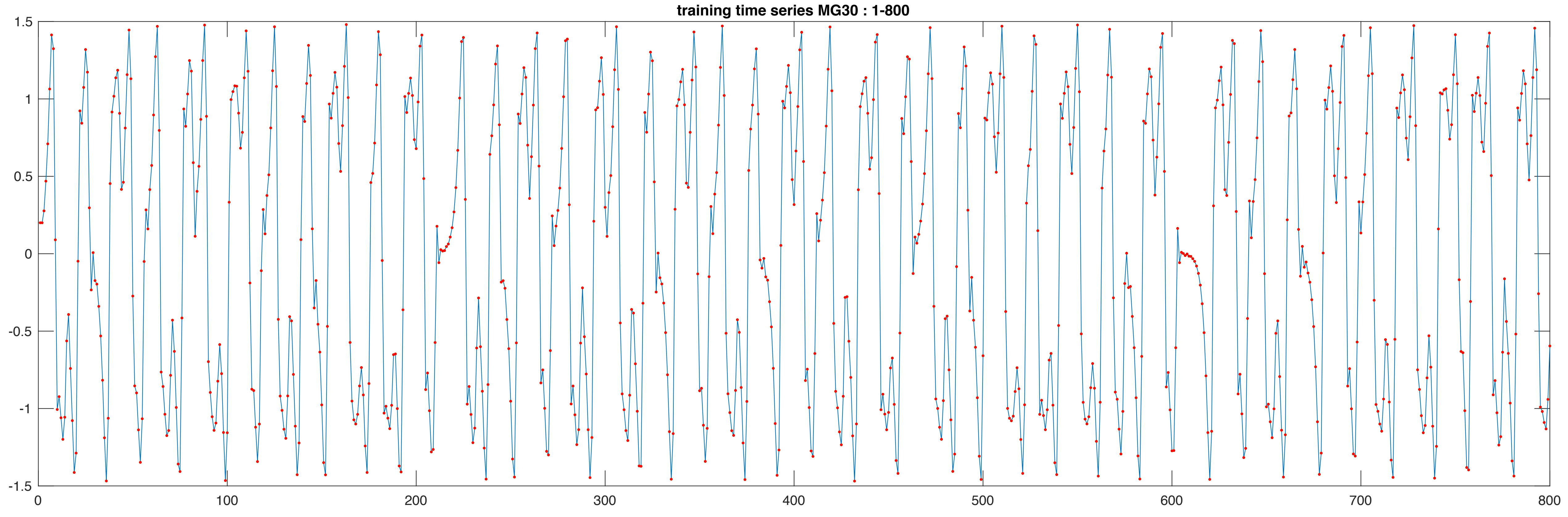
Neural Organization : Convolution

Mackey-Glass 30

$$\frac{\partial x}{\partial t} = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t),$$

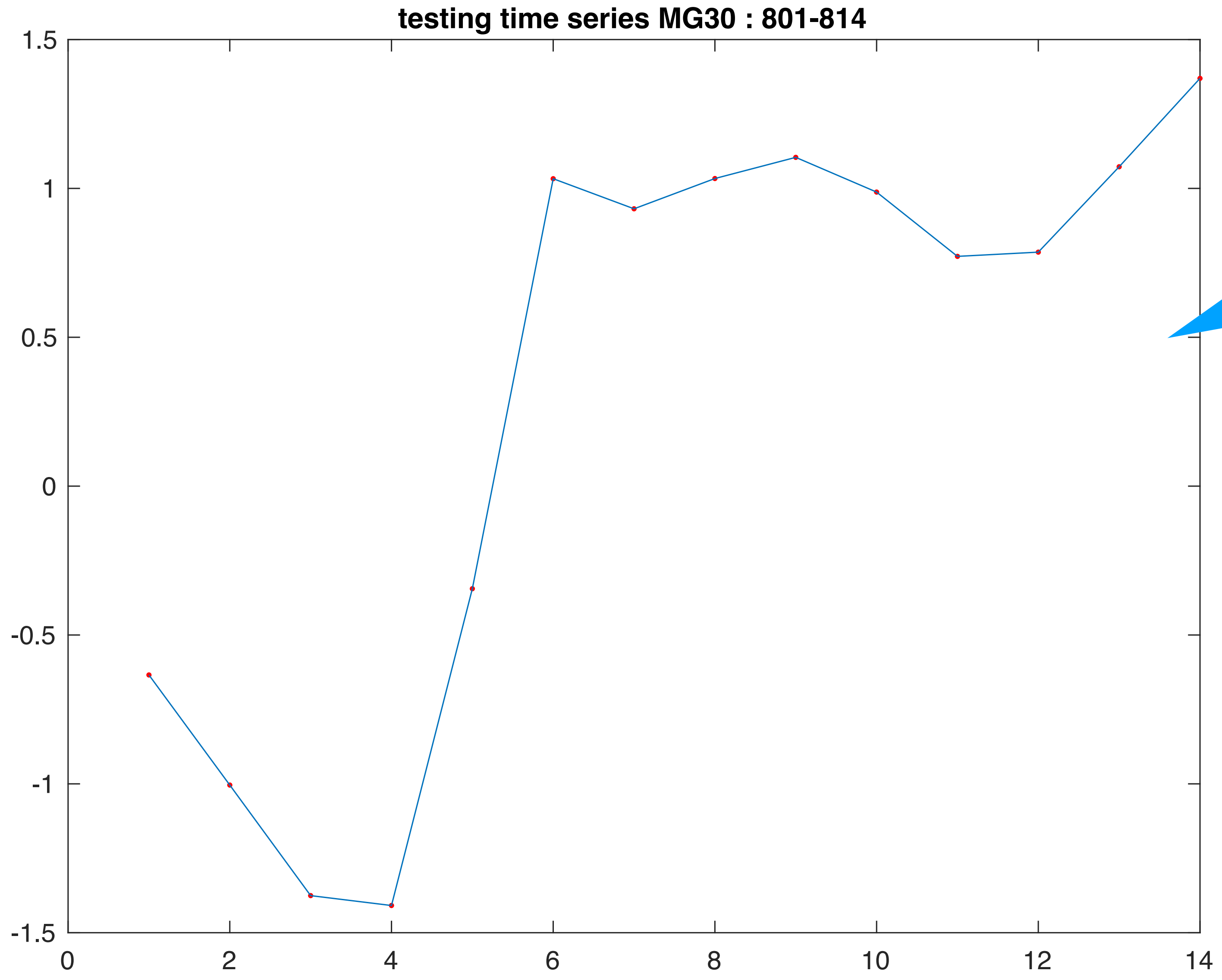
$\tau = 30$ $a = 0.2$, $c = 10$ and $b = 0.1$

Training time series MG30 : 1-800



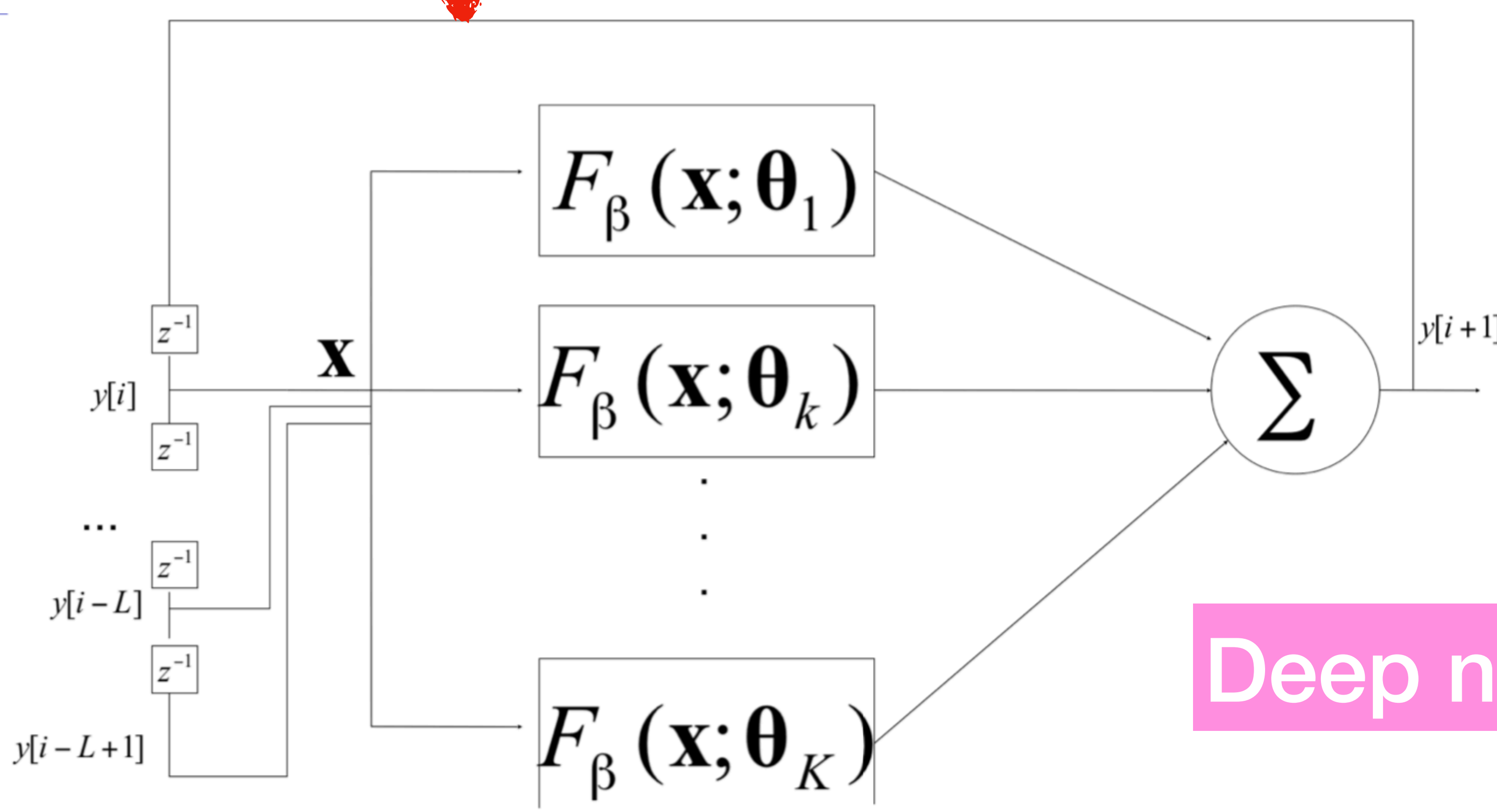
save MG30_time_series.mat x_training x_testing

Testing time series MG30 : 801-814



Training time series MG30 : 1-800

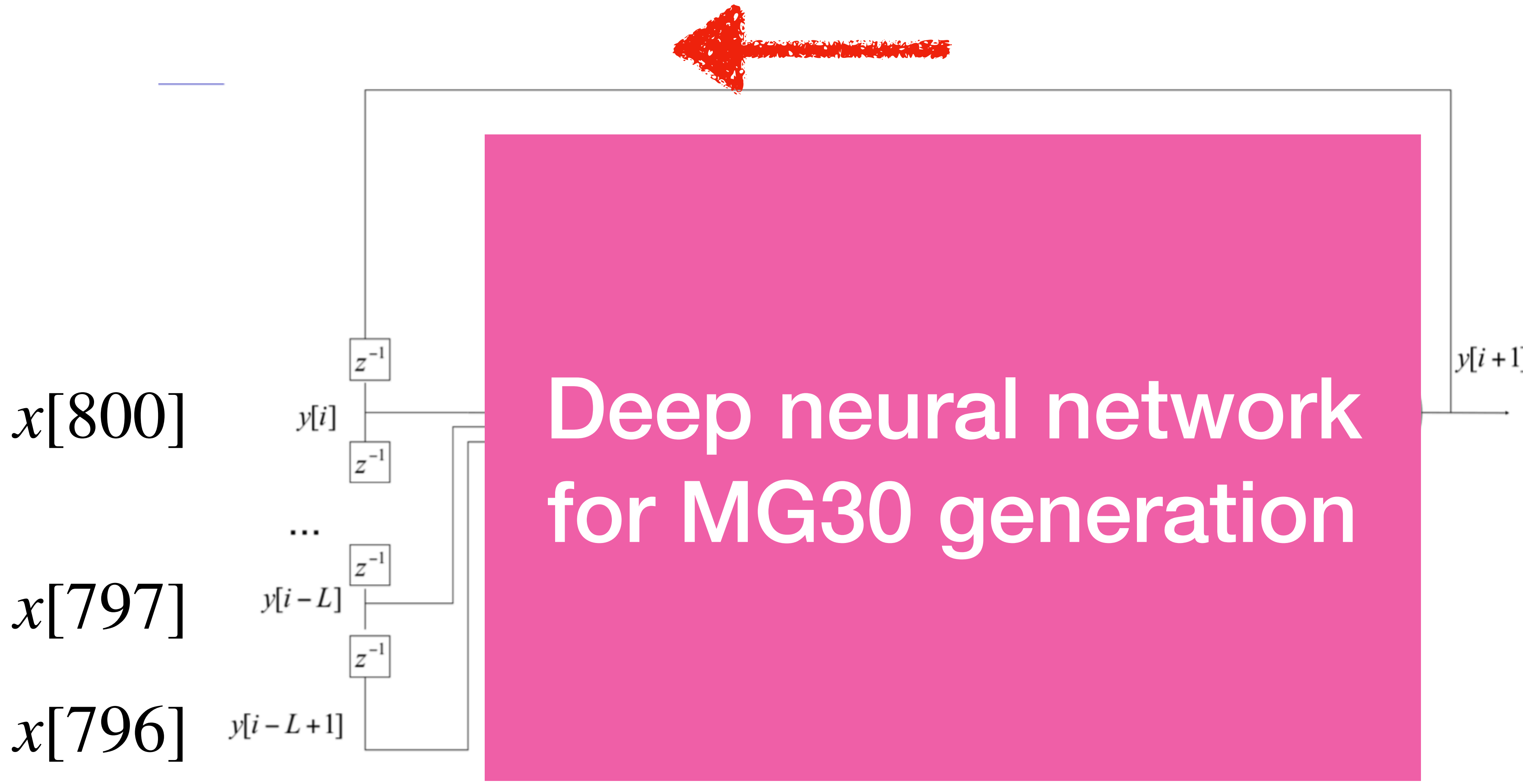
Deep Learning



Deep neural network

$$o_t = f(\mathbf{x}_t = (o_{t-L}, o_{t-L+1}, \ldots, o_{t-1})^T),$$

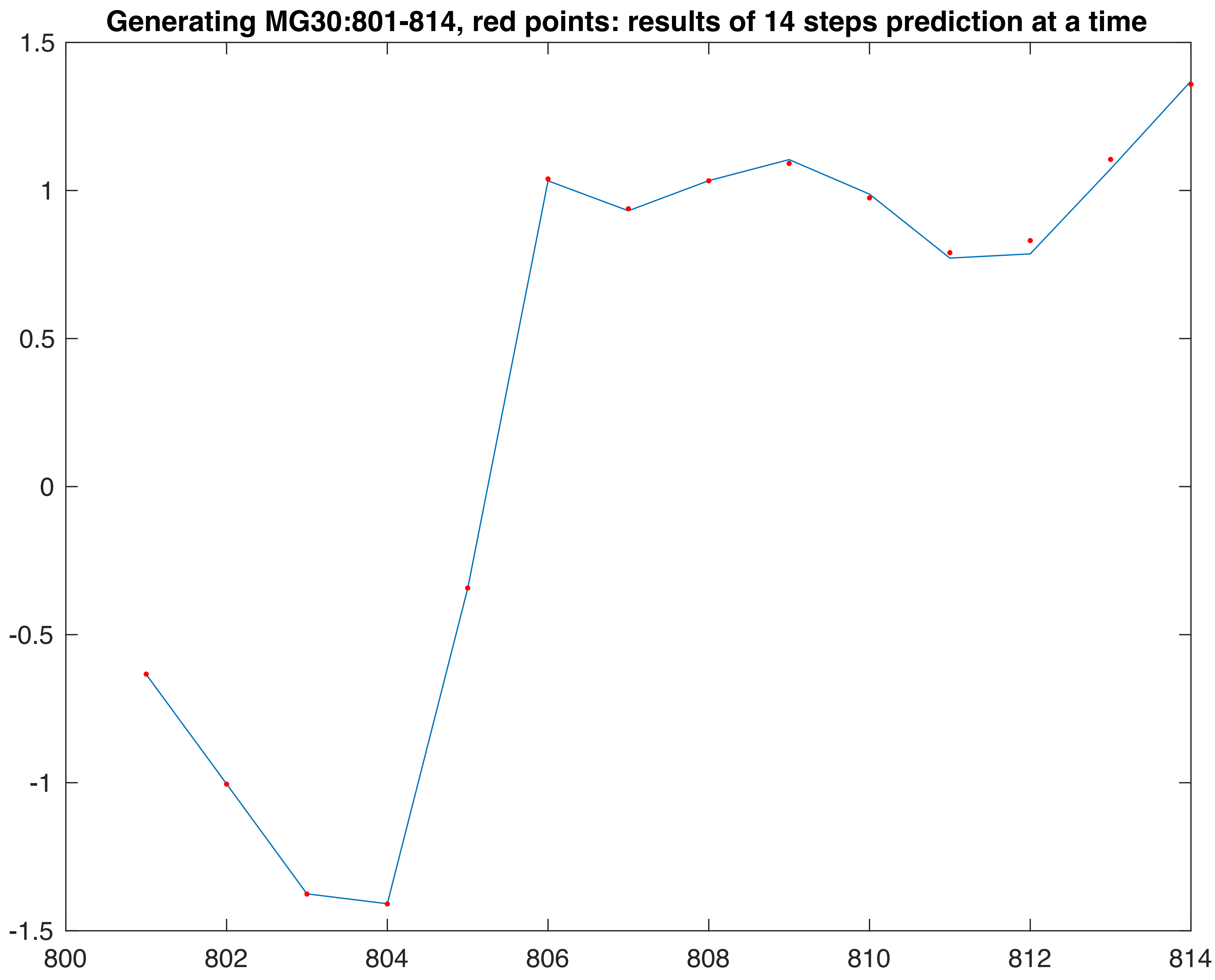
Initialization with $x[796] \ldots x[800]$

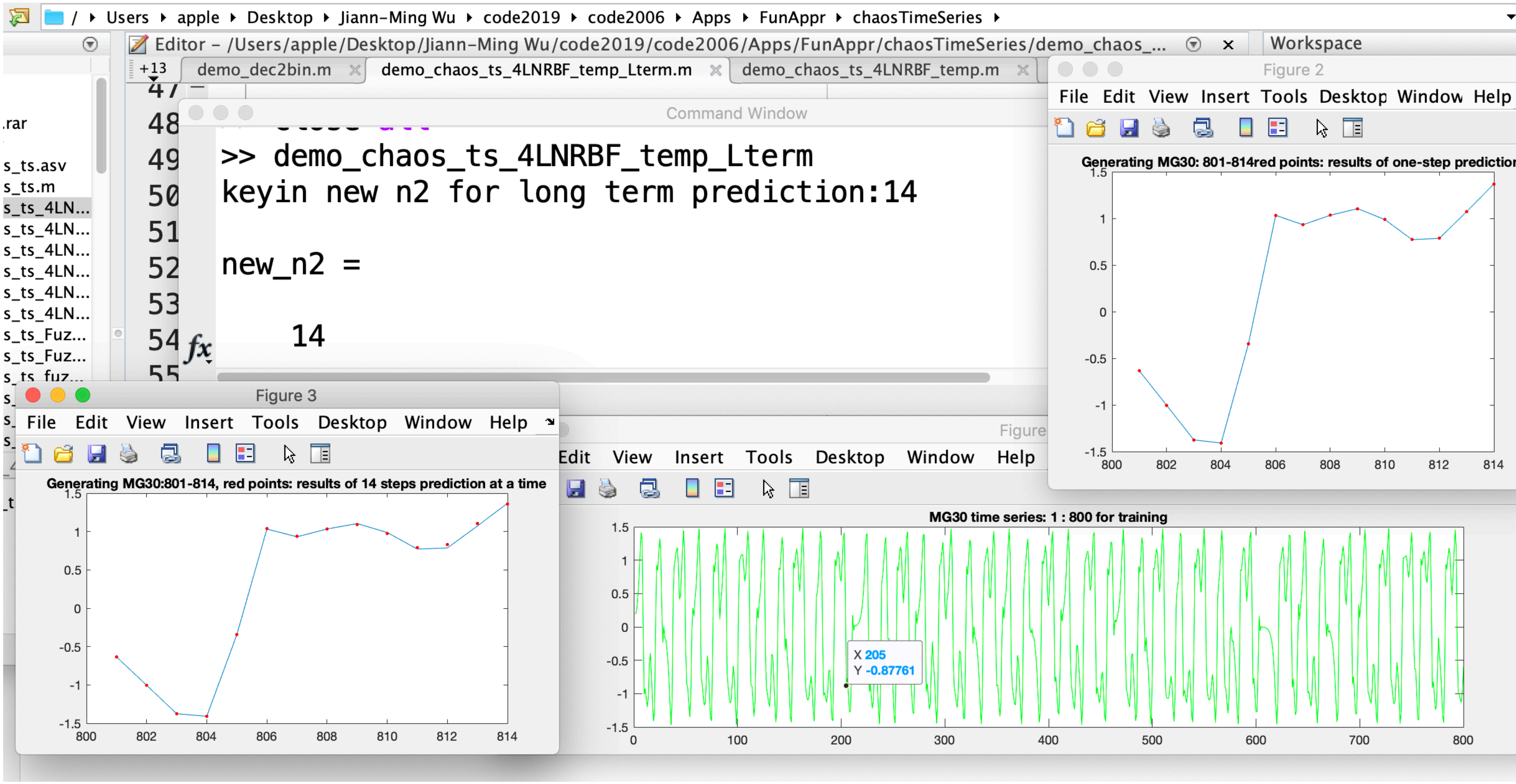


$$o_t = f(\mathbf{x}_t = (o_{t-L}, o_{t-L+1}, \ldots, o_{t-1})^T),$$

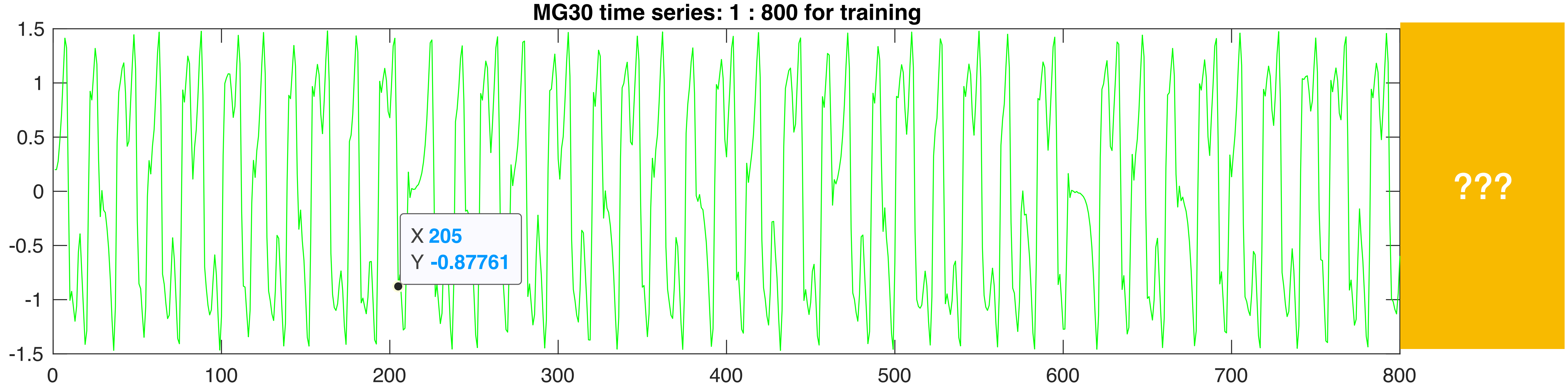
for generating testing time series MG30 : 801-814

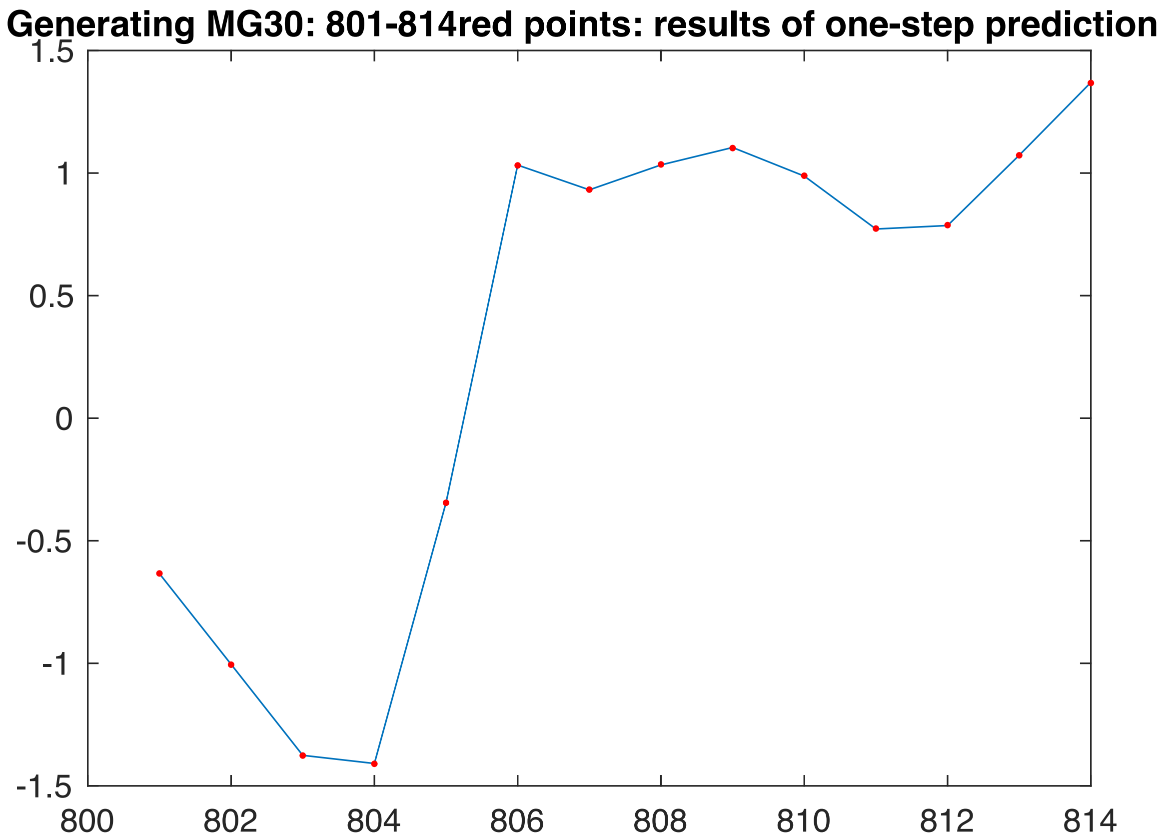
Generated MG30: 801-814 for prediction

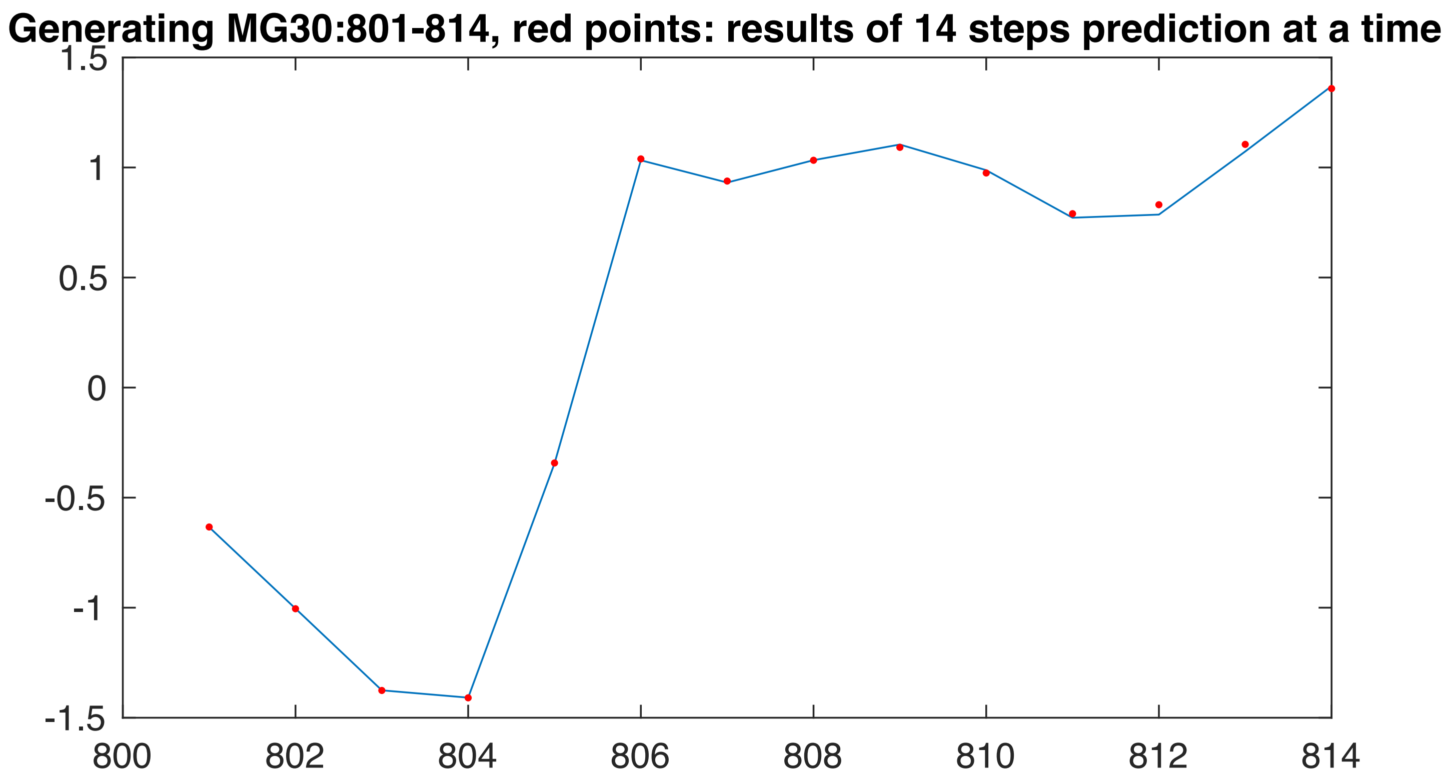


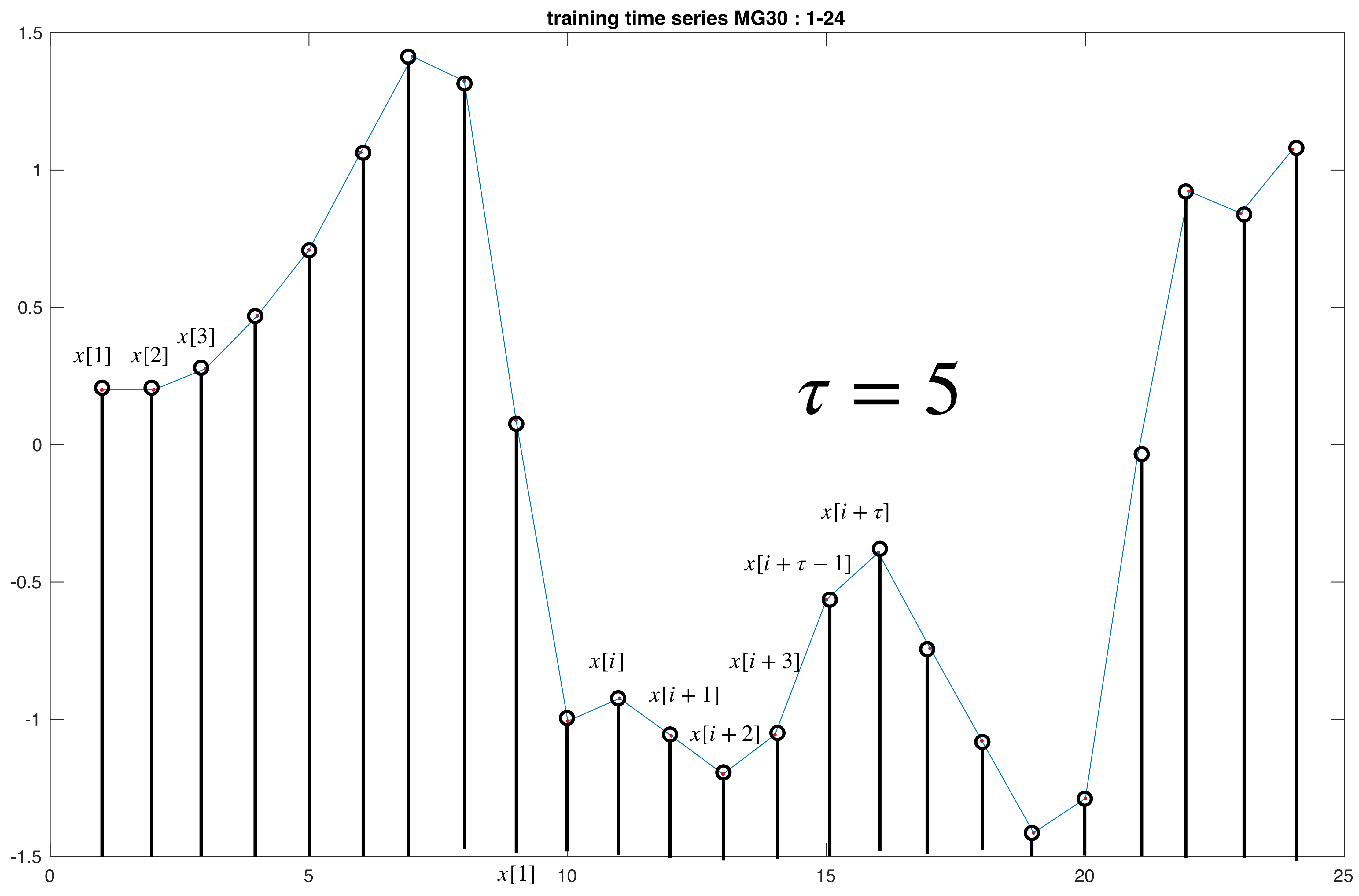

/ Users apple Desktop Jiann-Ming Wu code2019 code2006 Apps FunAppr chaosTimeSeries
Editor – /Users/apple/Desktop/Jiann-Ming Wu/code2019/code2006/Apps/FunAppr/chaosTimeSeries/demo_chaos_...
Workspace
demo_dec2bin.m
demo_chaos_ts_4LNRBF_temp_Lterm.m
demo_chaos_ts_4LNRBF_temp.m
Command Window
>> demo_chaos_ts_4LNRBF_temp_Lterm
keyin new n2 for long term prediction:14
new_n2 =
14
Figure 2
File Edit View Insert Tools Desktop Window Help
Generating MG30: 801-814red points: results of one-step prediction
Figure 3
File Edit View Insert Tools Desktop Window Help
Generating MG30:801-814, red points: results of 14 steps prediction at a time
MG30 time series: 1 : 800 for training
X 205
Y -0.87761

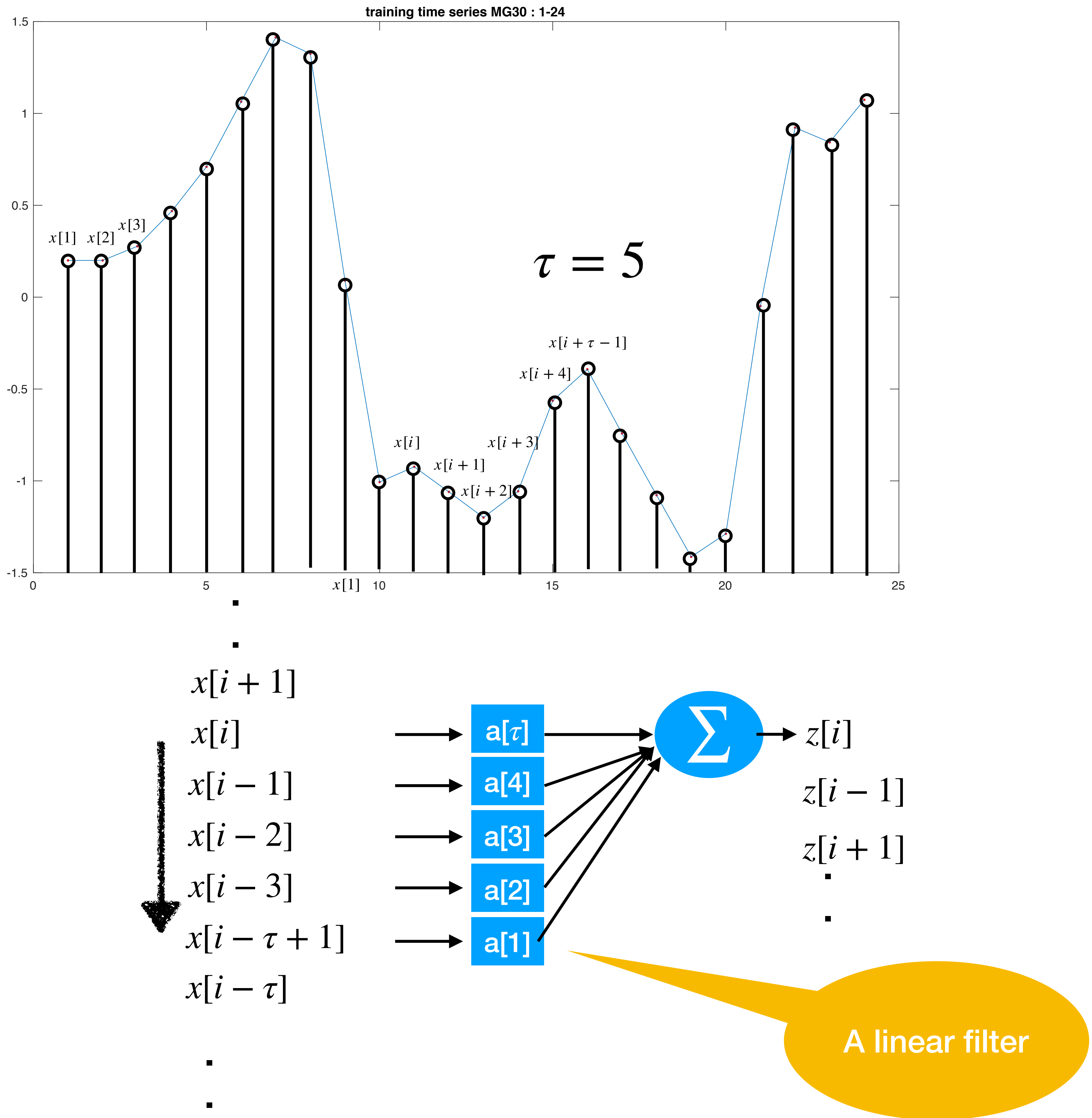
Generating MG30:801-814, red points: results of 14 steps prediction at a time





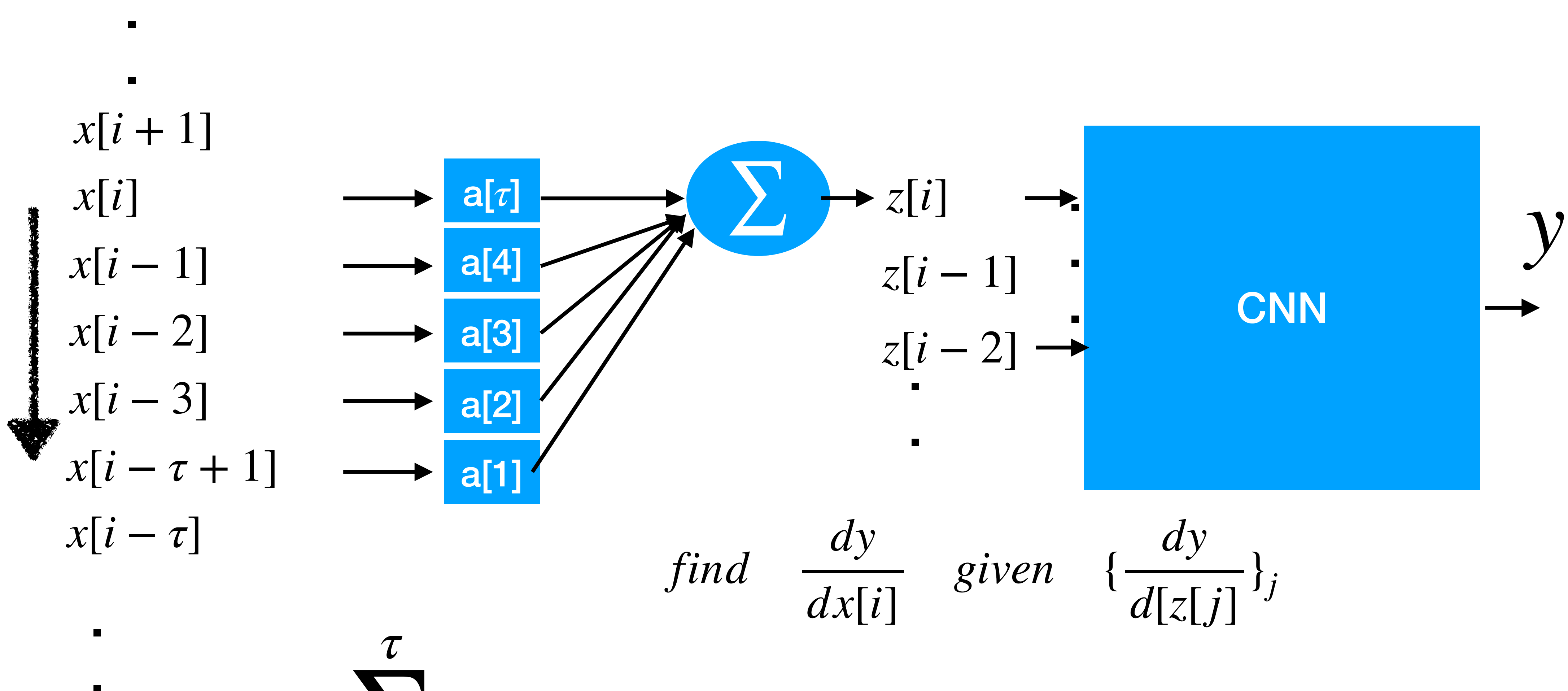



training time series MG30 : 1-24
$x[1]$
$x[2]$
$x[3]$
$\tau = 5$
$x[i+\tau]$
$x[i+\tau-1]$
$x[i]$
$x[i+1]$
$x[i+2]$
$x[i+3]$
$x[1]$
1.5
1
0.5
0
-0.5
-1
-1.5
0
5
10
15
20
25



training time series MG30 : 1-24
1.5
1
0.5
0
-0.5
-1
-1.5
0
5
10
15
20
25
x[1]
x[2]
x[3]
τ = 5
x[i + τ − 1]
x[i + 4]
x[i]
x[i + 1]
x[i + 3]
x[i + 2]
x[1]
x[i + 1]
x[i]
x[i − 1]
x[i − 2]
x[i − 3]
x[i − τ + 1]
x[i − τ]
a[τ]
a[4]
a[3]
a[2]
a[1]
Σ
z[i]
z[i − 1]
z[i + 1]
A linear filter

Feedforward calculation for 1D linear convolution

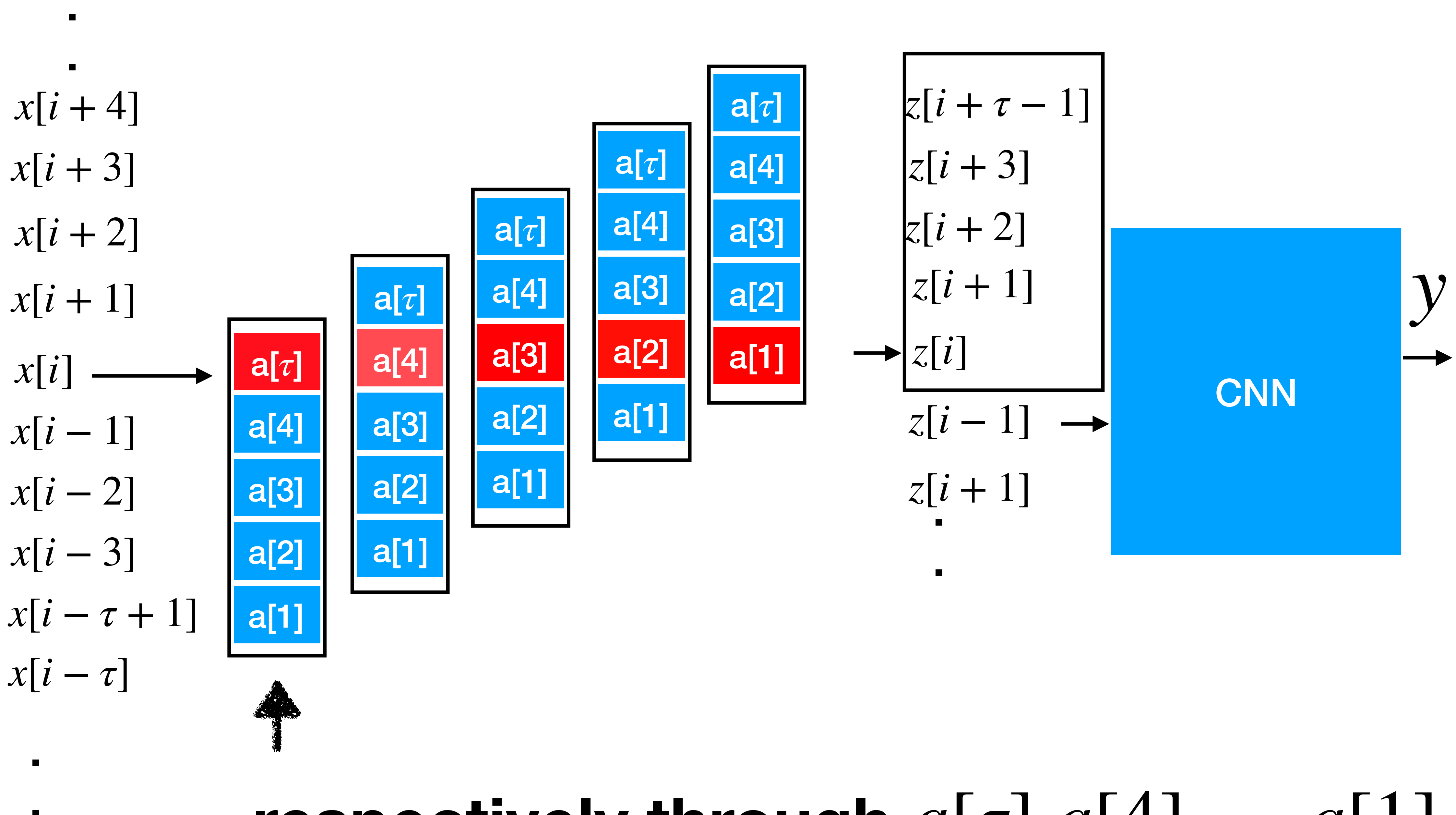


$$find \quad \frac{dy}{dx[i]} \quad given \quad \{\frac{dy}{d[z[j]}\}_j$$

$$z[i] = \sum_{k=1}^{\tau} a[k]x[i-\tau+k]$$

$$= a[1]x[i-\tau+1] + a[2]x[i-\tau+2] + a[3]x[i-\tau+3] + a[4]x[i-\tau+4] + a[\tau]x[i-\tau+\tau]$$

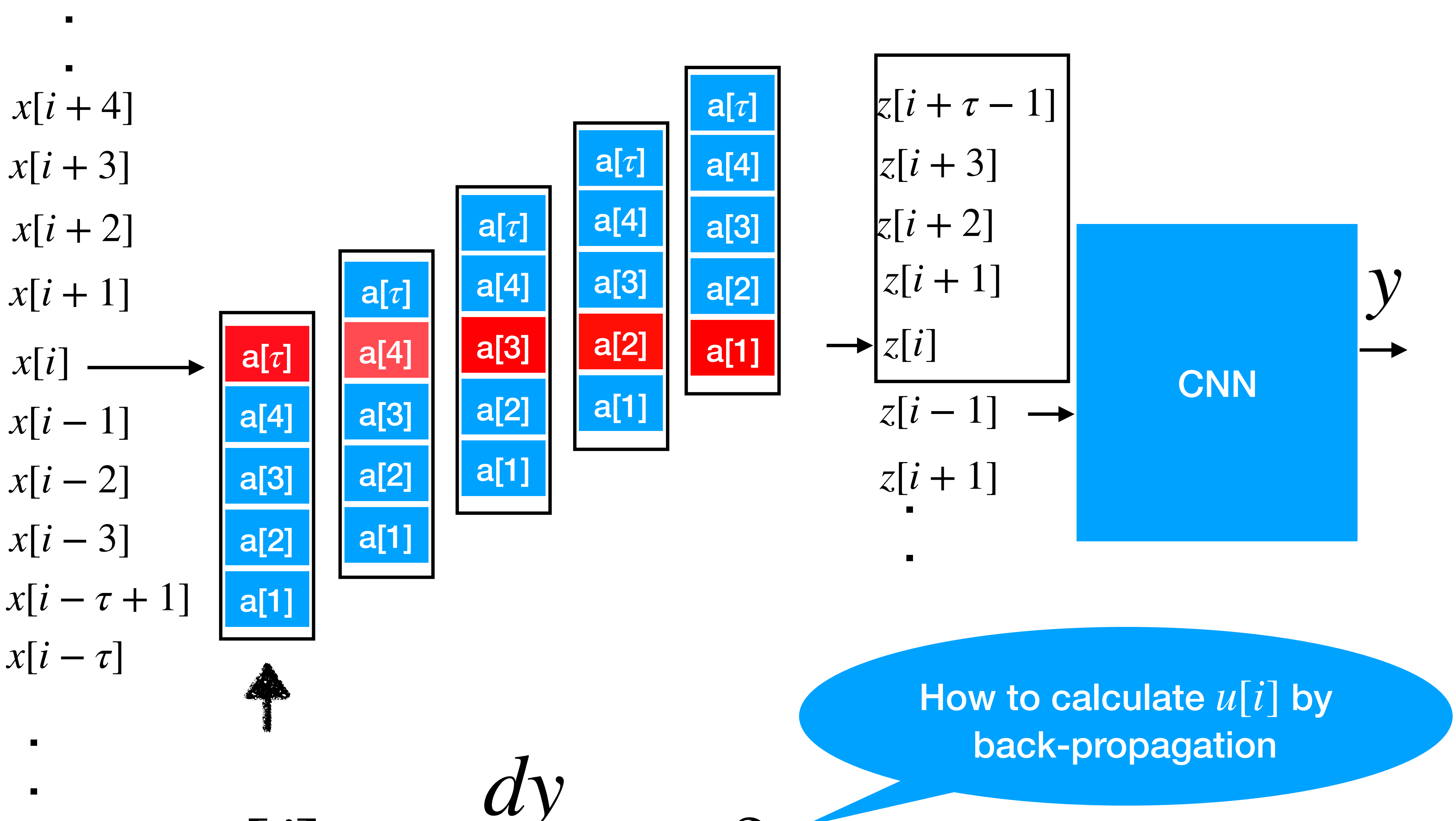
$x[i]$ contributes to $z[i],\ldots,z[i+\tau-1]$



respectively through $a[\tau], a[4], \ldots, a[1]$

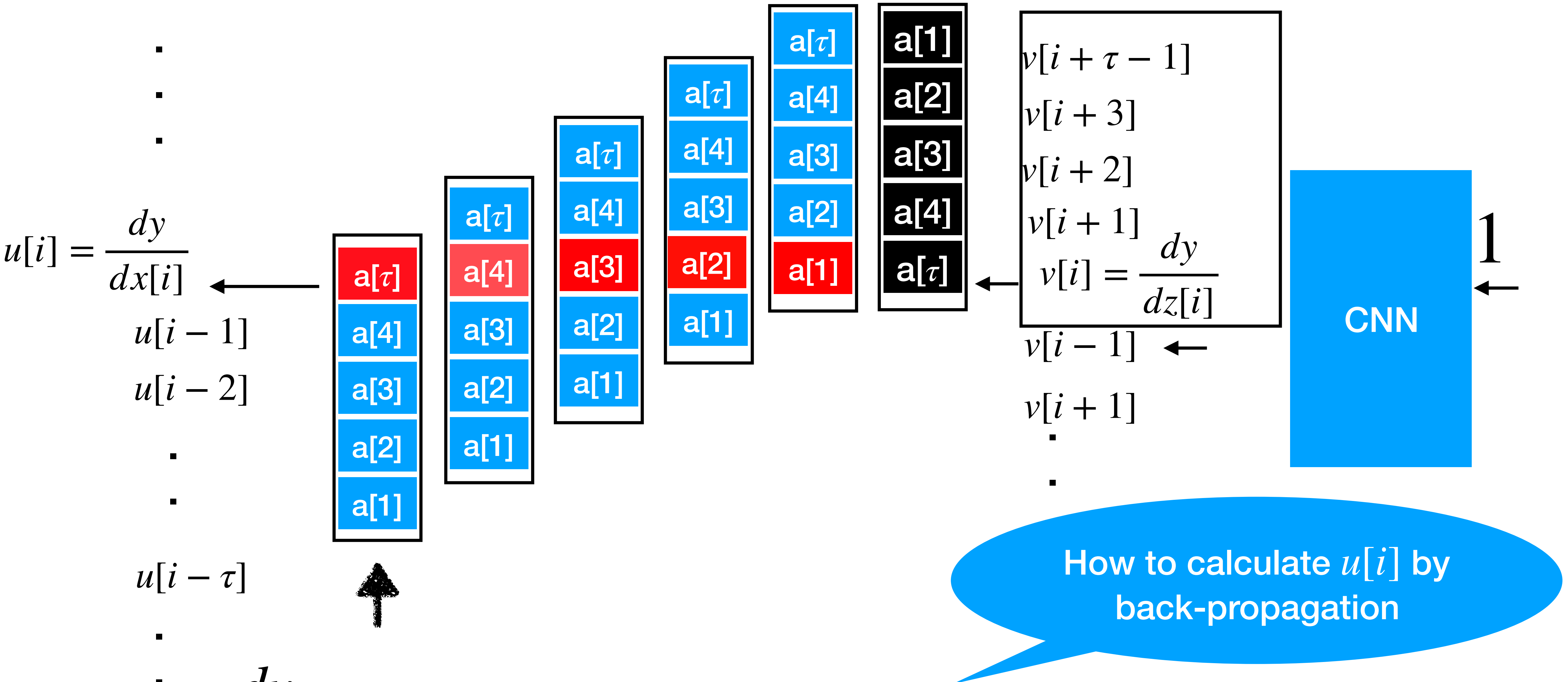
$$\frac{dz[i]}{dx[i]} = ? \quad \frac{dz[i+1]}{dx[i]} = ? \quad \frac{dz[i+2]}{dx[i]} = ? \quad \frac{dz[i+3]}{dx[i]} = ? \quad \frac{dz[i+\tau-1]}{dx[i]} = ?$$

Gradient back-propagation



$$u[i] = \frac{dy}{dx[i]} = ?$$

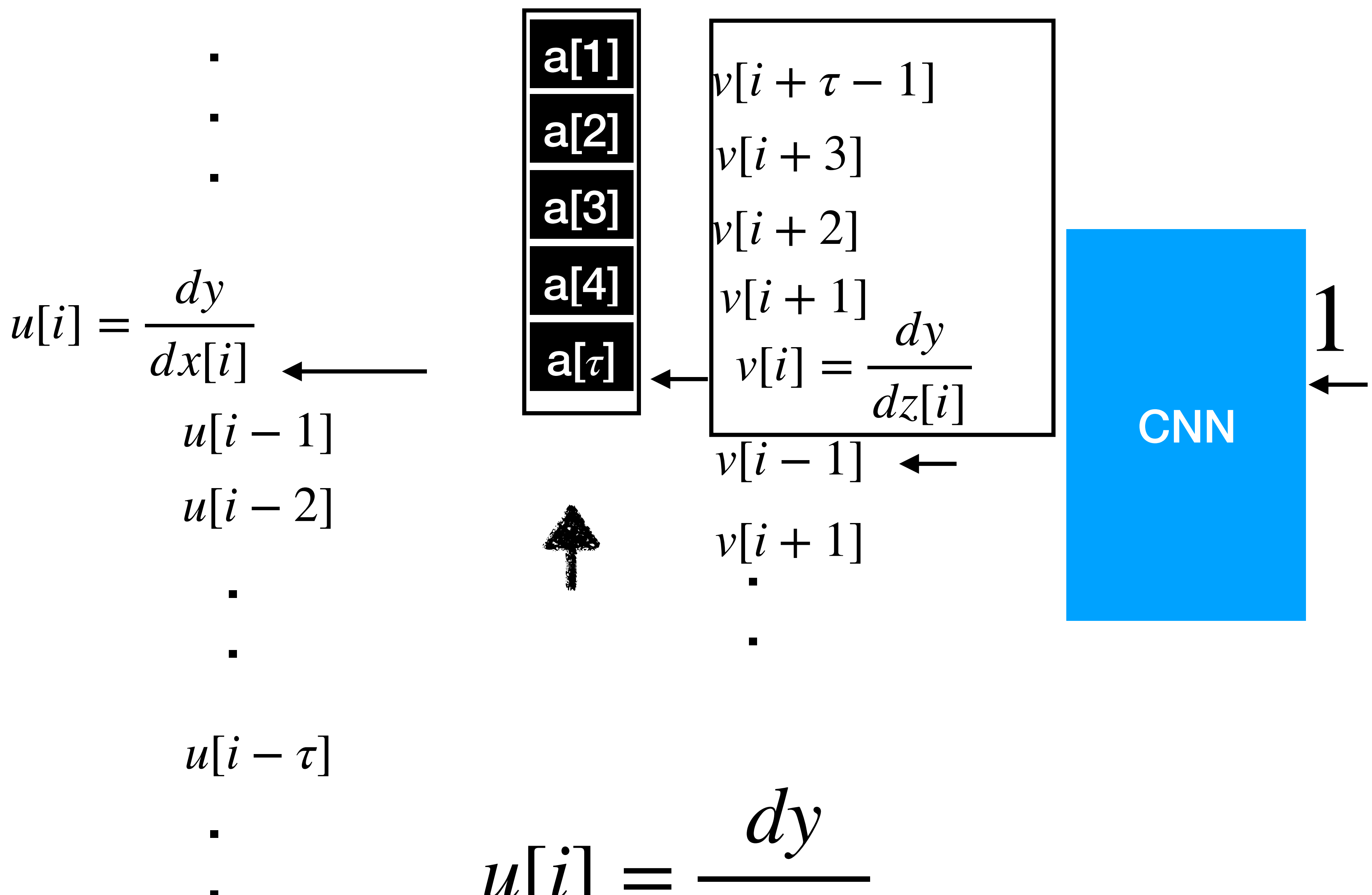
Back-propagation of Gradients



$$u[i] = \frac{dy}{dx[i]} = v[i] * a[\tau] + v[i+1] * a[4] + v[i+2] * a[3] + v[i+3] * a[2] + v[i+4] * a[1]$$

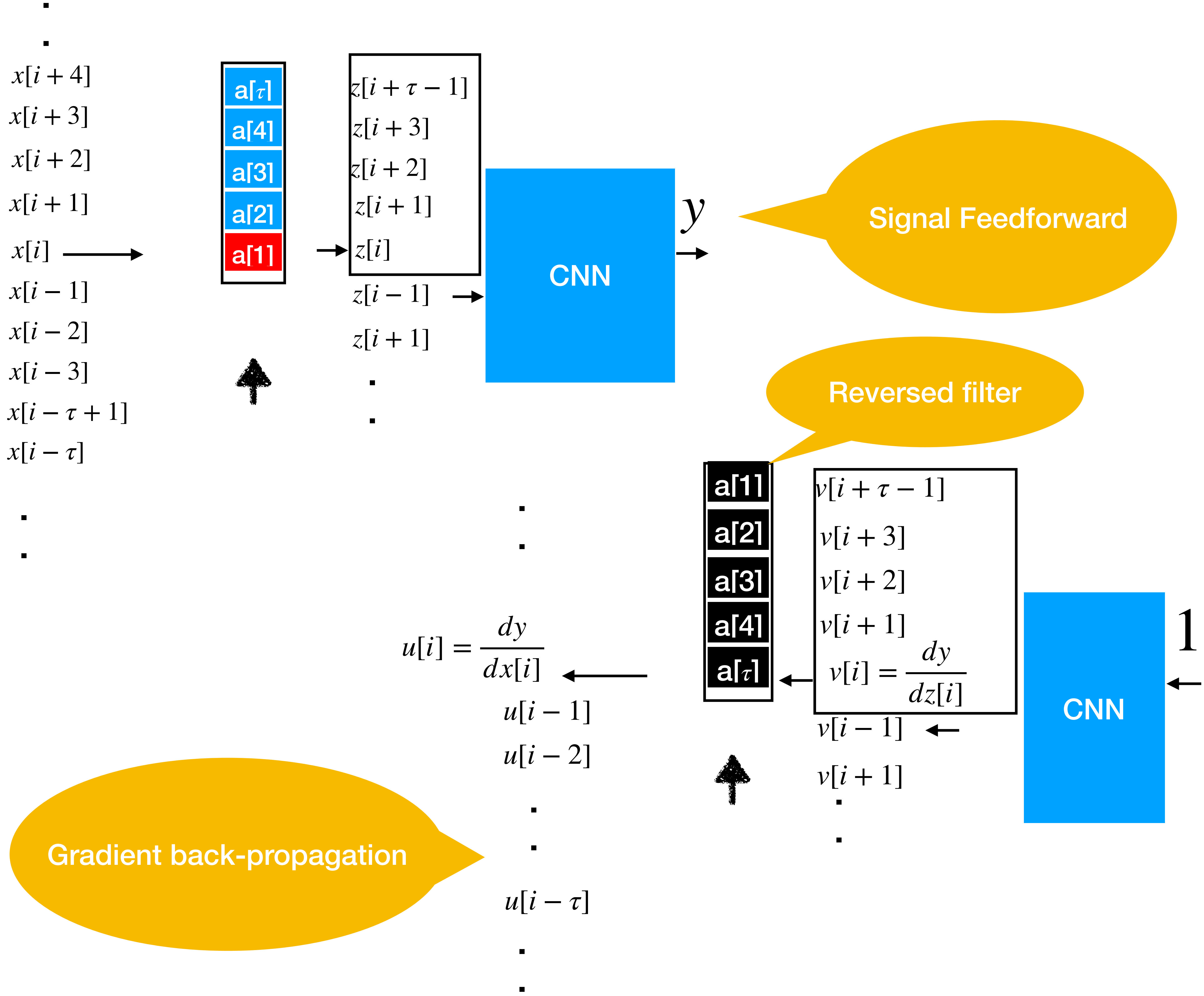
$$= \sum_{k=1}^{\tau} a[\tau-k+1] v[i+k-1]$$

Gradient back-propagation



$$u[i] = \frac{dy}{dx[i]}$$

$$= \sum_{k=1}^{\tau} a[\tau - k + 1] v[i + k - 1]$$



x[i + 4]
x[i + 3]
x[i + 2]
x[i + 1]
x[i]
x[i − 1]
x[i − 2]
x[i − 3]
x[i − τ + 1]
x[i − τ]
a[τ]
a[4]
a[3]
a[2]
a[1]
z[i + τ − 1]
z[i + 3]
z[i + 2]
z[i + 1]
z[i]
z[i − 1]
z[i + 1]
CNN
y
Signal Feedforward
Reversed filter
a[1]
a[2]
a[3]
a[4]
a[τ]
v[i + τ − 1]
v[i + 3]
v[i + 2]
v[i + 1]
v[i] = dy/dz[i]
v[i − 1]
v[i + 1]
CNN
1
u[i] = dy/dx[i]
u[i − 1]
u[i − 2]
u[i − τ]
Gradient back-propagation

2D convolution

Full Convolution

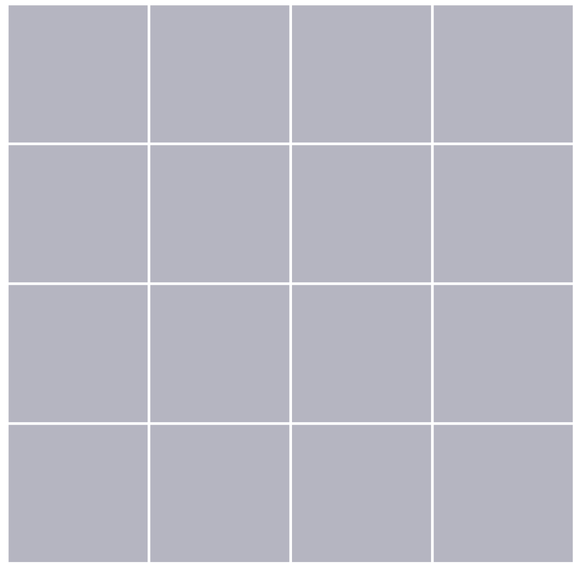
```
A = rand(3);
B = rand(4);
Cfull = conv2(A,B)
```

Cfull = *6×6*

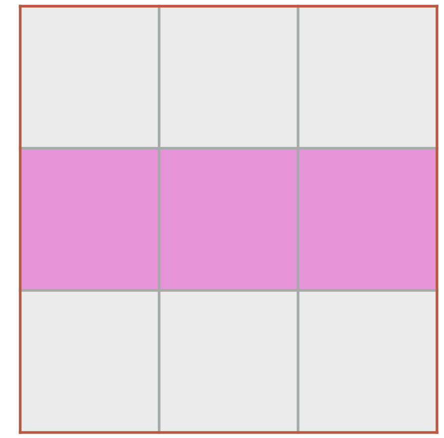
0.7861	1.2768	1.4581	1.0007	0.2876	0.0099
1.0024	1.8458	3.0844	2.5151	1.5196	0.2560
1.0561	1.9824	3.5790	3.9432	2.9708	0.7587
1.6790	2.0772	3.0052	3.7511	2.7593	1.5129
0.9902	1.1000	2.4492	1.6082	1.7976	1.2655
0.1215	0.1469	1.0409	0.5540	0.6941	0.6499

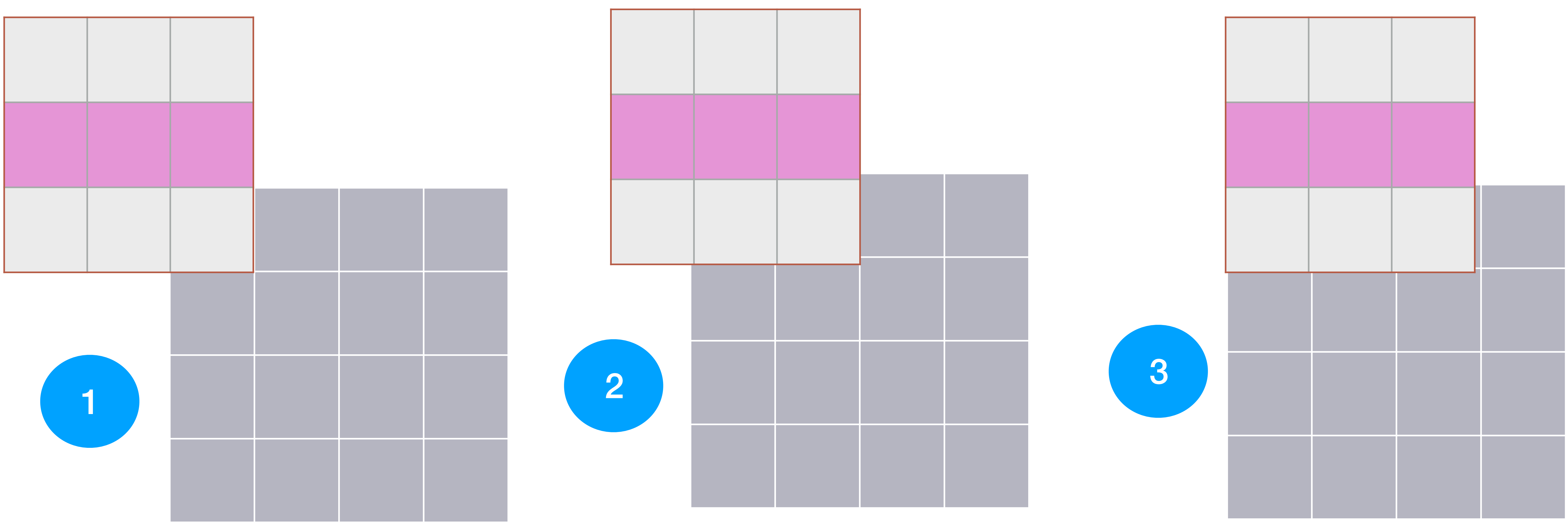
Full Convolution

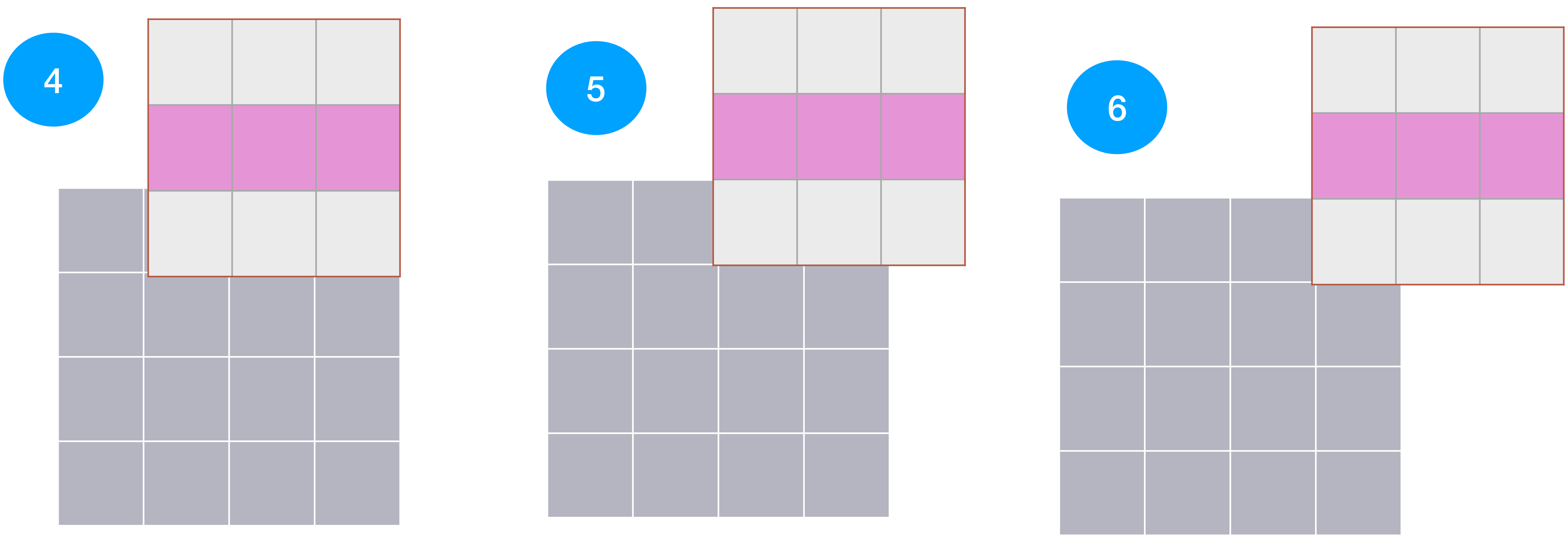
B



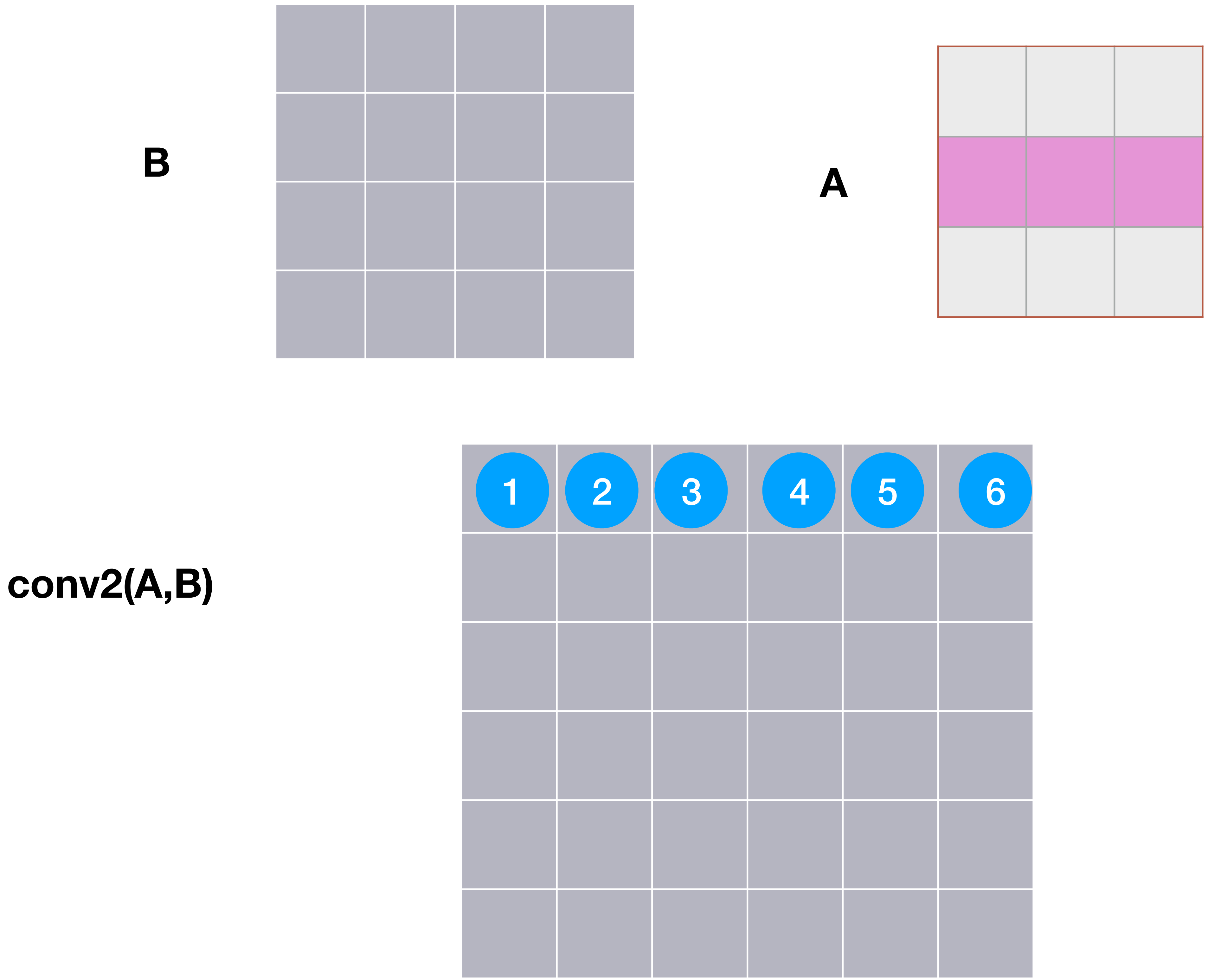
A



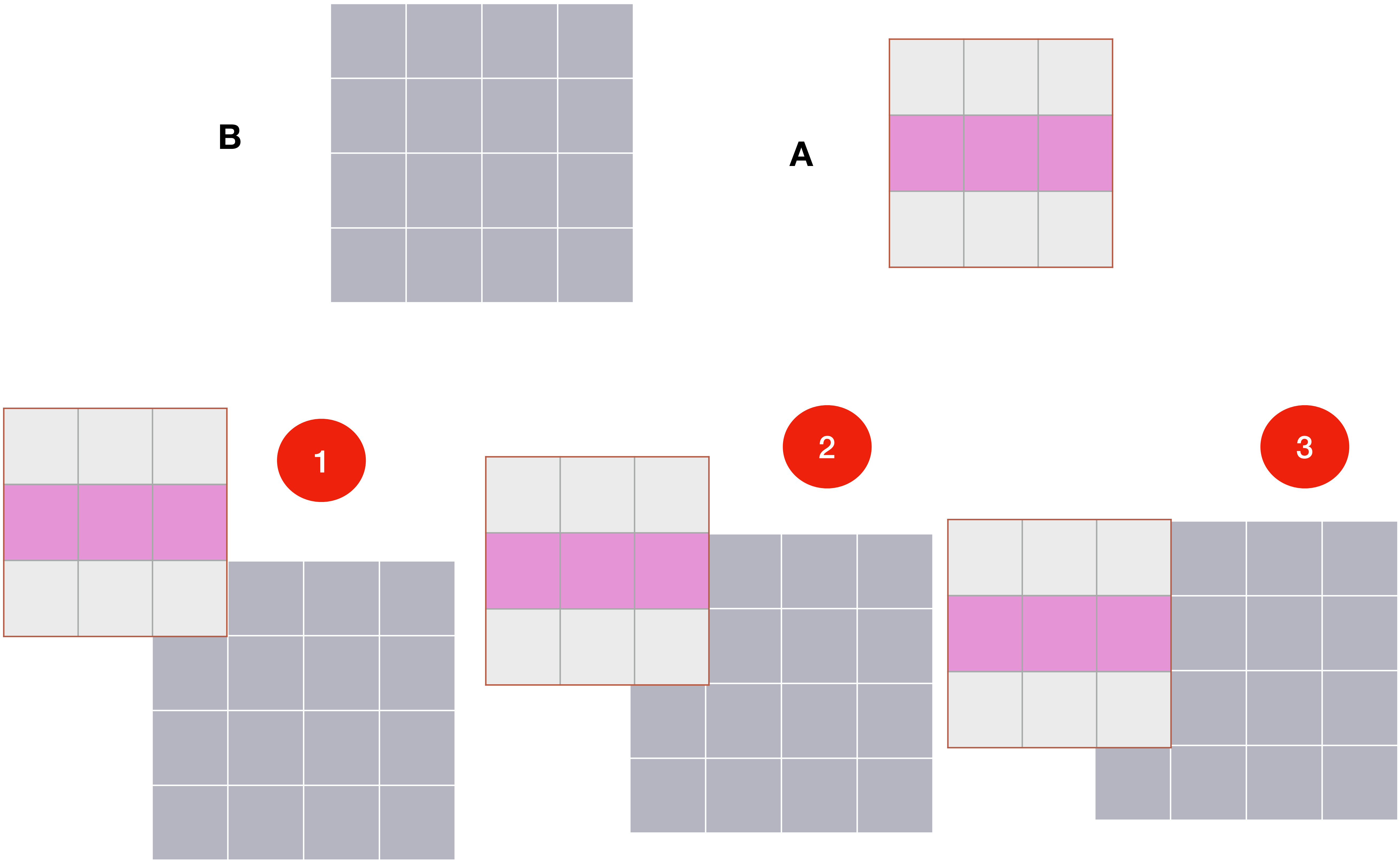



4
5
6

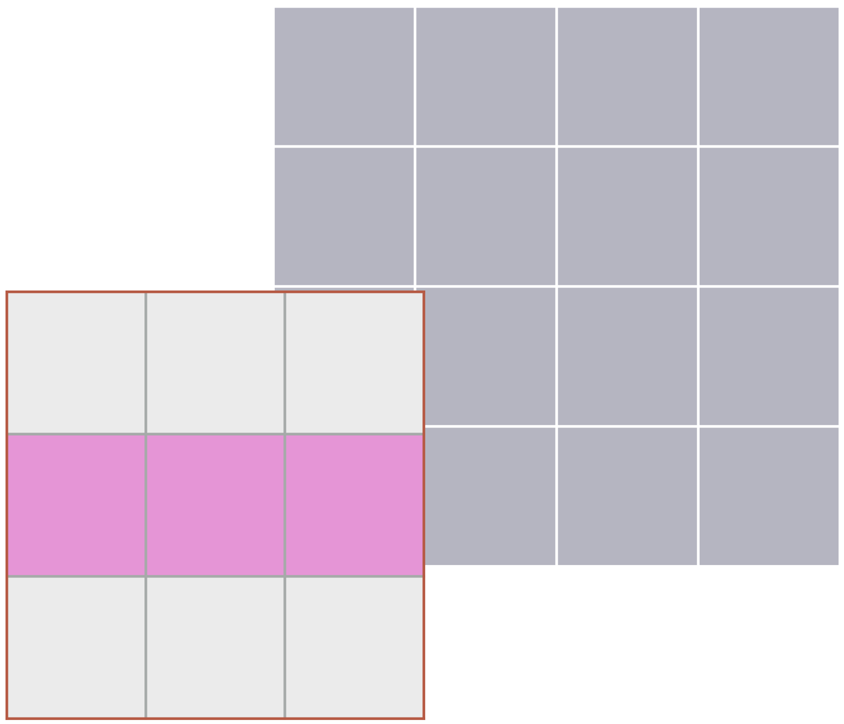
Full Convolution



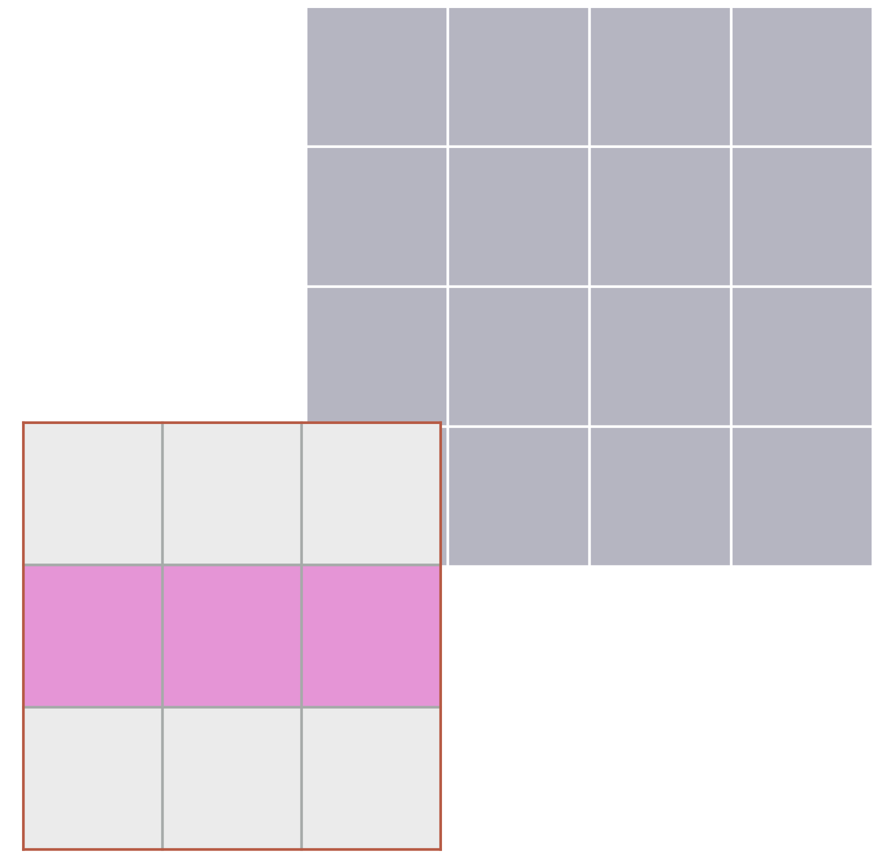
Full Convolution



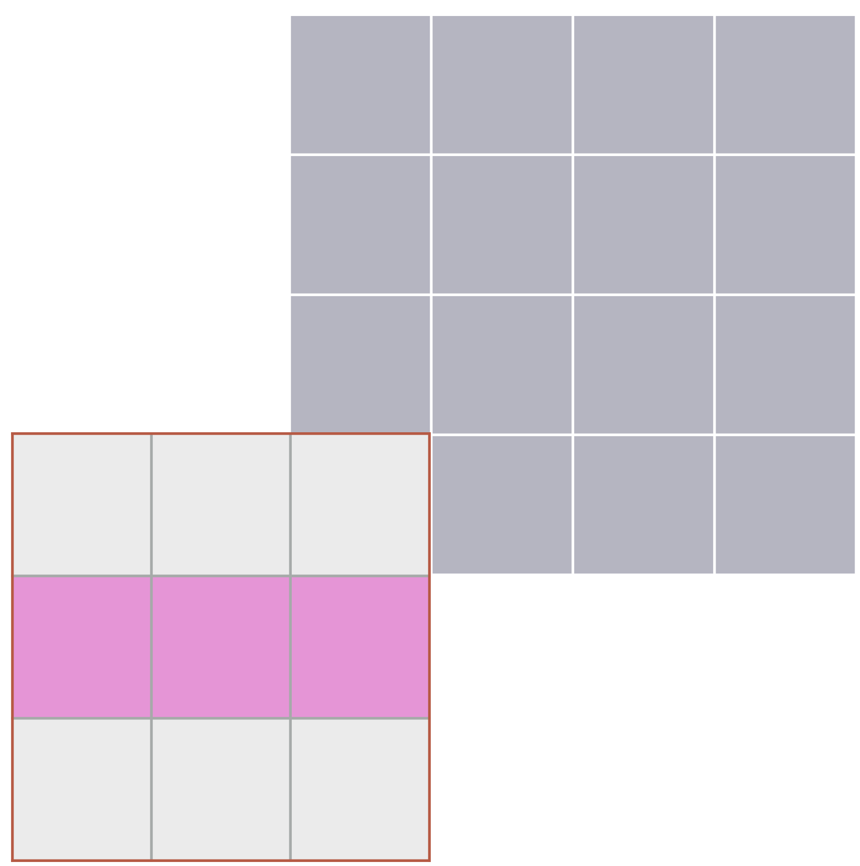
4



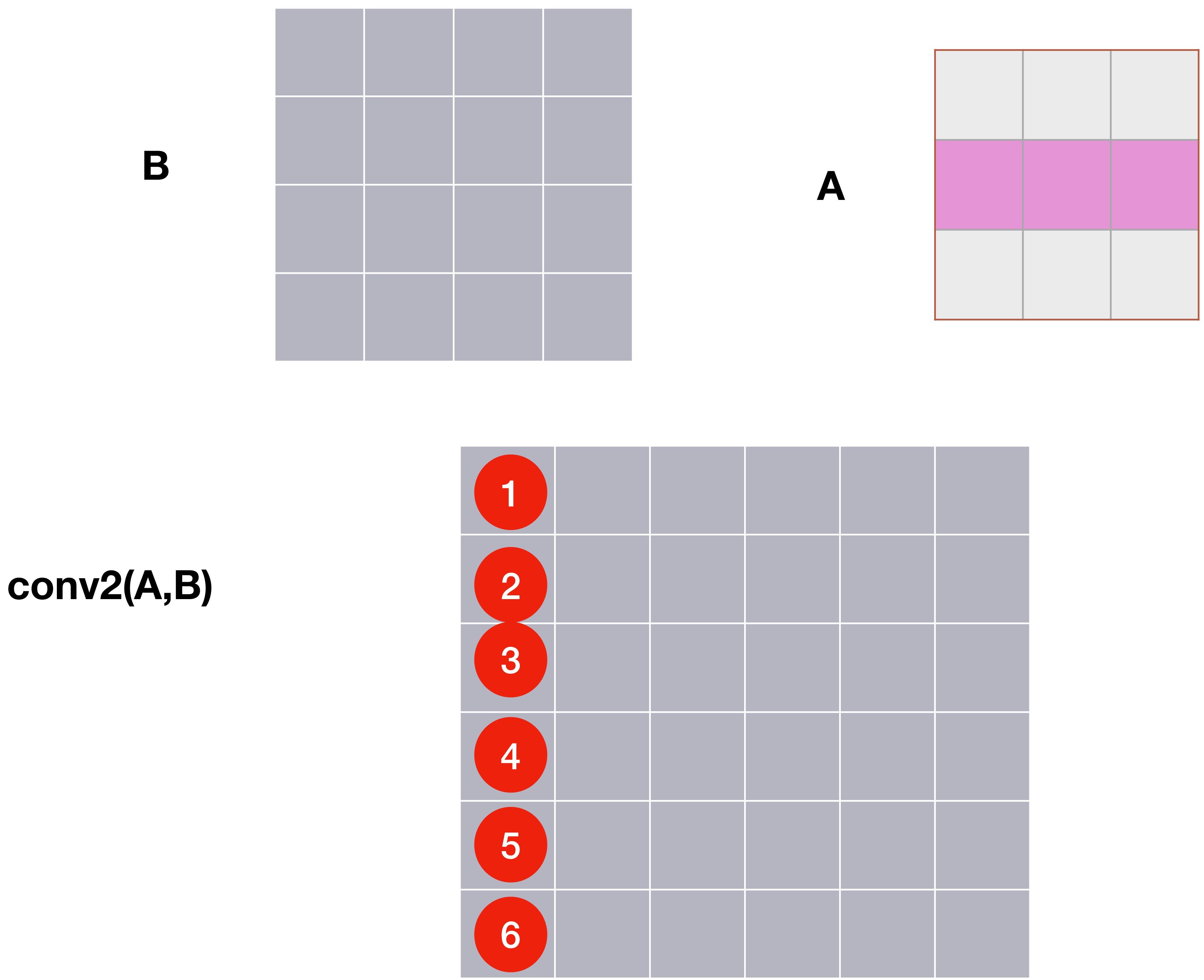
5



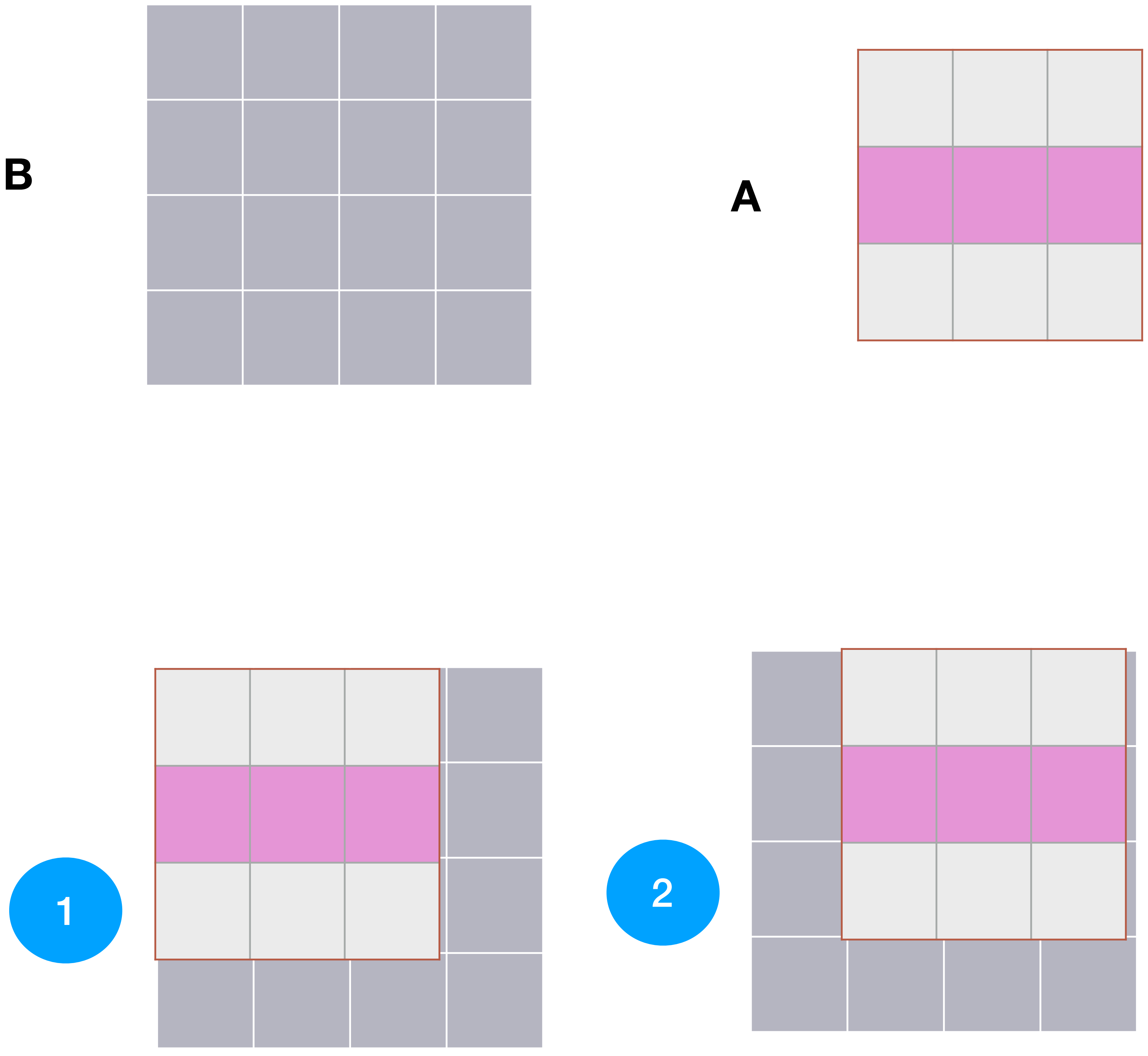
6



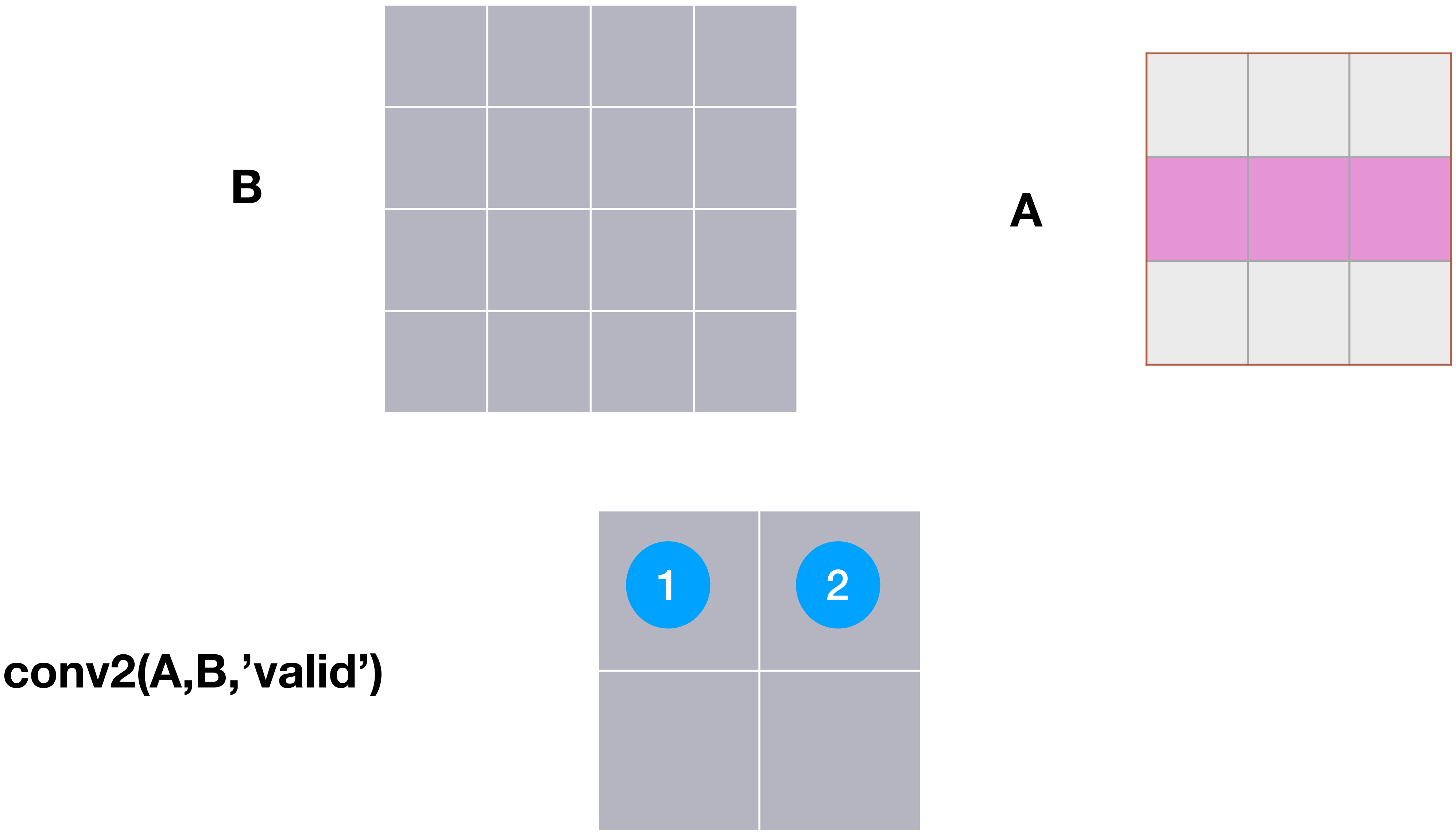
Full Convolution



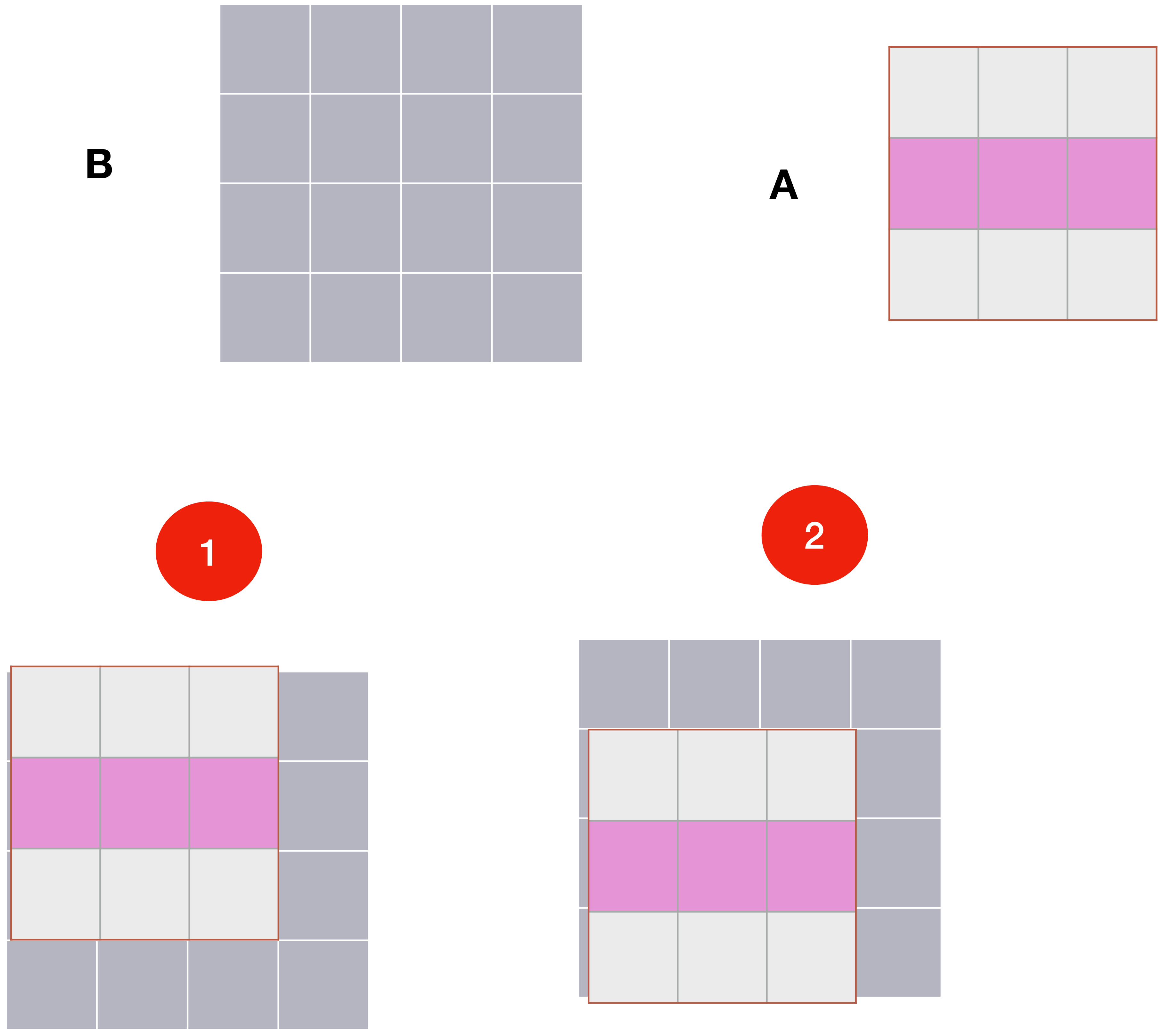
Valid Convolution



Valid Convolution

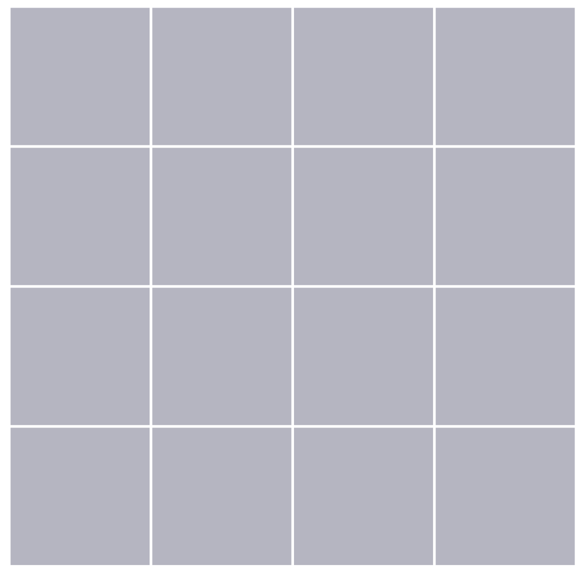


Valid Convolution

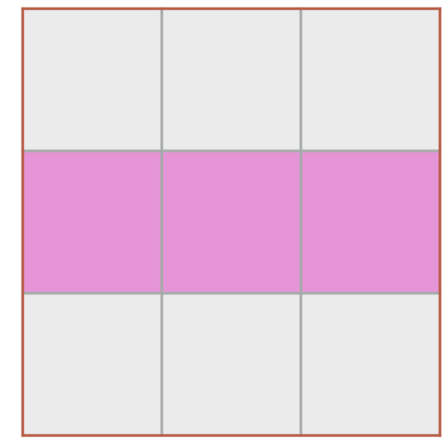


Valid Convolution

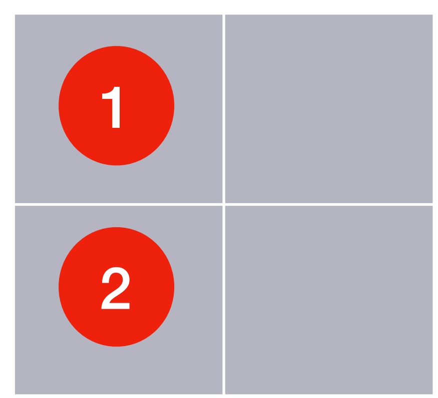
B



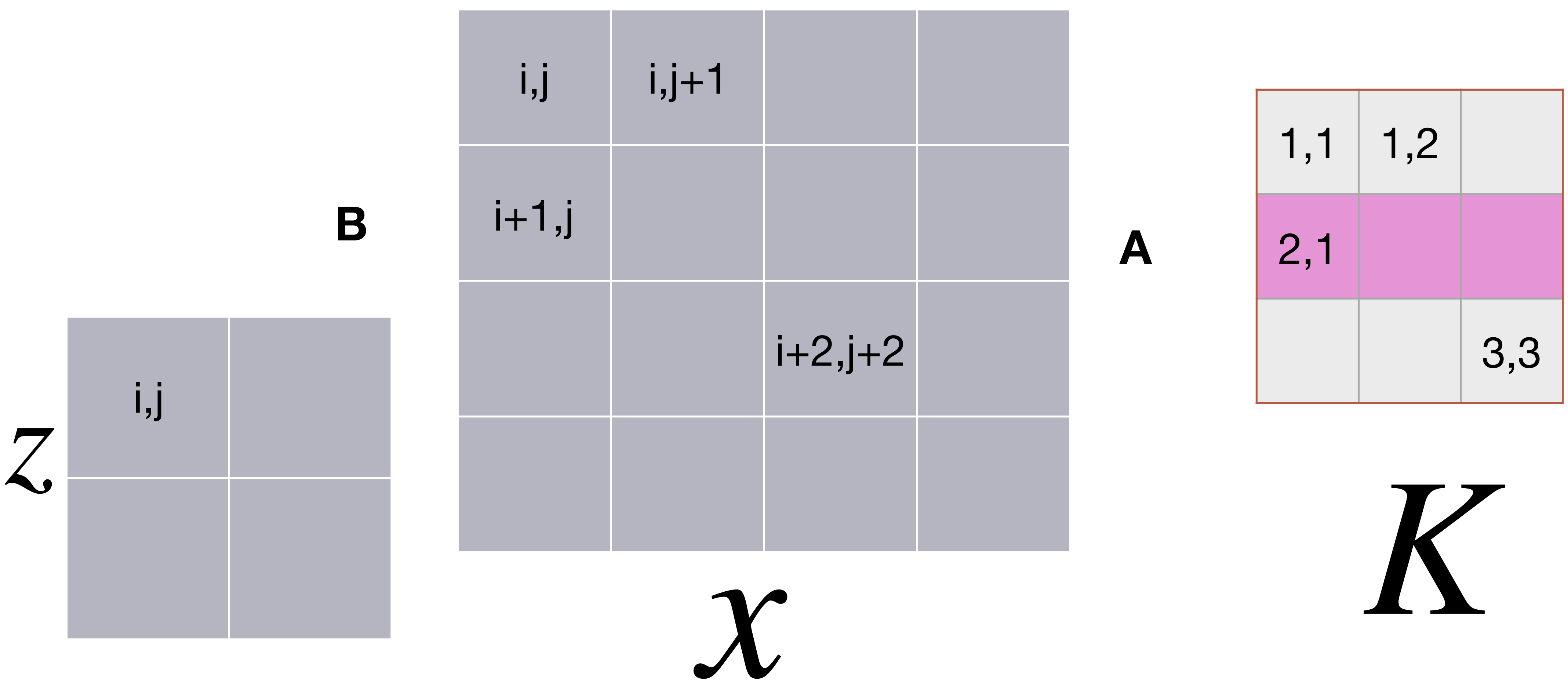
A



conv2(A,B,'valid')



Valid Convolution



conv2(A,B,'valid')

$$size(B,1) \geq size(A,1); \quad size(B,2) \geq size(A,2)$$

$$z[i,j] = \sum_{m}^{\tau} \sum_{n}^{\tau} K(m,n) * x[i+m-1, j+n-1]$$

where $i = 1 : size(B,1) - size(A,1) + 1$

$j = 1 : size(B,2) - size(A,2) + 1$

Valid convolution

表格 1

B	=			
	1	1	1	1
	1	1	1	1
	1	1	1	1
	1	1	1	1

```
>> B=ones(4,4);A=ones(3,3)/9;
>> z=conv2(B,A,'valid')

z =

    1.0000    1.0000
    1.0000    1.0000
```

表格 1-1

A =			
	0.1111	0.1111	0.1111
	0.1111	0.1111	0.1111
	0.1111	0.1111	0.1111

```
function z=my_valid_conv2(x,K)

assert(size(x,1) >= size(K,1) & size(x,2) >= size(K,2),"invalid matrix size")
fo

er
```

```
>> B=ones(4,4);A=ones(3,3)/9;
>> z=my_valid_conv2(B,A)

z =

     1     1
     1     1
```

表格 1

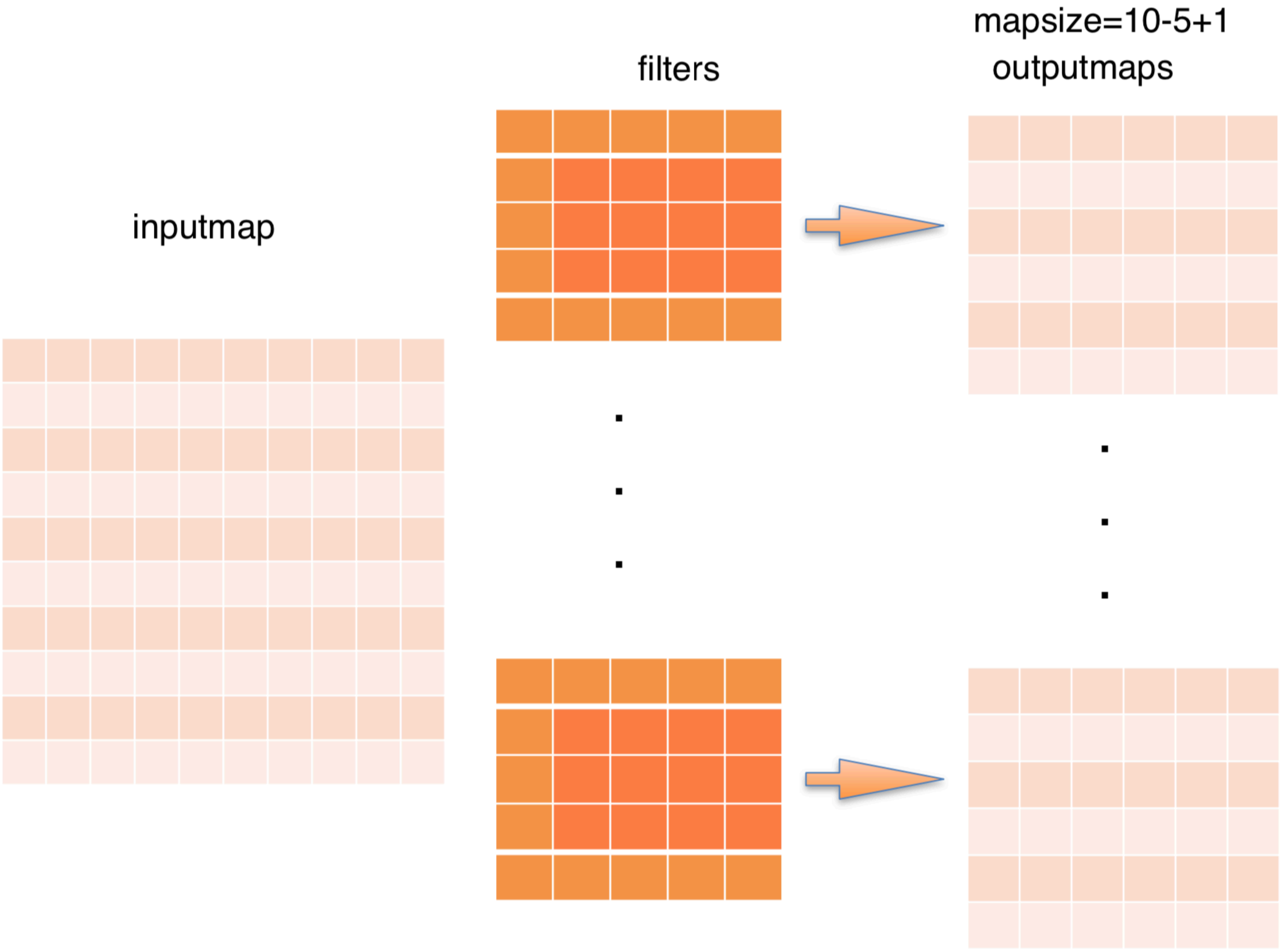
A =		
0.2348	0.0154	0.6491
0.3532	0.0430	0.7317
0.8212	0.1690	0.6477

表格 1-1

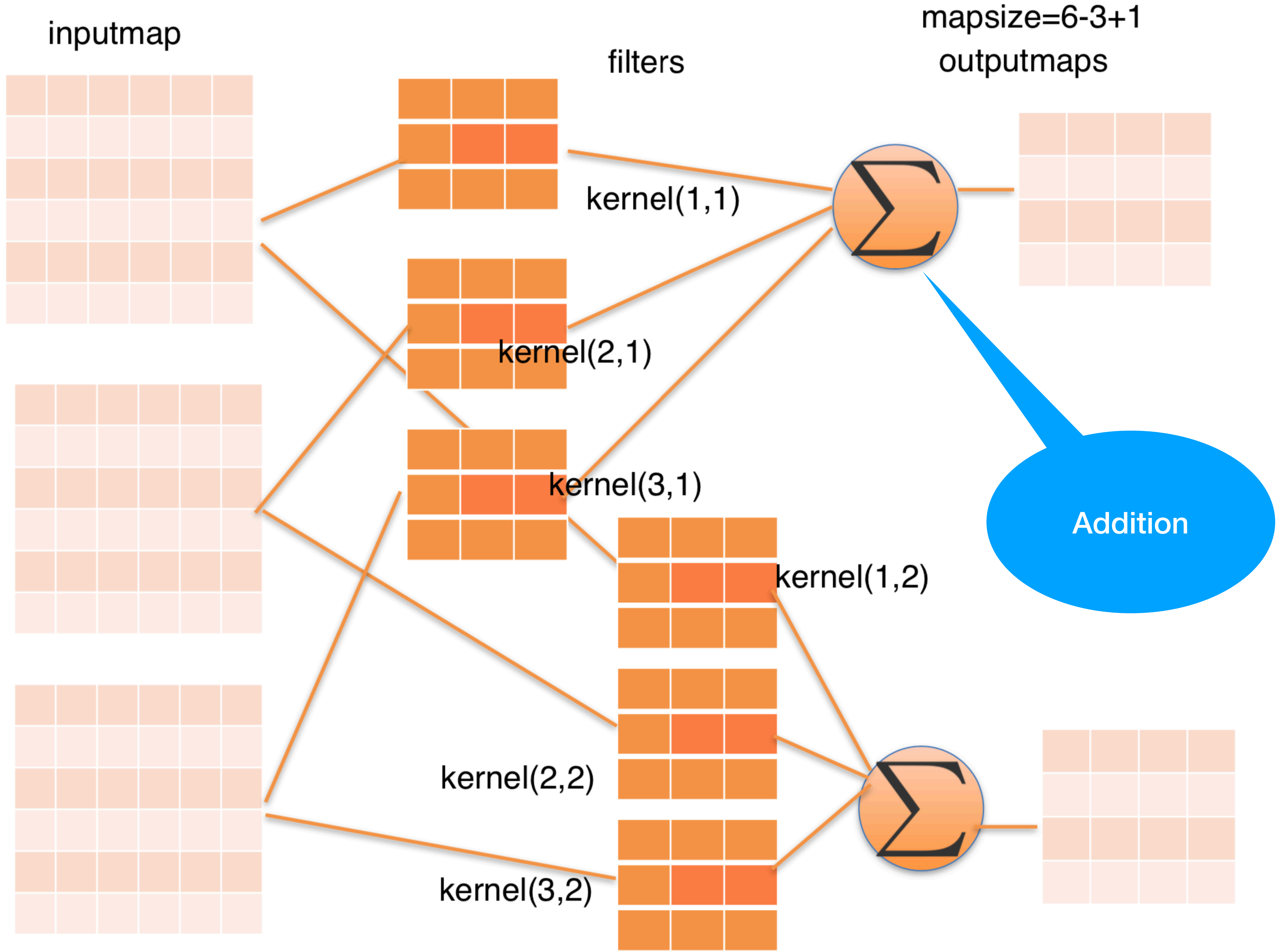
B =			
0.4509	0.1890	0.6256	0.7757
0.5470	0.6868	0.7802	0.4868
0.2963	0.1835	0.0811	0.4359
0.7447	0.3685	0.9294	0.4468

```
>> sum(sum(B(2:4,1:3).*A))

ans =

    2.0932
```

```
>> z2=my_valid_conv2(B,A)

z2 =

    1.6354    1.6366
    2.0932    1.6255

>> sum(sum(B(1:3,1:3).*A))

ans =

    1.6354

>> sum(sum(B(1:3,2:4).*A))

ans =

    1.6366

>> sum(sum(B(2:4,2:4).*A))

ans =

    1.6255
```



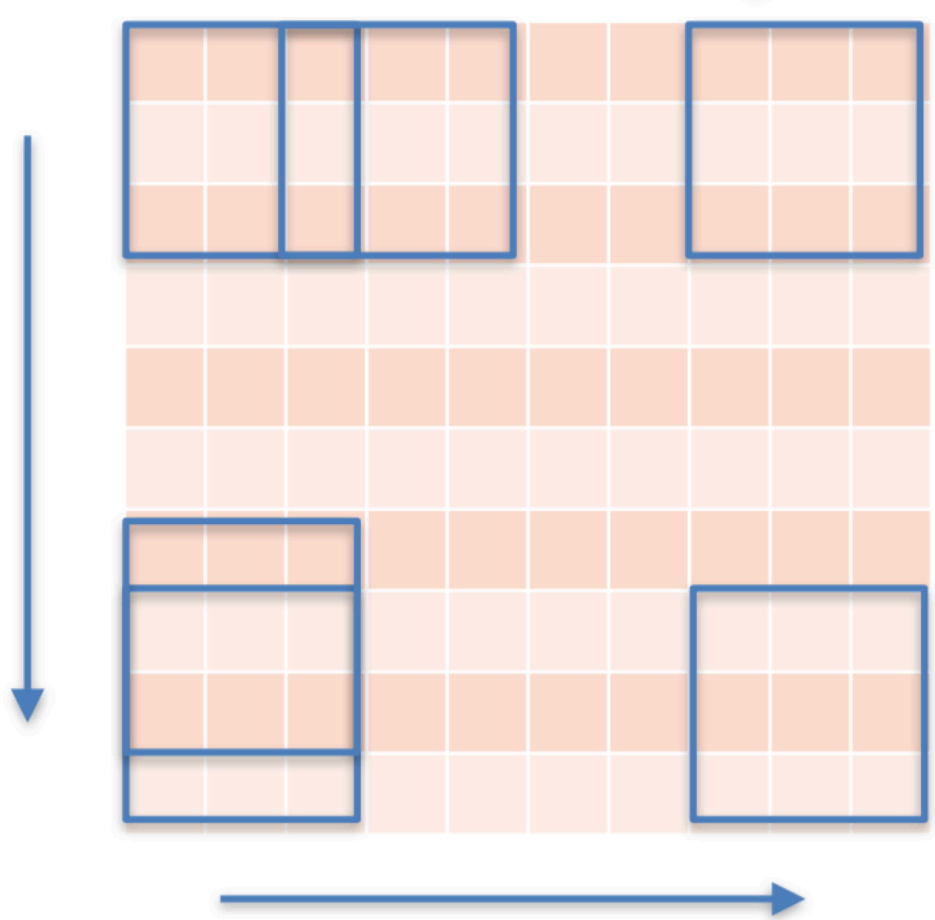
A stack of output maps

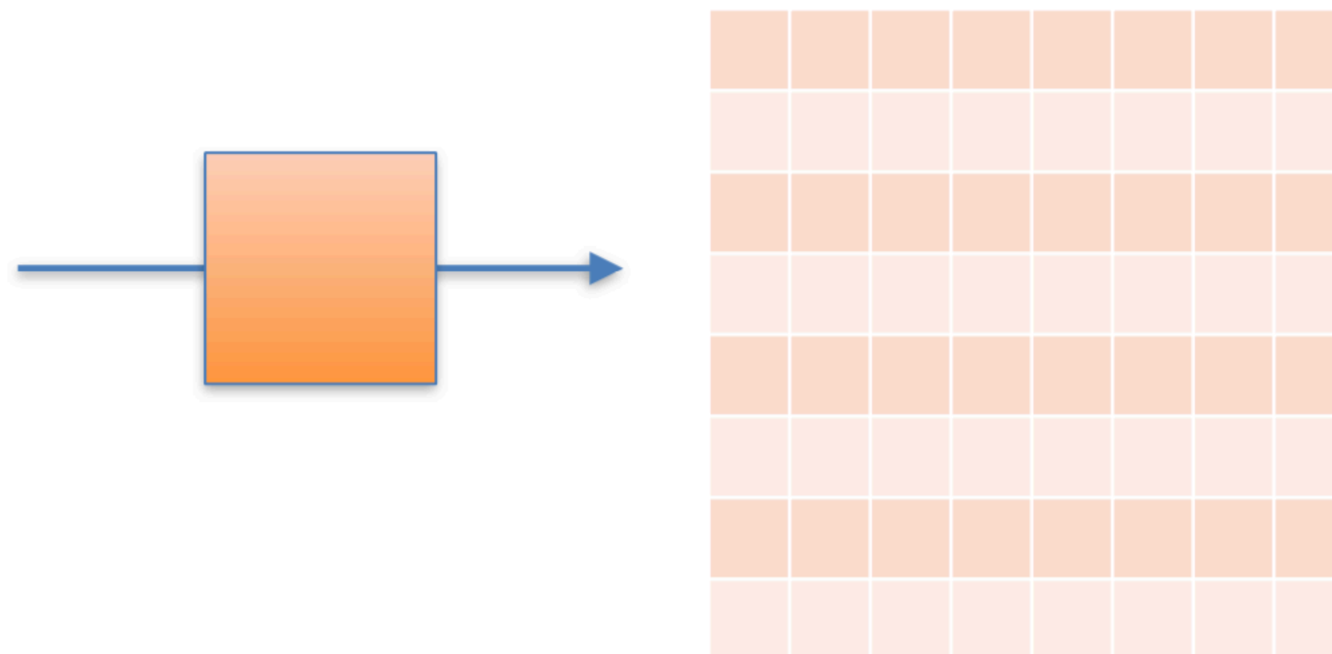

inputmap
filters
mapsize=6-3+1
outputmaps
kernel(1,1)
kernel(2,1)
kernel(3,1)
kernel(1,2)
kernel(2,2)
kernel(3,2)
Addition

2D Convolution

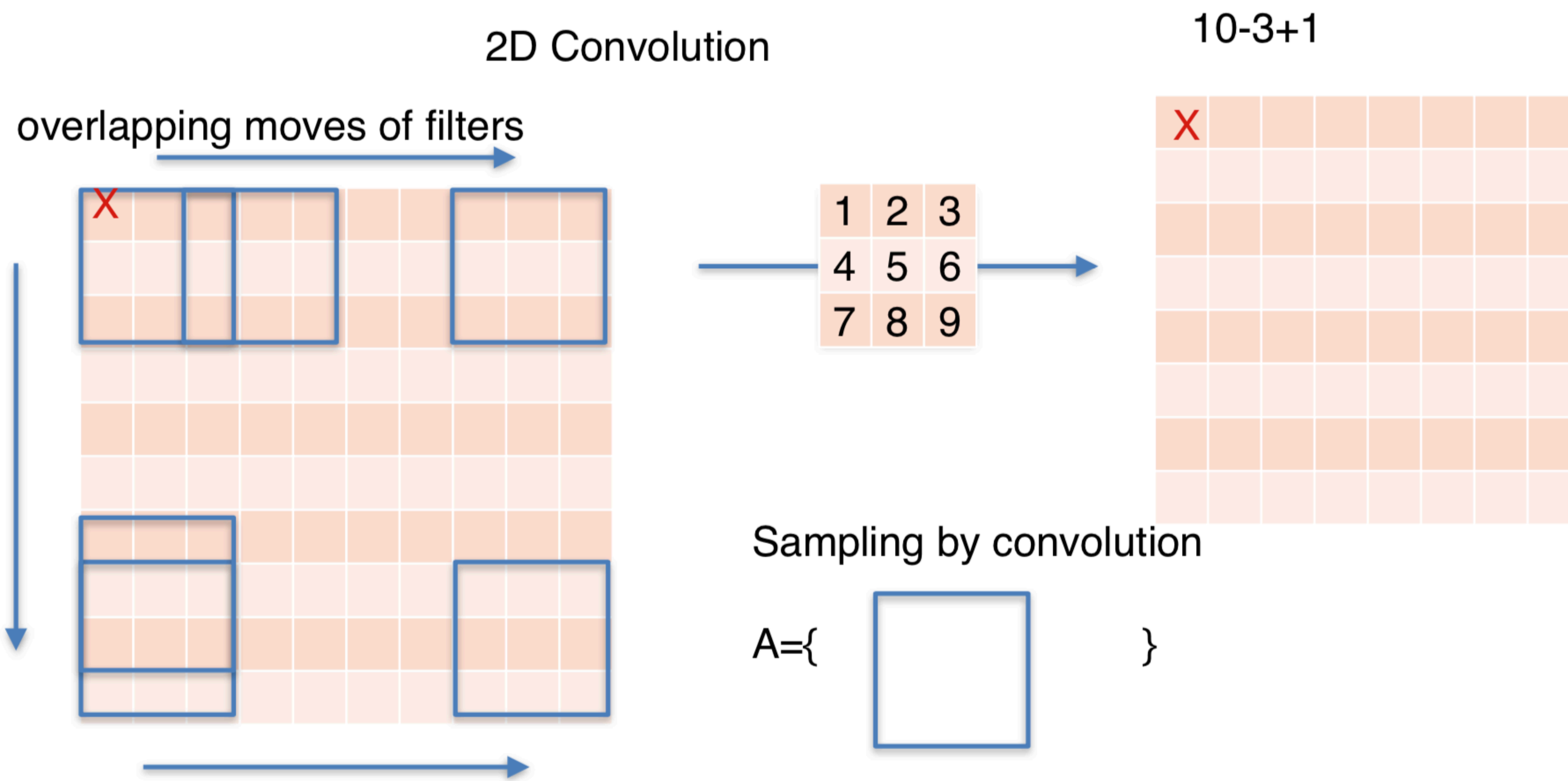
10-3+1

overlapping moving of filters





```
>> C=conv2(B,A,'valid')

C =

    1.0000    1.0000
    1.0000    1.0000

>> B

B =

     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1

>> A

A =

    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111

>>
```

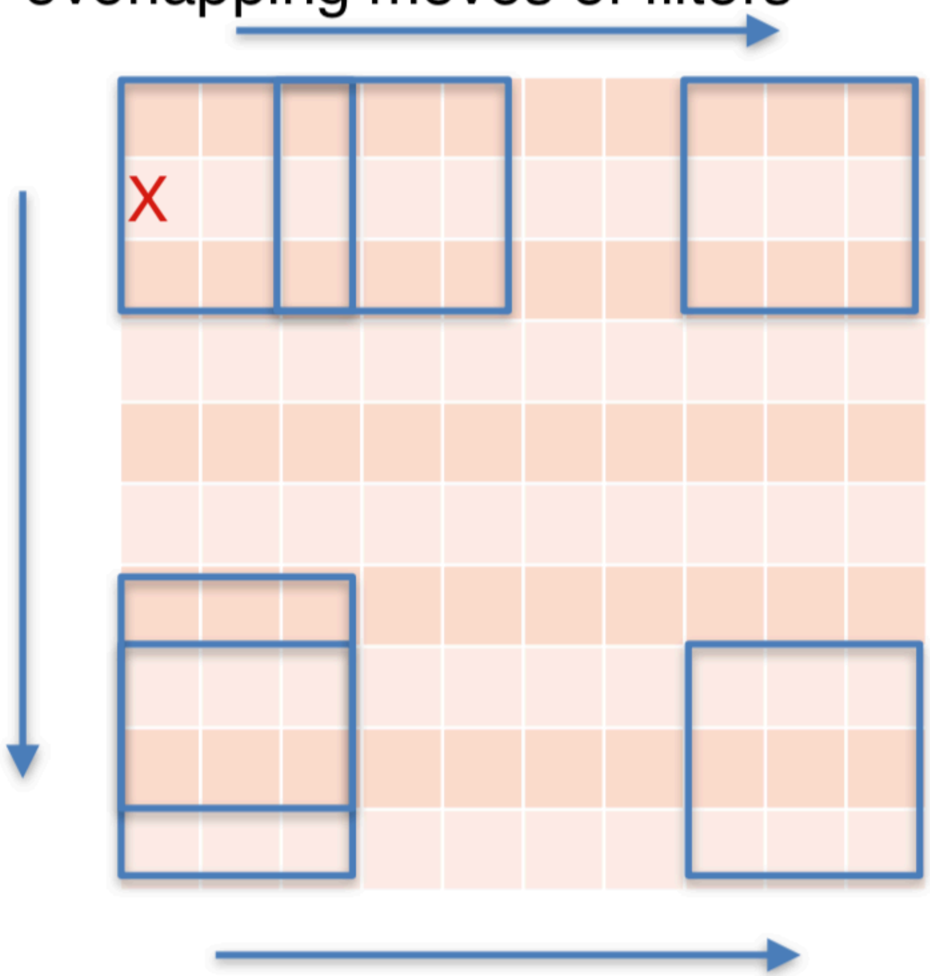


Position x on the activation map contributes to only position X of the resulting map through k(1)

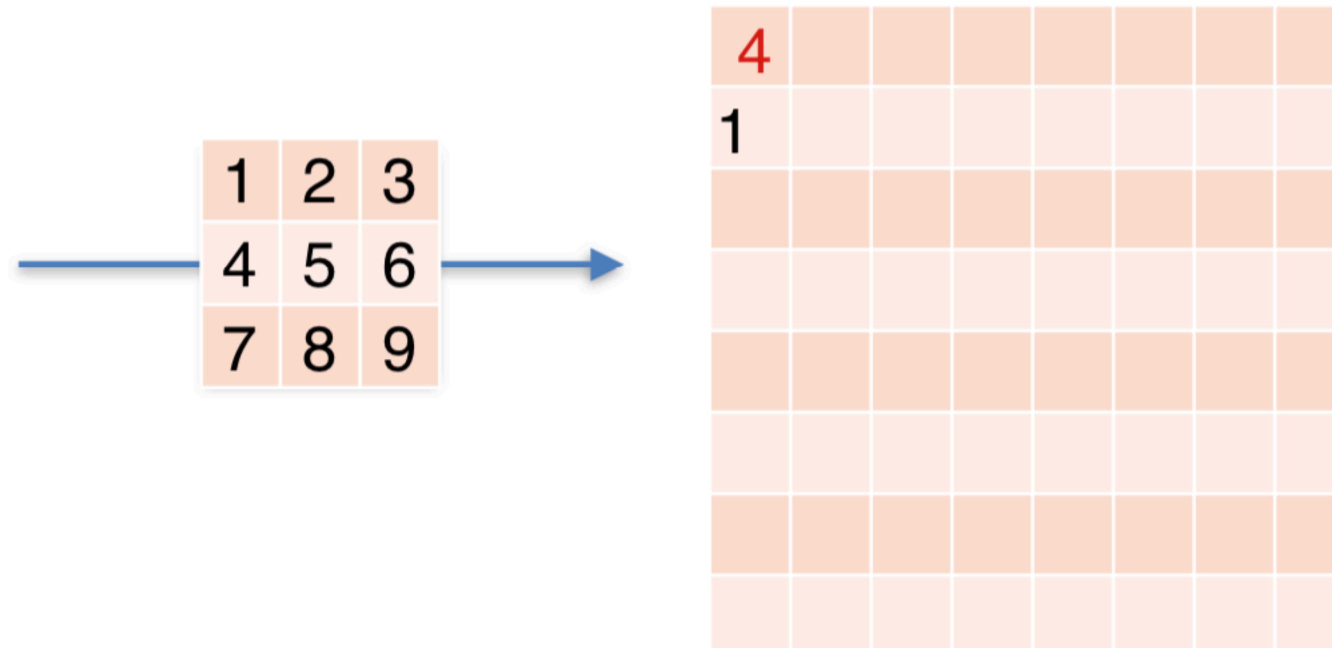
$$z[i,j] = \sum_{m}^{\tau} \sum_{n}^{\tau} K(m,n) * x[i+m-1, j+n-1]$$

2D Convolution

overlapping moves of filters



10-3+1



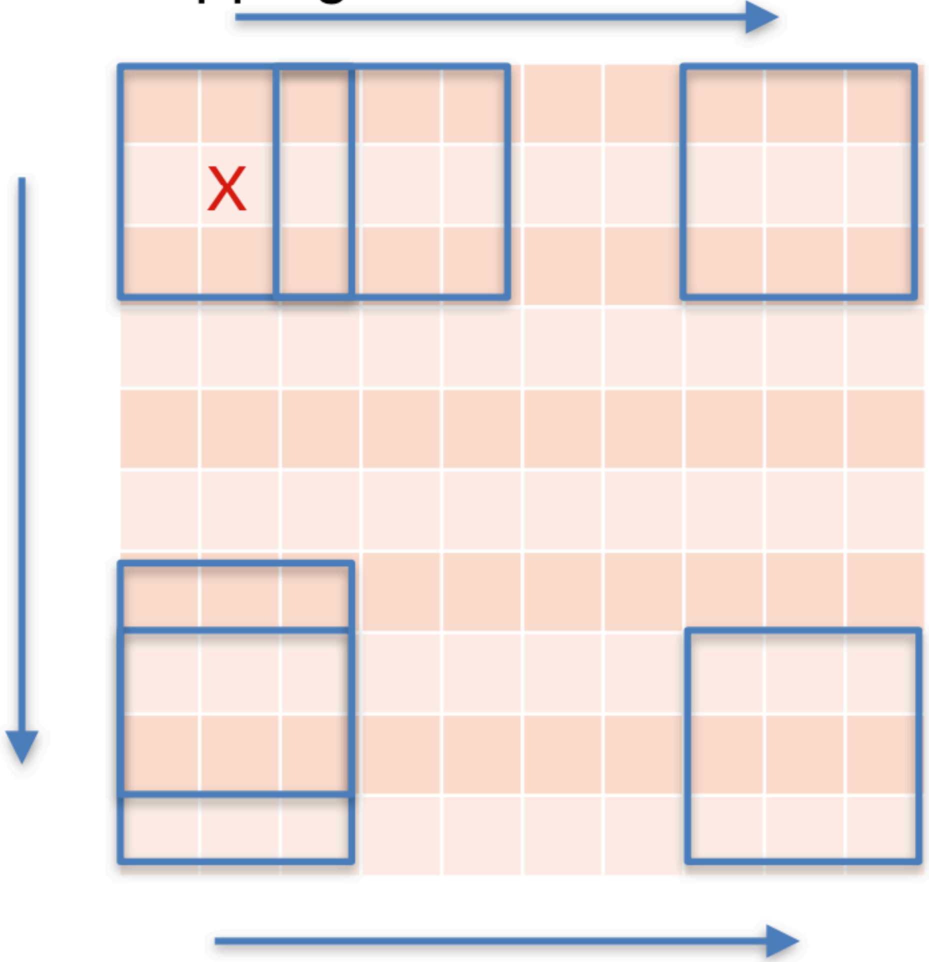
Sampling by convolution

A={  }

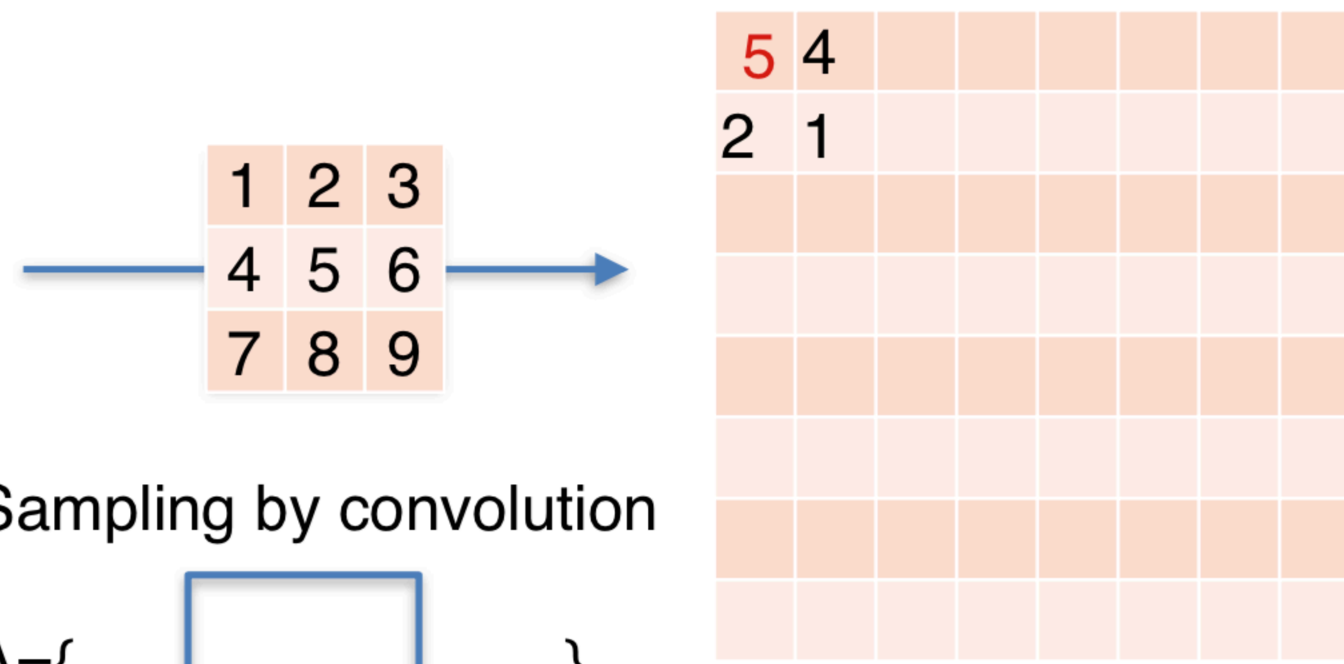
Position x on the activation map contributes to only positions with label 4 and 1 through k(4) and k(1) respectively

2D Convolution

overlapping moves of filters

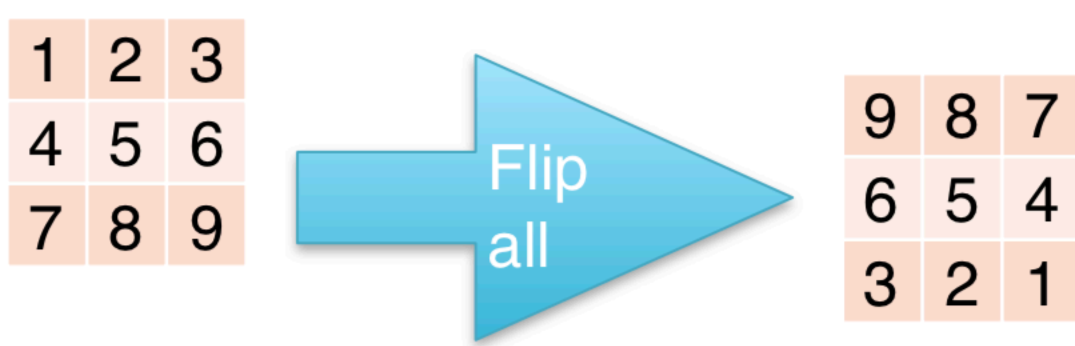


10-3+1



Sampling by convolution

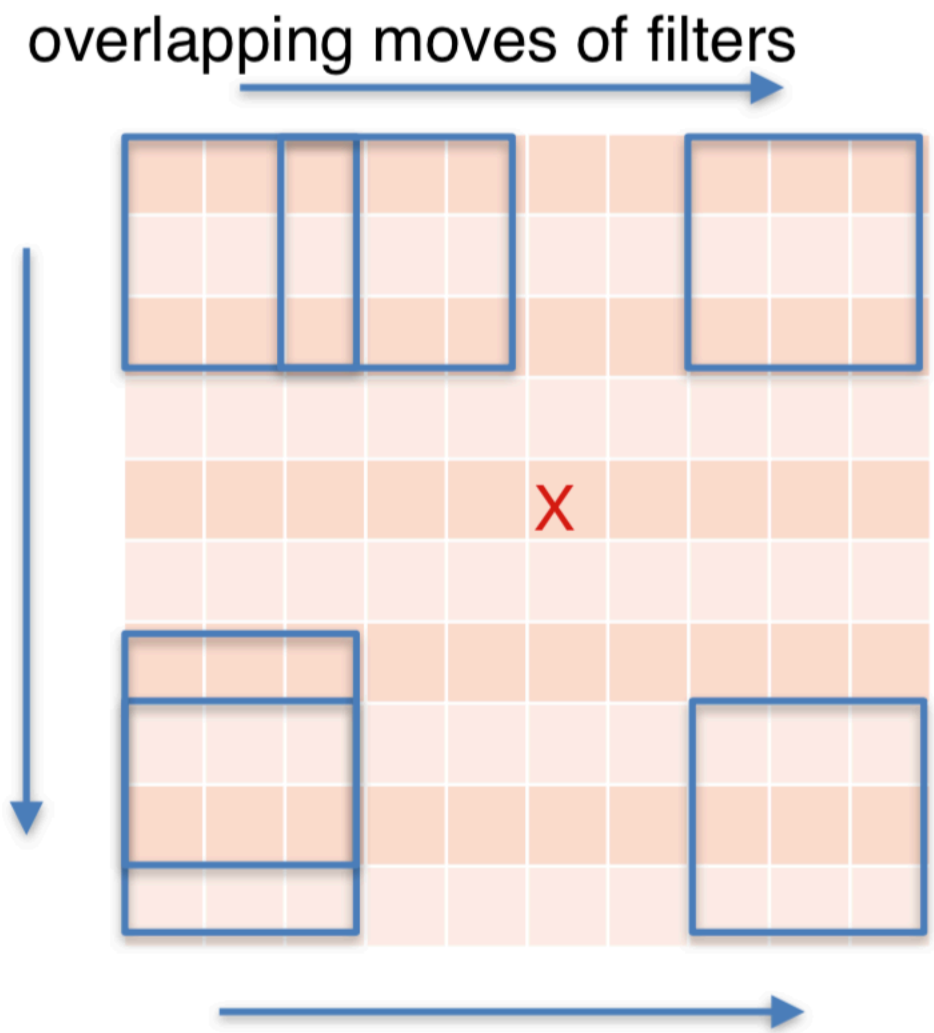
A={  }

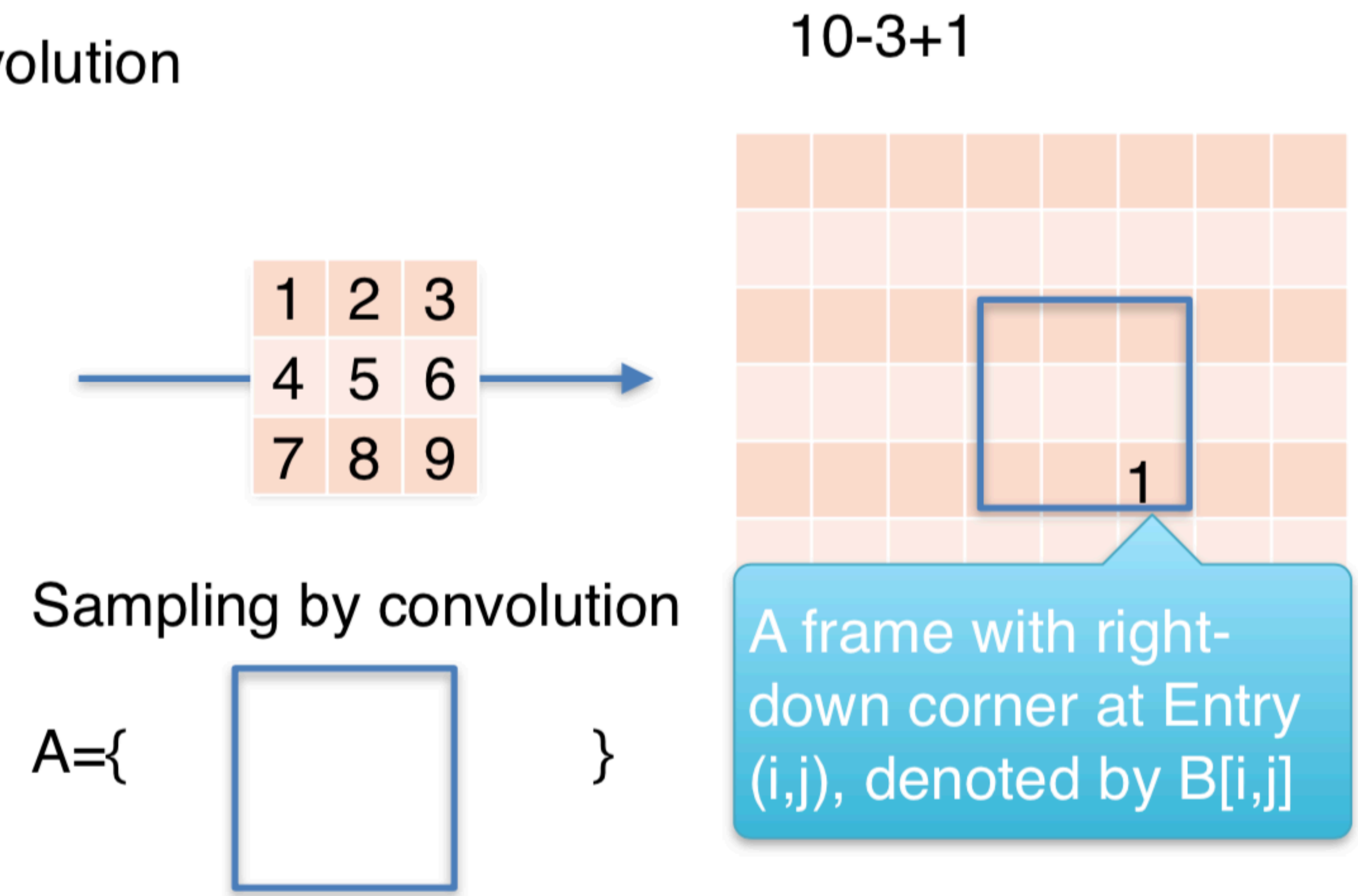


.There are four validly convoluted samples that contain position X

.Position x on the activation map contributes to only positions labeled 1,2,4 and 5 through k(5 4 / 2 1)

2D Convolution



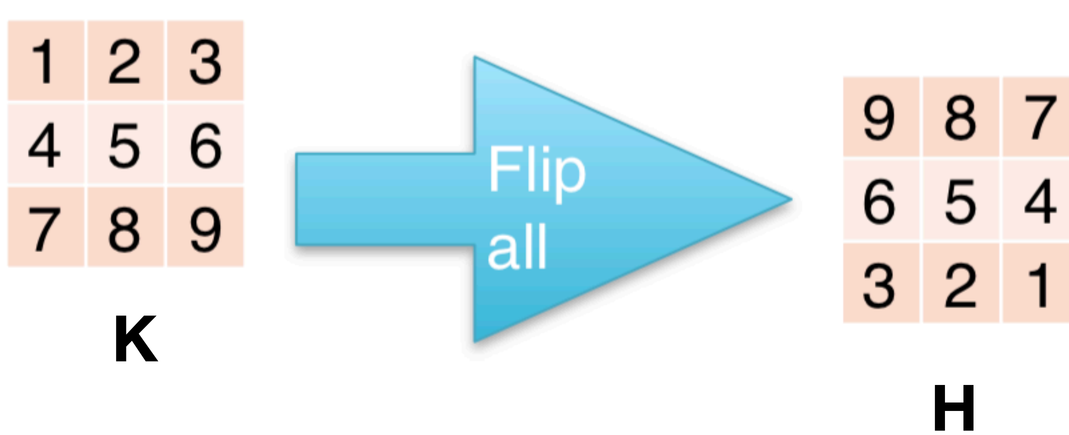


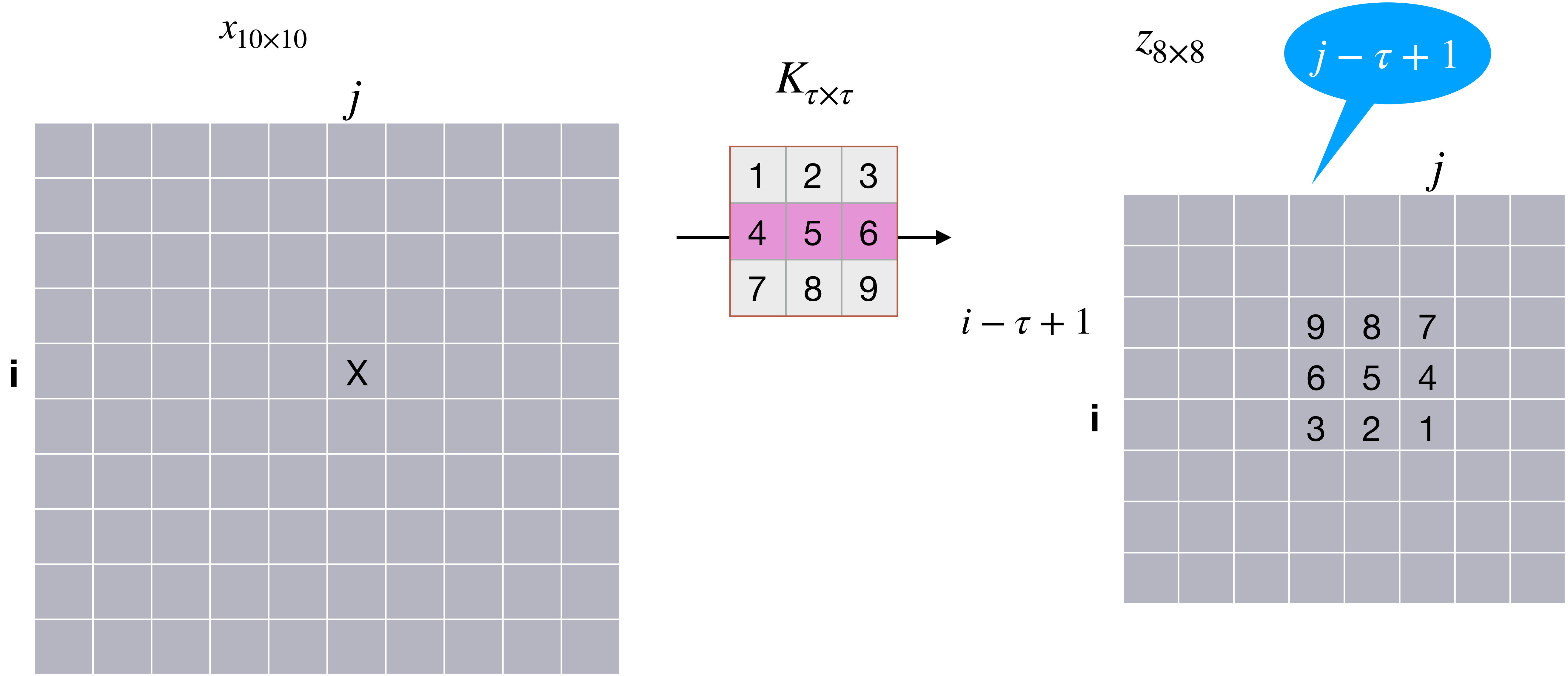
.There are 9 validly convoluted samples that contain position X at entry (i,j)

.Position x on the activation map contributes to positions labeled with 1-9 through k(

9	8	7
6	5	4
3	2	1

)

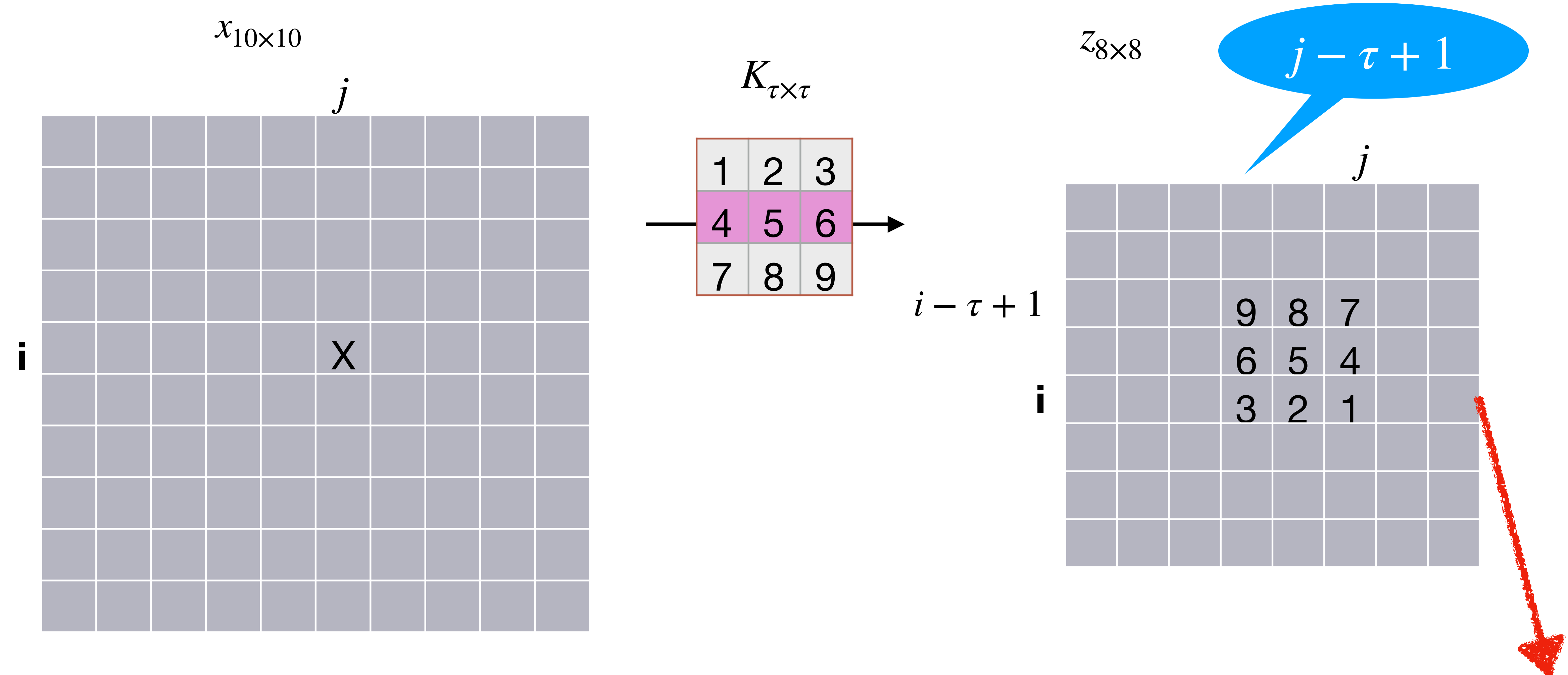


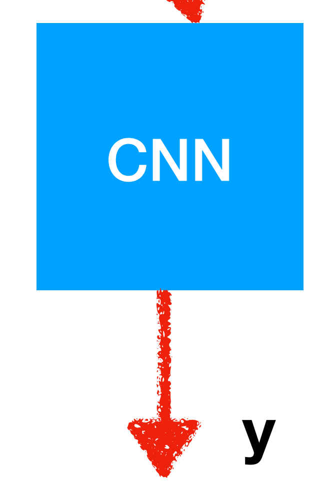


$x[i, j]$

contributes to

$$
\begin{array}{lll}
z[i-\tau+1, j-\tau+1] & \cdots & z[i-\tau+1, j] \\
\vdots & & z[i-1, j] \\
z[i, j-\tau+1] & z[i, j-1] & z[i, j]
\end{array}
$$





$x[i, j]$

contributes to

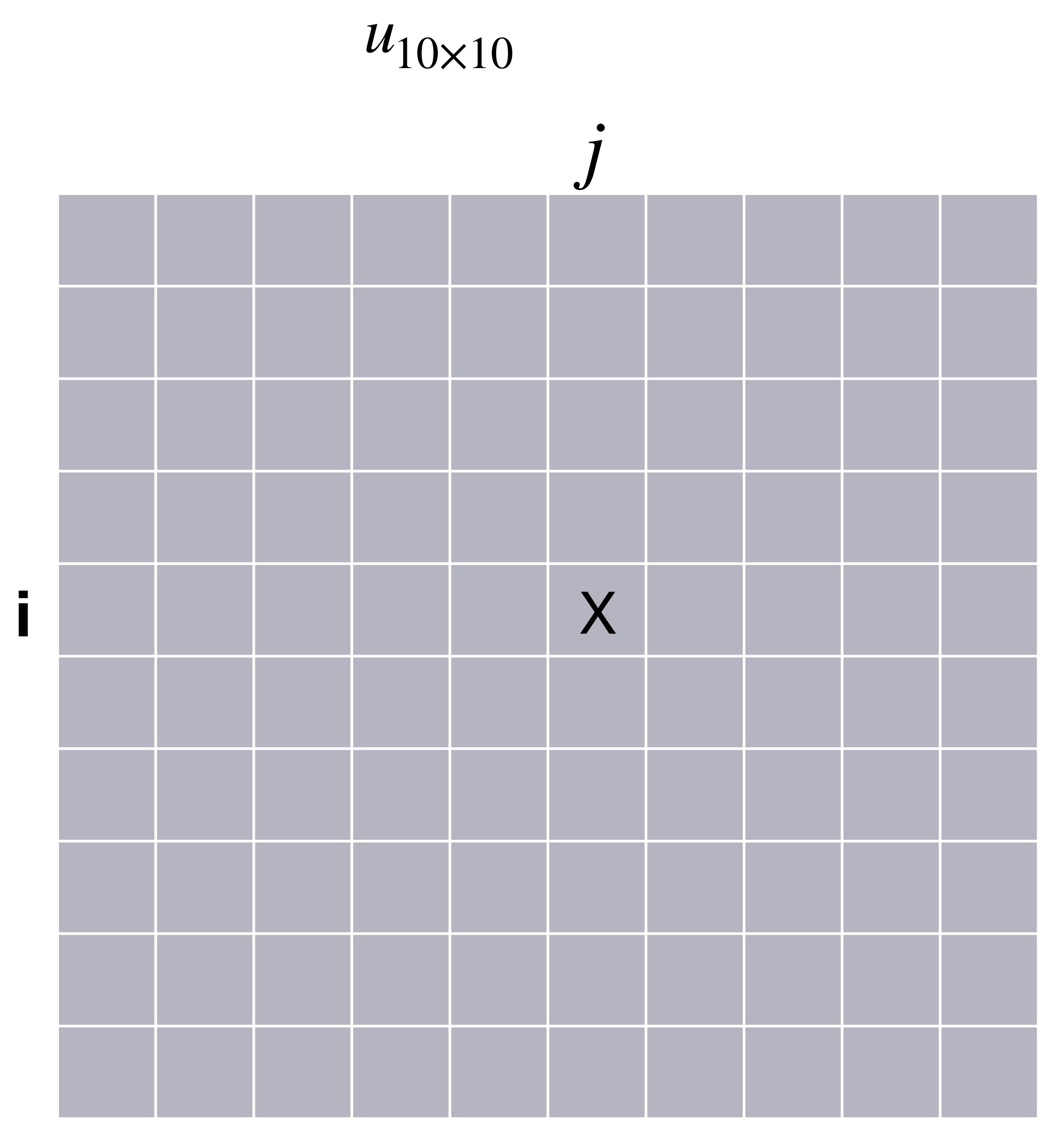
$z[i-\tau+1, j-\tau+1]$ ⋯ $z[i-\tau+1, j]$

⋮

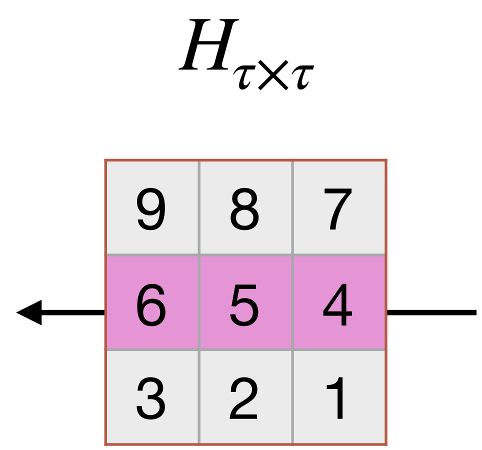
$z[i-1, j]$

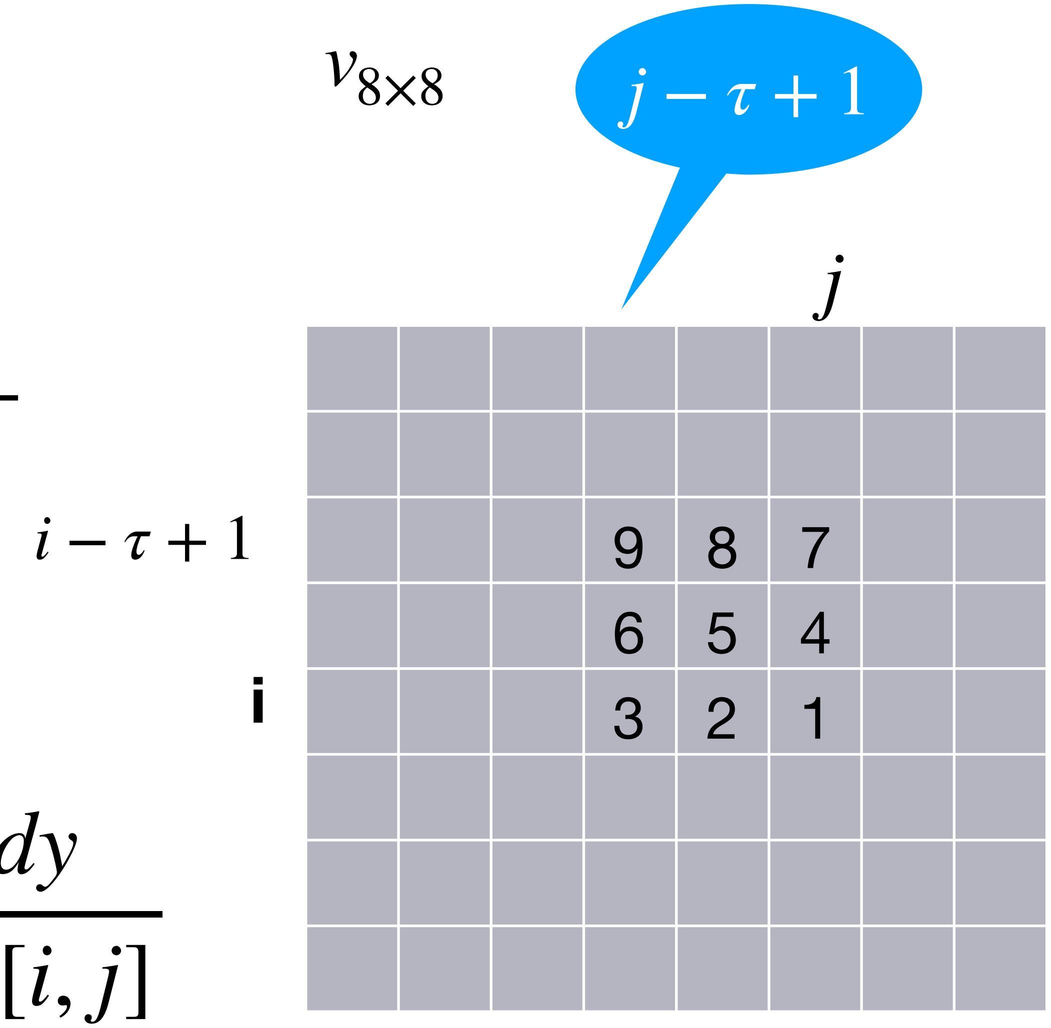
$z[i, j-\tau+1]$ $z[i, j-1]$ $z[i, j]$

Given $v[i,j] = \dfrac{dy}{dz[i,j]}$ $u[i,j] = \dfrac{dy}{dx[i,j]}$ **=?**



$x[i, j]$





$$v[i, j] = \frac{dy}{dz[i, j]}$$

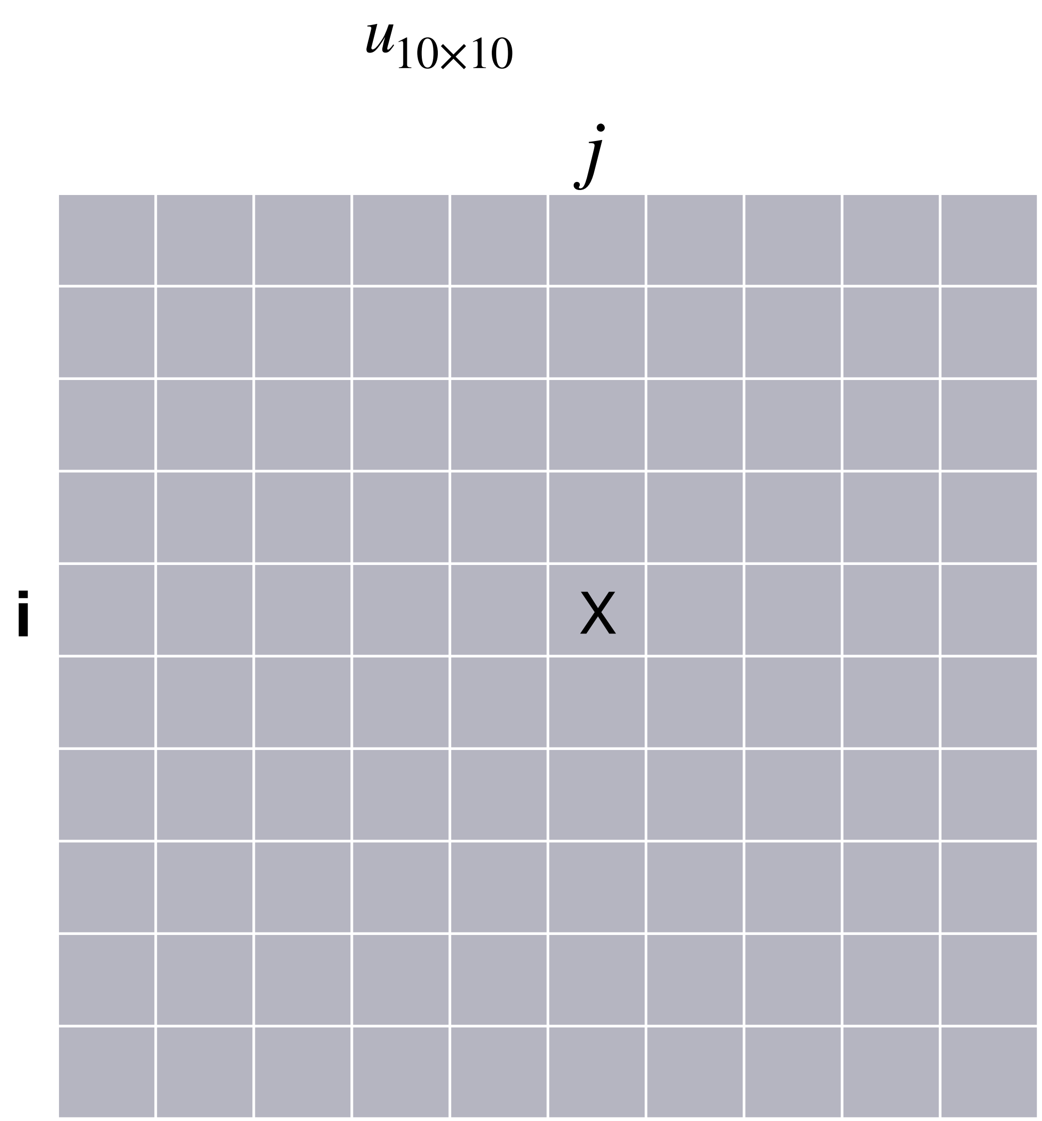
contributes to

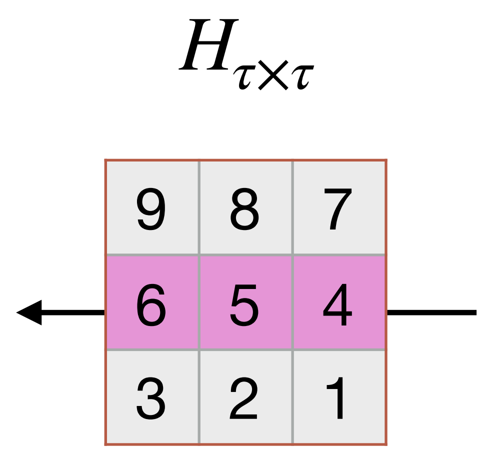
$z[i-\tau+1, j-\tau+1]$ $\cdots$ $z[i-\tau+1, j]$

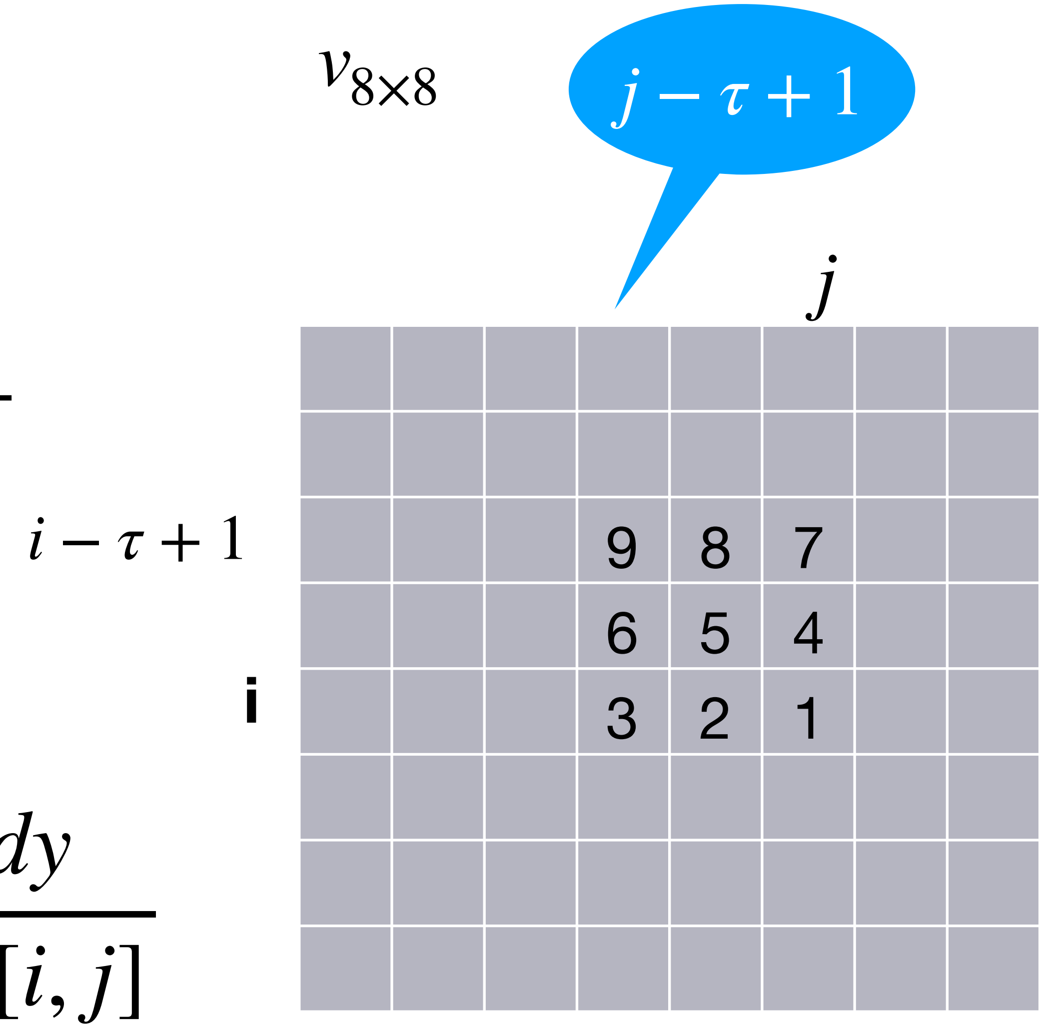
$\vdots$ $z[i-1, j]$

$z[i, j-\tau+1]$ $z[i, j-1]$ $z[i, j]$

$$u[i, j] = \frac{dy}{dx[i, j]} = \sum_{m}^{\tau} \sum_{n}^{\tau} H(m, n) * v[i-\tau+m, j-\tau+n]$$







$$v[i,j] = \frac{dy}{dz[i,j]}$$

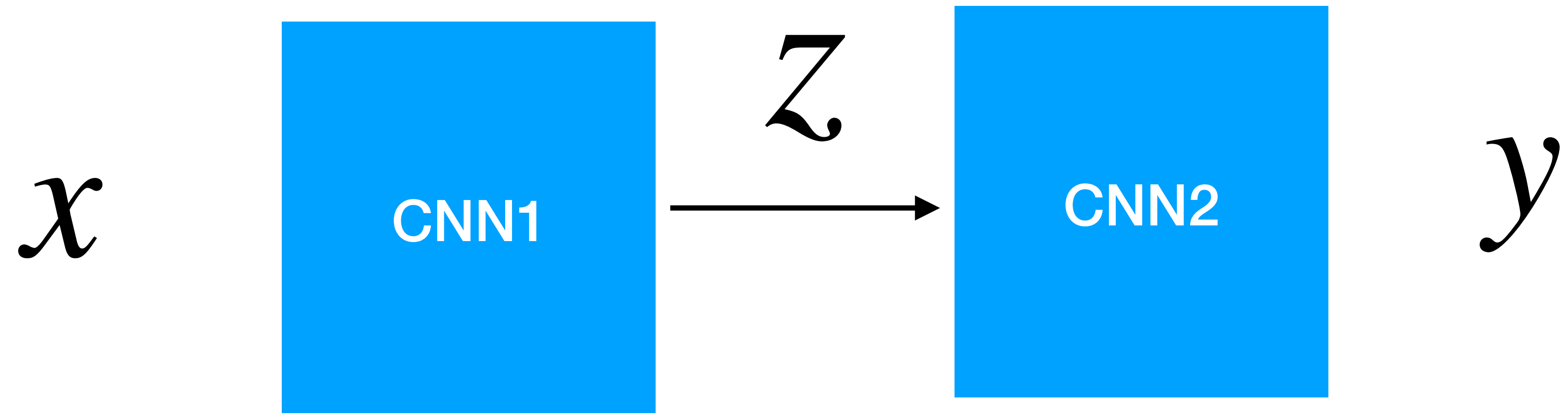
Signals translate forward

$$z[i,j] = \sum_{m}^{\tau} \sum_{n}^{\tau} K(m,n) * x[i+m-1, j+n-1]$$

Gradient back propagation

$$u[i,j] = \frac{dy}{dx[i,j]} = \sum_{m}^{\tau} \sum_{n}^{\tau} H(m,n) * v[i-\tau+m, j-\tau+n]$$

Our idea



Given $\frac{dy}{dz[i,j]}$ how to calculate $\frac{dy}{dx[i,j]}$ = ?