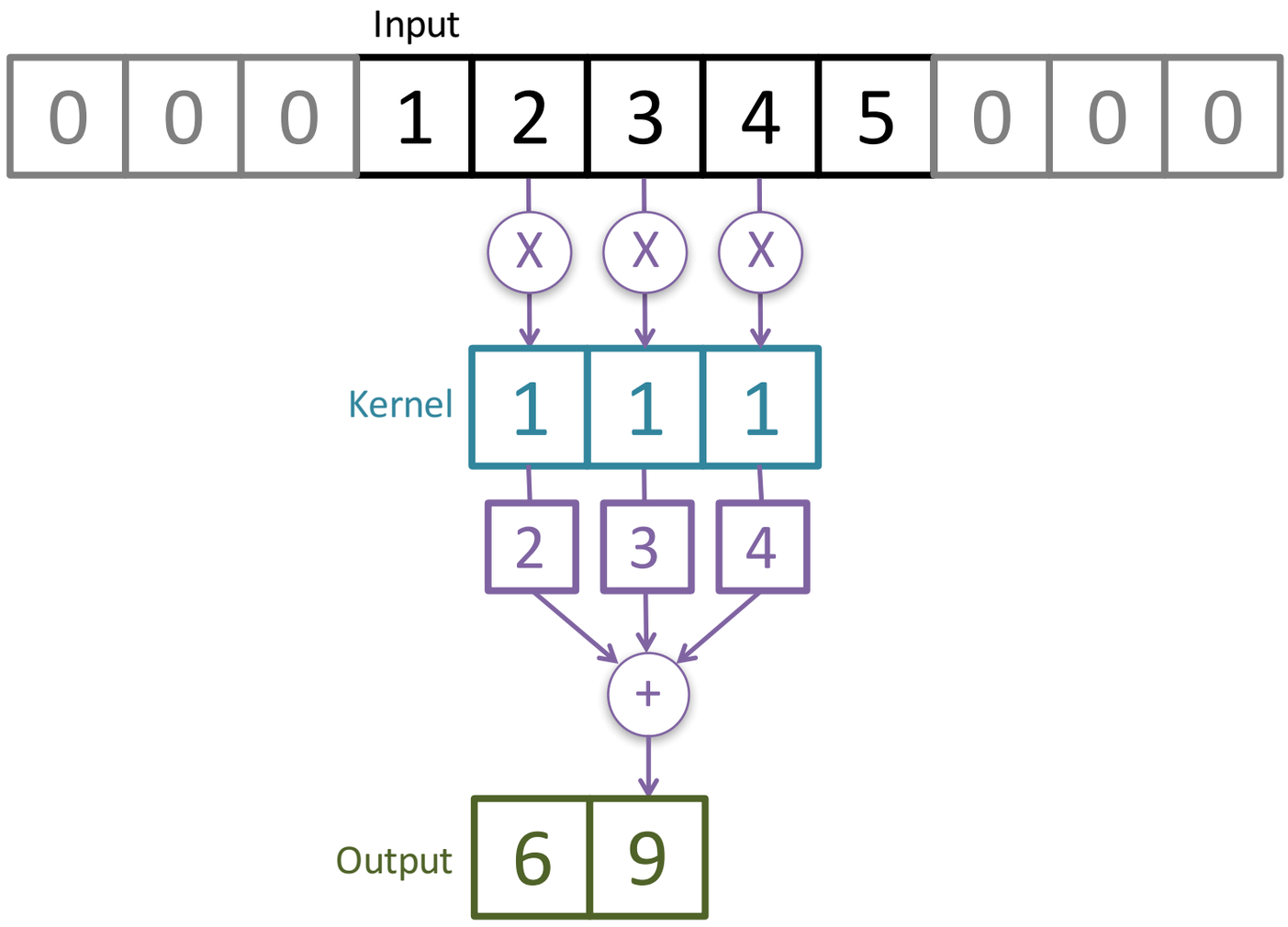
$r = n + k - 1$

BOUNDARY HANDLING: "VALID"

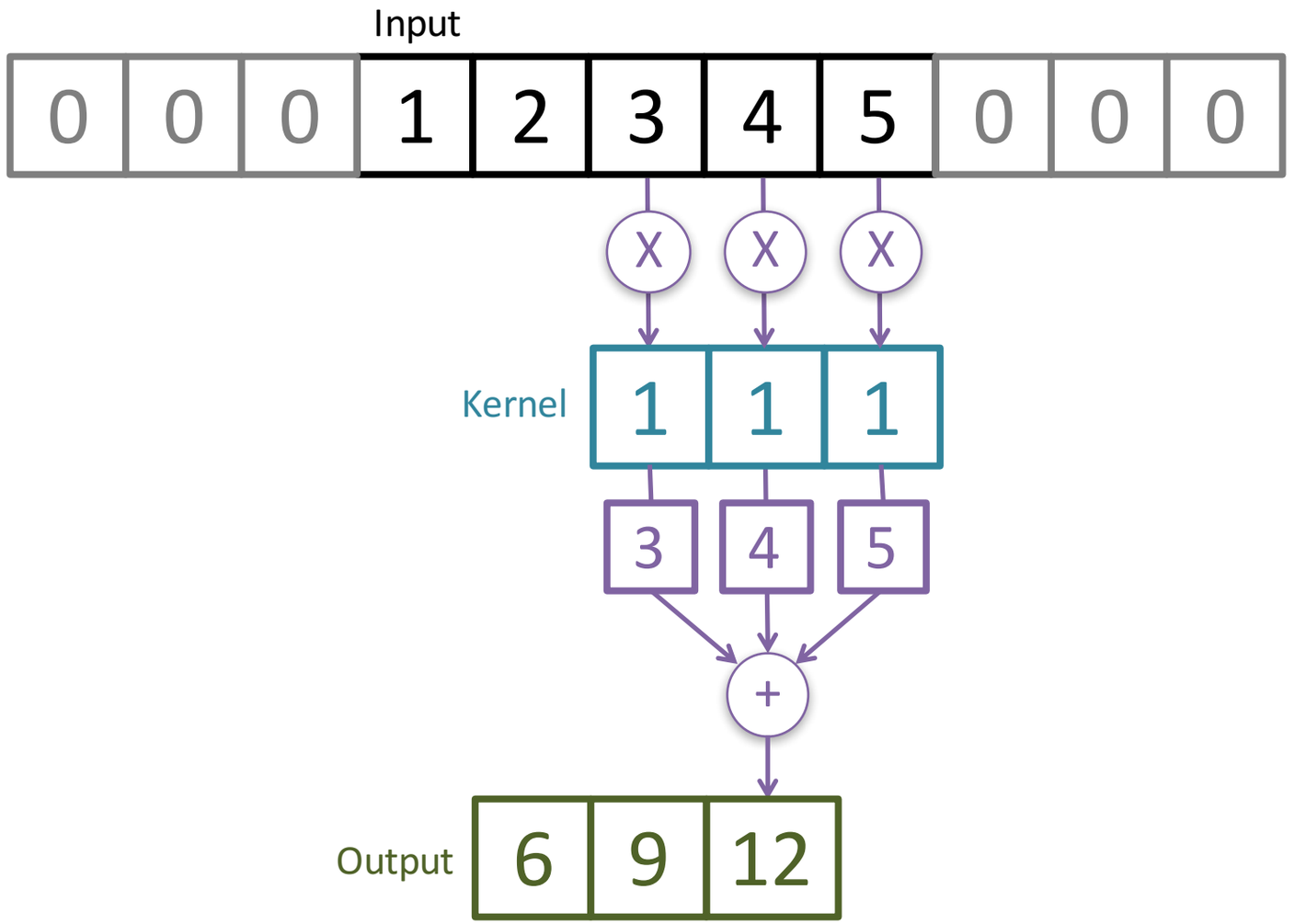
Doesn't make use of padding, leftmost part of kernel starts at input



BOUNDARY HANDLING: "VALID"



BOUNDARY HANDLING: "VALID"



BOUNDARY HANDLING: "VALID"

Input: n = 5

0	0	0	1	2	3	4	5	0	0	0

k = 3

Kernel

1	1	1

r = n – k + 1

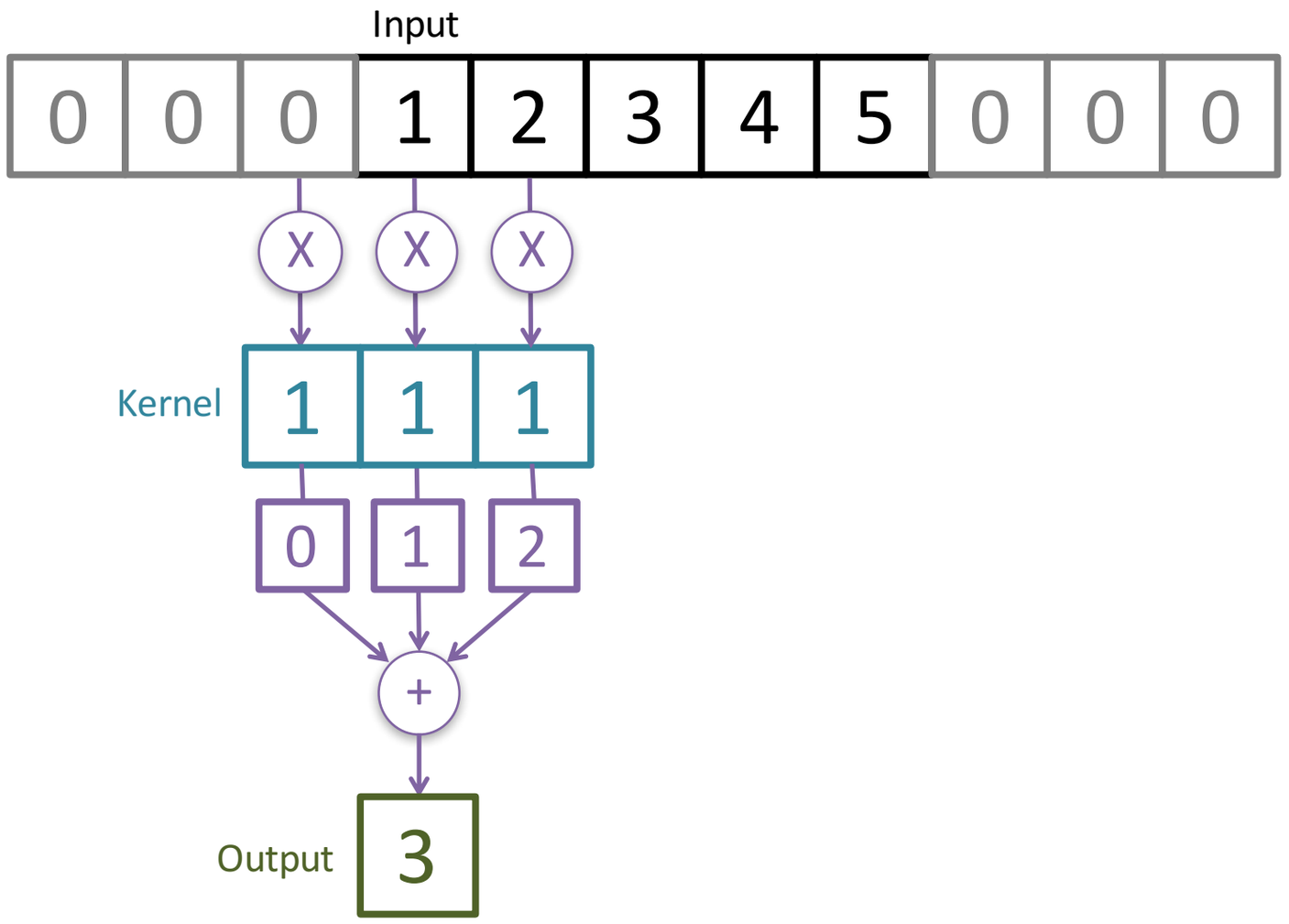
Output

6	9	12

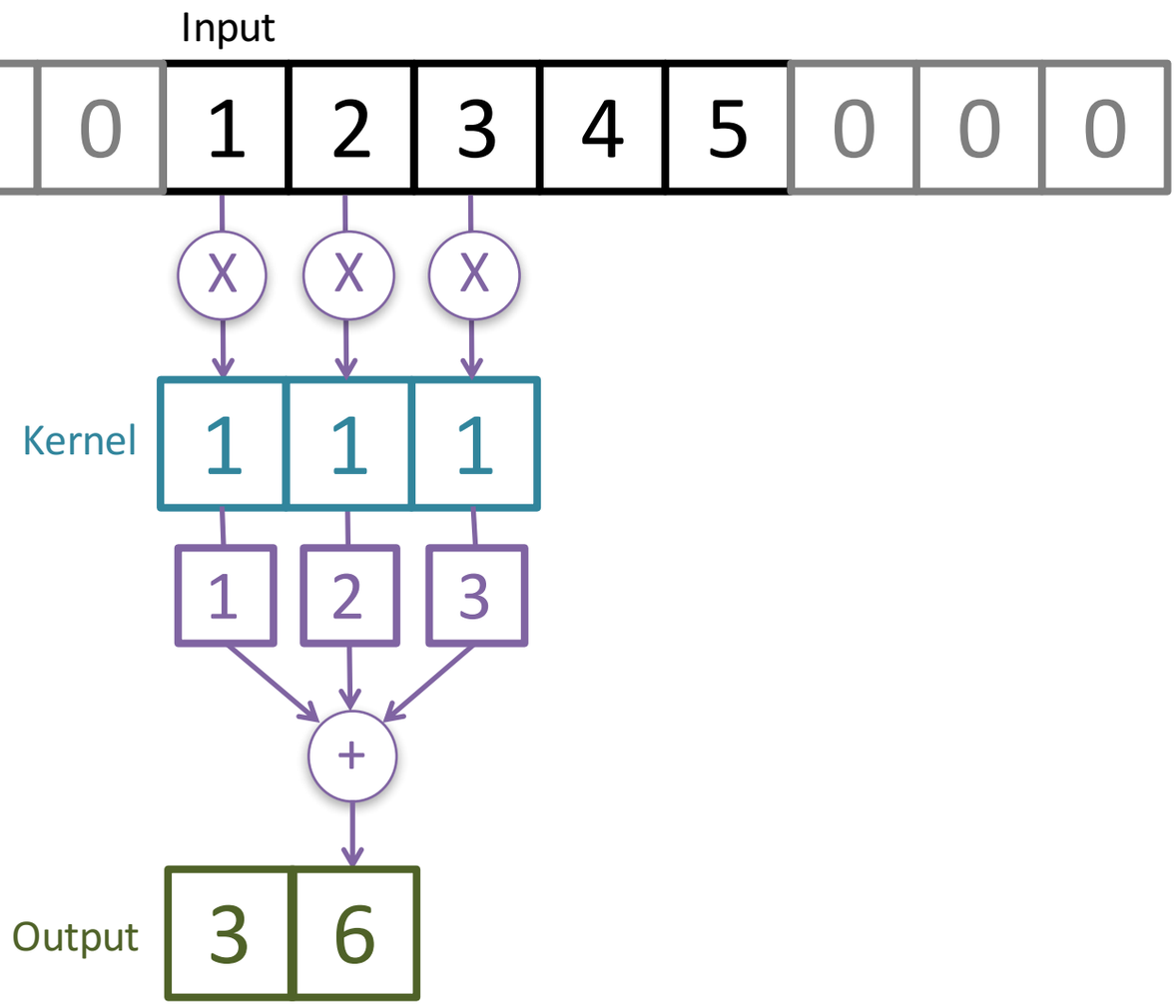
Output is shorter than input

BOUNDARY HANDLING: "SAME"

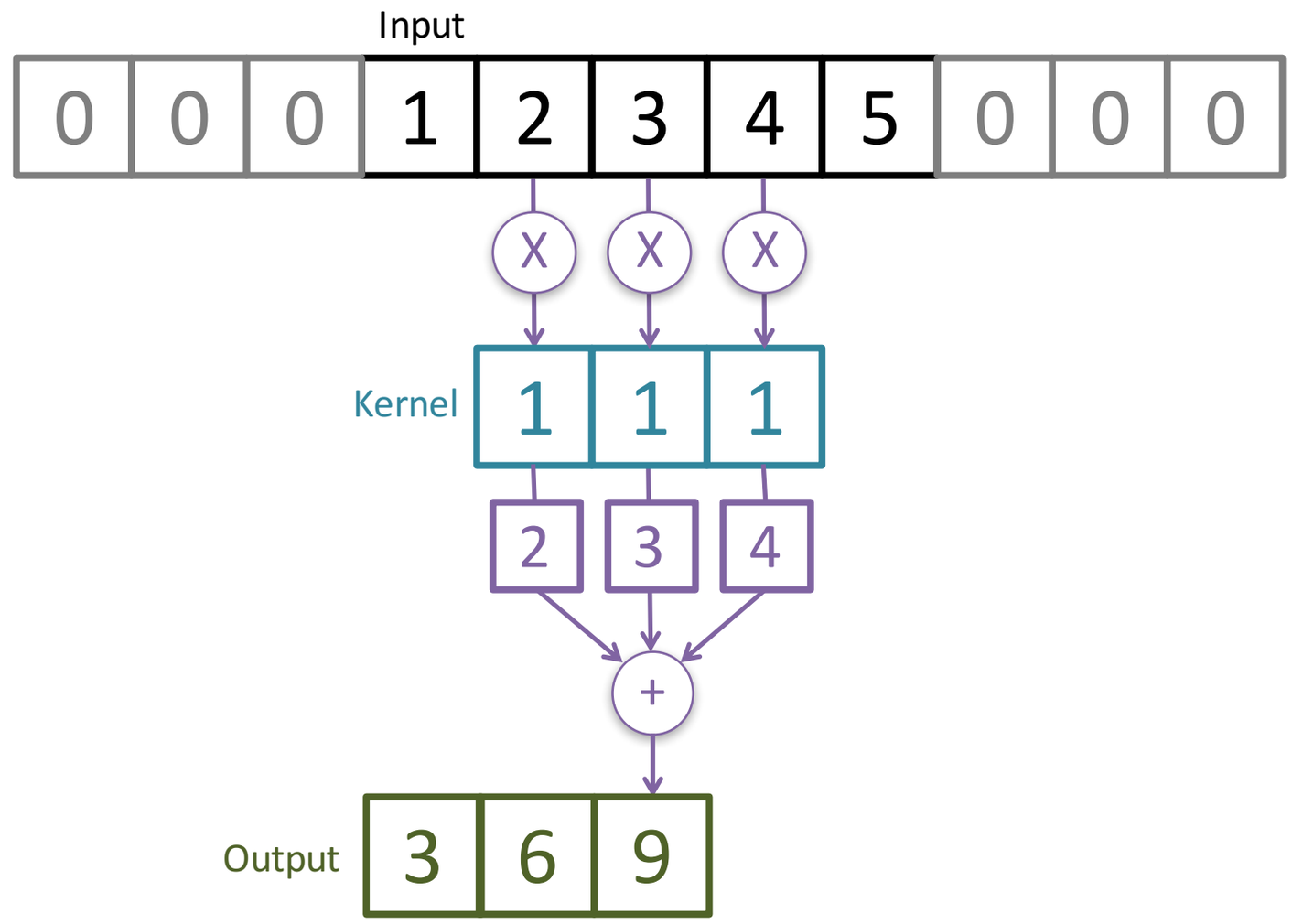
Makes use of padding such that input and output size are identical



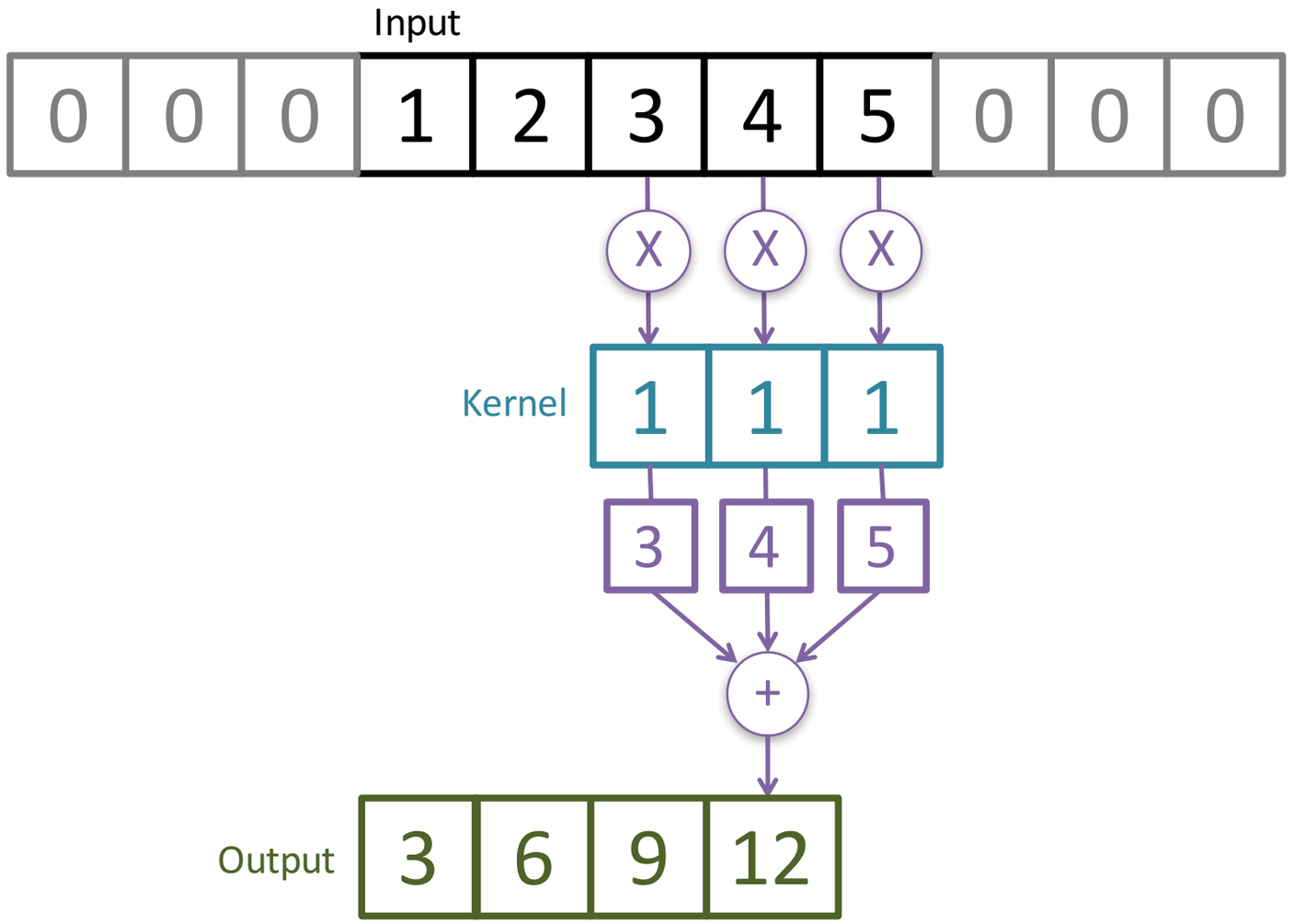
BOUNDARY HANDLING: "SAME"



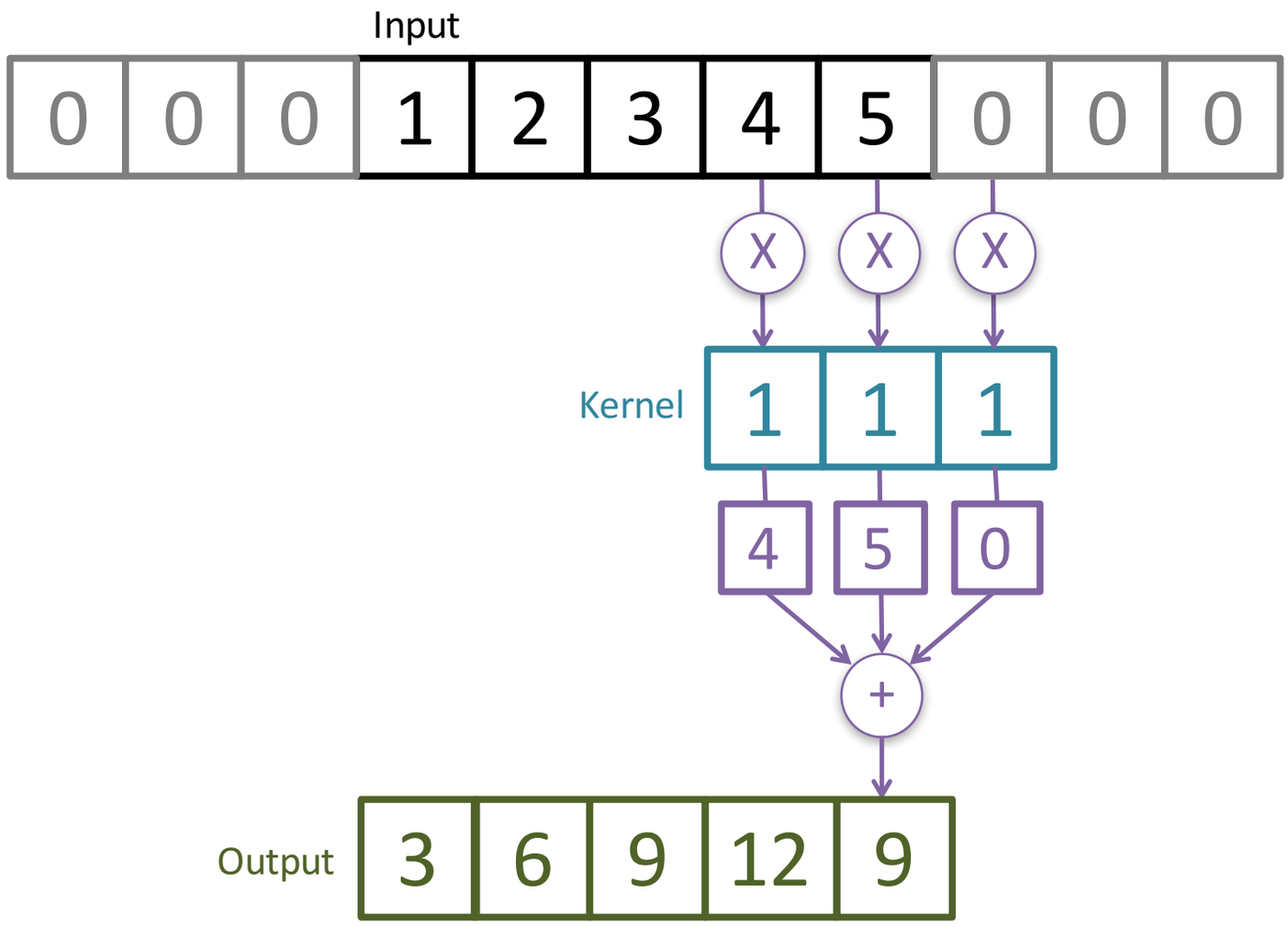
BOUNDARY HANDLING: "SAME"



BOUNDARY HANDLING: "SAME"



BOUNDARY HANDLING: "SAME"



BOUNDARY HANDLING: "SAME"

Input: n = 5

0	0	0	1	2	3	4	5	0	0	0

k = 3

Kernel

1	1	1

Output is same length as input

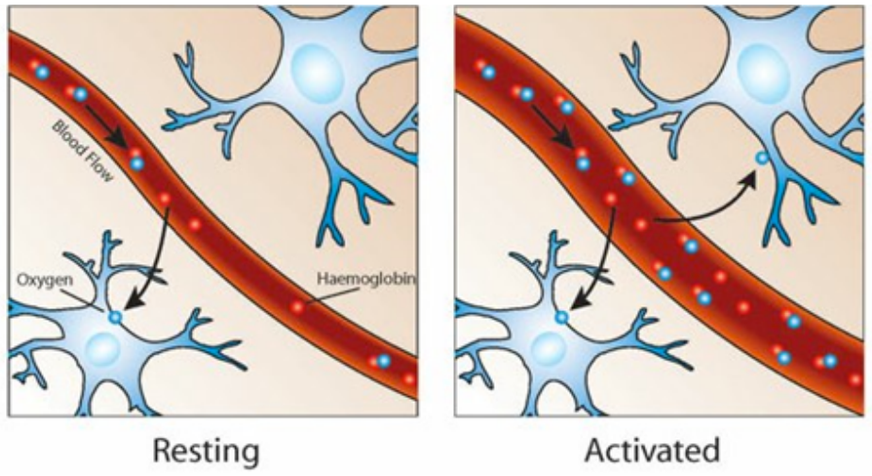
r = n

Output

3	6	9	12	9

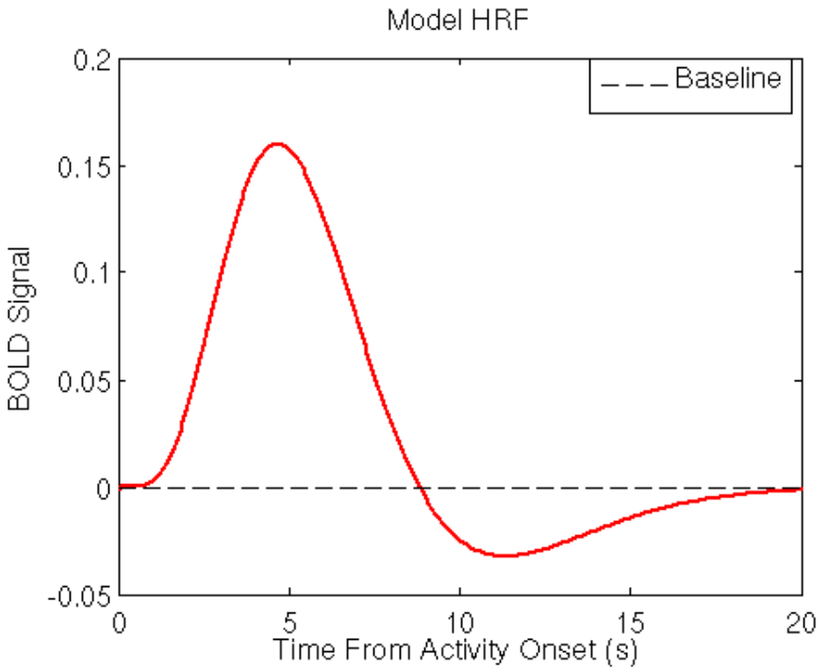
FMRI ANALYSIS: 1D CONVOLUTION

We can measure brain activity in awake, behaving humans with fMRI. Most fMRI analyses measure the changes in blood oxygen level dependent (BOLD) response



BOLD response is slow

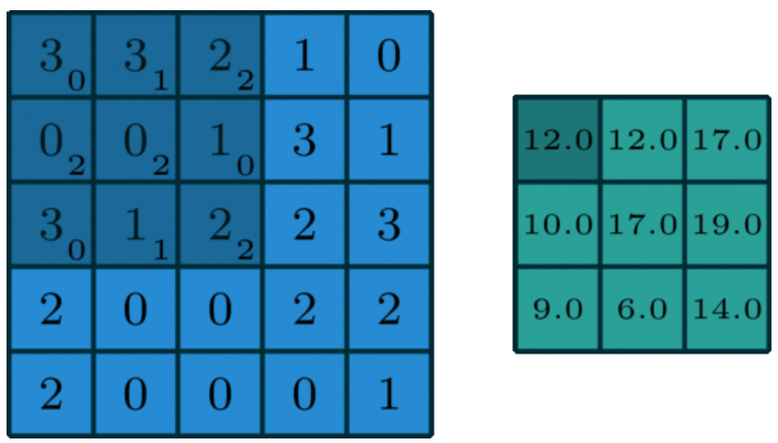
- Peaks ~6 seconds after neural activity
- Is roughly reliable within the same area
- Hemodynamic response is canonical



MATLAB PARTS 1-2

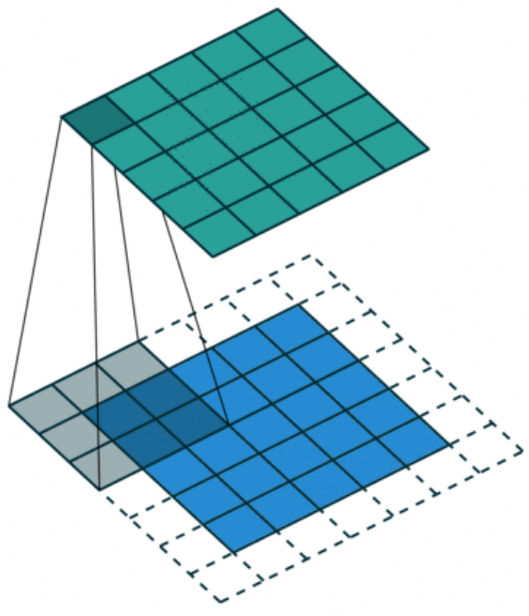
IMAGE PROCESSING: 2D CONVOLUTION

Think of 2D convolutions the same way as 1D convolutions: sliding on function on top of another, multiplying and adding

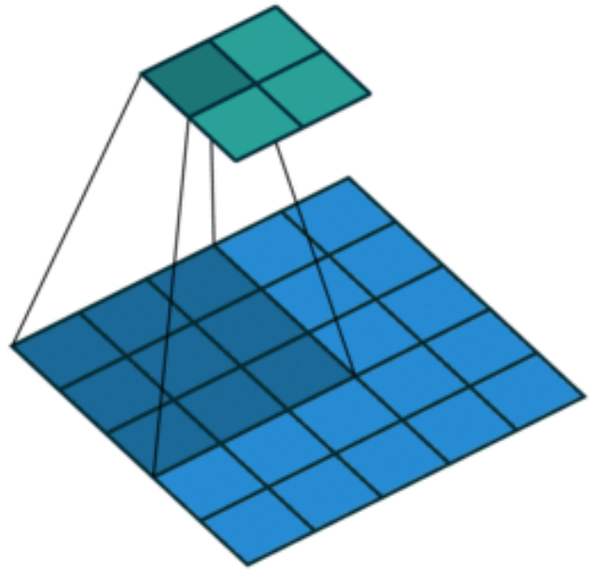


Otherwise, the same ideas that we've already been talking about apply

Padding



Striding



MATLAB PARTS 3-4

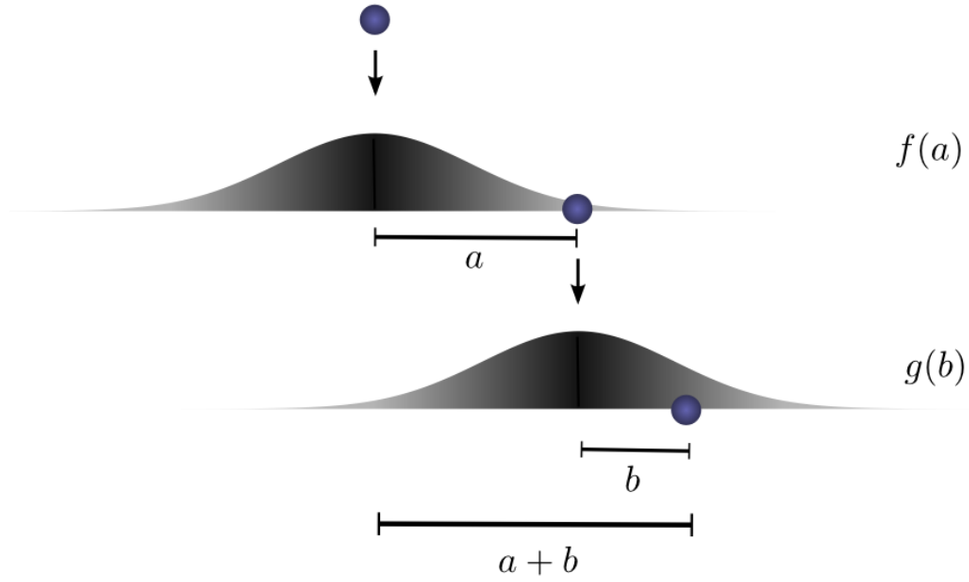
A PROBABILISTIC EXAMPLE

Let's say we're dropping a ball from a fixed height and are only considering how far the ball travels after it hits the ground in one dimension.

How likely is it that the ball will go a distance c if you drop it once, and then drop it a second time from the point at which it lands?

After the first drop: *f(a)* units from start
After the second drop: *g(b)* units from *a*

If we fix *a* and *b* and *a* + *b* = *c*, the probability that the ball will go a distance *c* is just *f(a)·g(b)*

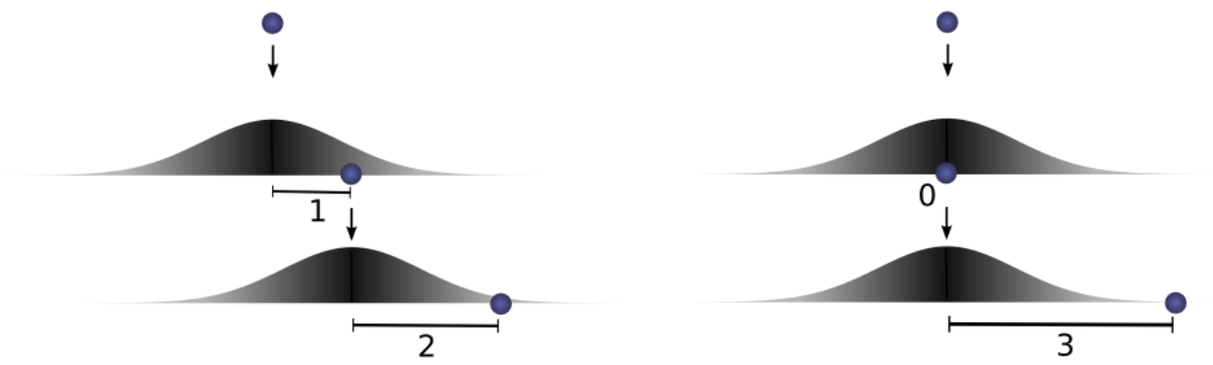


A PROBABILISTIC EXAMPLE

Here's a specific discrete example: if we want c to be 3 and the ball rolls a distance a=2 on the first drop, then it must roll b=1 on the second roll.

The probability of this is $f(2)\cdot g(1)$. But, we could've gotten a distance of 3 in many combinations: a=1 and b=2, a=0 and b=3, etc.

To find the total likelihood of the ball reaching distance c, we have to consider all of these partitions and sum the probability.

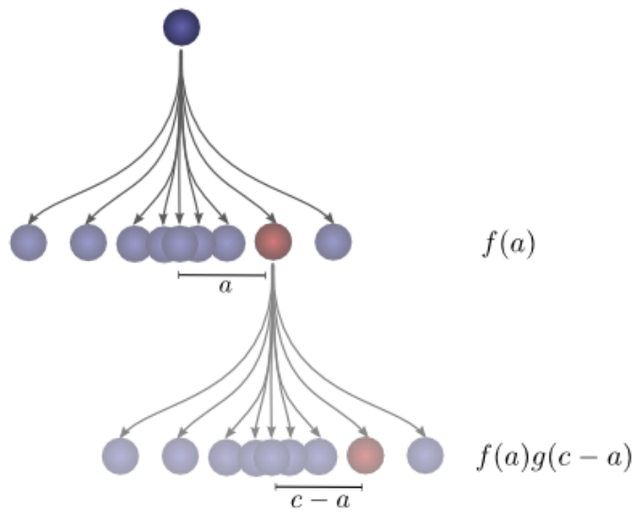


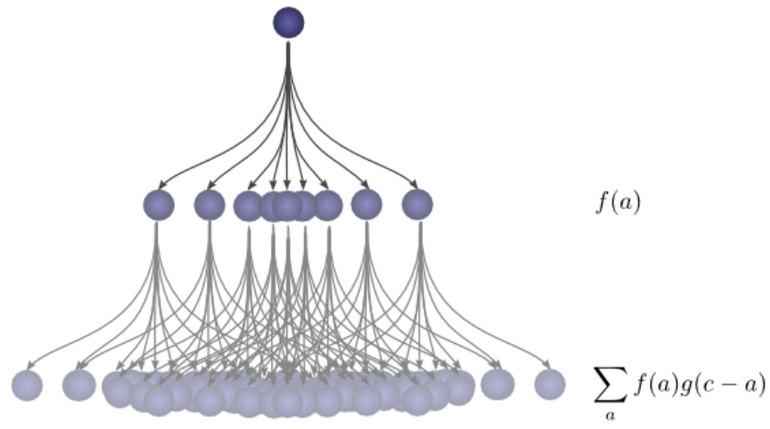
This is a convolution!

$$\ldots f(0)\cdot g(3)+f(1)\cdot g(2)+f(2)\cdot g(1)\ldots$$

$$(f*g)(c)=\sum_{a+b=c} f(a)\cdot g(b)$$

$$(f*g)(c)=\sum_{a} f(a)\cdot g(c-a)$$

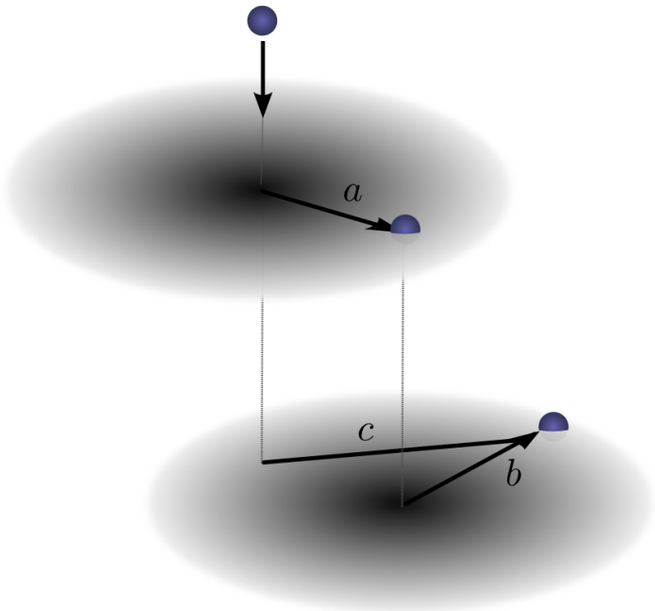




HIGHER DIMENSIONALITY

The idea of convolutions generalizes to higher dimensions. Consider our example from before, but the position of the ball shifts in two dimensions rather than one.

$$(f * g)(c) = \sum_{a+b=c} f(a) \cdot g(b)$$



$f(a)$

$g(b)$

Except now *a* and *c* are vectors

$$(f * g)(c_1, c_2) = \sum_{\substack{a_1+b_1=c_1,\\ a_2+b_2=c_2}} f(a_1, a_2) \cdot g(b_1, b_2)$$

$$(f * g)(c_1, c_2) = \sum_{a_1, a_2} f(a_1, a_2) \cdot g(c_1 - a_1, c_2 - a_2)$$