CARMINE-EMANUELE CELLA

INTRODUCTION TO GEOMETRIC SIGNAL THEORY

MUSIC 159

BASIC IDEA

We will interpret all signal theory using a single mathematical concept: **inner product**

This will be difficult at the beginning, because it requires to understand some results in linear algebra, but later on it will simplify all the other concepts in signal theory

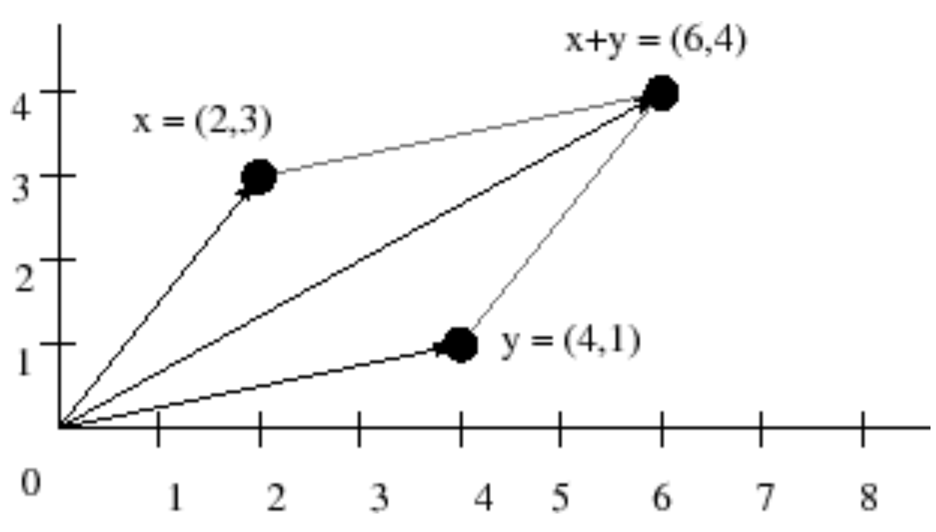
SIGNALS AS VECTORS

Discrete signals x of length n can be thought as a multidimensional vectors in $\mathbb{R}^n$ or $\mathbb{C}^n$. As such, they can be added with other vectors and multiplied by scalars as follows:

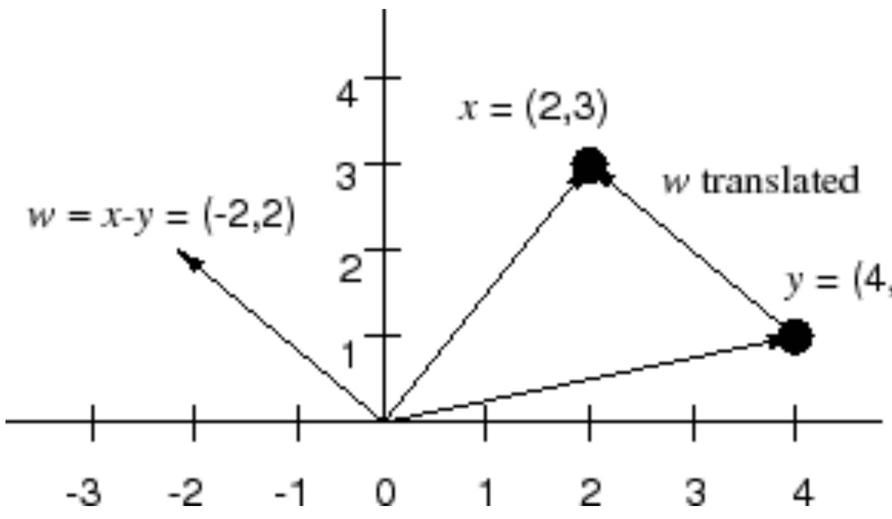
$$a_1 v_1 + a_2 v_2 \ldots + a_n v_n \tag{1}$$

where a_n are scalars and v_n are vectors.

LINEAR OPERATIONS AND VECTOR SPACES



Addition



Subtraction

$$\underline{y} = \alpha_1 \underline{x}_1 + \alpha_2 \underline{x}_2 + \cdots + \alpha_M \underline{x}_M$$

Linear combination

(sum of multiplications between scalars and vectors)

A set of vectors may be called a linear vector space if it is closed under linear combinations

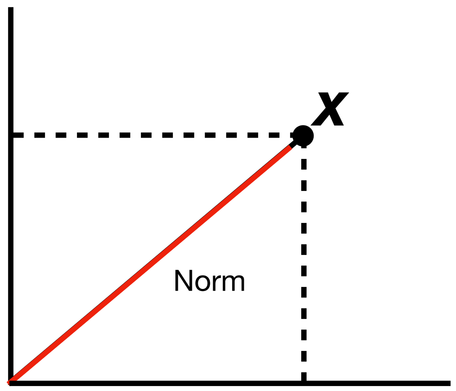
A vector space is a set of vectors that contains also the zero vector

Images from J. Smith, The mathematics of the DT

METRICS AND BANACH SPACES

On vector spaces can be computed several metrics:

- *mean*: $\mathcal{M} = \frac{1}{N} \sum_{n=0}^{N-1} x_n$;
- *energy*: $\mathcal{E} = \sum_{n=0}^{N-1} |x_n|^2$;
- *power*: $\mathcal{P} = \frac{\mathcal{E}}{N} = \frac{1}{N} \sum_{n=0}^{N-1} |x_n|^2$;
- L_2**-norm**: $\mathcal{N} = \sqrt{\mathcal{E}} = \sqrt{\sum_{n=0}^{N-1} |x_n|^2}$.



The latter metric, the norm, is often indicated as $\|x\|_2$ and represents the *length* of the vector in a space. When a vector space has a defined norm, it is called a *Banach space*. The L_2-norm is said to be *contractive*, that is:

$$\|x + y\|_2 \leq \|x\|_2 + \|y\|_2 \quad (2)$$

and it can be generalized to any order: $L_p = \|x\|_p = \left(\sum_{n=0}^{N-1} |x_n|^p\right)^{\frac{1}{p}}$.

This is also called normed space

INNER (SCALAR) PRODUCT AND HILBERT SPACES

A very important operation, called inner product, can defined on a Banach space as follows:

Remember that scalar product is your friend!

$$\langle x, y \rangle = \sum_{n=0}^{N-1} x_n \overline{y_n} \tag{3}$$

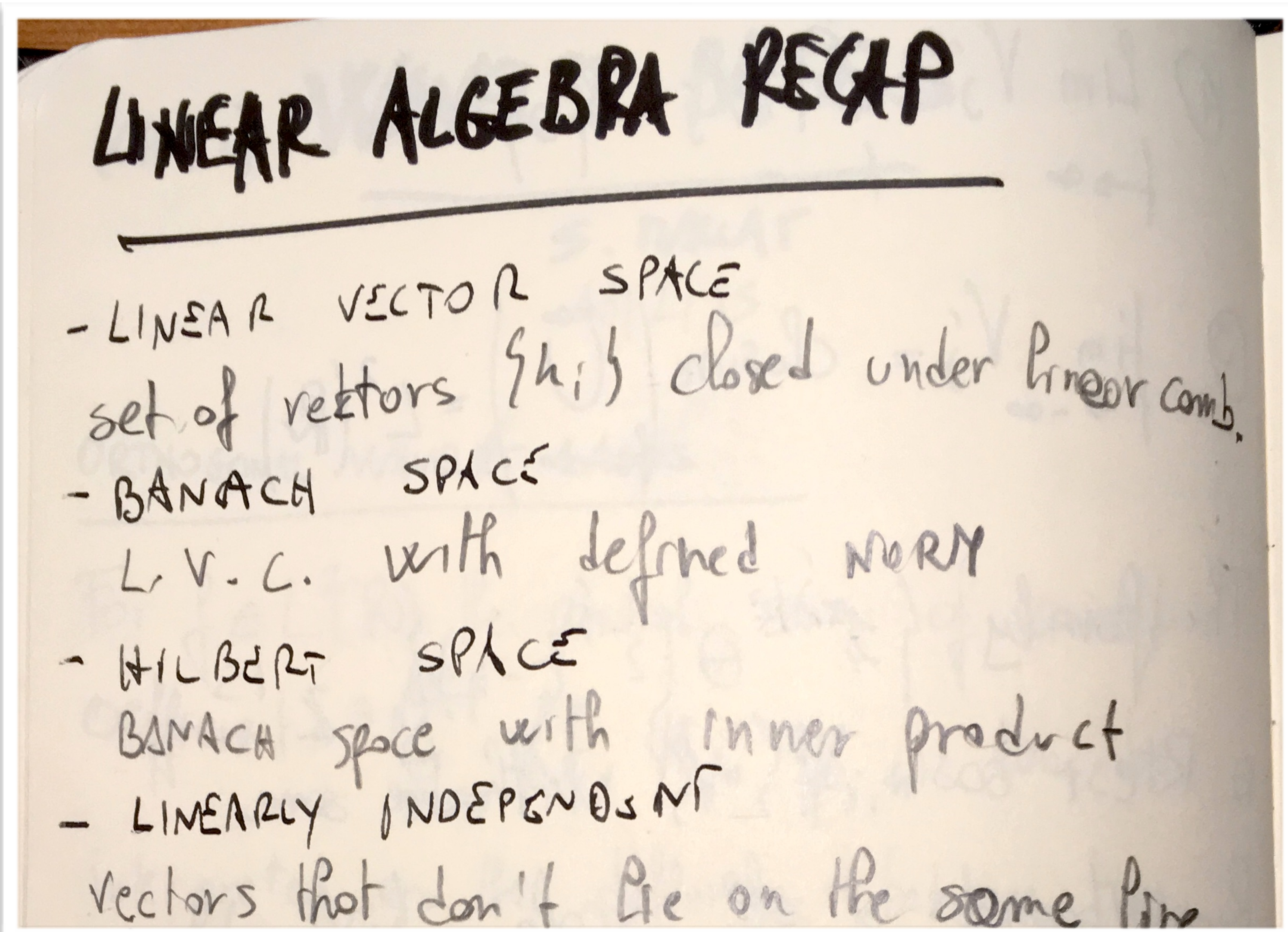
where $\overline{x}$ is the conjugate of x. A Banach space with a defined inner product is called a *Hilbert space*. The specific form of inner product shown in equation 3 is induced from the energy by taking the inner product of a vector with itself:

$$\langle x, x \rangle = \sum_{n=0}^{N-1} x_n \overline{x_n} = \sum_{n=0}^{N-1} |x_n|^2 = \|x\|_2^2. \tag{4}$$

Properties. The inner product has several important properties, among which:

- *Cauchy-Schwarz inequality*: $|\langle x, y \rangle| \leq \|x\| \cdot \|y\|$;
- *vector cosine*: $\frac{\langle x, y \rangle}{\|x\| \cdot \|y\|} \leq 1 = cos(\theta)$ where θ is the angle between the two vectors.

A SHORT SUMMARY



LINEAR ALGEBRA RECAP

- LINEAR VECTOR SPACE

set of vectors $\{h_i\}$ closed under linear comb.

- BANACH SPACE

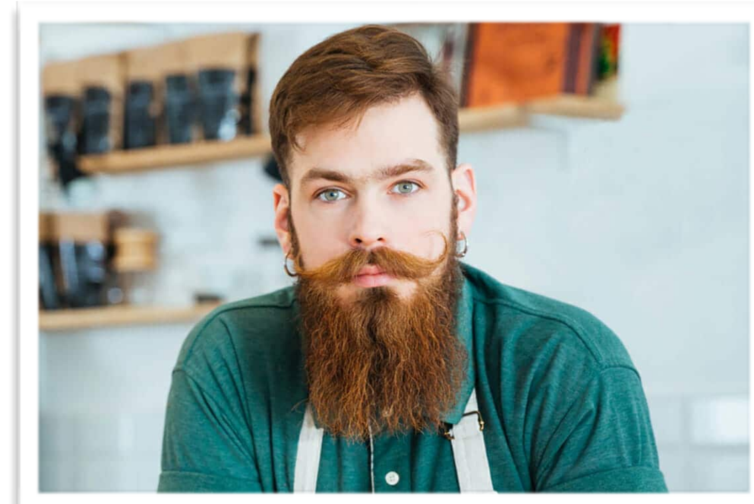
L.V.C. with defined NORM

- HILBERT SPACE

BANACH space with inner product

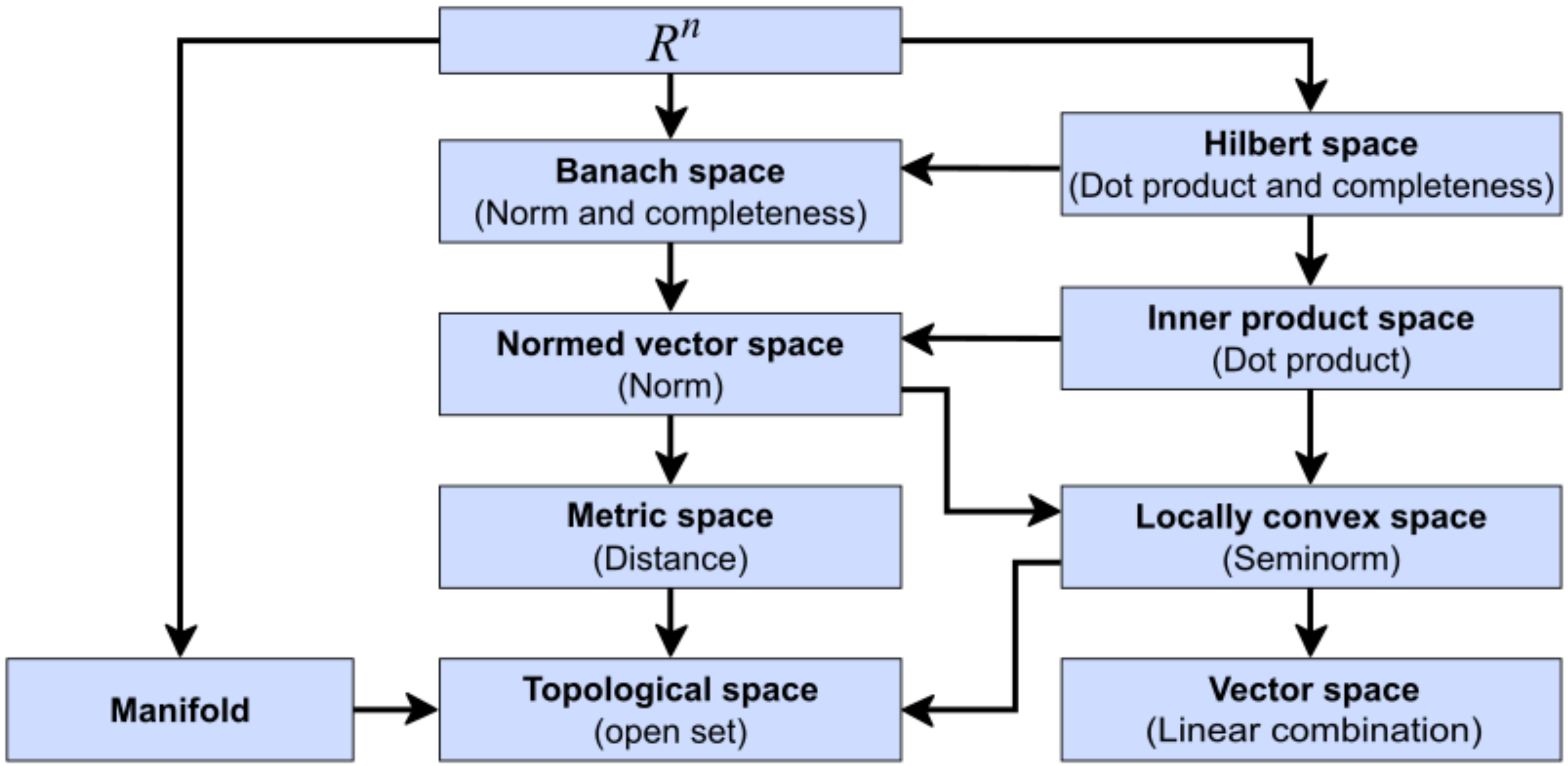
- LINEARLY INDEPENDENT

vectors that don't lie on the same line



Hand-written summaries are more hipster...

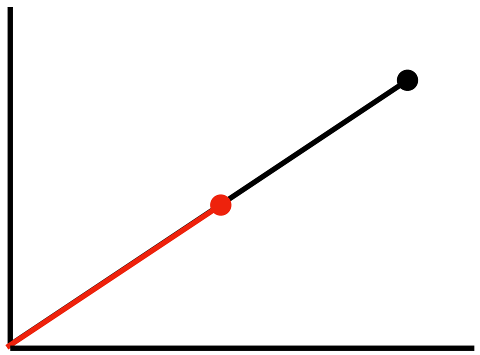
A TAXONOMY (JUST FOR FUN)



What's fun dude???

COLLINEARITY AND ORTHOGONALITY

Two vectors are said to be *collinear* if they both lie on the line that connects them to the centre (zero vector)

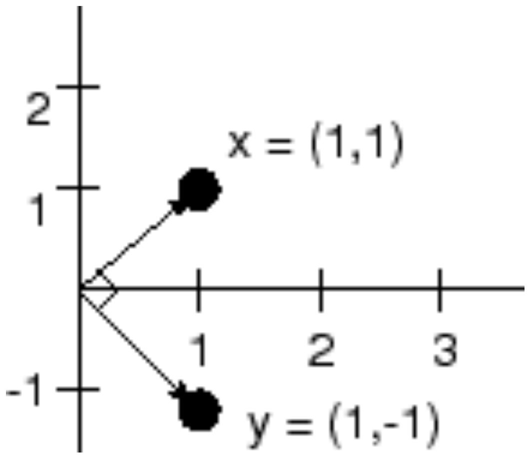


Collinear

Two vectors are said to be *orthogonal* (indicated as $x \perp y$) if their inner product is zero:

$$x \perp y \equiv \langle x, y \rangle = 0. \quad (5)$$

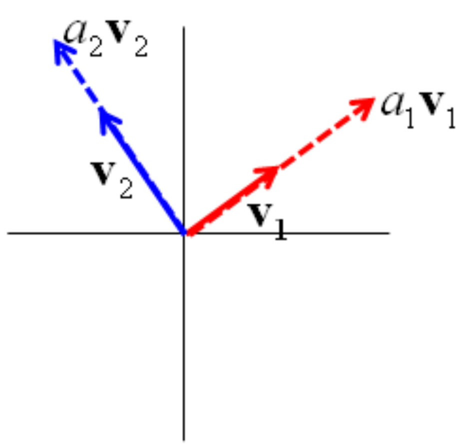
If two vectors are orthogonal and their norm is equal to 1, they are said *orthonormal.*



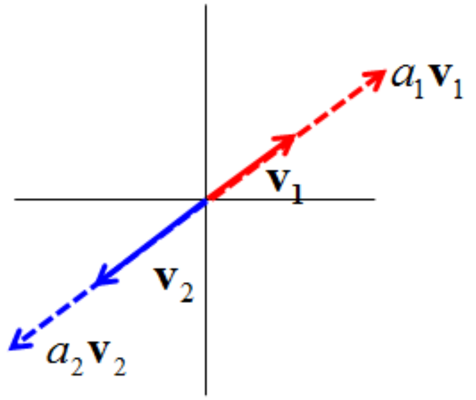
Orthogonal

LINEAR INDEPENDENCE

A set of vectors is **linearly independent** if none of them can be expressed as a linear combination of the others in the set. What this means intuitively is that they must point in *different directions* in space.



Conversely, they are said **linearly dependent** if one is a linear combination of the others.

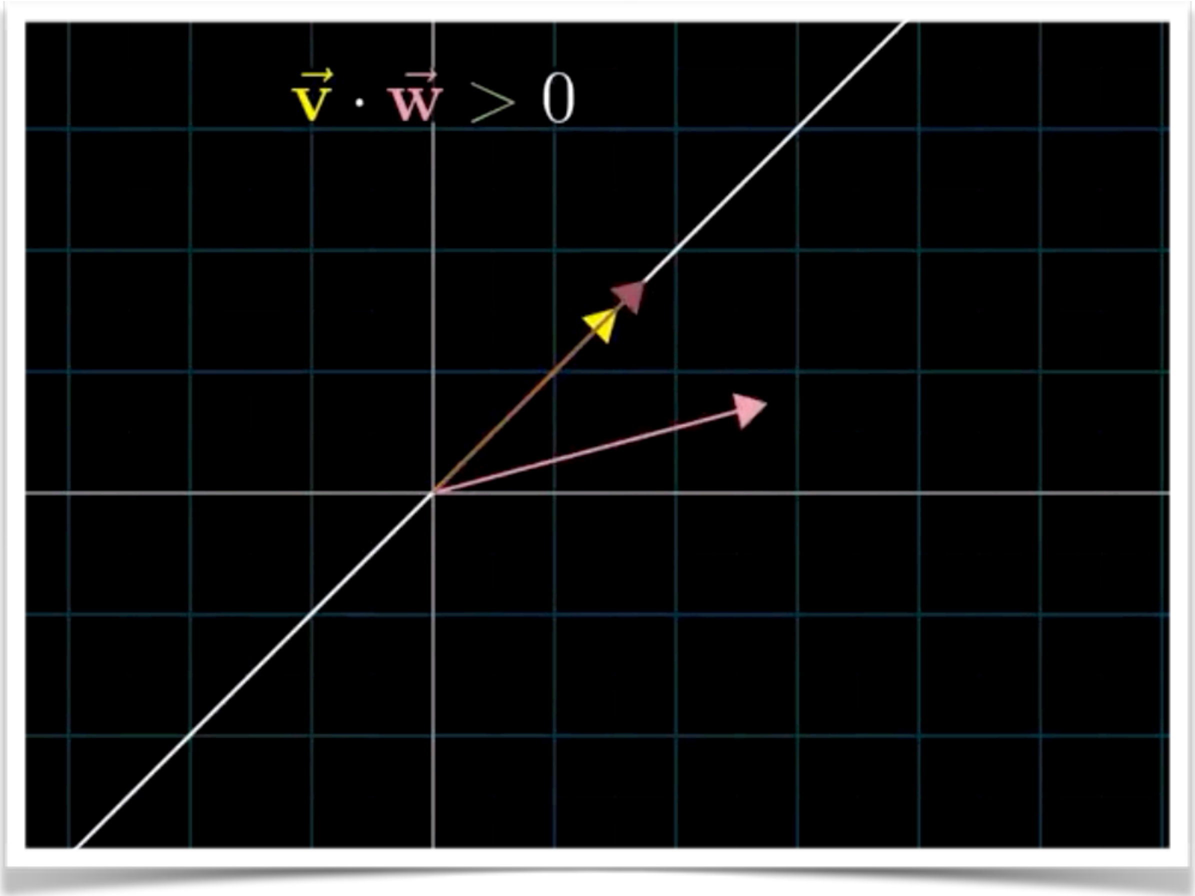


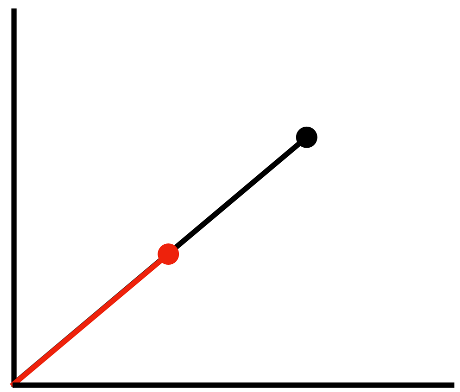
Example

The first three vectors are linearly independent; but the fourth vector equals **nine** times the first plus **five** times the second plus **four** times the third, so the four vectors together are linearly dependent.

$$\underbrace{\overbrace{\begin{bmatrix}0\\0\\1\end{bmatrix},\begin{bmatrix}0\\2\\-2\end{bmatrix},\begin{bmatrix}1\\-2\\1\end{bmatrix}}^{\text{independent}},\begin{bmatrix}4\\2\\3\end{bmatrix}}_{\text{dependent}}$$

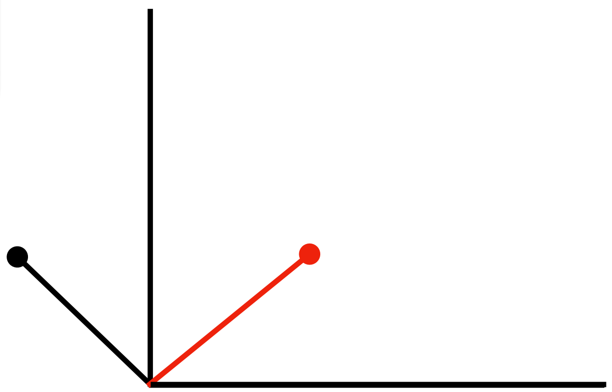
MEANING OF INNER PRODUCT





Collinear
Inner product high
(think about norm)

Inner product represents the *similarity* *between two vectors*



Orthogonal
Inner product zero

Clip from https://www.youtube.com/watch?v=LyGKycYT2v0&t=182s

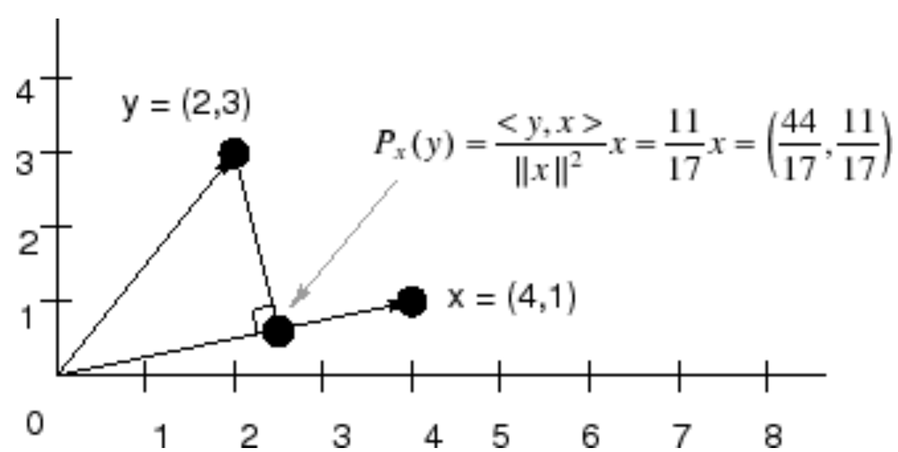
PROJECTIONS (1)

One of the most important applications of inner product is to project one vector over another. The projection of x on y is defined as:

$$\mathfrak{P}_x(y) = \boxed{\frac{\langle x, y\rangle}{\|x\|^2}} \cdot x \tag{6}$$

where the ratio between the inner product and the squared norm of x is called *coefficient of projection*. Important examples of projection will be shown in the following sections.

The coefficient of projection has an inner product at its core!

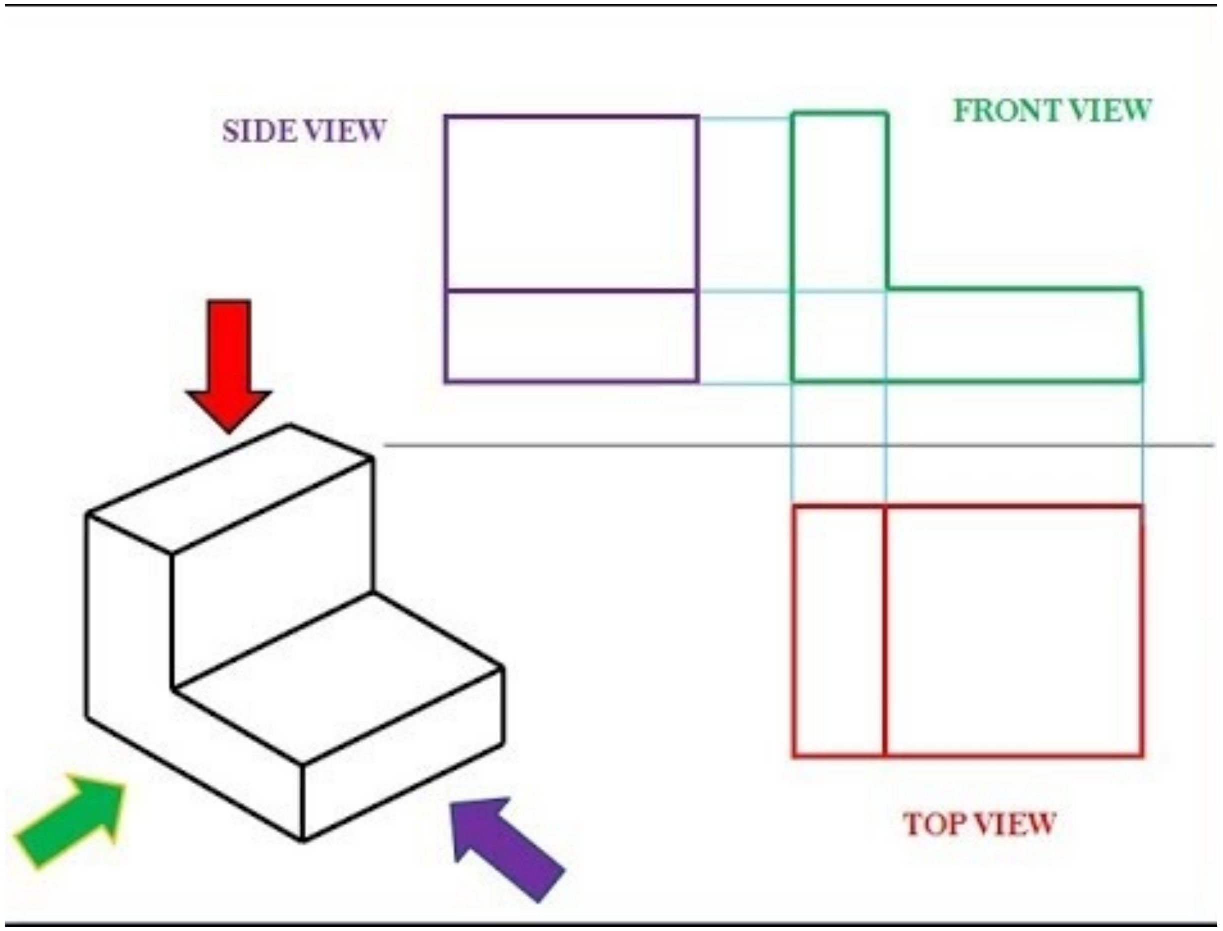


$$\mathbf{P}_x(y) \triangleq \frac{\langle y, x\rangle}{\|x\|^2}x = \frac{(2\cdot\overline{4} + 3\cdot\overline{1})}{4^2+1^2}x = \frac{11}{17}x = \left[\frac{44}{17}, \frac{11}{17}\right].$$

The basic idea of orthogonal projection of y onto x is to drop a perpendicular to define a new vector along x.

Image from J. Smith, The mathematics of the DT

PROJECTIONS (2)



A projection selects a different point of view on objects and highlights a specific property

RECONSTRUCTION FROM PROJECTIONS

Under specific conditions, a vector can
be reconstructed from some projections by means of linear combination.

Let $e_0(1,0)$ and $e_1(0,1)$ be to perpendicular vectors such as $\langle e_0, e_1 \rangle = 0$ and let x be a distinct vector in $\mathbb{R}^2$. The projection of x on e_0 is given by:

$$\begin{aligned}\mathfrak{P}_{e_0}(x) &= \frac{\langle e_0, x\rangle}{\|e_0\|^2} \cdot e_0 = \langle e_0, x\rangle \cdot e_0 \\ &= (x_0 \cdot \overline{1} + x_1 \cdot \overline{0}) = (x_0, 0)\end{aligned}$$

and, in the same way, the projection on e_1 is given by $\mathfrak{P}_{e_1}(x) = (0, x_1)$.

It is indeed possible to *recover* x by summing the computed projections:

$$\begin{aligned}x = \mathfrak{P}_{e_0}(x) + \mathfrak{P}_{e_1}(x) &= x_0 \cdot e_0 + x_1 \cdot e_1 \\ &= x_0 \cdot (1,0) + x_1 \cdot (0,1) = (x_0, x_1).\end{aligned}$$

It is important to remark that this reconstruction only works if the vectors on which the projections are done are pairwise orthogonal

RECONSTRUCTION – EXAMPLE

$$\underline{e}_0 \triangleq [1,0],$$
$$\underline{e}_1 \triangleq [0,1].$$

$\rightarrow$ $<e_0, e_1> = 0$, hence they are orthogonal

$x\ (x_0, x_1)$

Projection of x onto e_0

$$\begin{aligned}\mathbf{P}_{\underline{e}_0}(x) &\triangleq \frac{\langle x, \underline{e}_0\rangle}{\|\underline{e}_0\|^2}\underline{e}_0 \\ &= \langle x, \underline{e}_0\rangle\, \underline{e}_0 = \langle [x_0, x_1], [1,0]\rangle\, \underline{e}_0 = (x_0 \cdot \overline{1} + x_1 \cdot \overline{0})\underline{e}_0 = x_0\underline{e}_0 \\ &= [x_0, 0].\end{aligned}$$

Projection of x onto e_1

$$\begin{aligned}\mathbf{P}_{\underline{e}_1}(x) &\triangleq \frac{\langle x, \underline{e}_1\rangle}{\|\underline{e}_1\|^2}\underline{e}_1 \\ &= \langle x, \underline{e}_1\rangle\, \underline{e}_1 = \langle [x_0, x_1], [0,1]\rangle\, \underline{e}_1 = (x_0 \cdot \overline{0} + x_1 \cdot \overline{1})\underline{e}_1 = x_1\underline{e}_1 \\ &= [0, x_1].\end{aligned}$$

Reconstruction of x

$$x = \mathbf{P}_{\underline{e}_0}(x) + \mathbf{P}_{\underline{e}_1}(x) = x_0\underline{e}_0 + x_1\underline{e}_1 \triangleq x_0 \cdot [1,0] + x_1 \cdot [0,1] = (x_0, x_1)$$

PROJECTION ONTO NON-ORTHOGONAL VECTORS

$$\underline{s}_0 \triangleq [1, 1]$$
$$\underline{s}_1 \triangleq [0, 1]$$

→ are linearly independent, but not orthogonal

Projections

$$\mathbf{P}_{\underline{s}_0}(x) = \frac{x_0 + x_1}{2}\underline{s}_0$$
$$\mathbf{P}_{\underline{s}_1}(x) = x_1 \underline{s}_1.$$

Reconstruction

$$\begin{aligned}
\mathbf{P}_{\underline{s}_0}(x) + \mathbf{P}_{\underline{s}_1}(x) &= \frac{x_0 + x_1}{2}\underline{s}_0 + x_1 \underline{s}_1 \\
&\triangleq \frac{x_0 + x_1}{2}(1, 1) + x_1 \cdot (0, 1) \\
&= \left(\frac{x_0 + x_1}{2}, \frac{x_0 + 3x_1}{2}\right) \\
&\neq x.
\end{aligned}$$

Even if the vectors are linearly independent, the sum of projections onto them does not reconstruct the original vector.

The sum of projections onto a set of vectors will reconstruct the original vector only when the vector set is *orthogonal*.

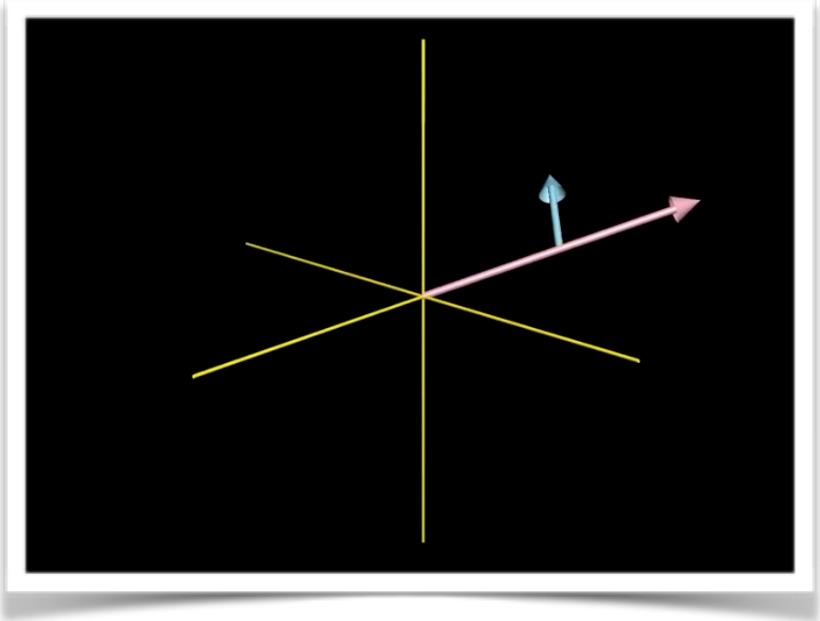
One can apply a process called Gram-Schmidt orthogonalization to any linearly independent vectors to form an orthogonal set

SPAN AND BASIS OF A VECTOR SPACE

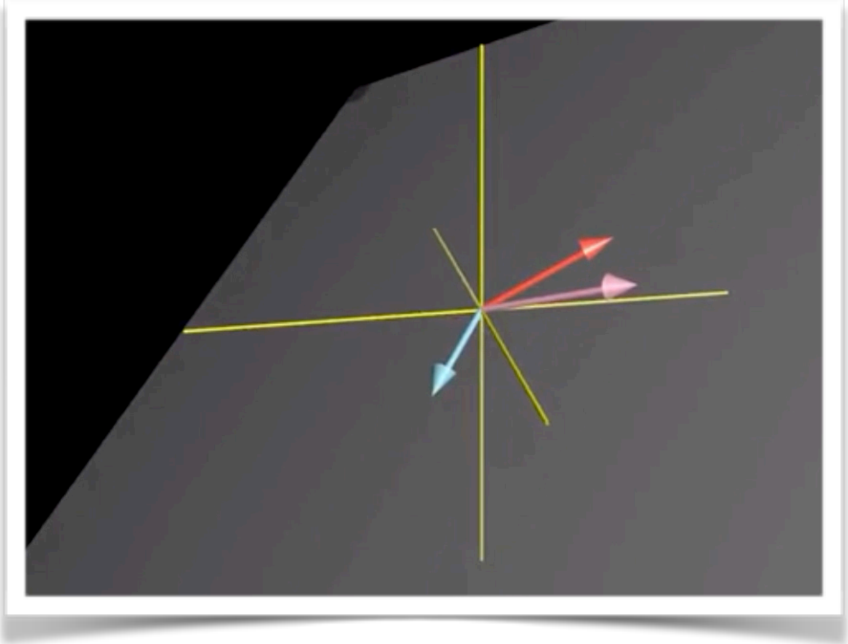
TIP: for a set of vector to be a basis of a n-dimensional space, it needs to have 3 vectors that are linearly independent

Basis. The subspace covered by all linear combinations of a set of vectors $\{s_0, \ldots, s_n\}$ is called *span*. If the set of vectors are linearly independent than the span is called *basis* of the vector space. It is easy to show that in a space in $\mathbb{R}^d$ there are d vectors in the basis for that space. Clearly, a vector can be reconstructed with a linear combination from its projections on another set of vectors if and only if the set used is a basis.

Span example (2 vectors in 3D)

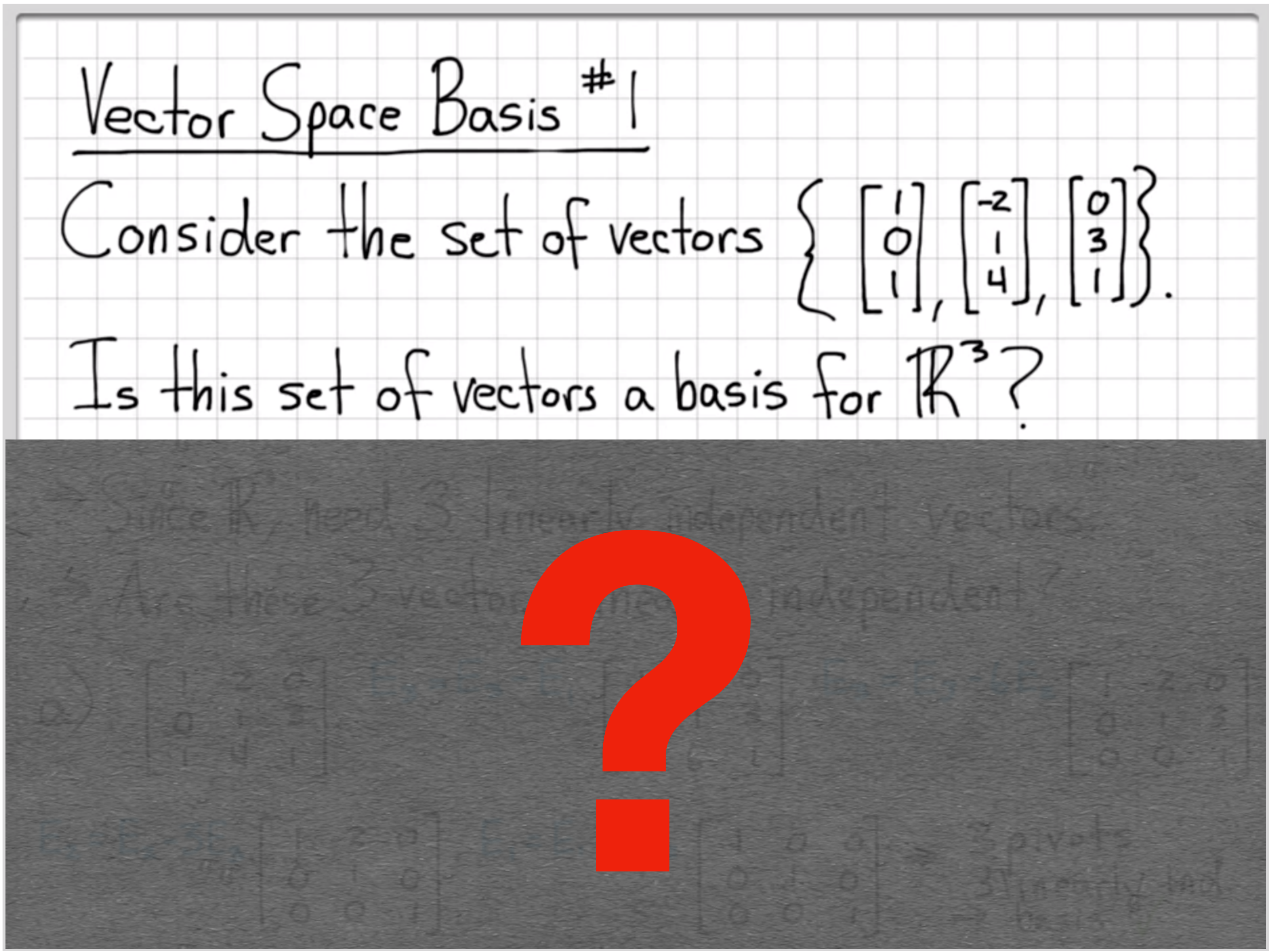


Basis example (3 vectors in 3D)



Clips from https://www.youtube.com/watch?v=k7RM-ot2NWY

EXERCISE



https://www.youtube.com/watch?v=3l3qfs2vINE&t=65s

ANOTHER SUMMARY (LESS HIPSTER)

A **vector** is a point in an *n*-dimensional space

A **linear combination** is the sum of the multiplications between scalars and vectors

A set of vectors (with the zero vector) closed under linear combination is called **vector space**

The **norm** represents the *length of a vector*

A vector space on which is defined the norm is called **Banach space**

The **inner product** is the sum of the multiplications of the elements of two vectors; it represents their *similarity*

Two vectors are called **orthogonal** if their inner product is zero

A Banach space on which is defined the inner product is called **Hilbert space**

ANOTHER SUMMARY (OH GEE.. REALLY NOT HIPSTER)

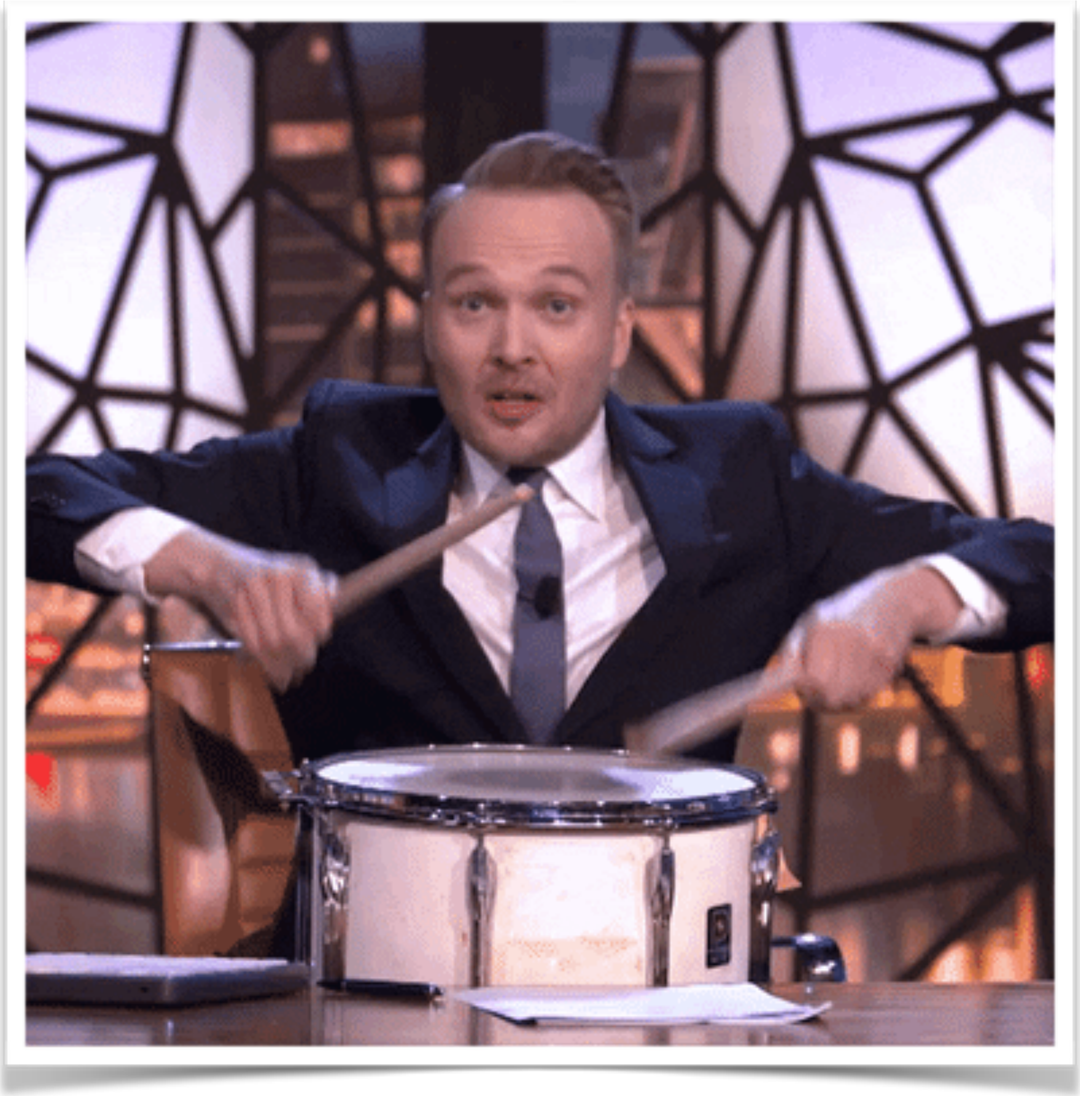
A **projection** (that is essentially an inner product) is a change of perspective on a vector that highlights specific properties

A **basis** of a *n*-dimensional vector space is a set of pairwise orthogonal *n* vectors

If a vector is projected on a basis it can be **reconstructed** by summing all the projections

In summary, **a vector in vector space can be written as a linear combination of a basis for that space, by multiplying it by some constants and summing the products**

Are you ready for the "big" thing???



ANALYSIS AND SYNTHESIS (1)

It is possible to define an **analysis** as the estimation of the constants and a **synthesis** as the linear combination equation that recover the signal; following subsections will clarify this important concept.

Analysis. The analysis is the representation ϕ_x of a signal given by the inner product of it by a basis in a vector space; it is therefore given by the projection

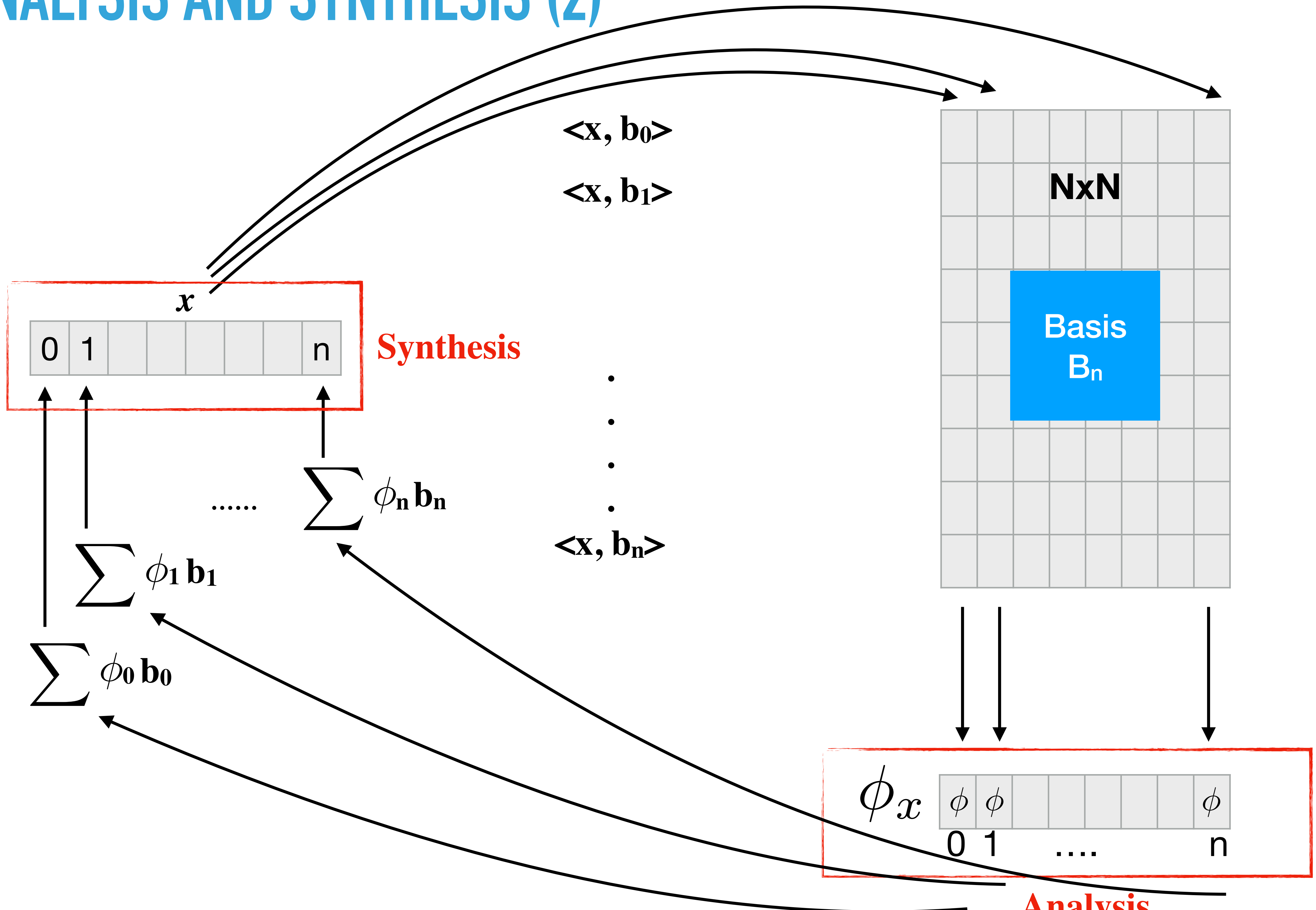
$$\phi_x = \sum_t x(t) * \overline{b_k} = \langle x, b_k \rangle \tag{7}$$

where b_k is a given basis and t is time.

Synthesis. The synthesis is the reconstruction of the original signal x by the summation of the products with the representation ϕ_x created by the analysis:

$$x(t) = \sum_k \phi_x b_k(t) = \sum_k \langle x, b_k \rangle b_k(t). \tag{8}$$

ANALYSIS AND SYNTHESIS (2)



DISCRETE FOURIER TRANSFORM (DFT)

The discrete Fourier transform. The Fourier representation is interpretable in the following context as a specific case of analysis and synthesis, where the basis is given by a set of complex sinusouids: $b_k = e^i 2\pi k$ (where i is the imaginary unit).

Analysis (projections onto basis)

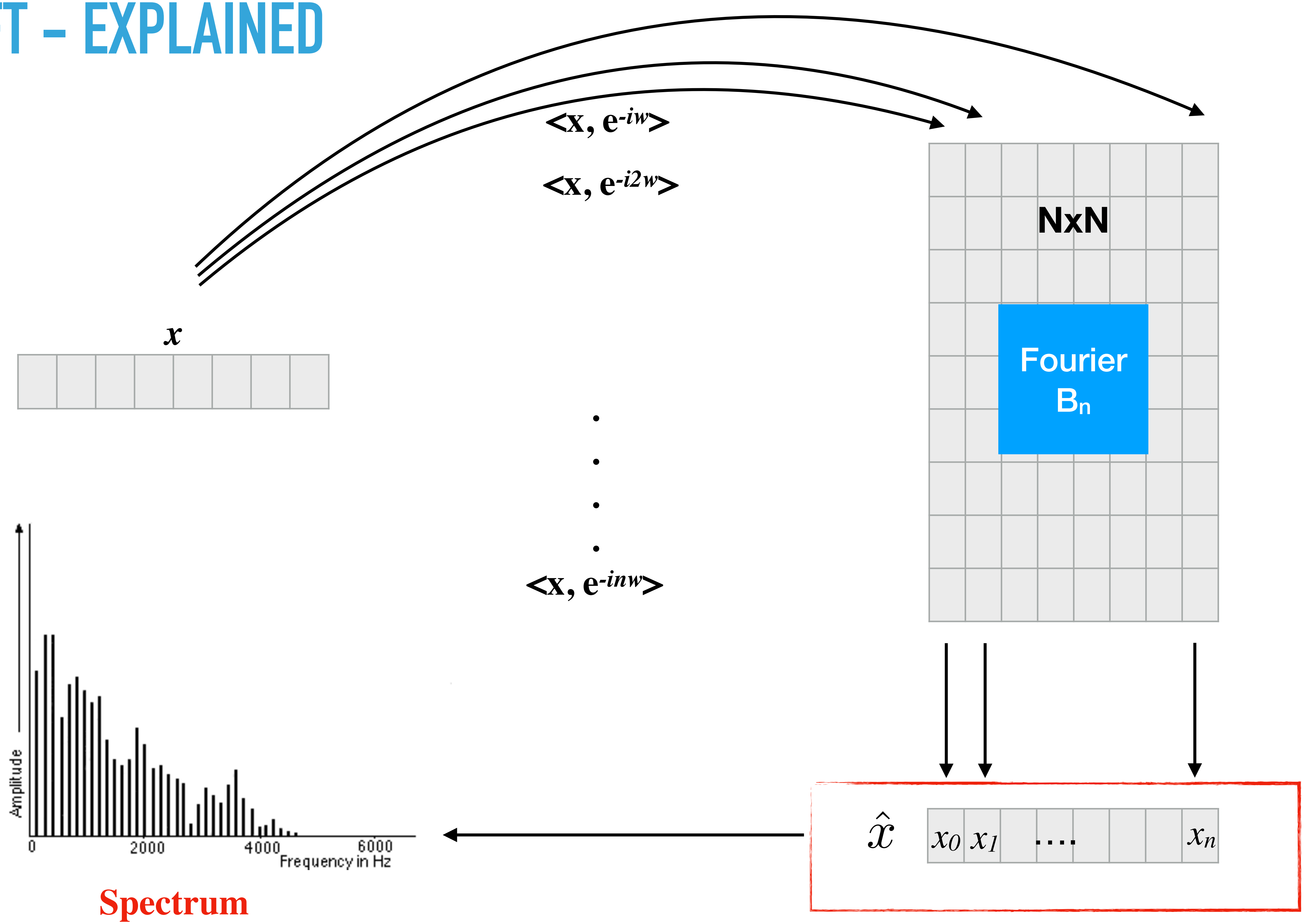
$$\boxed{\hat{x}(k) = \sum_t x(t) e^{\frac{-i2\pi kt}{T}}} \tag{9}$$

Synthesis (reconstruction by sum)

$$\boxed{x(t) = \frac{1}{T} \sum_k \hat{x}(k) e^{\frac{i2\pi kt}{T}}}. \tag{10}$$

The basis of Fourier is defined in frequency,
hence it "highlights" that parameter of the sound
that are projected onto it

DFT - EXPLAINED



CONVOLUTION

Convolution is discussed in Music 158b!

The convolution then, can be defined as an inner product that is repeated over time:

$$(x * h)_t = \sum_m x(t-m) * h(m) \tag{11}$$

where h is called *kernel* and is of length m. The convolution is therefore:

(1) the time series given by a signal weighted by another that slides along;
(2) the cross-variance between two signals (similarity in time);
(3) the time series given the mapping between to signals;
(4) a filtering process.

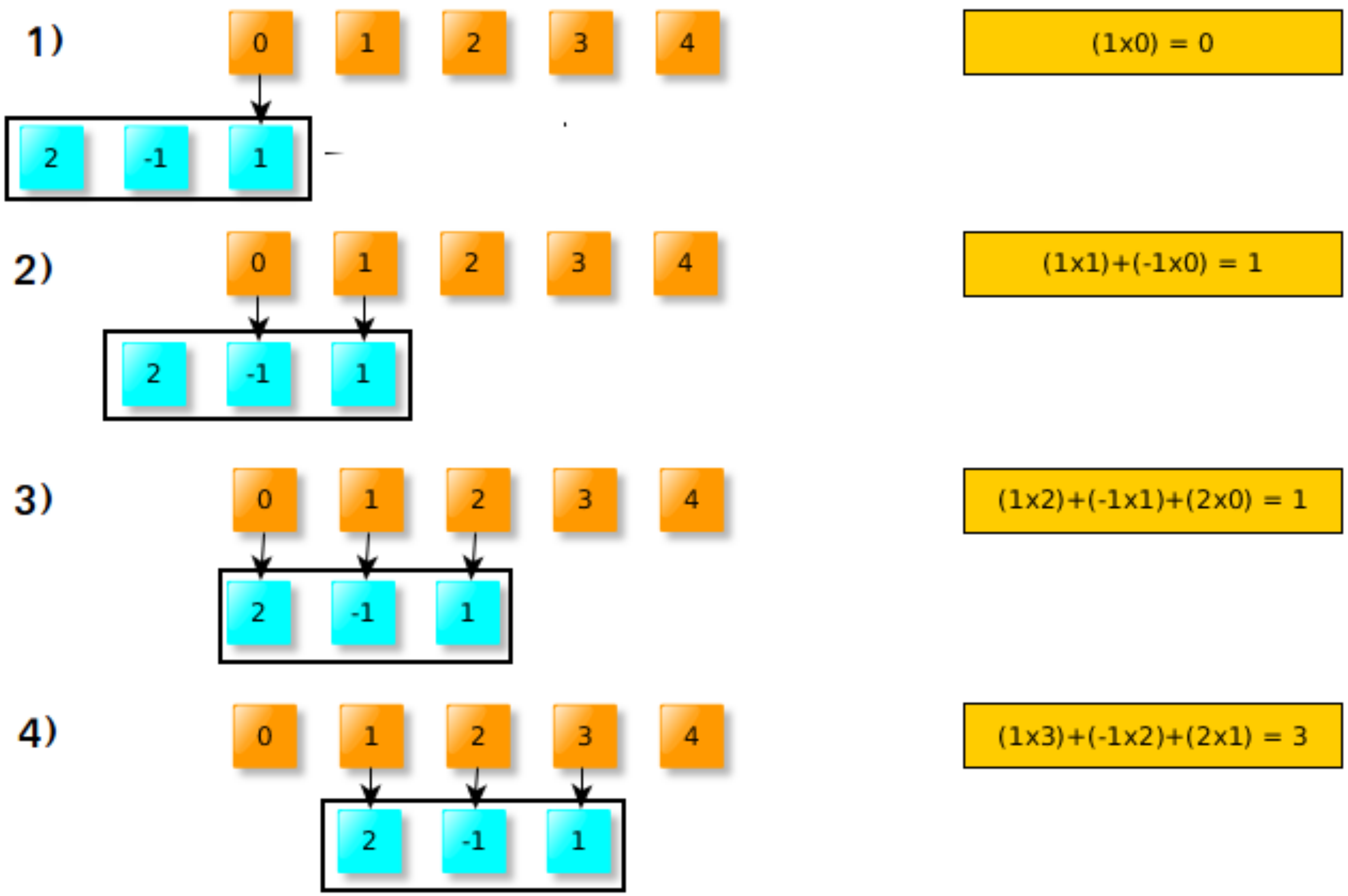
The convolution theorem. There is an important relation between the DFT and convolution: the convolution in time domain between to signals is equal to the product of the DFT of them. Formally:

$$x * h \equiv \hat{x} \cdot \hat{h} \tag{12}$$

where $\hat{x}, \hat{h}$ are DFTs of respective signals.

VISUAL EXPLANATION (1)

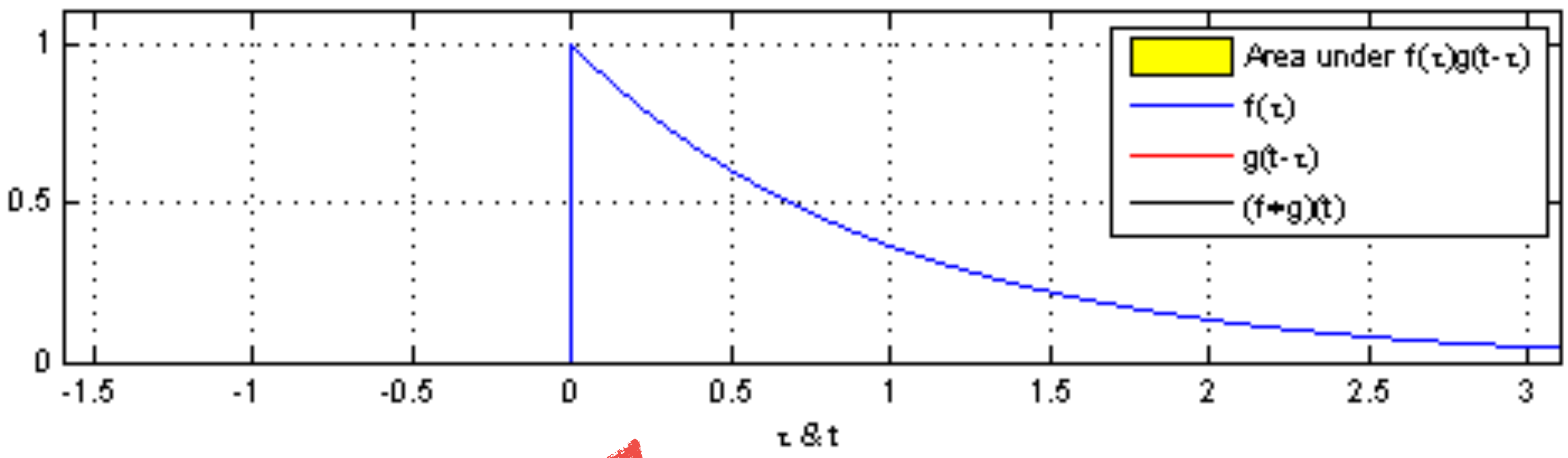
A signal sliding over another



https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/convolution.html

VISUAL EXPLANATION (2)

A signal sliding over another



But why??

https://en.wikipedia.org/wiki/Convolution

INTERPRETATION

Convolution can be interpreted in several ways:

- The *mapping* between to signals (ie. the transmissions of the characteristics of a signal onto another)
- The cross-variance between two signals (*similarity* in time/space) - this is the interpretation in convolutional neural networks
- The weighted sum of a signal, where the weights are given by another signal
- A filtering process

This will be more clear when we will see filters

MORE ON CONVOLUTION THEOREM

A convolution in time domain can be tought as a complex multiplication in frequency domain:

$$y(t) = x(t) * h(t) = X(k)H(k) \tag{1}$$

Convolving two signals in time domain is the same thing as multiplying their spectra

This is the basic principle behind the so called fast convolution

EXAMPLE: CONVOLUTION-BASED REVERB

Anechoic orchestral recording

Impulse response of Concert Gebouw in Amsterdam

Convolution of the two signals

We will implement this in a minute!

(WIENER) DECONVOLUTION

As consequence, by divinding two signals in frequency domain we remove the effect of convolution (*Wiener deconvolution*)

$$W(k) = \frac{Y(k)\overline{X(k)}}{|X(k)|^2 + \sigma^2} \tag{3}$$

where $\overline{X}$ denotes the complex conjuage and $\sigma = \lambda \max(|X(k)|)$ (λ is a scaling parameter).

All the extra parameters are normalisation factors; we will see this in practice

Questions:

Can we use deconvolution to remove a reverb?

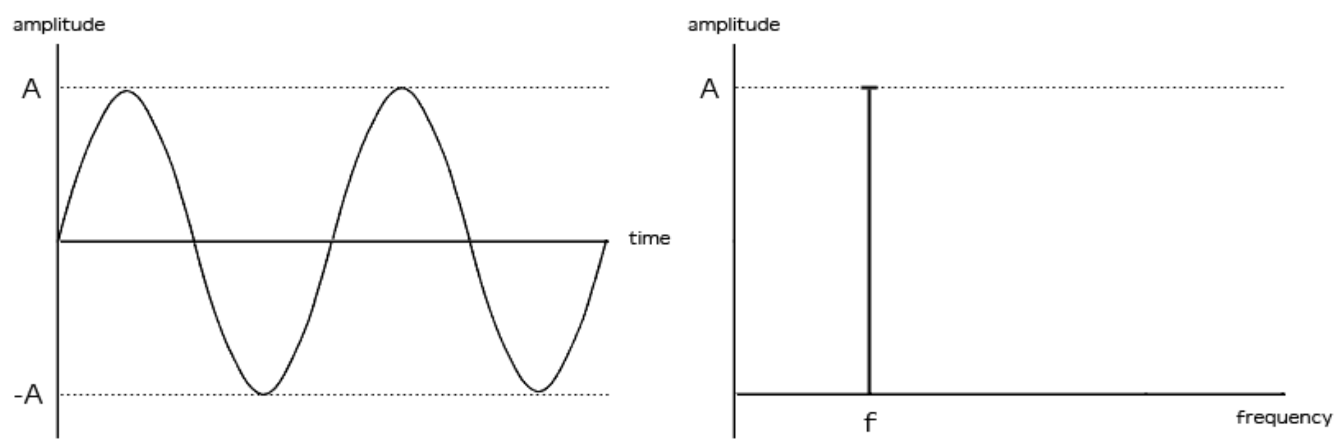
Can we use deconvolution to compute the impulse response of a room?

FEATURE MAPS

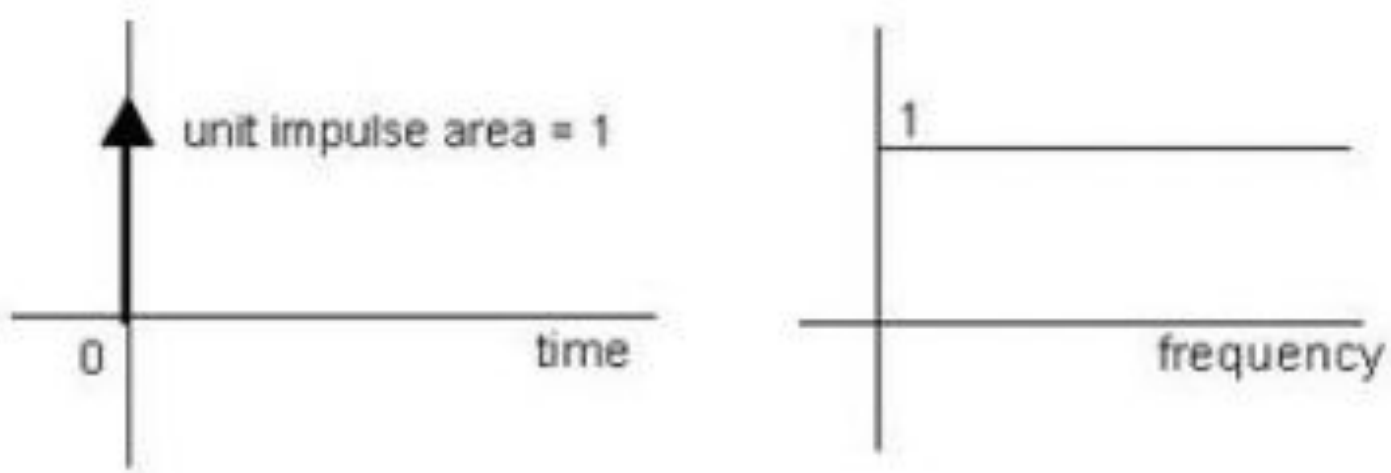
TIME VS FREQUENCY

3.4. **Different bases.** The basis made of complex sinusoids is only a possible basis among infinite. This basis focus on representig correctly frequencies and is therefore well localized in frequency but is not localized in time. On the other hand, it is possible to create a basis made of Dirac's pulses that will provide perfect localization in time but no localization in frequency.

Fourier



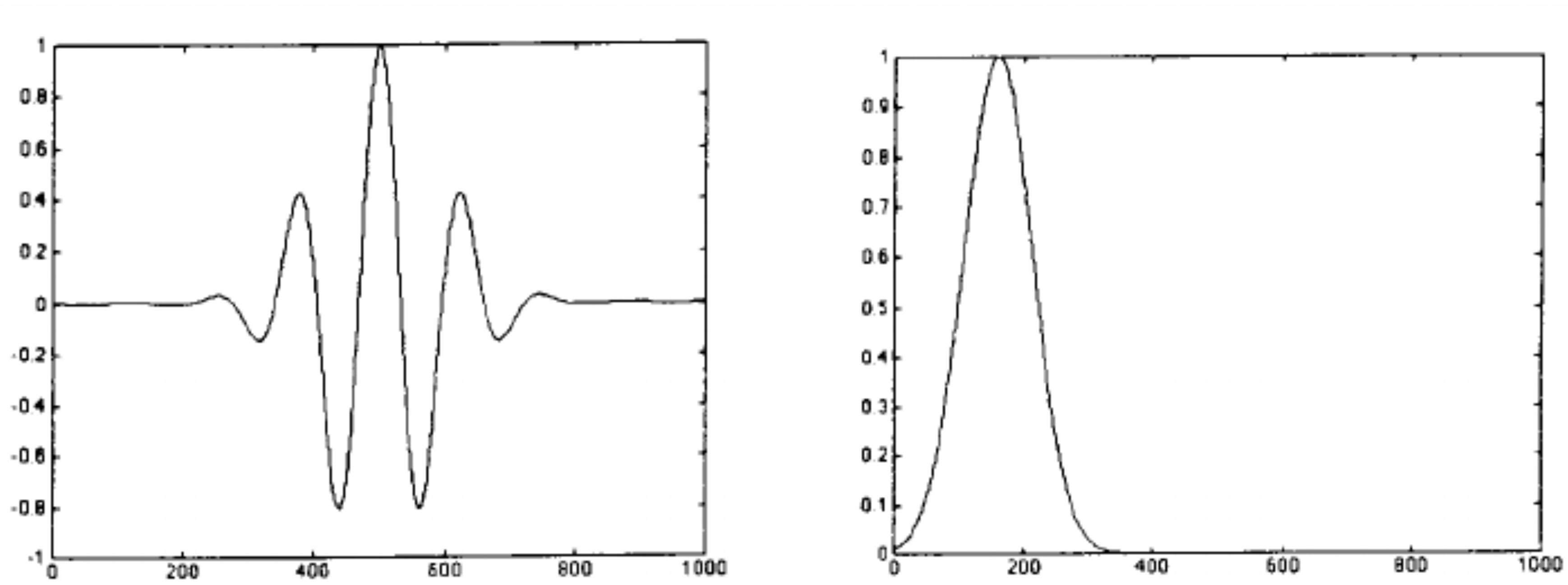
Dirac



WAVELETS

A compromise between sinusoids and impulses is given by bases made of oscillating signals with a temporal limitation, such as *wavelets*. A wavelet is a bandpass filter centered on a specific frequency with a specific bandwidth that has therefore a localization both in time and frequency. The Gabor wavelet represent the best compromise, in term of Heisenberg uncertainty principle, between time and frequency. More information on this vast subject can be found in the references.

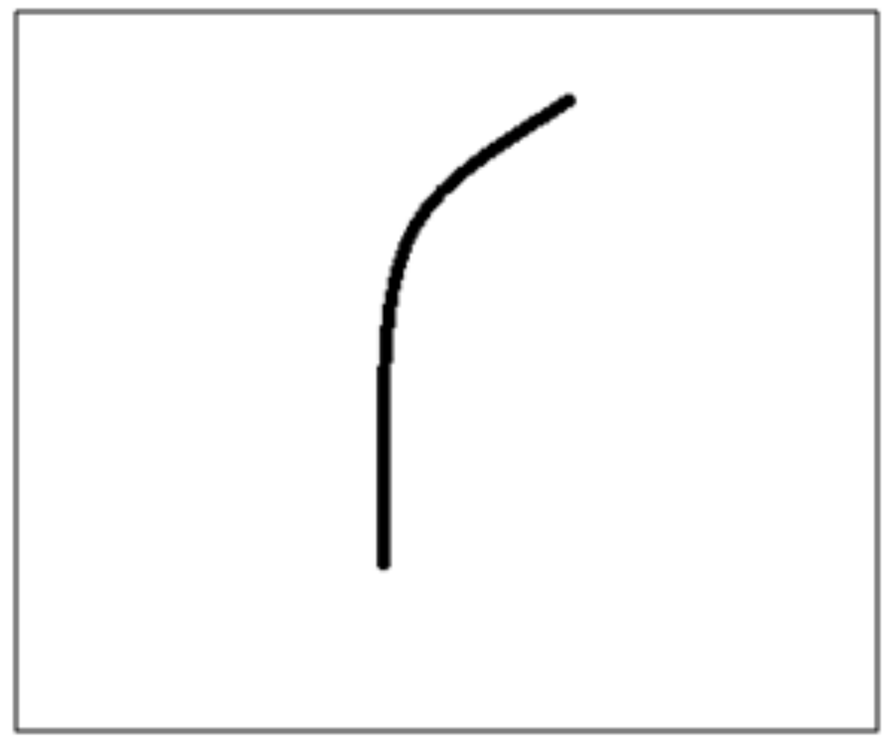
Wavelets



FILTER (KERNEL)

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

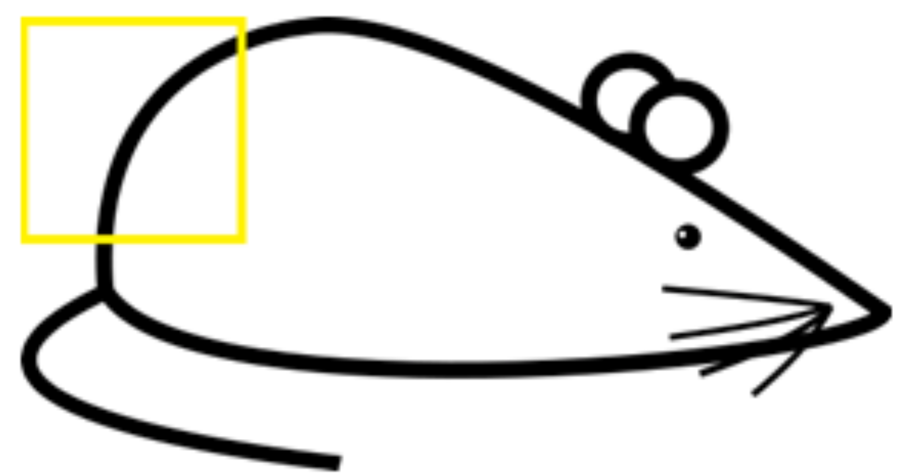


Visualization of a curve detector filter

https://medium.com/@himadrisankarchatterjee/a-basic-introduction-to-convolutional-neural-network-8e39019b27c4

UNKNOWN OBJECT

Original image



Visualization of the filter on the image

https://medium.com/@himadrisankarchatterjee/a-basic-introduction-to-convolutional-neural-network-8e39019b27c4

SUCCESSFUL CONVOLUTION WITH KERNEL

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

https://medium.com/@himadrisankarchatterjee/a-basic-introduction-to-convolutional-neural-network-8e39019b27c4

UNSUCCESSFUL CONVOLUTION WITH KERNEL

Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

$*$

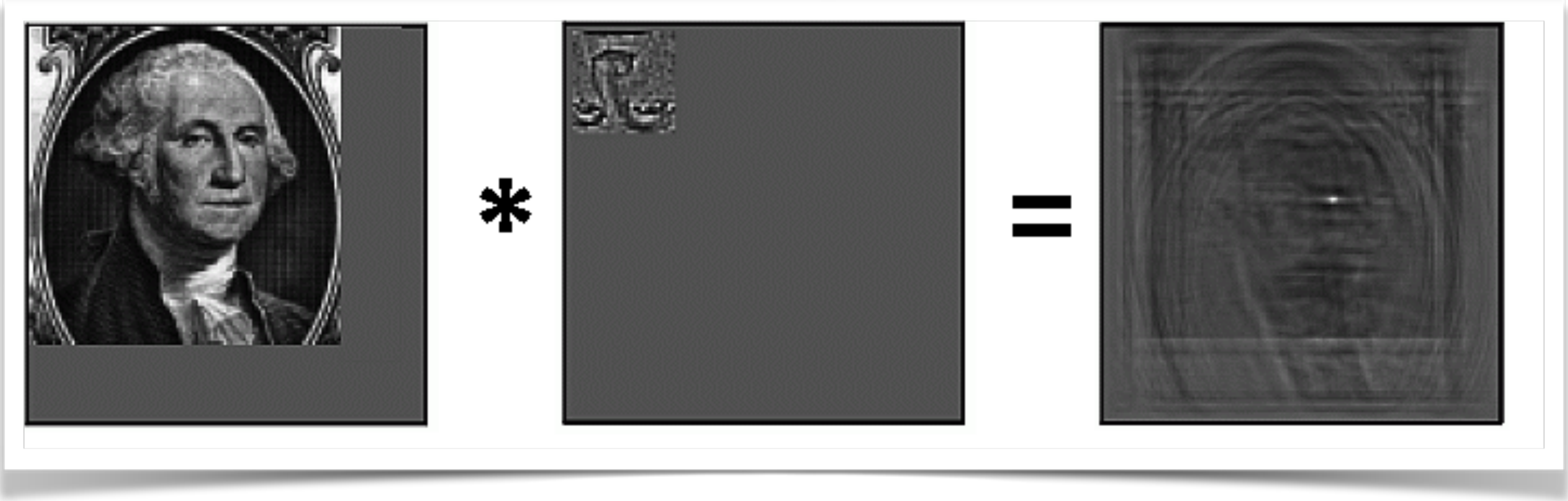
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

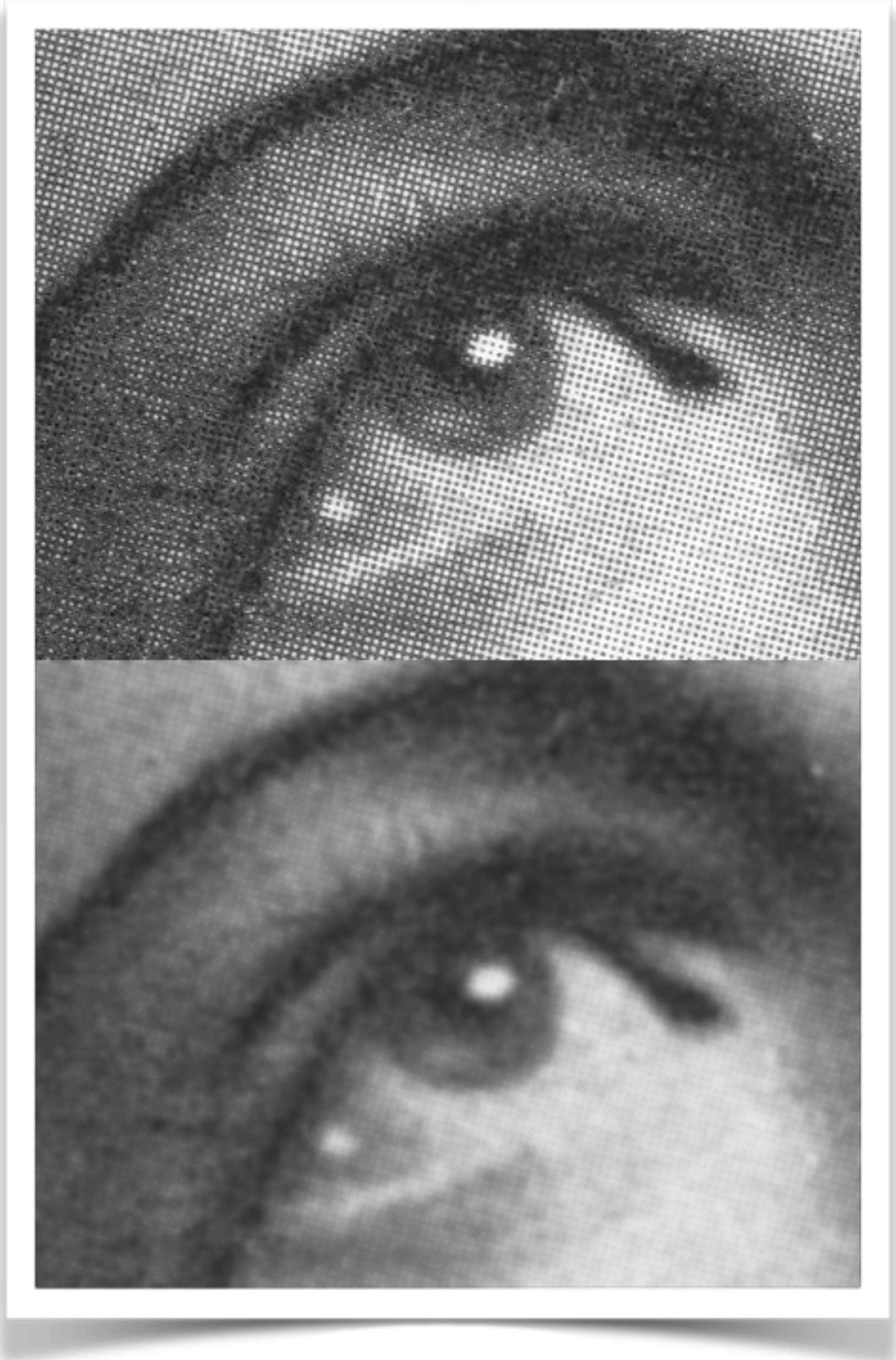
Multiplication and Summation = 0

https://medium.com/@himadrisankarchatterjee/a-basic-introduction-to-convolutional-neural-network-8e39019b27c4

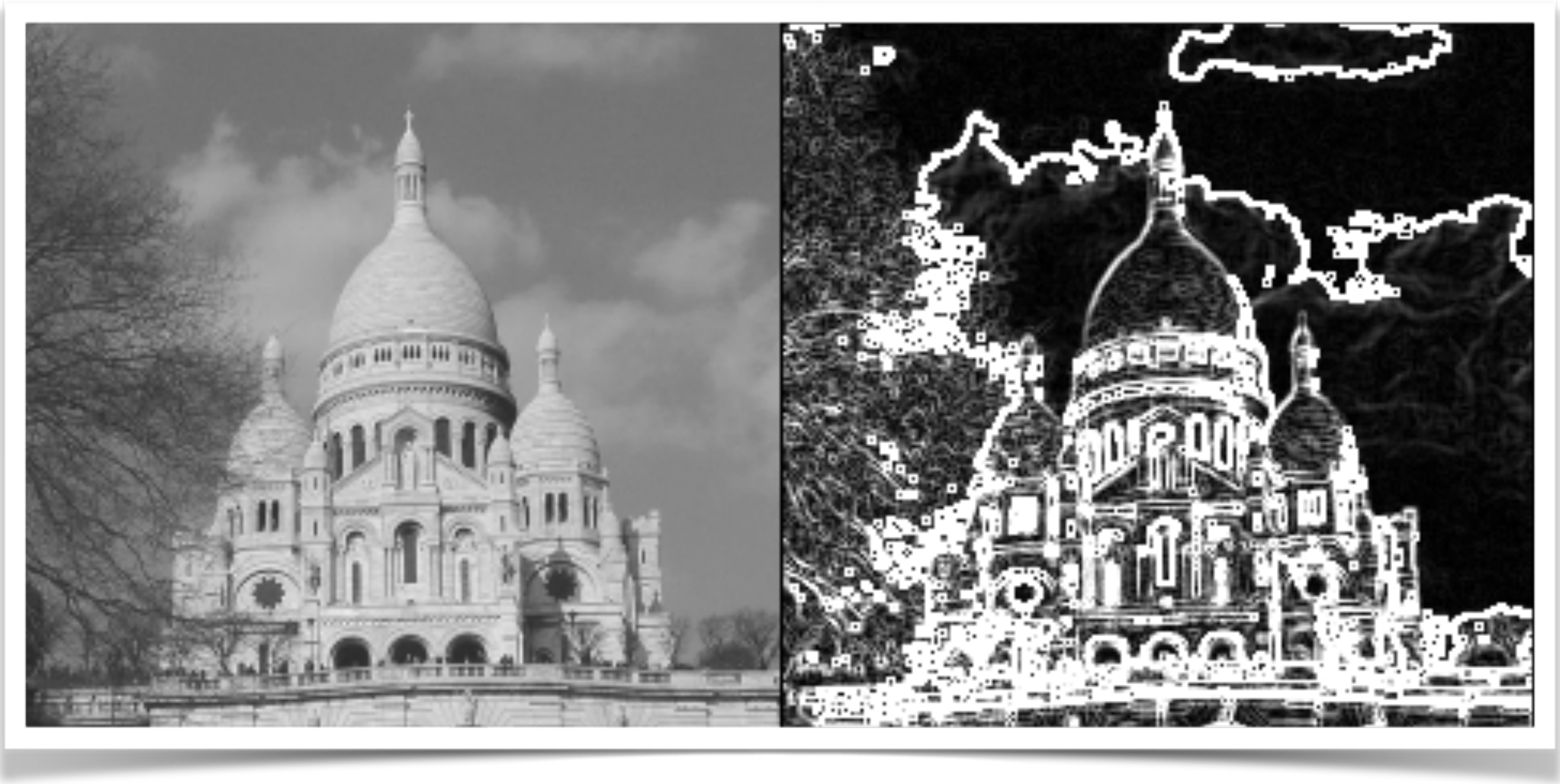
PATTERN MATCHING



GAUSSIAN BLUR



EDGE DETECTION



EIGENVECTORS – NOT EVERYTHING REACTS

Eigenvalues and eigenvectors are often introduced to students in the context of linear algebra courses focused on matrices.[23][24] Furthermore, linear transformations can be represented using matrices,[1][2] which is especially common in numerical and computational applications.[25]

Consider n-dimensional vectors that are formed as a list of n scalars, such as the three-dimensional vectors

$$x = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} -20 \\ -60 \\ -80 \end{bmatrix}.$$

These vectors are said to be scalar multiples of each other, or parallel or collinear, if there is a scalar λ such that

$$x = \lambda y.$$

In this case $\lambda = -1/20$.

Now consider the linear transformation of n-dimensional vectors defined by an n by n matrix A,

$$Av = w,$$

or

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

where, for each row,

$$w_i = A_{i1}v_1 + A_{i2}v_2 + \cdots + A_{in}v_n = \sum_{j=1}^{n} A_{ij}v_j.$$

If it occurs that v and w are scalar multiples, that is if

$$Av = w = \lambda v, \qquad (1)$$

then v is an **eigenvector** of the linear transformation A and the scale factor λ is the **eigenvalue** corresponding to that eigenvector. Equation (1) is the **eigenvalue equation** for the matrix A.

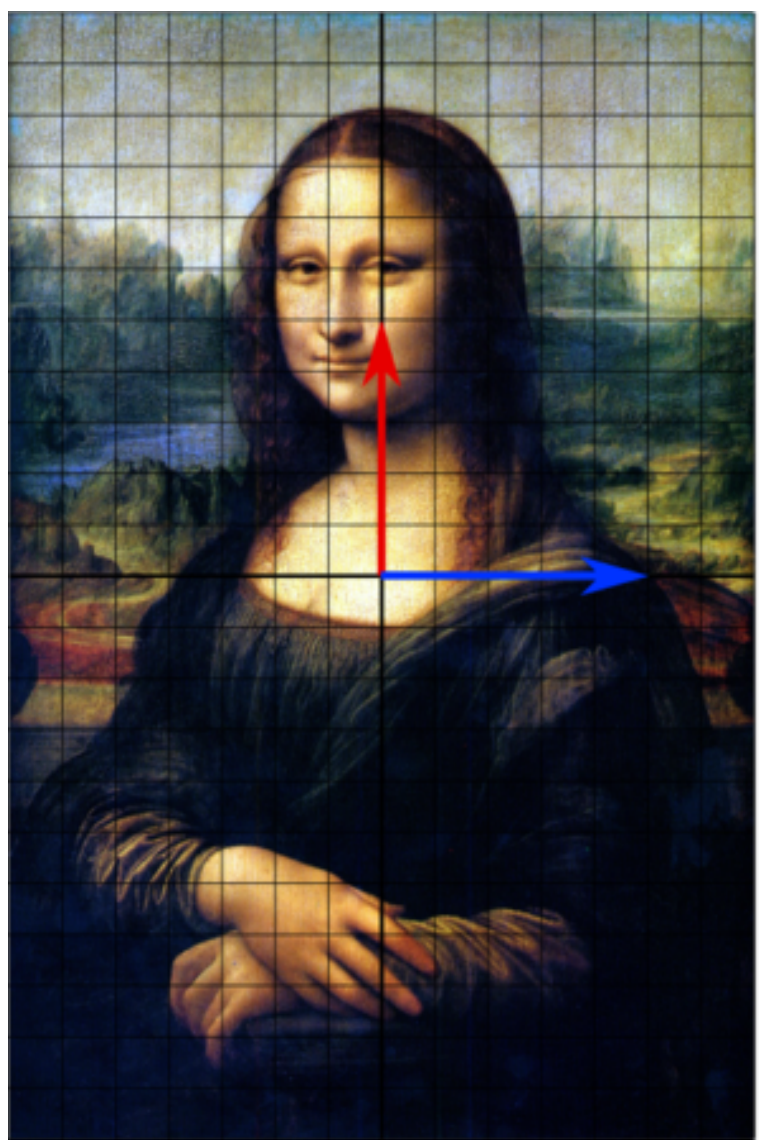
Equation (1) can be stated equivalently as

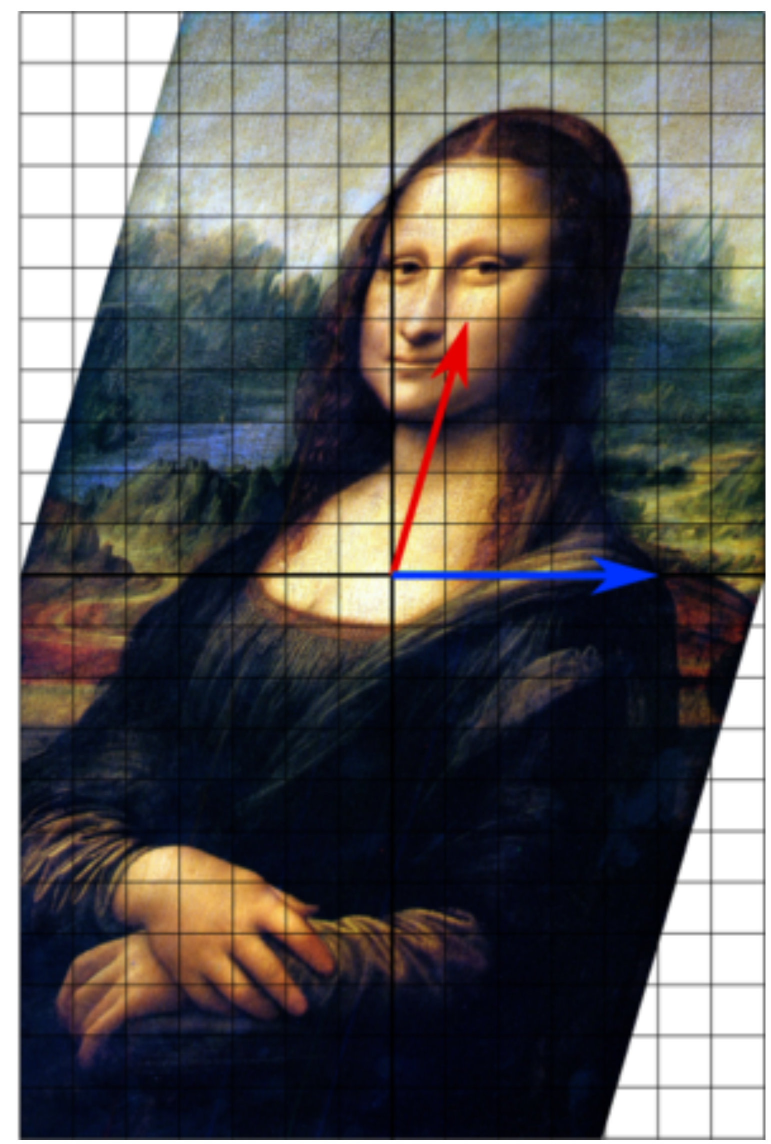
$$(A - \lambda I)v = 0, \qquad (2)$$

where I is the n by n identity matrix.

Matrix A acts by stretching the vector x, not changing its direction, so x is an eigenvector of A.

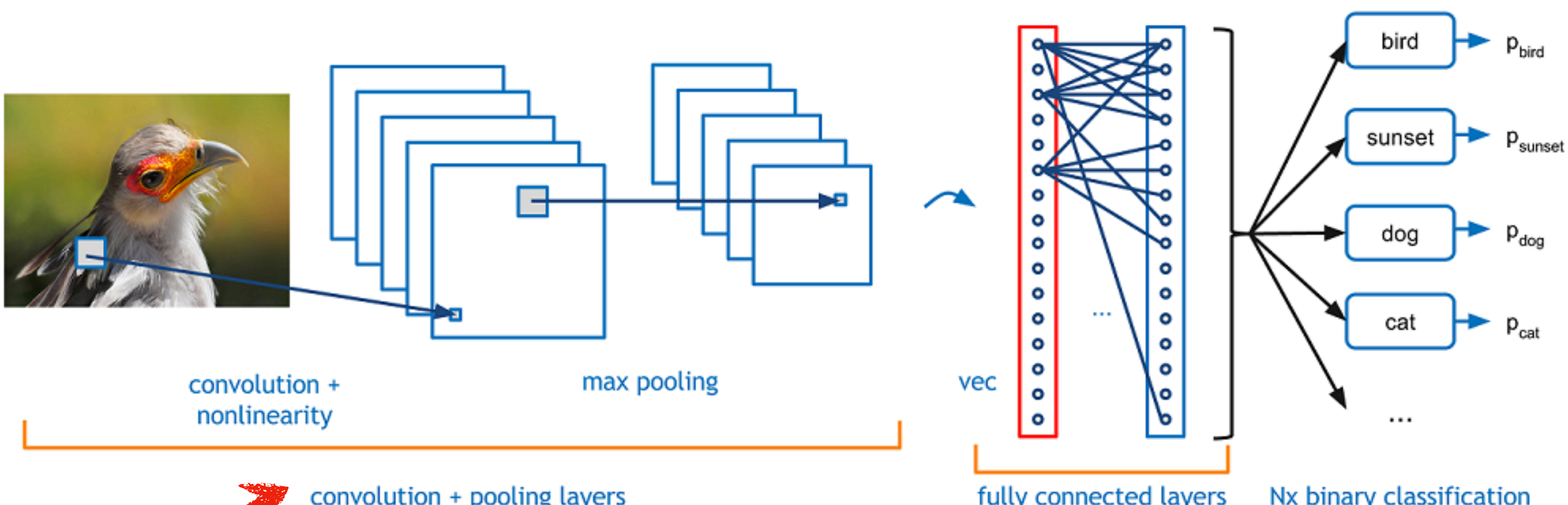
EIGENVECTORS FOR SHEAR





CONVOLUTIONAL NEURAL NETWORKS (CNN)

CNN use the same principle as Fourier (projection), but the *kernel* is learned on a specific problem (and highlights a specific property)



We will see CNN more in details soon!

https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

FINAL SUMMARY

Yeah.. that's quite hipster...

All transformations on signals are made of inner products!!

$$\langle x, y \rangle$$

Convolution

CNN

Fourier Analysis

Wavelet Analysis

Digital Filters

...and more...

THANK YOU!

Suggested exercise:
try to write in Python the code of the DFT (without using Google)!