Lab 05

11-785/685/485

CNN: Basics

Instructors

Aarya Makwana Ketan Chaudhary ChooWon Sir

Sep 27 2024

Carnegie Mellon University

Agenda

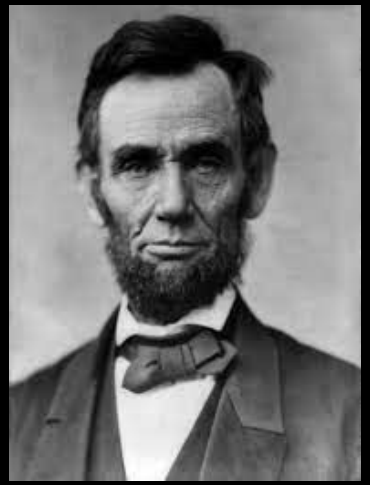
1. What is CNN?
2. Components of a CNN
3. Practice Quiz

Agenda

1. What is CNN?
2. Components of a CNN
3. Practice Quiz

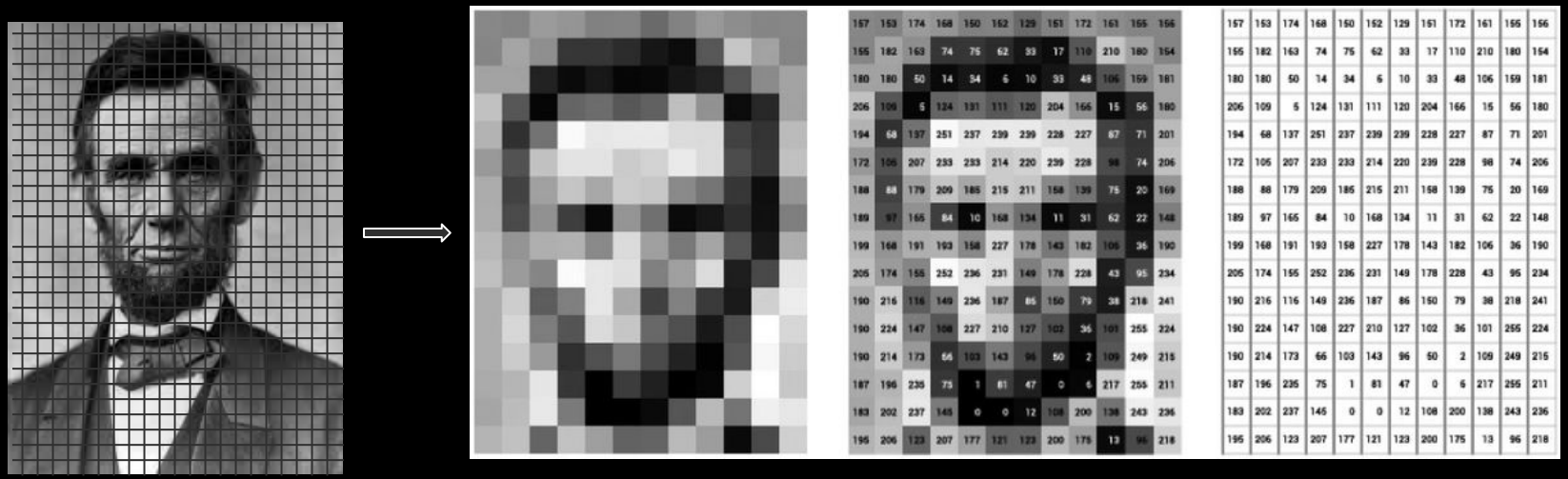
What is an image?

A visual representation



What is an image for a computer?

A matrix `I` of dimensions `(M,N)` with `I[i][j] = intensity(pixel(i,j))`

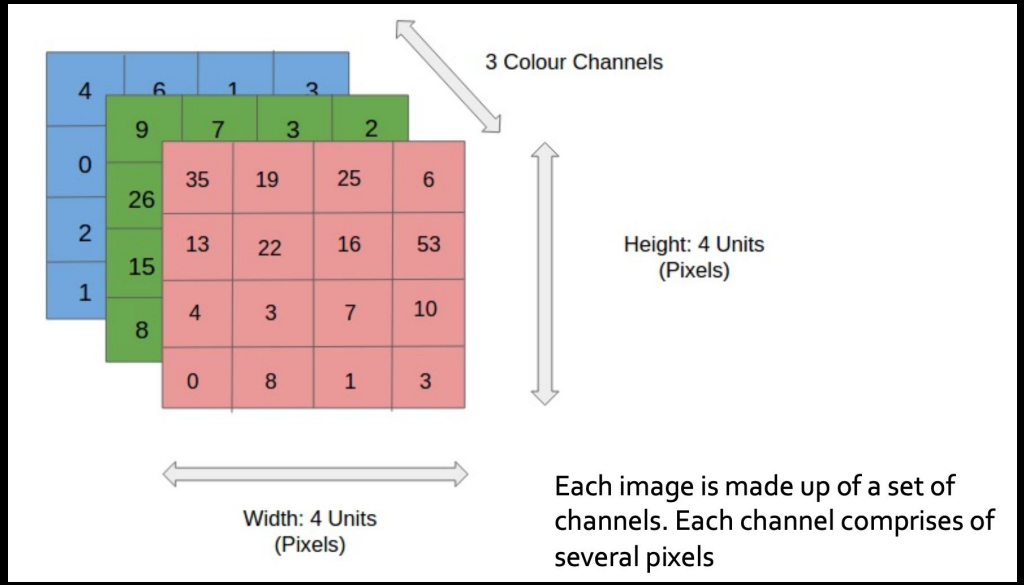


How to represent an image?

I → (3,M,N)

I[c][i][j] =

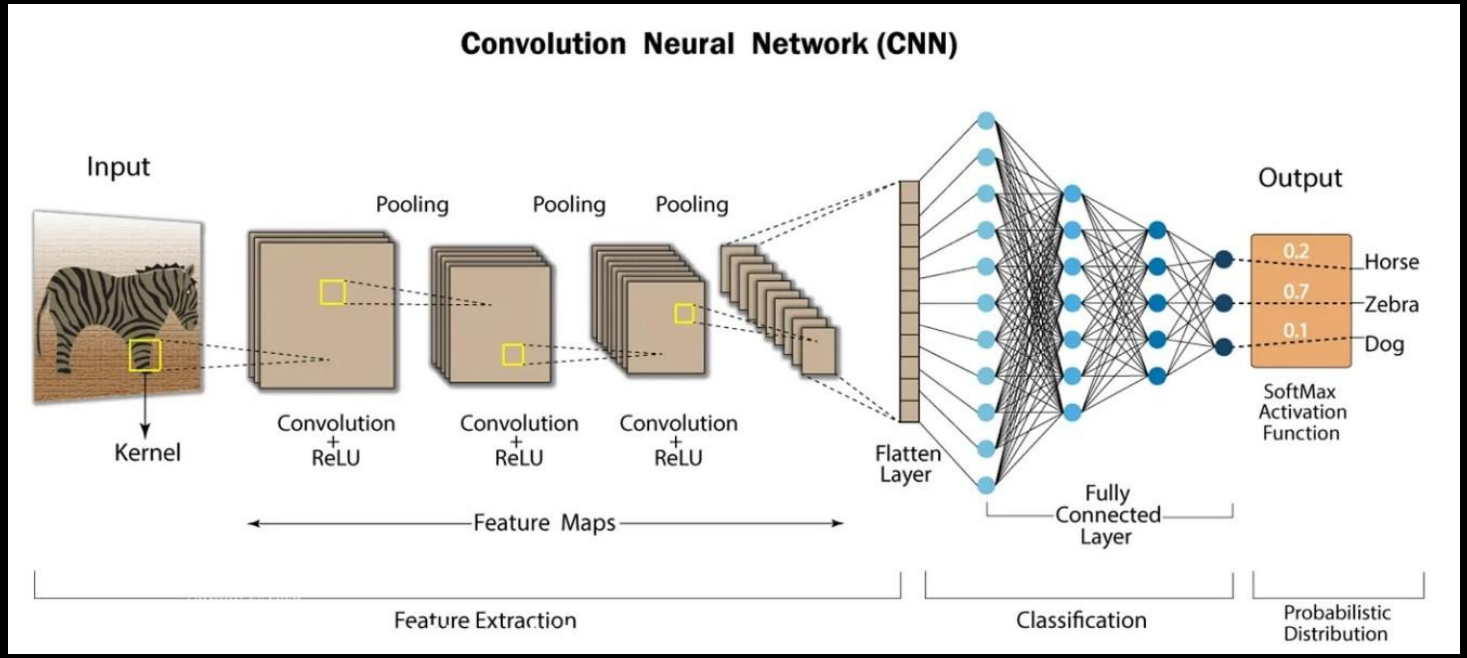
Intensity at **pixel(i,j)** for channel **c**



3 for a colored image, 1 for B&W.

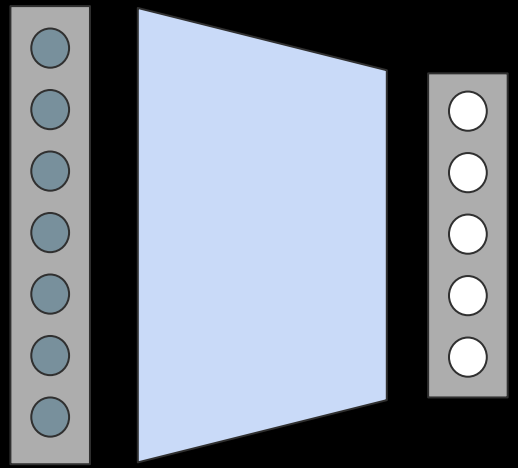
The number of channels you encounter could even increase!

What is a CNN?

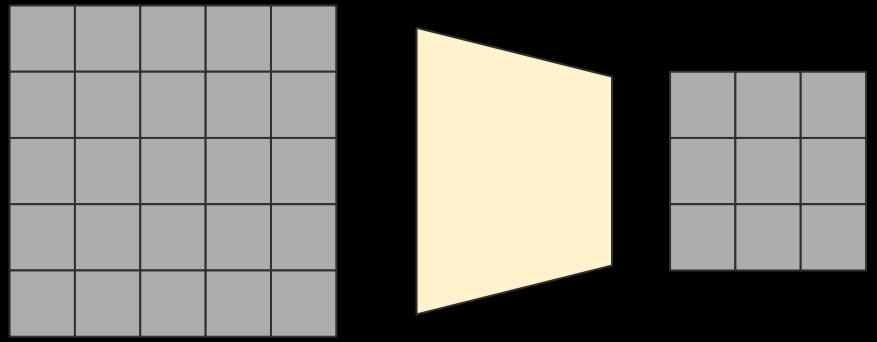


Source: https://www.linkedin.com/pulse/what-convolutional-neural-network-cnn-deep-learning-nafiz-shahriar/

Why not use MLP?



Vector to Vector



Feature map to Feature map

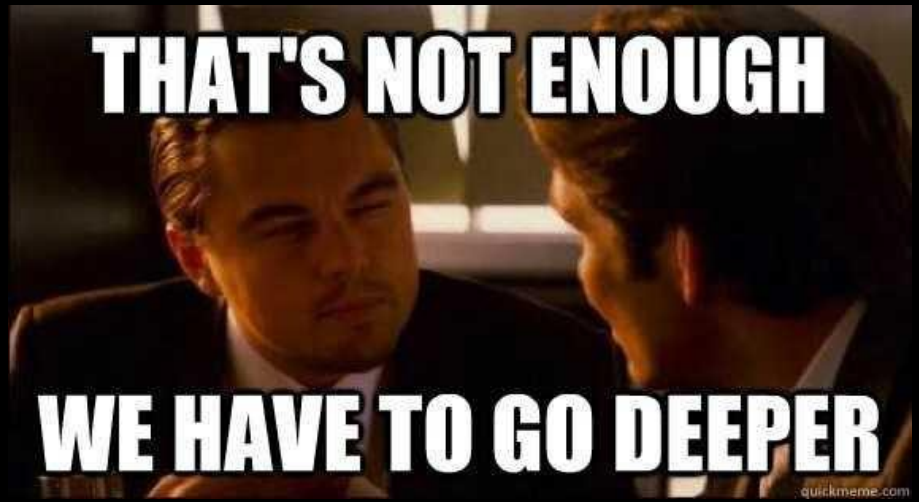
Building Blocks of a CNN

Main Building blocks

- Convolution Layer
- Pooling Layer

Others(can also be found in MLP)

- Activation Layer
- Normalization Layer(LayerNorm, etc)
- Batch Normalization (BatchNorm)



Building Blocks of a CNN

Hyperparameters

Conv layer:

- Filter/kernel size
- Stride
- Number of filters
- Padding

Pooling layer:

- Pooling type & size(pool size)
- Stride

Number of layers

Convolutional Layer (Conv layer)

Convolutional layers are the core components of CNNs. They apply convolution operations using learnable filters (kernels) to the input data. These filters slide across the input to detect patterns, edges, and features.

Kernel/Filters

The size of the convolutional kernels (filters) determines the spatial extent over which the convolution operation is applied. Common kernel sizes are 3x3, 5x5, or 7x7.

Stride

The stride specifies the step size at which the convolutional kernel/filter is moved across the input data. A larger stride reduces the spatial dimensions of the output feature maps.

Padding

0	0	0	0	0	0
0	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	0
0	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	0
0	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$	0
0	$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$	0
0	0	0	0	0	0

Convolution

Essentially element-wise (Hadamard) multiplications and summations**(Dot product)**

Input - **A**

0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1

*

Kernel - **W**

-1	0	1
-2	0	2
-1	0	1

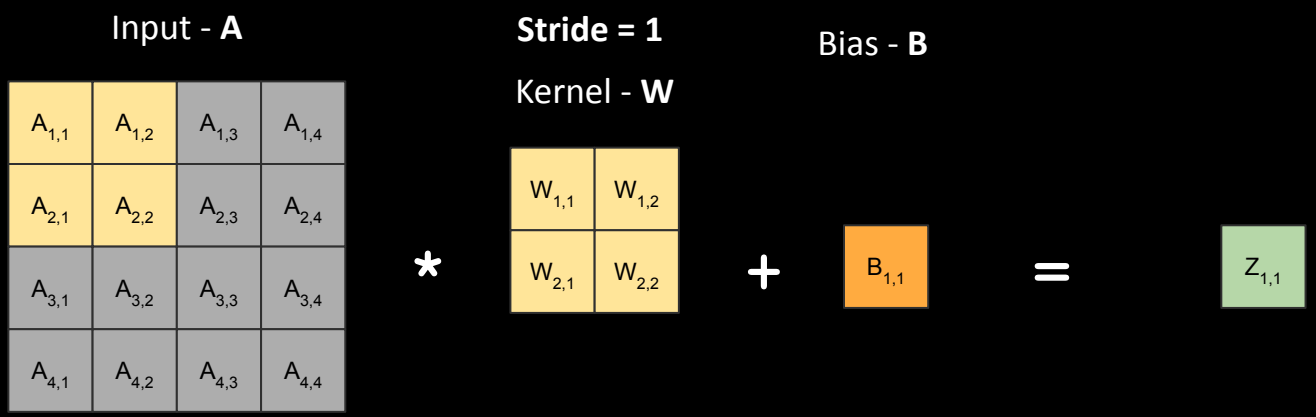
=

0	0	4	4	0	0
0	0	4	4	0	0
0	0	4	4	0	0
0	0	4	4	0	0
0	0	4	4	0	0
0	0	4	4	0	0

Here the stride is 1

Convolution

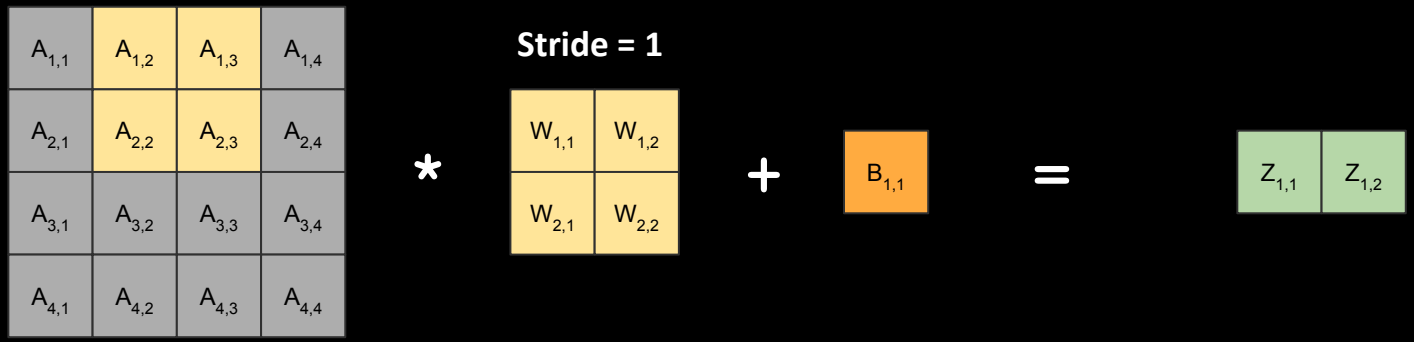
Essentially element-wise (Hadamard) multiplications and summations



$$Z_{1,1} = (A_{1,1} * W_{1,1}) + (A_{1,2} * W_{1,2}) + (A_{2,1} * W_{2,1}) + (A_{2,2} * W_{2,2}) + B$$

Convolution

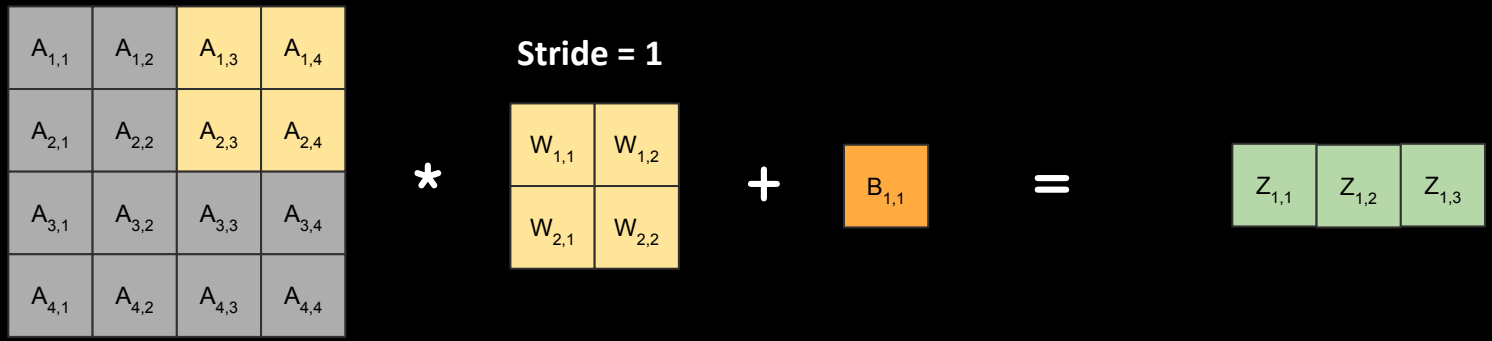
Essentially element-wise (Hadamard) multiplications and summations



$$Z_{1,2} = (A_{1,2} * W_{1,1}) + (A_{1,3} * W_{1,2}) + (A_{2,2} * W_{2,1}) + (A_{2,3} * W_{2,2}) + B$$

Convolution

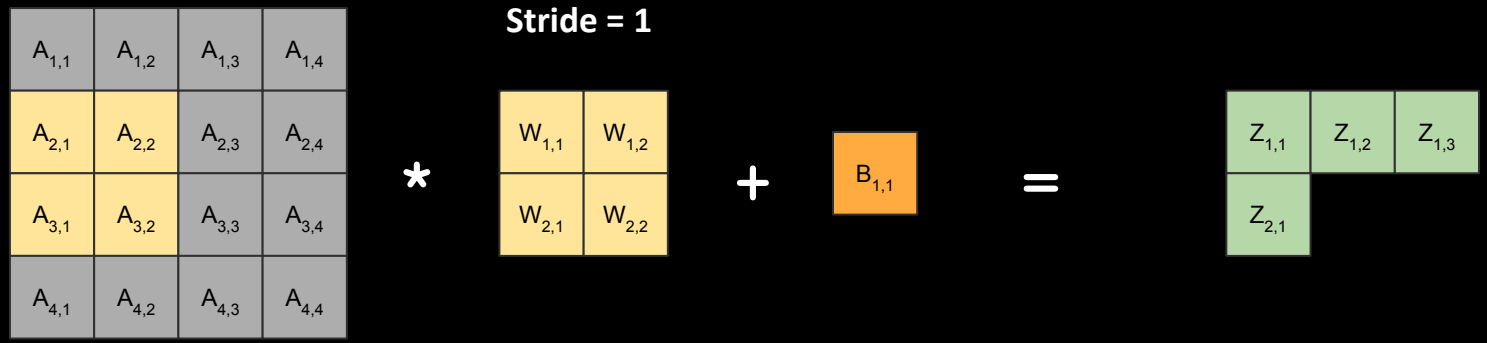
Essentially element-wise (Hadamard) multiplications and summations



$$Z_{1,3} = (A_{1,3} * W_{1,1}) + (A_{1,4} * W_{1,2}) + (A_{2,3} * W_{2,1}) + (A_{2,4} * W_{2,2}) + B$$

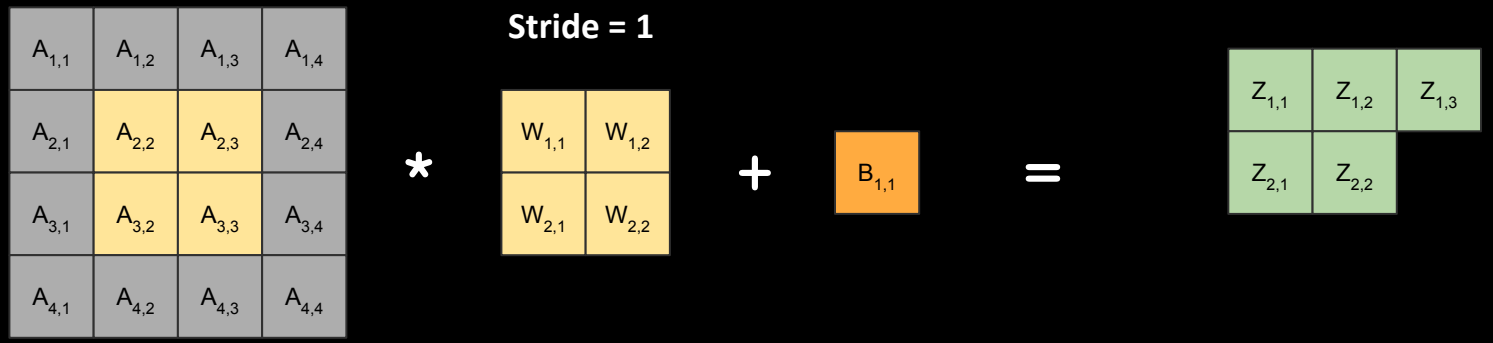
Convolution

Essentially element-wise (Hadamard) multiplications and summations



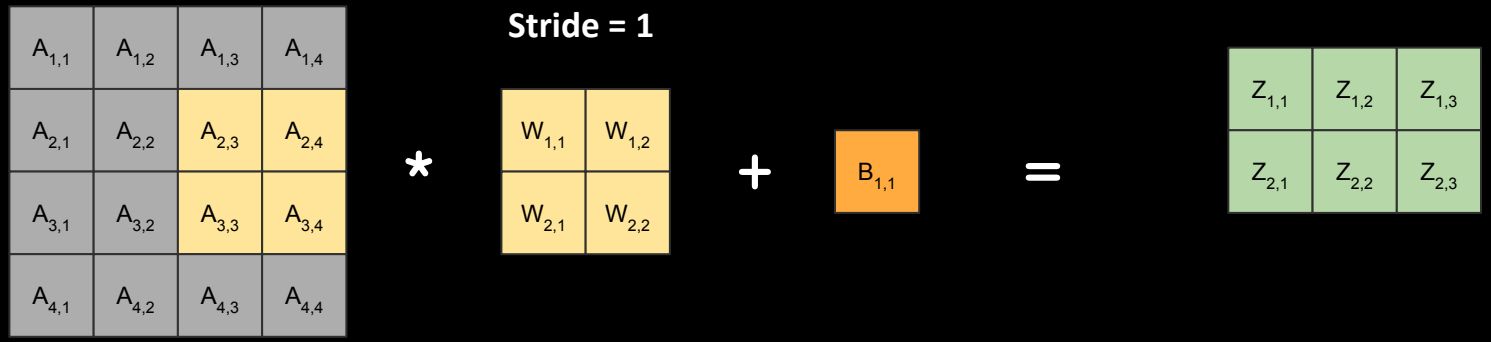
Convolution

Essentially element-wise (Hadamard) multiplications and summations



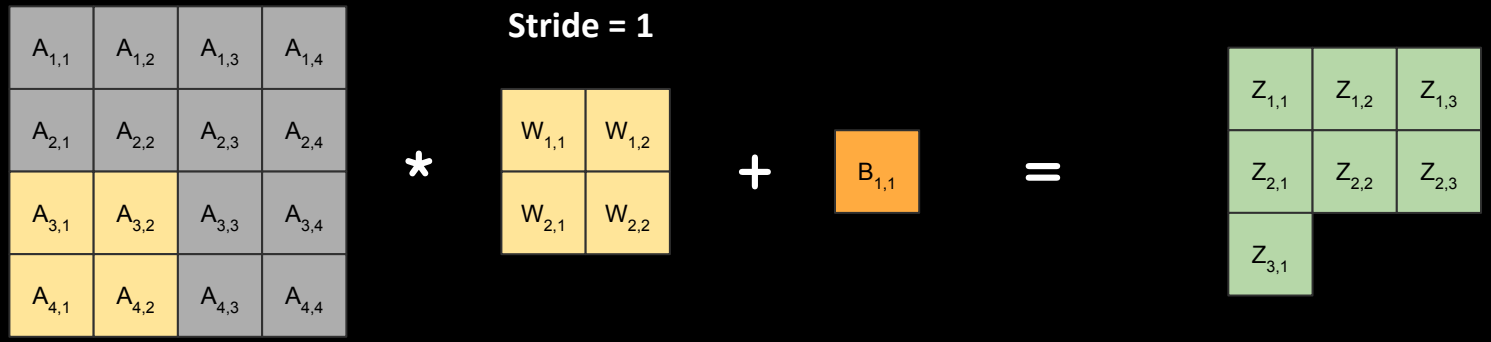
Convolution

Essentially element-wise (Hadamard) multiplications and summations



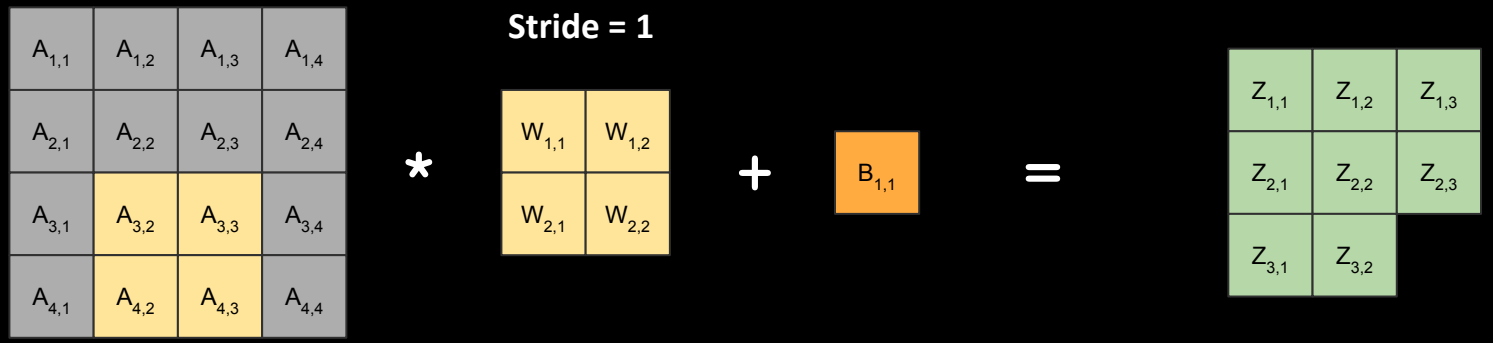
Convolution

Essentially element-wise (Hadamard) multiplications and summations



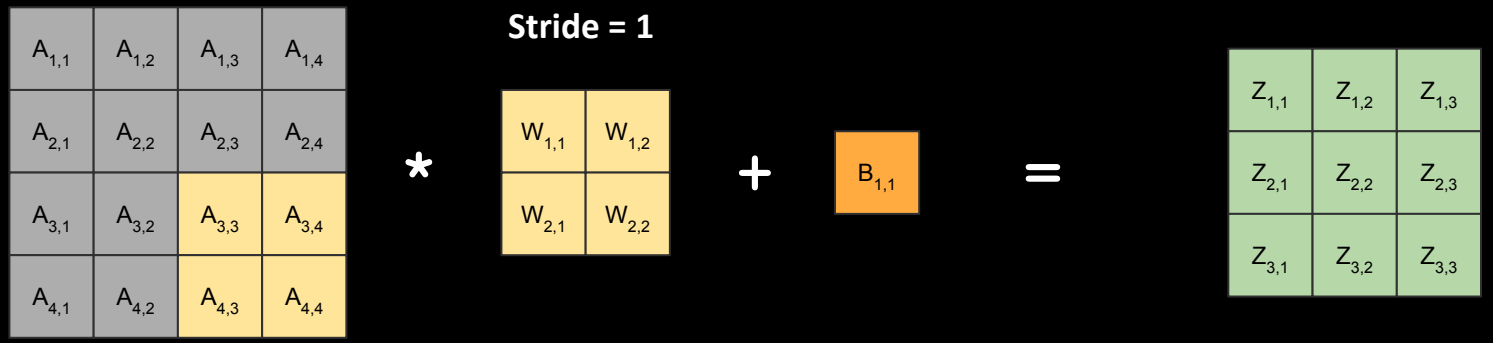
Convolution

Essentially element-wise (Hadamard) multiplications and summations

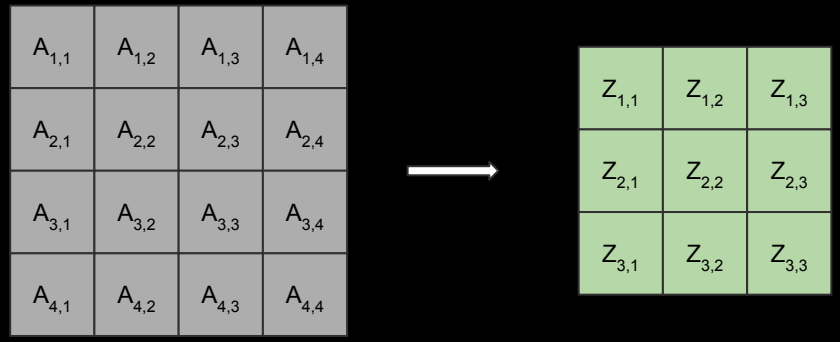


Convolution

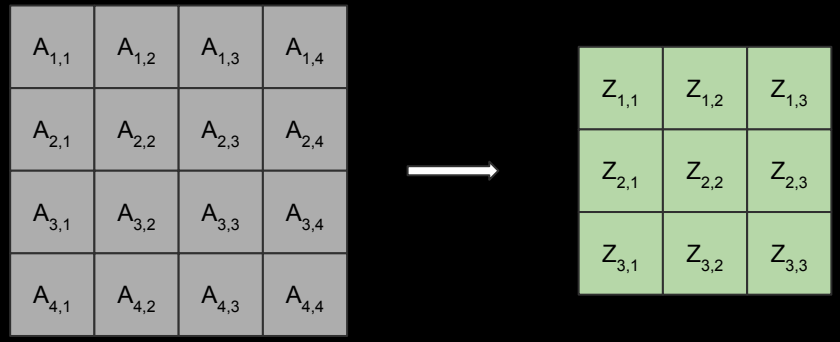
Essentially element-wise (Hadamard) multiplications and summations



Output Size


$A_{1,1}$ $A_{1,2}$ $A_{1,3}$ $A_{1,4}$
$A_{2,1}$ $A_{2,2}$ $A_{2,3}$ $A_{2,4}$
$A_{3,1}$ $A_{3,2}$ $A_{3,3}$ $A_{3,4}$
$A_{4,1}$ $A_{4,2}$ $A_{4,3}$ $A_{4,4}$
$Z_{1,1}$ $Z_{1,2}$ $Z_{1,3}$
$Z_{2,1}$ $Z_{2,2}$ $Z_{2,3}$
$Z_{3,1}$ $Z_{3,2}$ $Z_{3,3}$

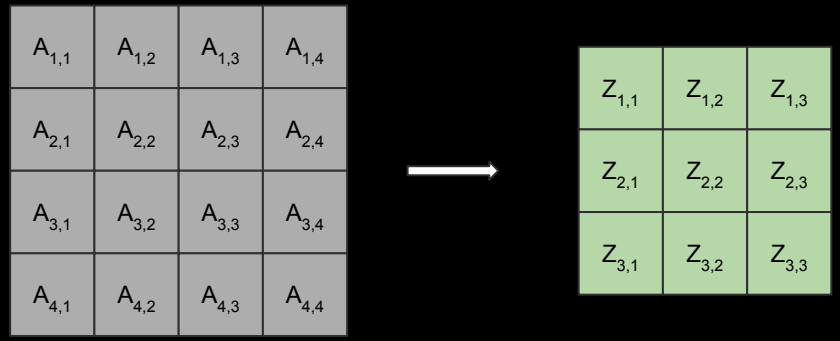
Output Size



Output Width =
$[(W_{in} - W_k + 2P) // (S)] + 1$

Same goes for Height.

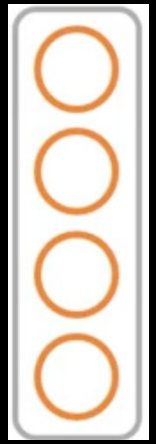
Output Size



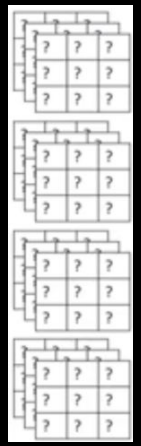
$$\text{Output Width} = [(W_{in} - W_k + 2P) \;//\; (S)] + 1$$

P: Padding (here - 0)
S: Stride (here - 1)

Convolution Neural Networks



Kernels



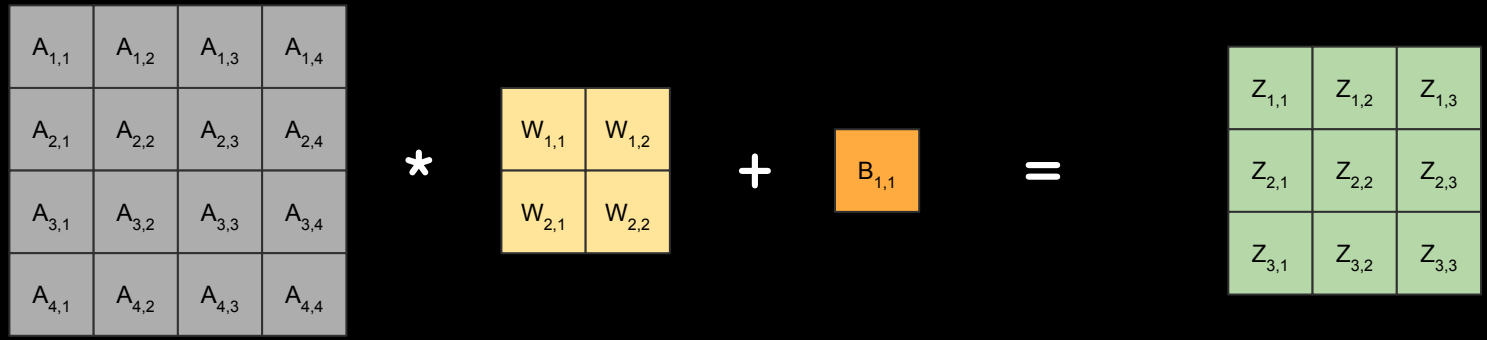
Convolution Layer

Activation Function

Output channels from Convolutional Layer

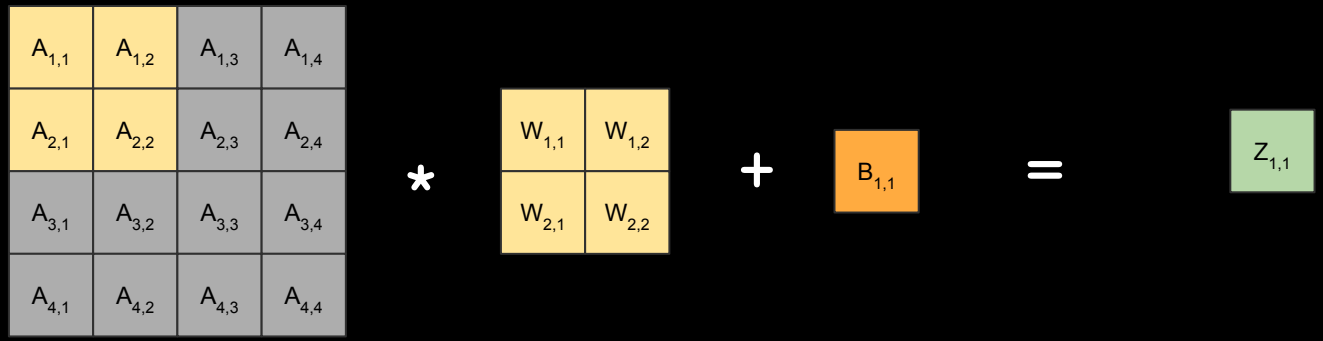
Stride = 1

What we did before - The kernel “moves” one pixel (or element) at a time.



Stride = 2

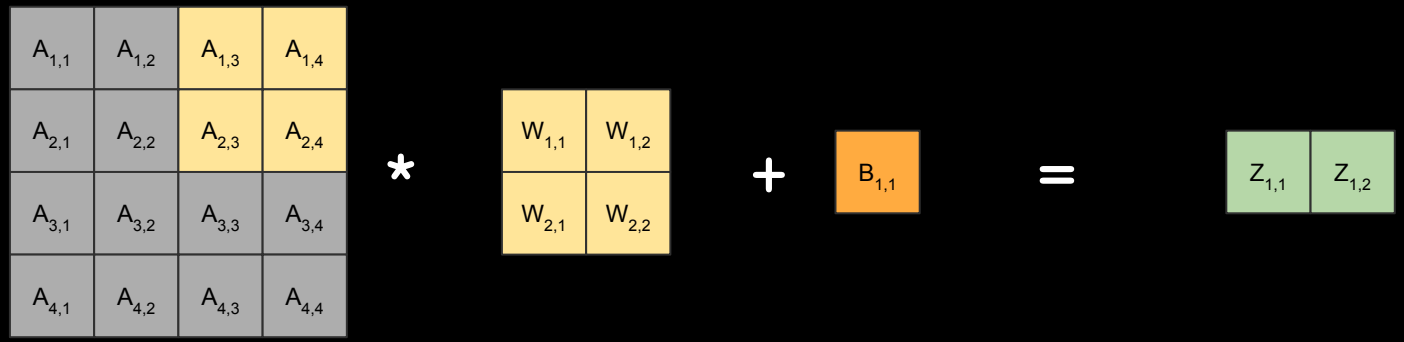
Start at the same place



$$Z_{1,1} = (A_{1,1} * W_{1,1}) + (A_{1,2} * W_{1,2}) + (A_{2,1} * W_{2,1}) + (A_{2,2} * W_{2,2}) + B$$

Stride = 2

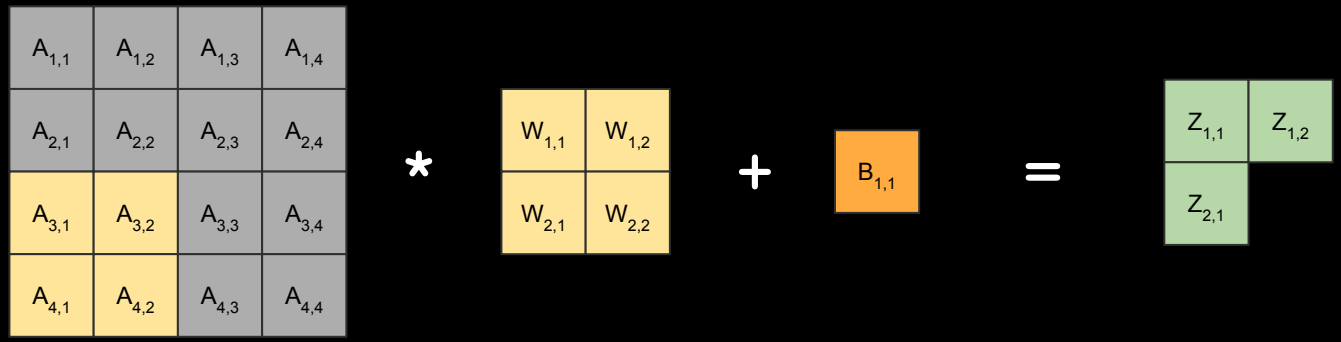
Move two elements to the right



$$Z_{1,2} = (A_{1,3} * W_{1,1}) + (A_{1,4} * W_{1,2}) + (A_{2,3} * W_{2,1}) + (A_{2,4} * W_{2,2}) + B$$

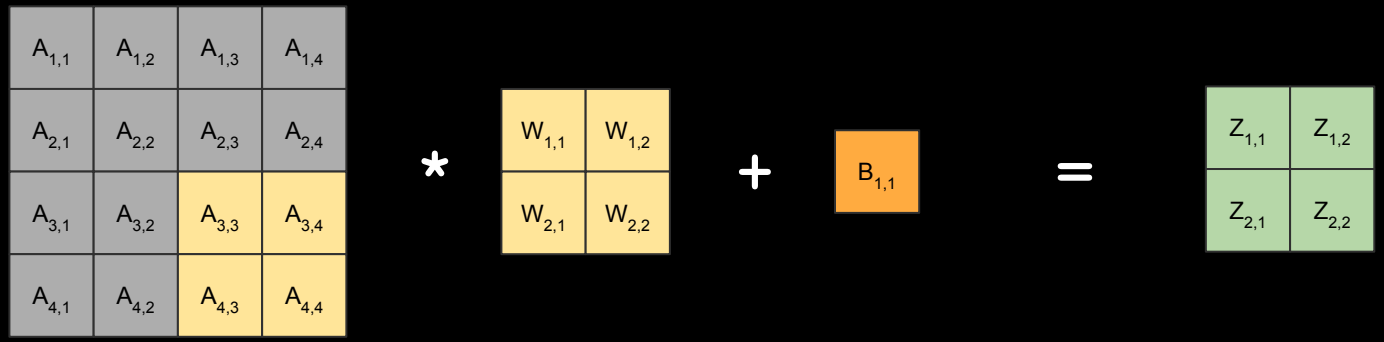
Stride = 2

Move two elements down.

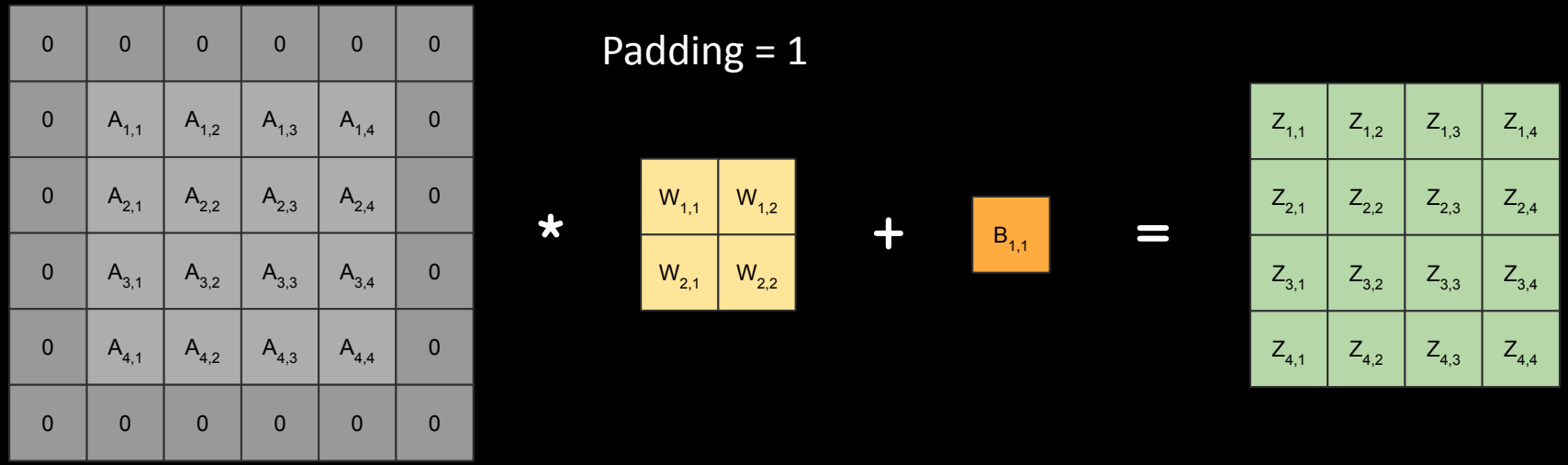


Stride = 2

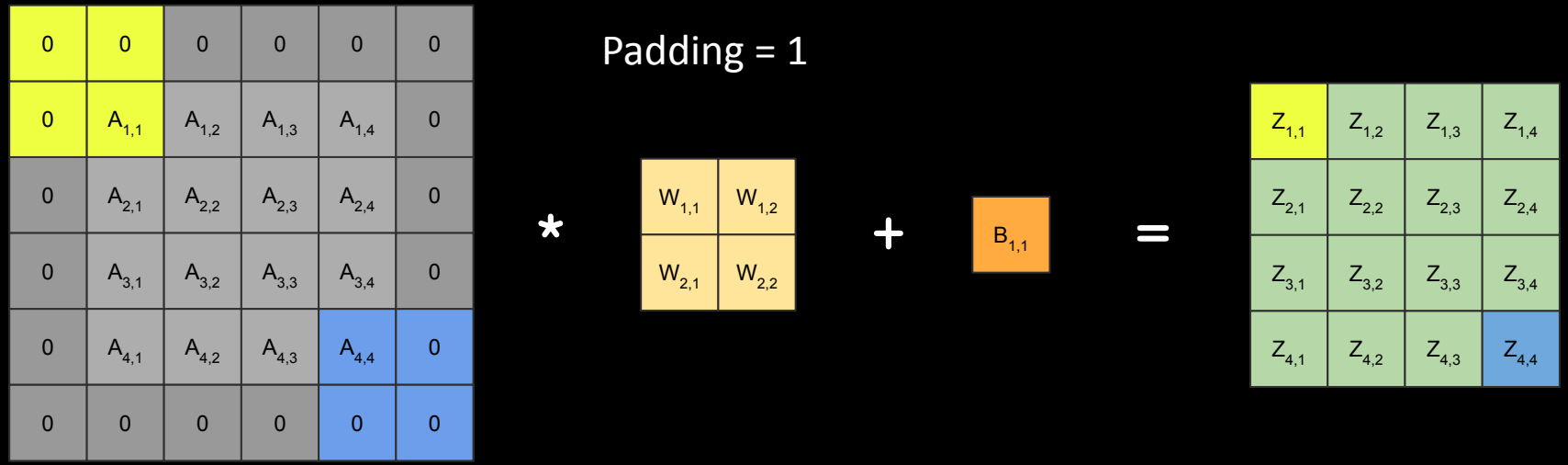
Move two elements to the right.



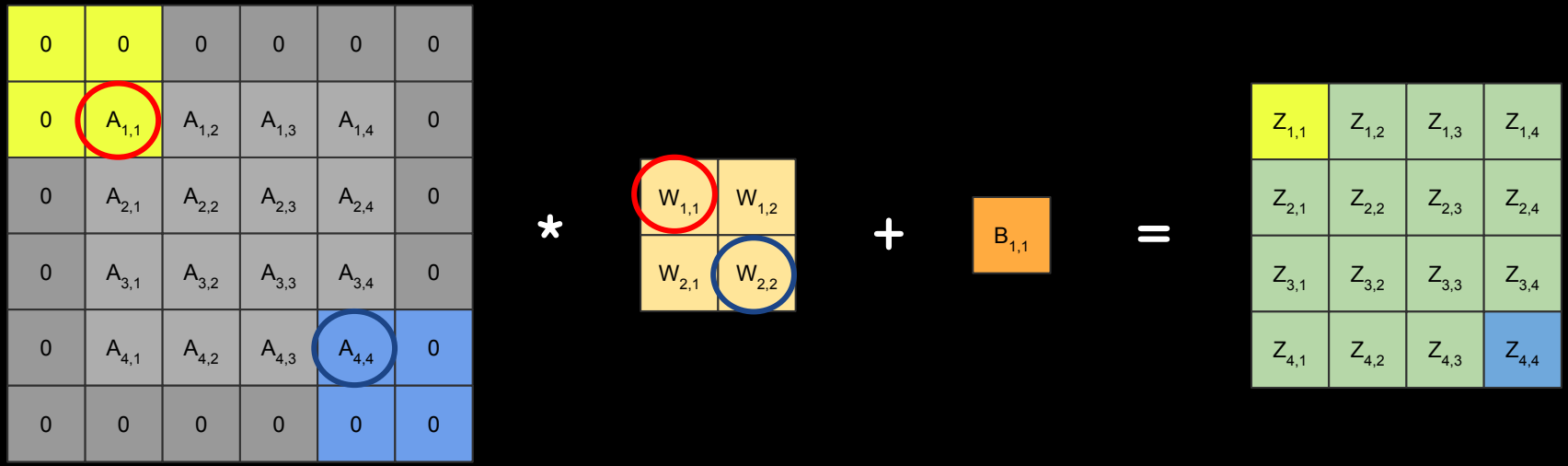
Padding


Padding = 1
0 0 0 0 0 0
0 $A_{1,1}$ $A_{1,2}$ $A_{1,3}$ $A_{1,4}$ 0
0 $A_{2,1}$ $A_{2,2}$ $A_{2,3}$ $A_{2,4}$ 0
0 $A_{3,1}$ $A_{3,2}$ $A_{3,3}$ $A_{3,4}$ 0
0 $A_{4,1}$ $A_{4,2}$ $A_{4,3}$ $A_{4,4}$ 0
0 0 0 0 0 0
*
$W_{1,1}$ $W_{1,2}$
$W_{2,1}$ $W_{2,2}$
+
$B_{1,1}$
=
$Z_{1,1}$ $Z_{1,2}$ $Z_{1,3}$ $Z_{1,4}$
$Z_{2,1}$ $Z_{2,2}$ $Z_{2,3}$ $Z_{2,4}$
$Z_{3,1}$ $Z_{3,2}$ $Z_{3,3}$ $Z_{3,4}$
$Z_{4,1}$ $Z_{4,2}$ $Z_{4,3}$ $Z_{4,4}$

Padding

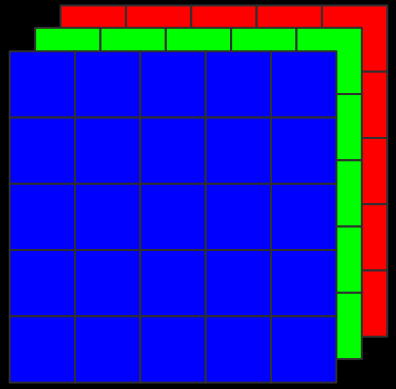

Padding = 1
0 0 0 0 0 0
0 A1,1 A1,2 A1,3 A1,4 0
0 A2,1 A2,2 A2,3 A2,4 0
0 A3,1 A3,2 A3,3 A3,4 0
0 A4,1 A4,2 A4,3 A4,4 0
0 0 0 0 0 0
*
W1,1 W1,2
W2,1 W2,2
+
B1,1
=
Z1,1 Z1,2 Z1,3 Z1,4
Z2,1 Z2,2 Z2,3 Z2,4
Z3,1 Z3,2 Z3,3 Z3,4
Z4,1 Z4,2 Z4,3 Z4,4

Padding

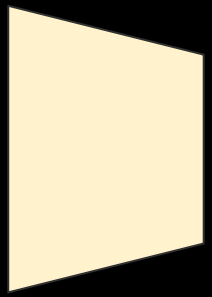

0 0 0 0 0 0
0 A1,1 A1,2 A1,3 A1,4 0
0 A2,1 A2,2 A2,3 A2,4 0
0 A3,1 A3,2 A3,3 A3,4 0
0 A4,1 A4,2 A4,3 A4,4 0
0 0 0 0 0 0
*
W1,1 W1,2
W2,1 W2,2
+
B1,1
=
Z1,1 Z1,2 Z1,3 Z1,4
Z2,1 Z2,2 Z2,3 Z2,4
Z3,1 Z3,2 Z3,3 Z3,4
Z4,1 Z4,2 Z4,3 Z4,4

Multi-channel CNN

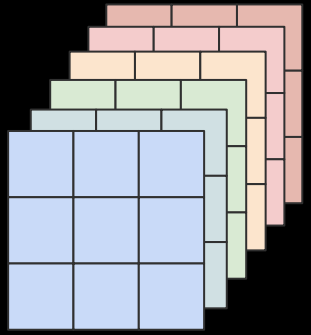
Input channels



CNN/Conv layer



Output channels

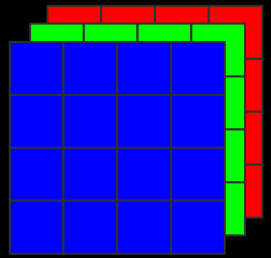


Multi-channel CNN

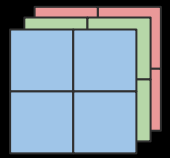
- Each kernel (or **filter**) has as many channels as the input does.

  `[kernel channels = Input channels]`

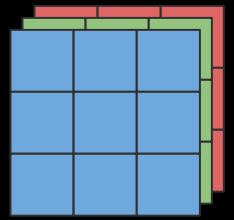
- Channel **c** of **the kernel** convolves with channel **c** (corresponding) of **the input**.
- The number of output channels from the convolution = number of **filters(kernels)** applied to the input.



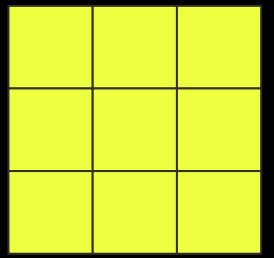
Input
# of channels = 3



Kernel/filter
# of channels = 3

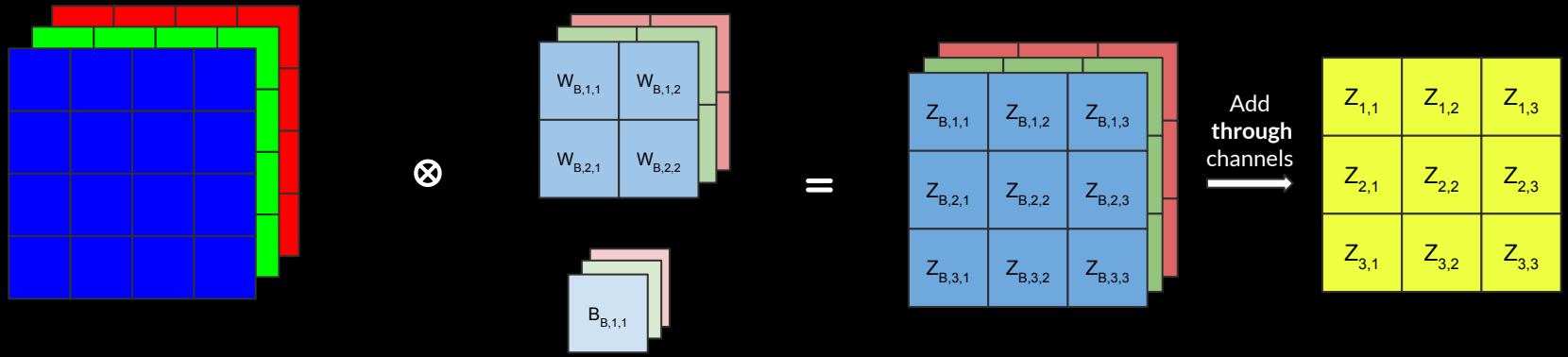


output maps



output map

1 Filter with 3-channel input



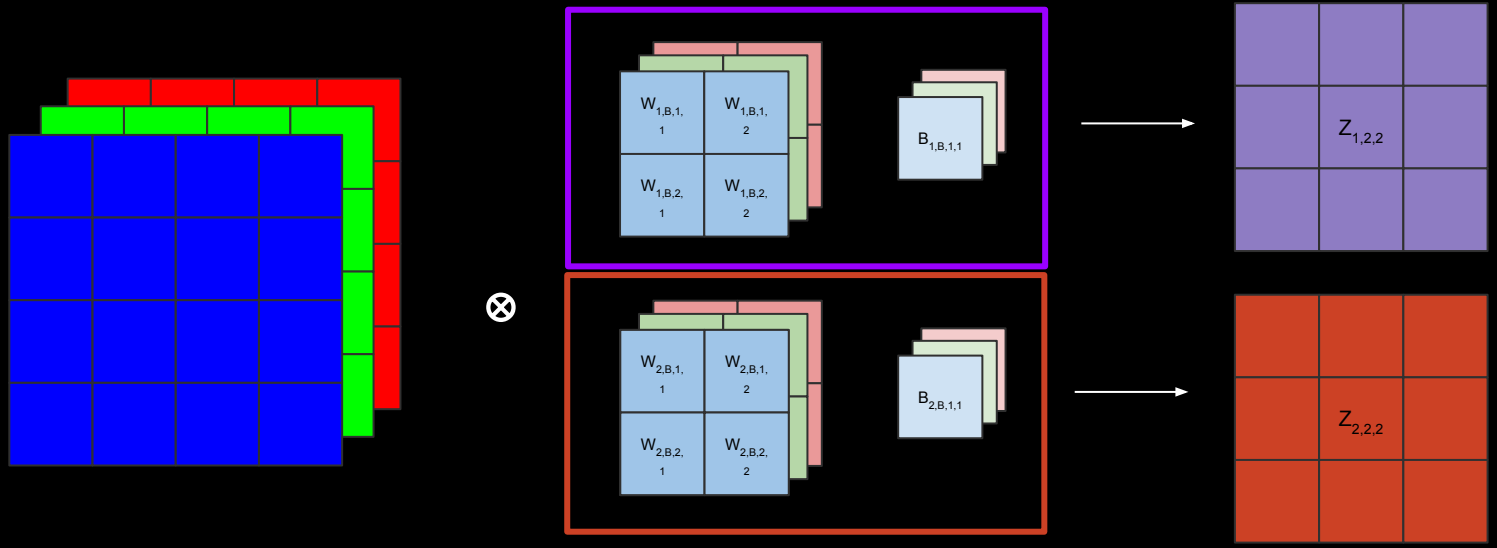
3 channel input

Kernel/filter

3 almost-output maps

1 output map

2 Filters with 3-channel input



3 channel input

2 Kernels

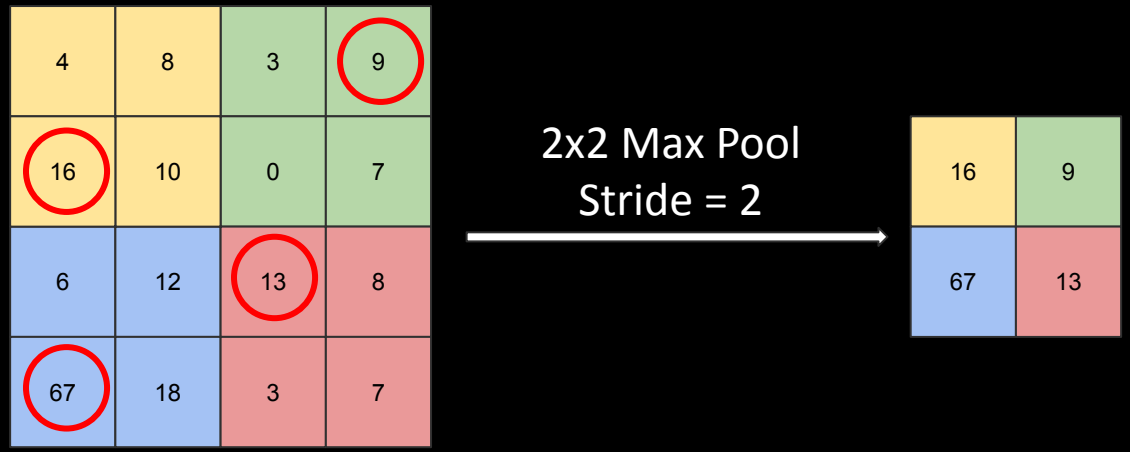
2 output maps

Pooling

Pooling Layer

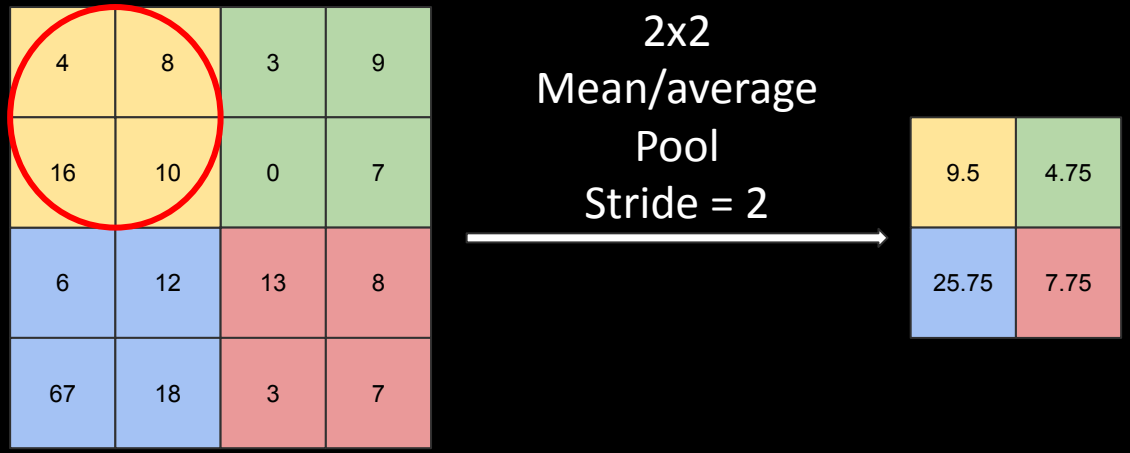
- Downsamples the spatial dimensions of the feature maps produced by convolutional layers
- Retains important features while reducing computational load
- Max-pooling and average-pooling are common pooling operations
- Introduces Jitter Invariance

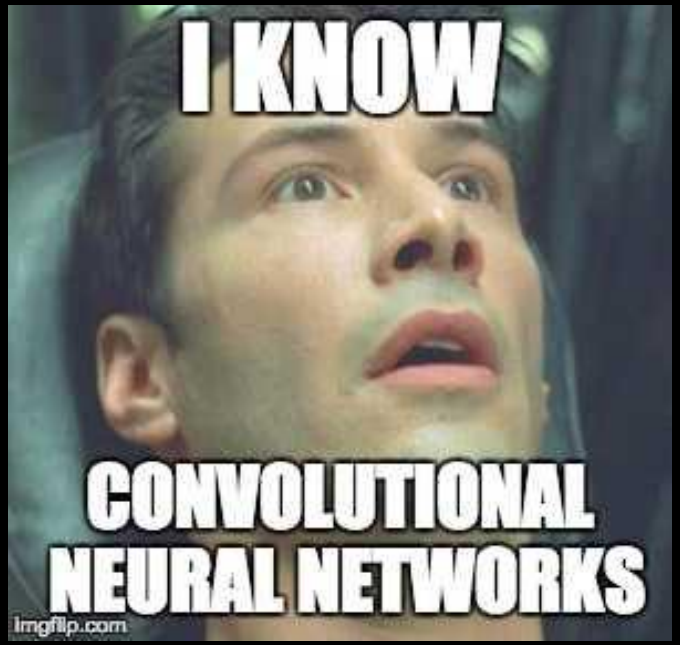
Pooling (Max Pooling)

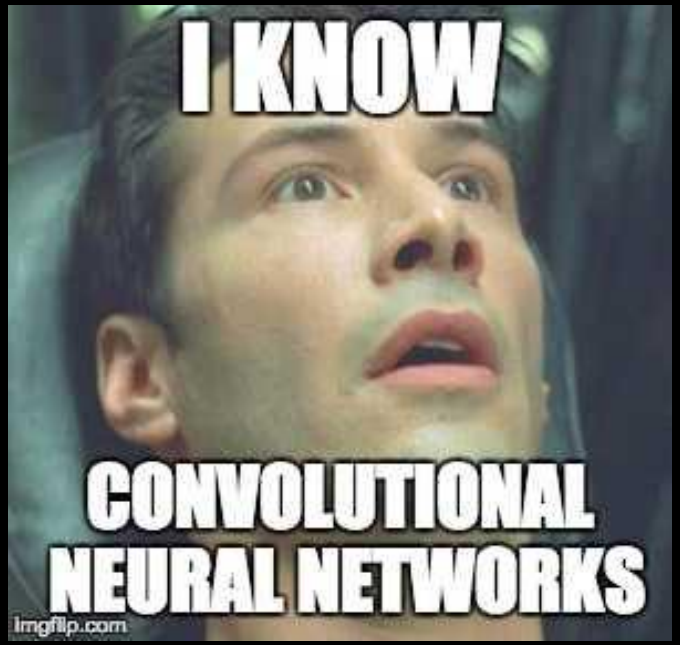


Pooling (Average Pooling)

Calculate the mean




I KNOW
CONVOLUTIONAL
NEURAL NETWORKS
imgflip.com



Do you?

Some Questions to Practice now!

Question 1

If you apply a 3x3 filter with a stride of 1 and no padding to a 6x6 input image, what will be the dimensions of the output feature map?

a) 6x6

b) 4x4

c) 3x3

d) 5x5

Question 1

If you apply a 3x3 filter with a stride of 1 and no padding to a 6x6 input image, what will be the dimensions of the output feature map?

a) 6x6

b) 4x4

c) 3x3

d) 5x5

Question 2

1. A convolution layer of a 1D CNN (a TDNN) operates on the following input:

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

with the following two convolutional filters:

$$W_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The filters operate on input Y to compute the output Z. The convolution uses a stride of 1.

What is the second value of the second channel of Z (i.e.Z(2, 2) where the first index represents channel)? The indexing is 1-based (so indices start at 1).

(a) 3 **(b) 2** **(c) 0** **(d) 1**

Question 2

1. A convolution layer of a 1D CNN (a TDNN) operates on the following input:

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

with the following two convolutional filters:

$$W_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The filters operate on input Y to compute the output Z. The convolution uses a stride of 1.

What is the second value of the second channel of Z (i.e.Z(2, 2) where the first index represents channel)? The indexing is 1-based (so indices start at 1).

(a) 3 **(b) 2** **(c) 0** **(d) 1**

Question 3

A convolution layer of a 2D CNN operates on the following input (the input is two channels of 4x4, represented by the two 4x4 matrices, where the left matrix represents the first channel and the right matrix represents the second channel):

$$Y = \left[\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}\right]$$

with the following two convolutional filters:

$$W_1 = \left[\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\right]$$

$$W_2 = \left[\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right]$$

The filters operate on input Y to compute the output Z . The convolution uses a stride of 2. What is the value at position (2,2) of the first channel of Z (i.e. Z(1, 2, 2) where the first index represents the channel)? The indexing is 1-based (so indices start at 1).

(a) 8 **(b) 4** **(c) 2** **(d) 0**

Question 3

A convolution layer of a 2D CNN operates on the following input (the input is two channels of 4x4, represented by the two 4x4 matrices, where the left matrix represents the first channel and the right matrix represents the second channel):

$$Y = \left[\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}\right]$$

with the following two convolutional filters:

$$W_1 = \left[\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\right]$$

$$W_2 = \left[\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right]$$

The filters operate on input Y to compute the output Z . The convolution uses a stride of 2. What is the value at position (2,2) of the first channel of Z (i.e. Z(1, 2, 2) where the first index represents the channel)? The indexing is 1-based (so indices start at 1).

(a) 8 **(b) 4** **(c) 2** **(d) 0**

Question 4

A convolution layer of a 1D CNN (a TDNN) operates on the following input:

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

with the following three convolutional filters:

$$W_1 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$W_3 = \begin{bmatrix} 3 & 2 \\ 1 & 0 \end{bmatrix}$$

The filters operate on input Y to compute the output Z . The convolution uses a stride of 2. No zero padding is employed in the forward pass. What is the (2, 2) element of the output Z?

(a) 2 **(b) 1** **(c) 3** **(d) 0**

Question 4

A convolution layer of a 1D CNN (a TDNN) operates on the following input:

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

with the following three convolutional filters:

$$W_1 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$W_3 = \begin{bmatrix} 3 & 2 \\ 1 & 0 \end{bmatrix}$$

The filters operate on input Y to compute the output Z . The convolution uses a stride of 2. No zero padding is employed in the forward pass. What is the (2, 2) element of the output Z?

(a) 2 **(b) 1** **(c) 3** **(d) 0**

Question 5

What is the convolution of (1, 2, 3) * (4, 5, 6)?

Question 5

What is the convolution of (1, 2, 3) * (4, 5, 6)?

> **(4, 13, 28, 27, 18)**

Question 6

Which of the following kernels is most likely to be an edge detection filter?

a) [0.11, 0.11, 0.11]
[0.11, 0.11, 0.11]
[0.11, 0.11, 0.11]

b) [0, -1, 0]
[-1, 4, -1]
[0, -1, 0]

c) [1, 1, 1]
[1, 1, 1]
[1, 1, 1]

d) [-1, -1, -1]
[-1, 8, -1]
[-1, -1, -1]

Question 6

Which of the following kernels is most likely to be an edge detection filter?

a) [0.11, 0.11, 0.11]
[0.11, 0.11, 0.11]
[0.11, 0.11, 0.11]

b) [0, -1, 0]
[-1, 4, -1]
[0, -1, 0]

c) [1, 1, 1]
[1, 1, 1]
[1, 1, 1]

d) [-1, -1, -1]
[-1, 8, -1]
[-1, -1, -1]

Question 7

Which of the following kernels is most likely to be a blur filter?

a) [0, -1, 0]
[-1, 5, -1]
[0, -1, 0]

b) [-1, -1, -1]
[-1, 8, -1]
[-1, -1, -1]]

c) [1/9, 1/9, 1/9]
[1/9, 1/9, 1/9]
[1/9, 1/9, 1/9]

d) [0, 1, 0]
[1, -4, 1]
[0, 1, 0]

Question 7

Which of the following kernels is most likely to be a blur filter?

a) [0, -1, 0]
[-1, 5, -1]
[0, -1, 0]

b) [-1, -1, -1]
[-1, 8, -1]
[-1, -1, -1]]

c) [1/9, 1/9, 1/9]
[1/9, 1/9, 1/9]
[1/9, 1/9, 1/9]

d) [0, 1, 0]
[1, -4, 1]
[0, 1, 0]

Question 8

If you perform a tensordot operation between a tensor A of shape (10, 20, 30) and a tensor B of shape (30, 40, 50), along the last axis of A and the first axis of B, what will be the shape of the resulting tensor?

a) (10, 20, 40, 50)

b) (10, 20, 50)

c) (30, 40, 50)

d) (10, 20, 30, 40, 50)

Question 8

If you perform a tensordot operation between a tensor A of shape (10, 20, 30) and a tensor B of shape (30, 40, 50), along the last axis of A and the first axis of B, what will be the shape of the resulting tensor?

a) (10, 20, 40, 50)

b) (10, 20, 50)

c) (30, 40, 50)

d) (10, 20, 30, 40, 50)

Question 9

If you have an input tensor with shape (32, 28, 28, 3) and apply 64 filters of size 5x5, what will be the shape of the weight tensor for this convolutional layer?

a) (5, 5, 3, 64)

b) (64, 5, 5, 3)

c) (28, 28, 3, 64)

d) (32, 5, 5, 64)

Question 9

If you have an input tensor with shape (32, 28, 28, 3) and apply 64 filters of size 5x5, what will be the shape of the weight tensor for this convolutional layer?

a) (5, 5, 3, 64)

b) (64, 5, 5, 3)

c) (28, 28, 3, 64)

d) (32, 5, 5, 64)

Convolutional Filters

Operation	Kernel ω	Image result g(x,y)
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Ridge or edge detection	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

Source: https://en.wikipedia.org/wiki/Kernel_(image_processing)

Thank you!