Matrix and Array Math

- Scalar
- Matrix
- 3D Array
- 4D Array

- Array arithmetic
- Matrix arithmetic
- Dot product
- Cross product
- Convolution

Matrix and array math underlie digital image processing.

To understand the math we need, you need to learn several concepts and operations.

Most of the concepts are surprisingly easy.

You'll learn how to use these ideas later, and how they are implemented in various programs (like matlab), later.

What's a **matrix**?

A **matrix** is a rectangular table of values

1	**2**	**3**
3	**5**	**6**
7	**9**	**2**

with rows

1	2	3
3	5	6
7	9	2

and columns.

A matrix with m rows and n columns is called an m-by-n matrix (written m n)

and m and n are called its **dimensions**.

The dimensions of a matrix are always given with the number of rows first,

then the number of columns.

3 x 4

1	2	4	1
5	6	7	1
3	3	1	2

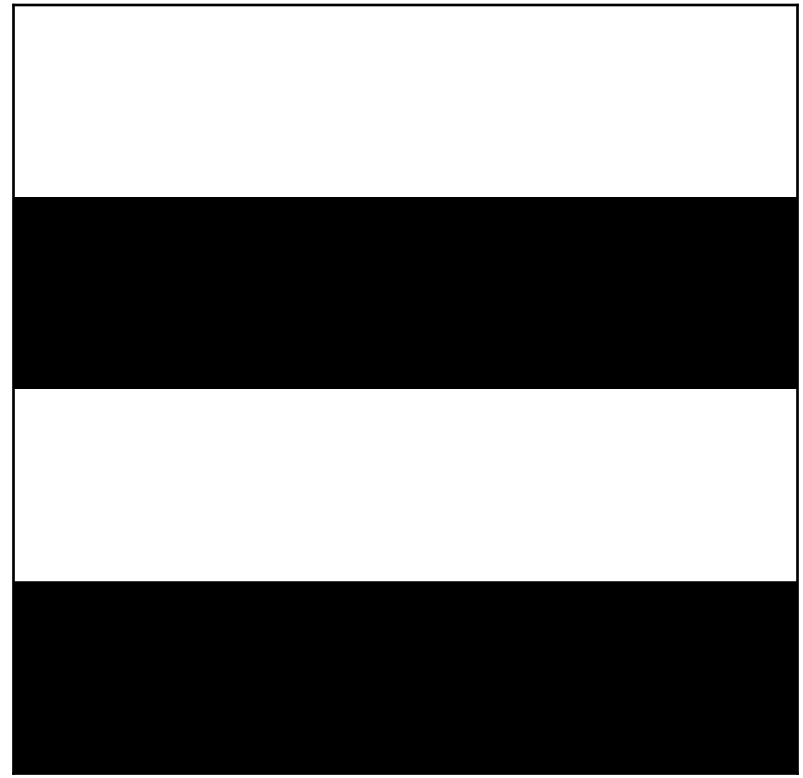
4 x 3

3	44	6
18	7	0.5
5	11	6
0	1	2

But, why do we care?

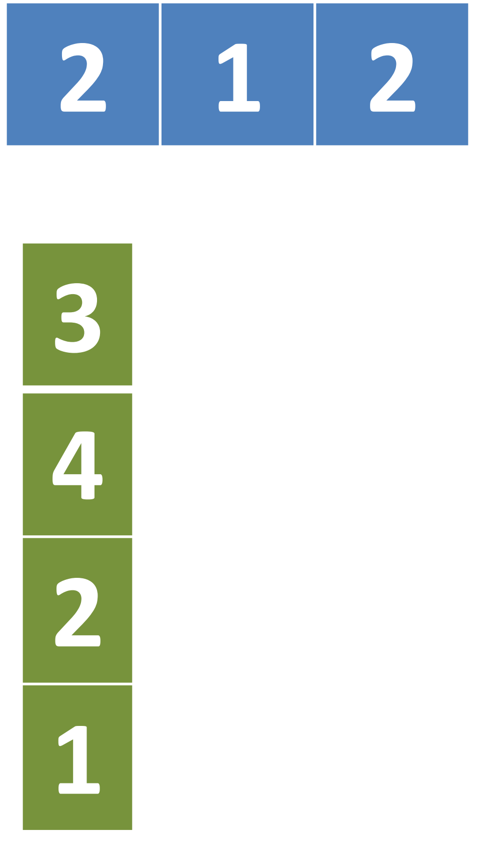
Because a matrix can be (and is) used to represent an image:

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



What's a **vector**?

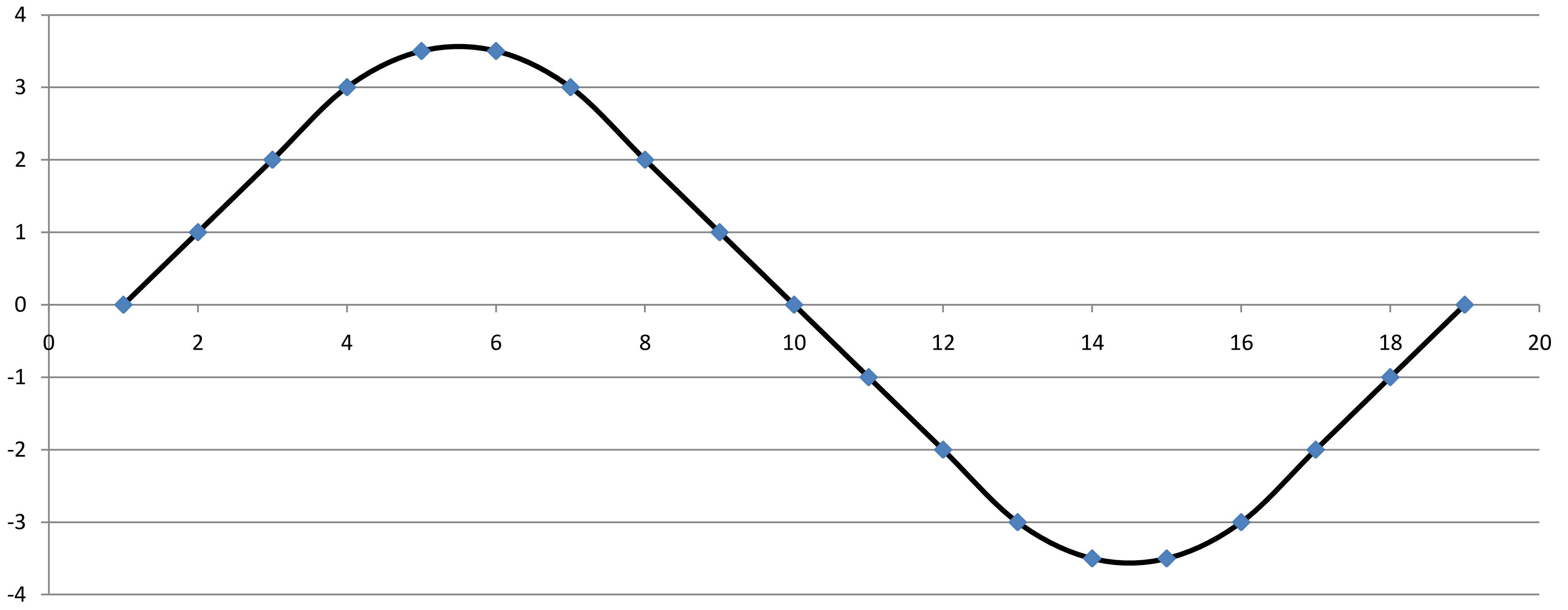
A vector is a row or column of numbers.



And, what do we do with a vector?

A vector can be used to represent a wave.

0	1	2	3	3.5	3.5	3	2	1	0	-1	-2	-3	-3.5	-3.5	-3	-2	-1	0



An array is a more general concept than a matrix.

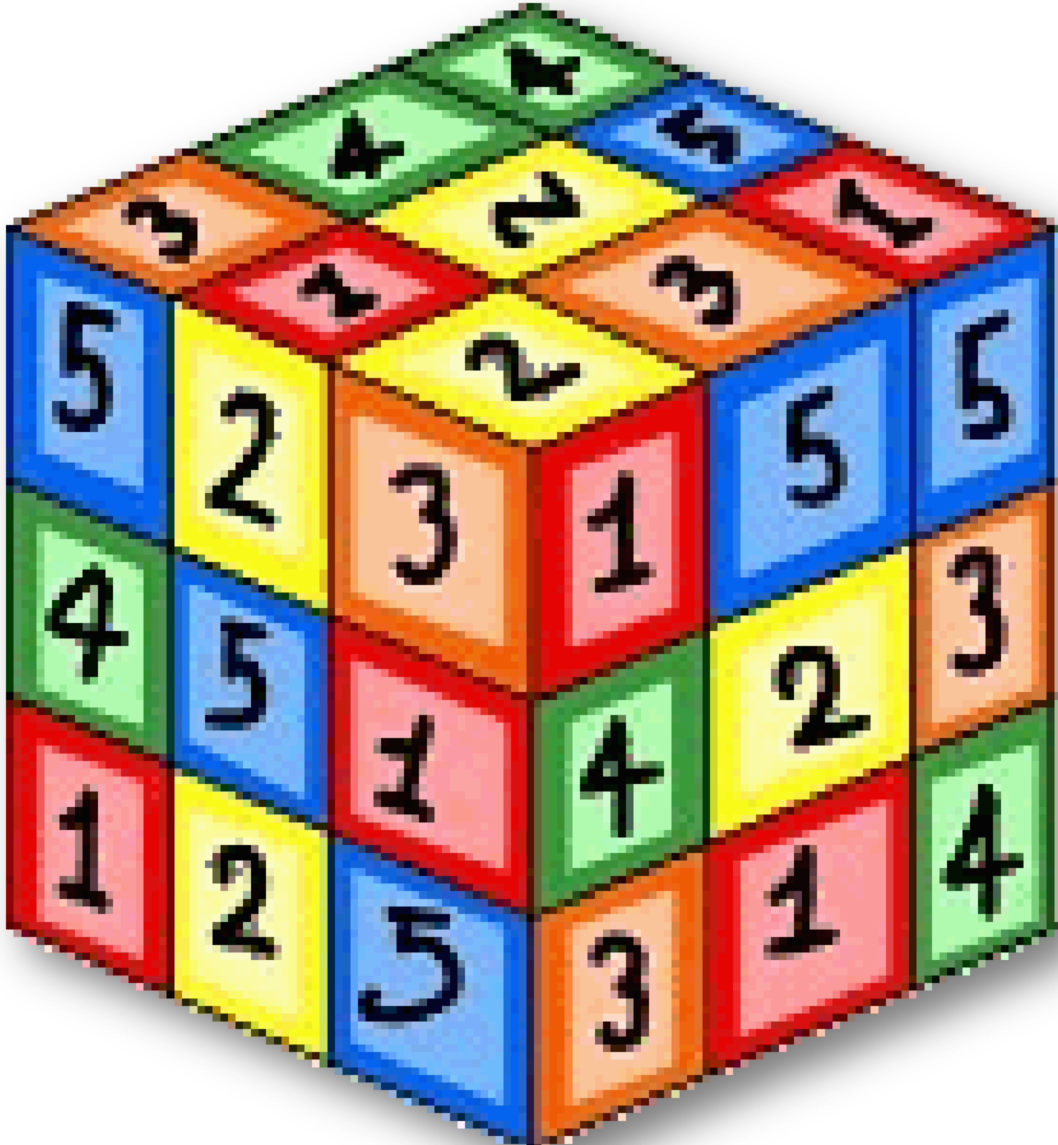
(Specifically, an **array** is a data structure consisting of a group of elements that are accessed by indexing.)

For example,
a matrix is a 2D array.

…and a vector is a
1D array.

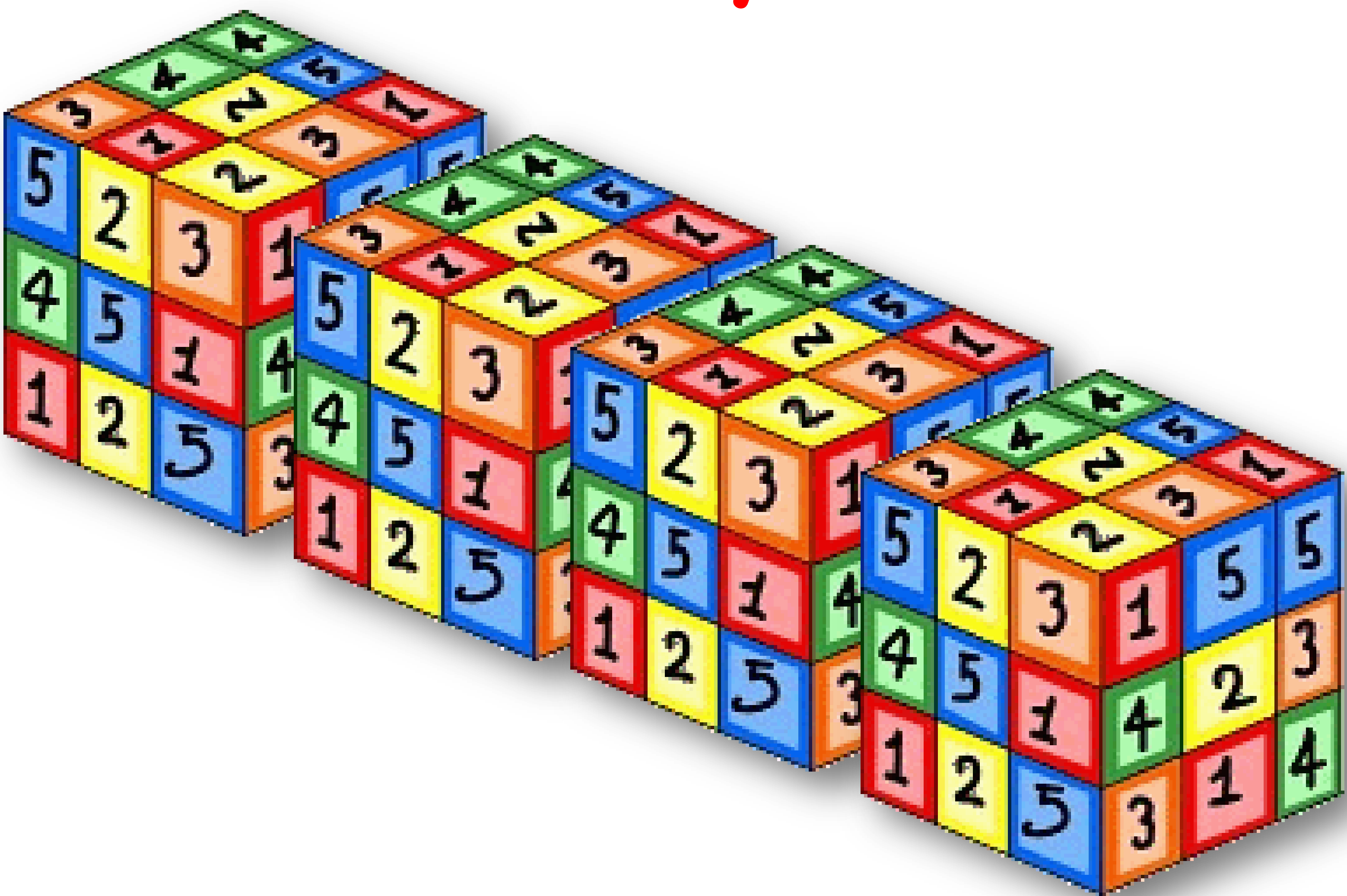
An array can have any number of dimensions.

In image processing, we often work with **3D arrays**,



because a 3D array can represent a volume.

And we work with **4D arrays**:



because a 4D array can represent, for example, a volume through time.

Scalars are individual numbers:

Let's Review

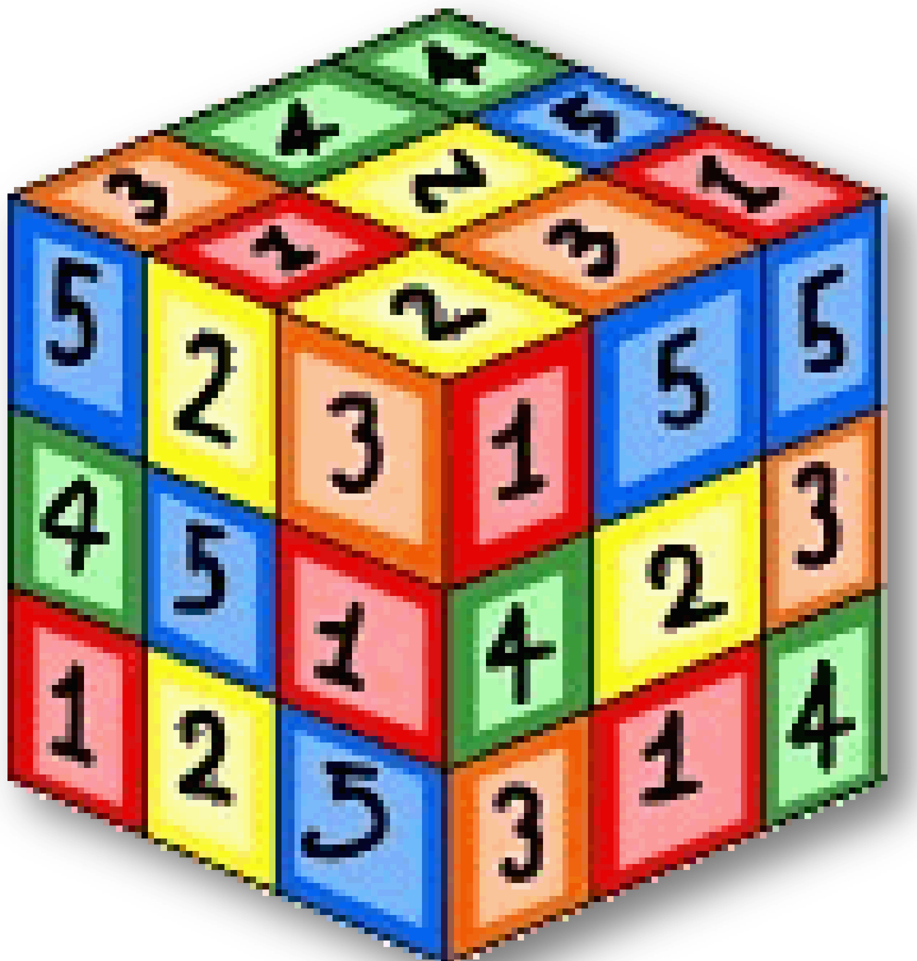
Scalar

Vector

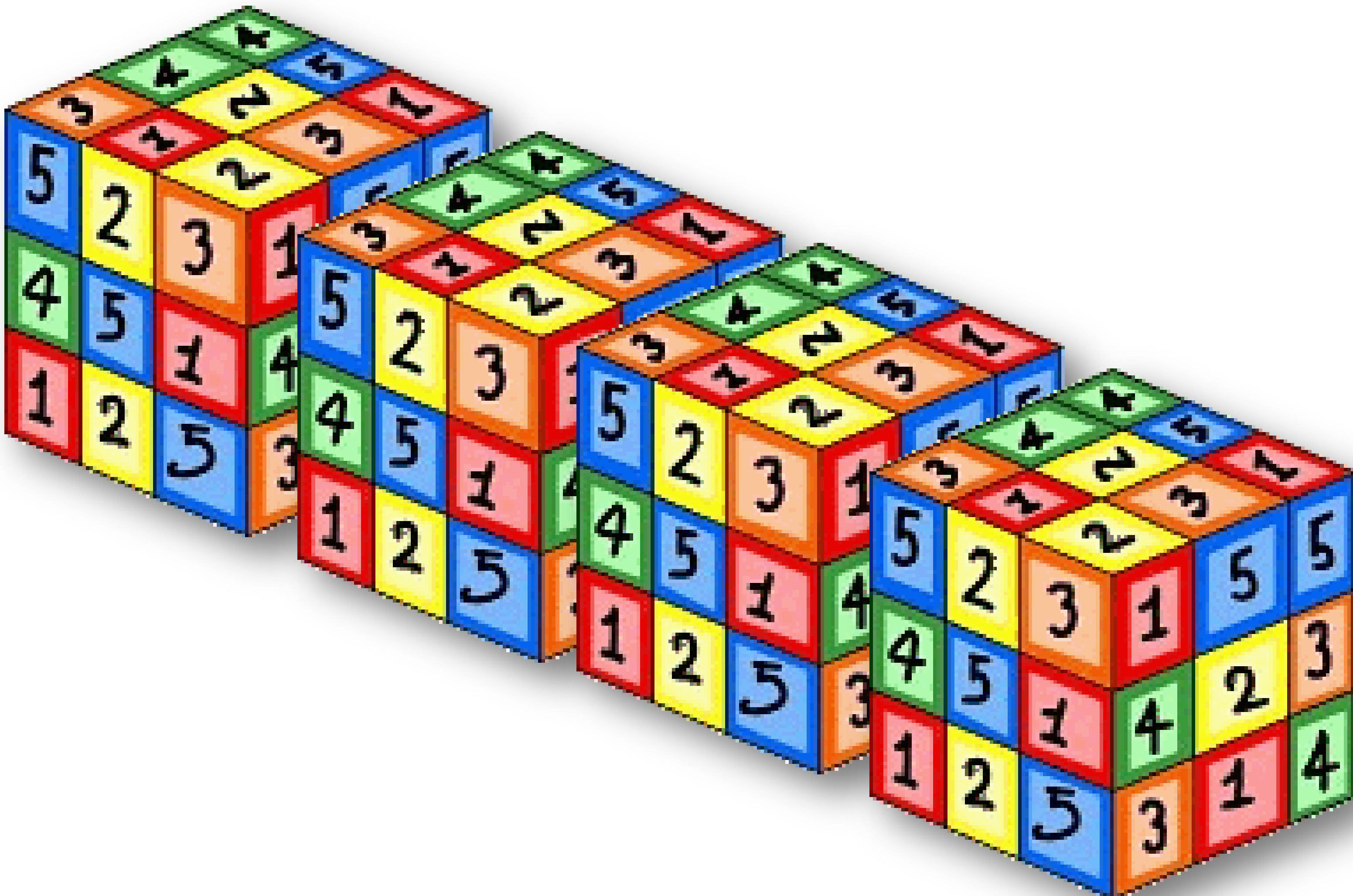
Matrix

1	2	3
3	5	6
7	9	2

3D Array



4D Array



[redacted]

Matrix arithmetic operations are defined by the rules of linear algebra.

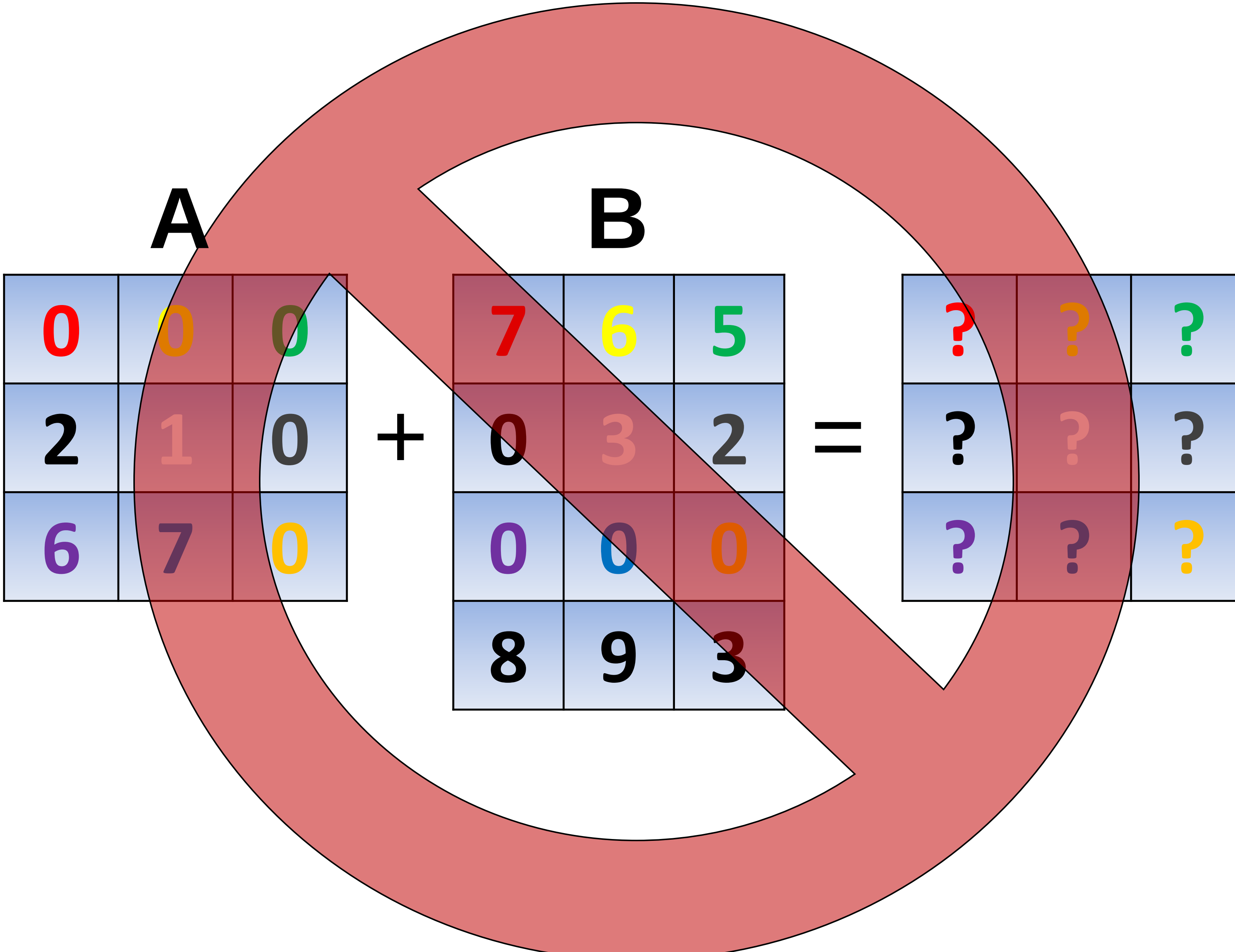
Array arithmetic operations are carried out element by element, and can be used with multidimensional arrays.

All the major image processing programs have some kind of "image math" that uses array operations.

Array Addition:
Array A + Array B
adds A and B,
element by element.

$$\overset{A}{\begin{bmatrix} 0 & 0 & 0 \\ 2 & 1 & 0 \\ 6 & 7 & 0 \end{bmatrix}} + \overset{B}{\begin{bmatrix} 7 & 6 & 5 \\ 0 & 3 & 2 \\ 0 & 0 & 0 \end{bmatrix}} = \overset{C}{\begin{bmatrix} 7 & 6 & 5 \\ 2 & 4 & 2 \\ 6 & 7 & 0 \end{bmatrix}}$$

A and B must be the same size,


A
B
0 0 0
2 1 0
6 7 0
+
7 6 5
0 3 2
0 0 0
8 9 3
=
? ? ?
? ? ?
? ? ?

…unless one is a scalar.

A scalar can be added to a matrix of any size.

$$\overset{A}{\begin{bmatrix}0 & 0 & 0\\ 2 & 1 & 0\\ 6 & 7 & 0\end{bmatrix}} + \overset{B}{4} = \overset{C}{\begin{bmatrix}4 & 4 & 4\\ 6 & 5 & 4\\ 10 & 11 & 4\end{bmatrix}}$$

Array Subtraction: Array A - Array B subtracts B from A, element by element.

$$\overset{A}{\begin{bmatrix}0&0&0\\2&1&0\\6&7&0\end{bmatrix}} - \overset{B}{\begin{bmatrix}7&6&5\\0&3&2\\0&0&0\end{bmatrix}} = \overset{C}{\begin{bmatrix}-7&-6&-5\\2&-2&-2\\6&7&0\end{bmatrix}}$$

A and B must be the same size,

…unless one is a scalar.

And a scalar can be subtracted from a matrix of any size.

$$\begin{array}{ccccc} A & & B & & C \\ \begin{bmatrix} 0 & 0 & 0 \\ 2 & 1 & 0 \\ 6 & 7 & 0 \end{bmatrix} & - & 4 & = & \begin{bmatrix} -4 & -4 & -4 \\ -2 & -3 & -4 \\ 2 & 3 & -4 \end{bmatrix} \end{array}$$

Array Multiplication: Array A .* Array B multiplies A and B, element by element.

$$\underset{A}{\begin{bmatrix} 0 & 0 & 0 \\ 2 & 1 & 0 \\ 6 & 7 & 0 \end{bmatrix}} \;.*\; \underset{B}{\begin{bmatrix} 7 & 6 & 5 \\ 0 & 3 & 2 \\ 0 & 0 & 0 \end{bmatrix}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

A and B must be the same size, unless one is a scalar.

A matrix of any size can be multiplied by a scalar.

$$\underset{A}{\begin{bmatrix} 0 & 0 & 0 \\ 2 & 1 & 0 \\ 6 & 7 & 0 \end{bmatrix}} \;.*\; \underset{B}{4} \;=\; \underset{C}{\begin{bmatrix} 0 & 0 & 0 \\ 8 & 4 & 0 \\ 24 & 28 & 0 \end{bmatrix}}$$

Array Multiplication is sometimes called the '**dot product**' to differentiate it from matrix multiplication.

Matrix multiplication (sometimes called the "**cross-product**")

multiplies each
element in *A*,
by every element in B

Cross Product

3
4
5

A

*

1	2	6

B

=

3*1	3*2	3*6
4*1	4*2	4*6
5*1	5*2	5*6

i.e.,

3	6	18
4	8	24
5	10	30

Array Division is element by element division of Array A by Array B

$$\overset{\textbf{A}}{\begin{bmatrix} 2 & 4 & 6 \\ 1 & 3 & 2 \\ 6 & 8 & 0 \end{bmatrix}} \;./\; \overset{\textbf{B}}{\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 2 \\ 4 & 1 & 6 \end{bmatrix}} = \begin{bmatrix} 2 & 2 & 2 \\ 0.5 & 1 & 1 \\ 1.5 & 8 & 0 \end{bmatrix}$$

Again, A and B must have the same size, unless B is a scalar.

$$\overset{A}{\begin{bmatrix} 2 & 4 & 6 \\ 2 & 1 & 0 \\ 6 & 7 & 3 \end{bmatrix}} \;./\; \overset{B}{2} = \overset{C}{\begin{bmatrix} 1 & 2 & 3 \\ 1 & 0.5 & 0 \\ 3 & 3.5 & 1.5 \end{bmatrix}}$$

By far, the most useful array operations in day to day imaging are…

addition,

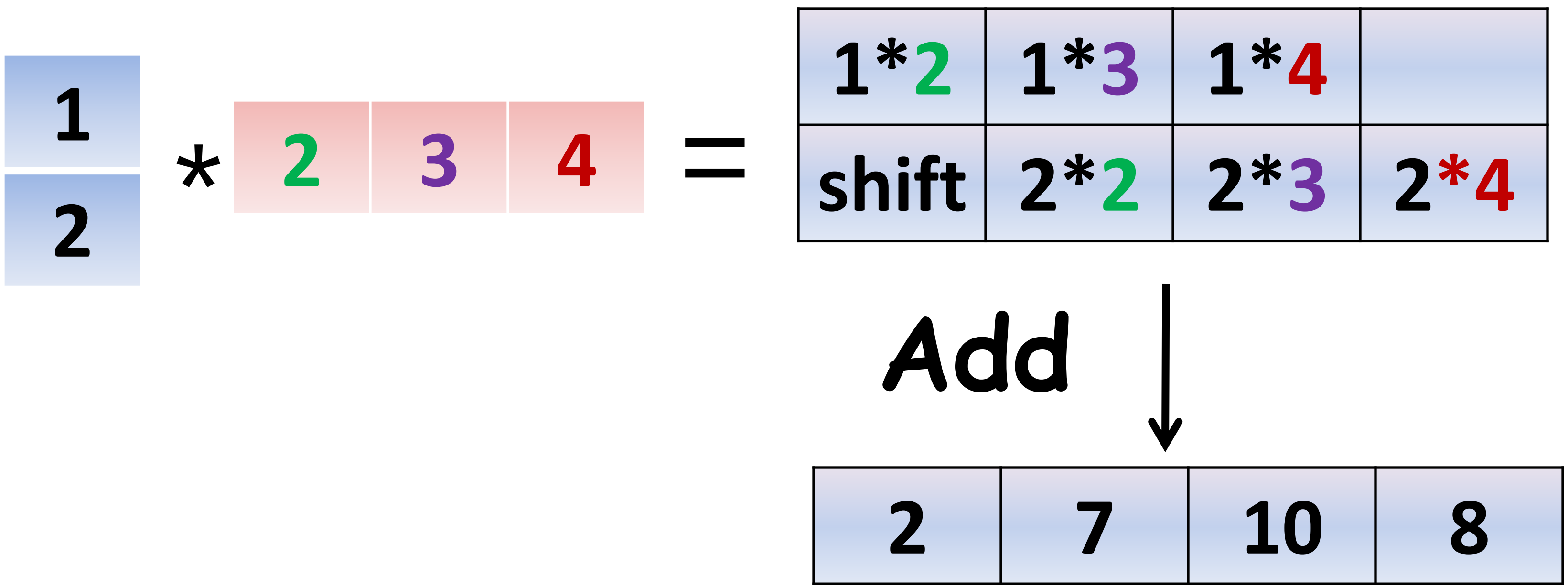
subtraction, &

dot-product multiplication

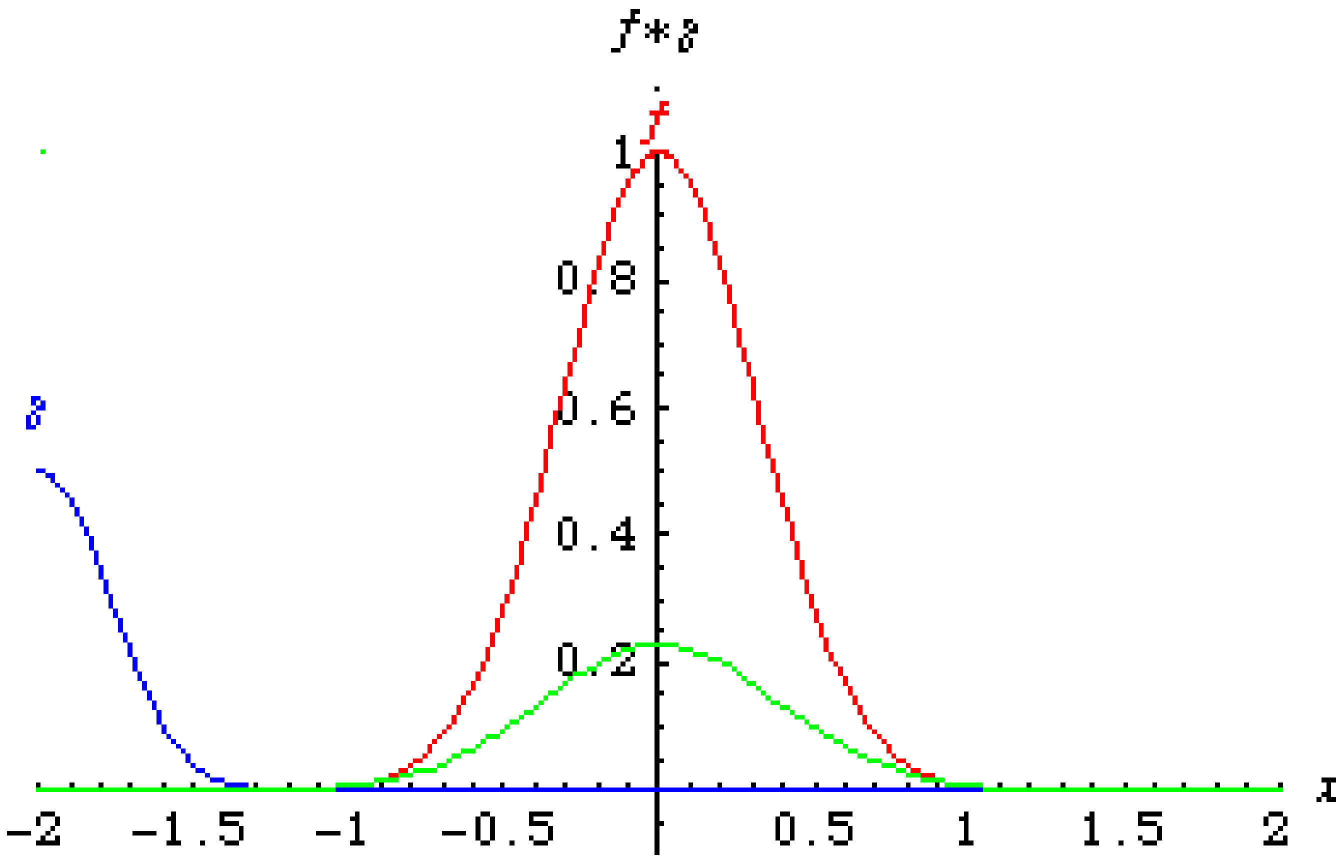
Another important matrix operation to be familiar with is **convolution**

Convolution uses multiplication, shifting and addition to blend two functions (or waves) into a third.

Convolution

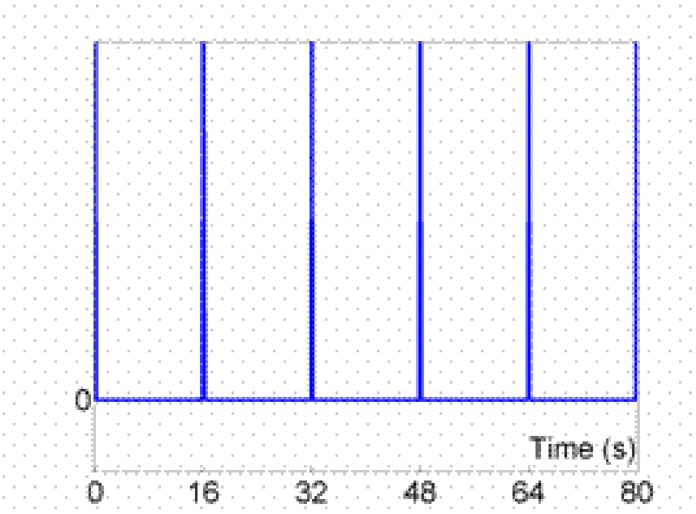


Convolution of Gaussians: green is the result of convolving red and blue



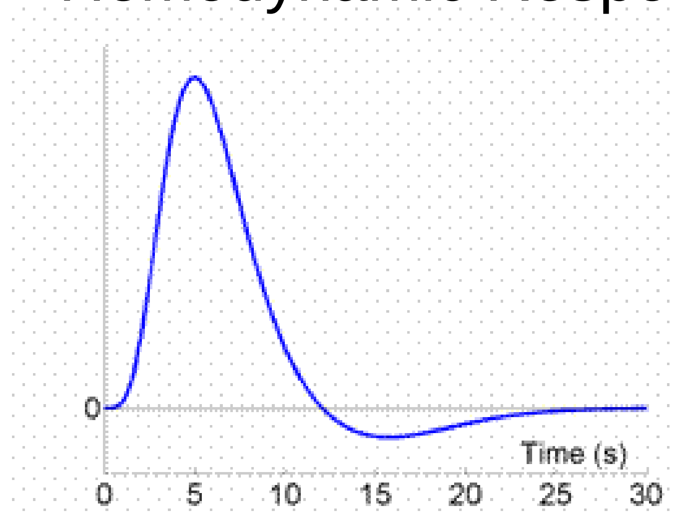
Convolution is used extensively by neuroimaging software, e.g., Convolution combines fmri stimuli with the hemodynamic response to model the expected waveform

Stimulus



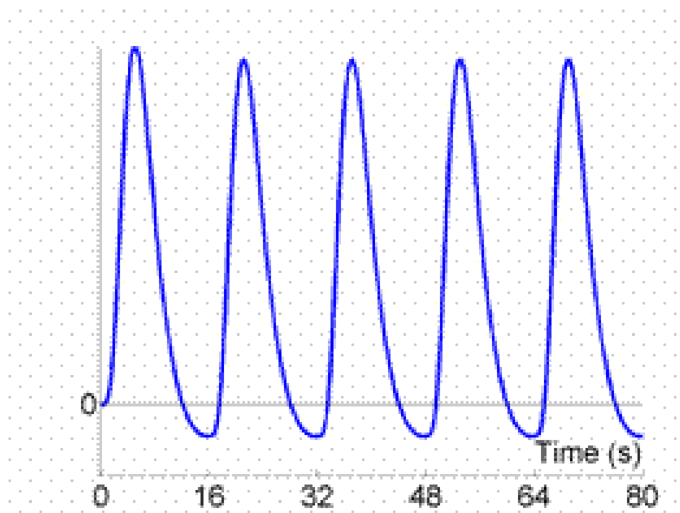
$\otimes$

Hemodynamic Response



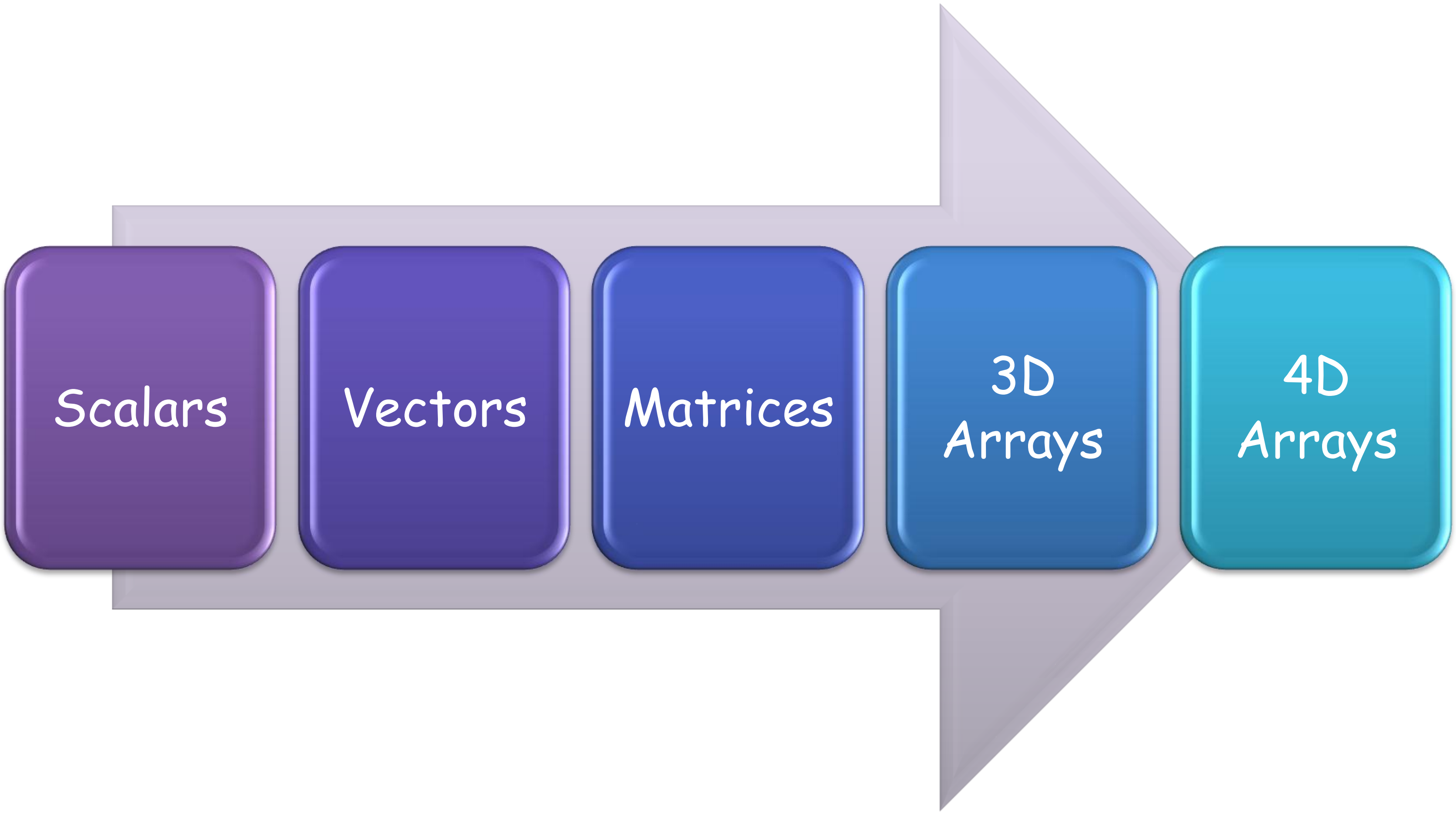
$=$

Convolved model



Summary

We examined data structures of increasing complexity:


Scalars
Vectors
Matrices
3D Arrays
4D Arrays

We distinguished array arithmetic (which works element by element) from matrix arithmetic (based on linear algebra).

We examined several array operations

Addition

Subtraction

Multiplication

Division

We also examined matrix multiplication

and convolution

Scalar

Matrix

3D Array

4D Array

Array arithmetic

Matrix arithmetic

Dot product

Cross product

Convolution

Cross Product

3
4
5

A

*

1	2	6

B

=

3*1	3*2	3*6
4*1	4*2	4*6
5*1	5*2	5*6

i.e.,

3	6	18
4	8	24
5	10	30

Convolution

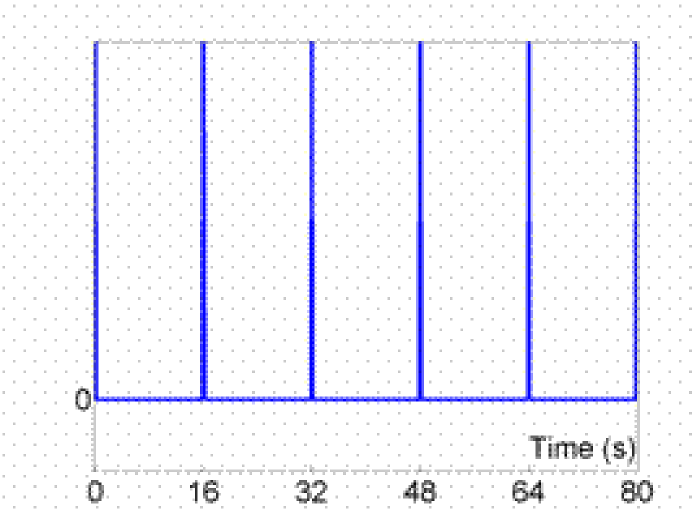
$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 4 \end{bmatrix} =$$

1*2	1*3	1*4	
shift	2*2	2*3	2*4

Add ↓

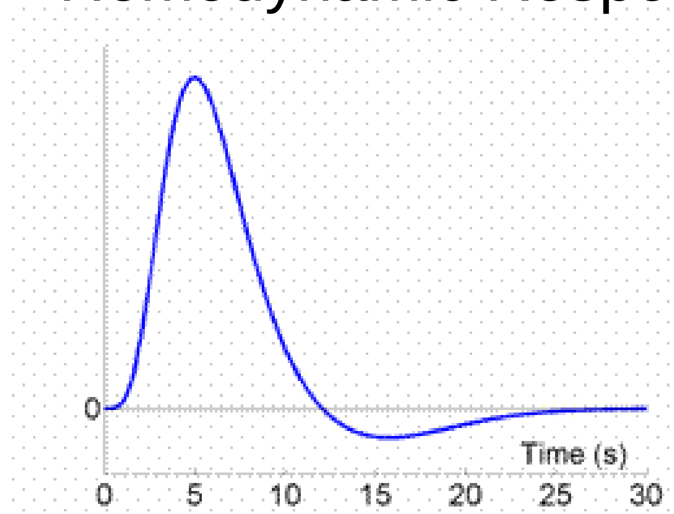
2	7	10	8

Stimulus



$\otimes$

Hemodynamic Response



Convolved model

$=$