Homework 8

due Wed Nov 9 at start of class

`Homework8.pdf`

`nashville.jpg`

download from Brightspace

`Signals.zip` (Jupyter notebook)
`Psychopy.zip` (Python files)

More on Images and Signals

Convolution

$$I_{out}(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} F(u,v)\ I_{in}(i-u, j-v)$$

discrete version

$$I_{out}(x,y) = \int\int F(u,v)\ I_{in}(x-u, y-v)\ du\ dv$$

continuous version

convolutions in Python

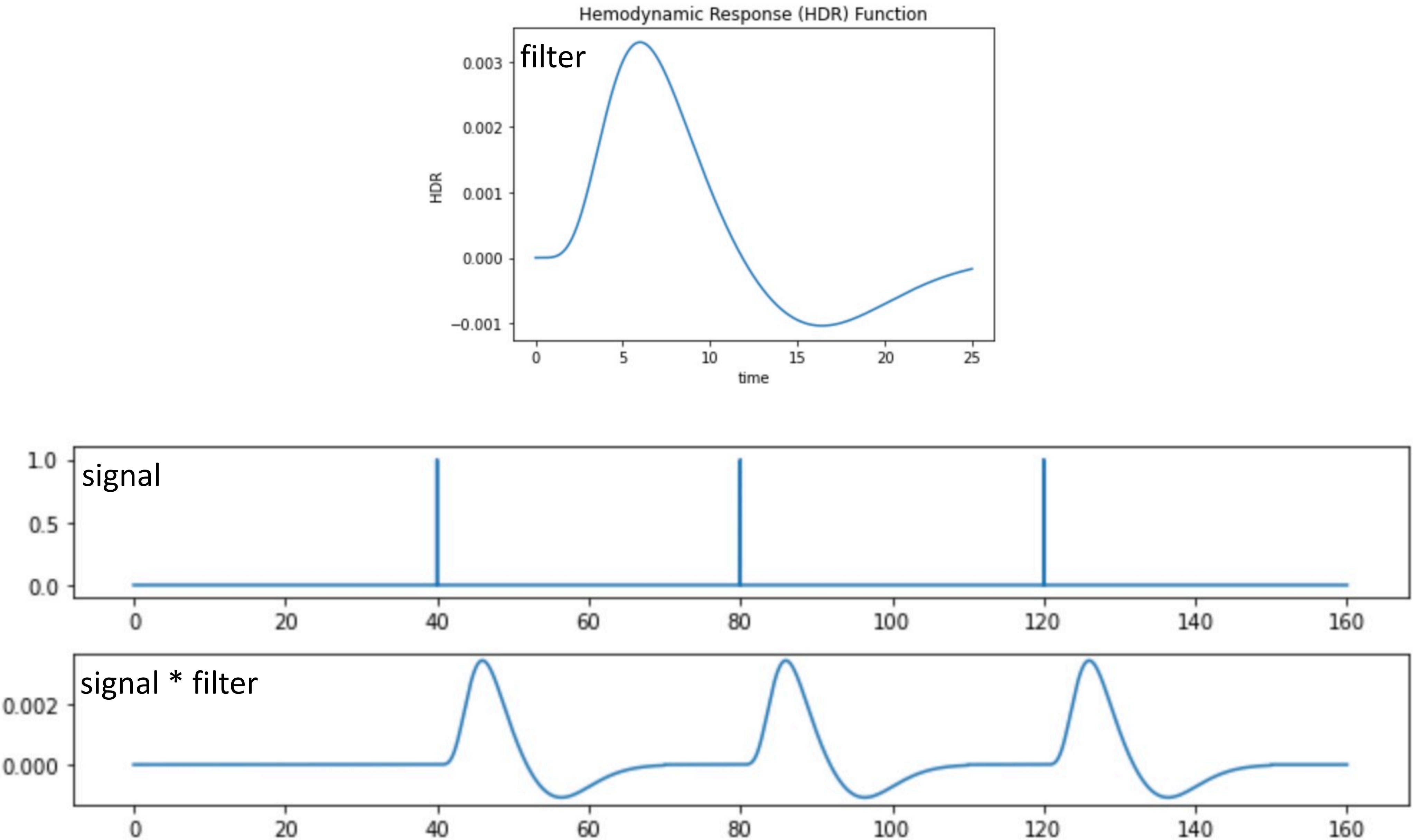
```
from scipy import signal

filtered = signal.convolve(sig, fil, mode='same')
```

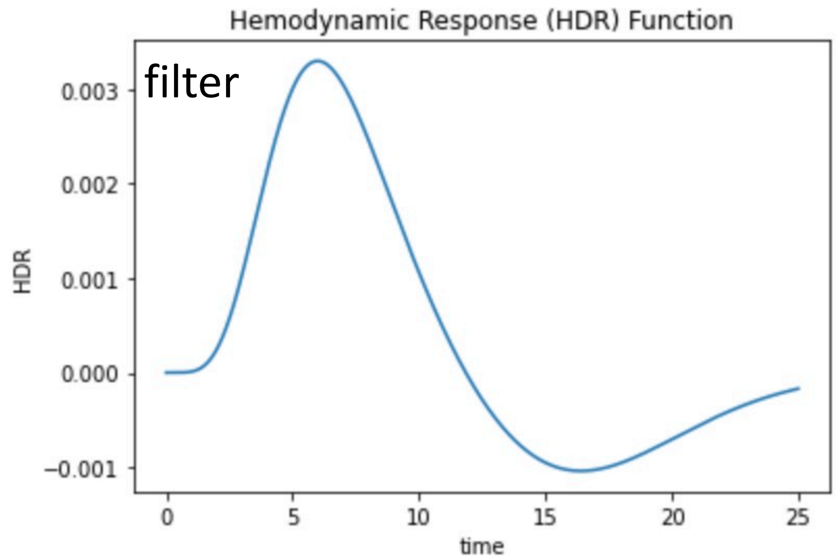
`'same'` means that the output is *not* padded …
the output is the *same* length as the signal

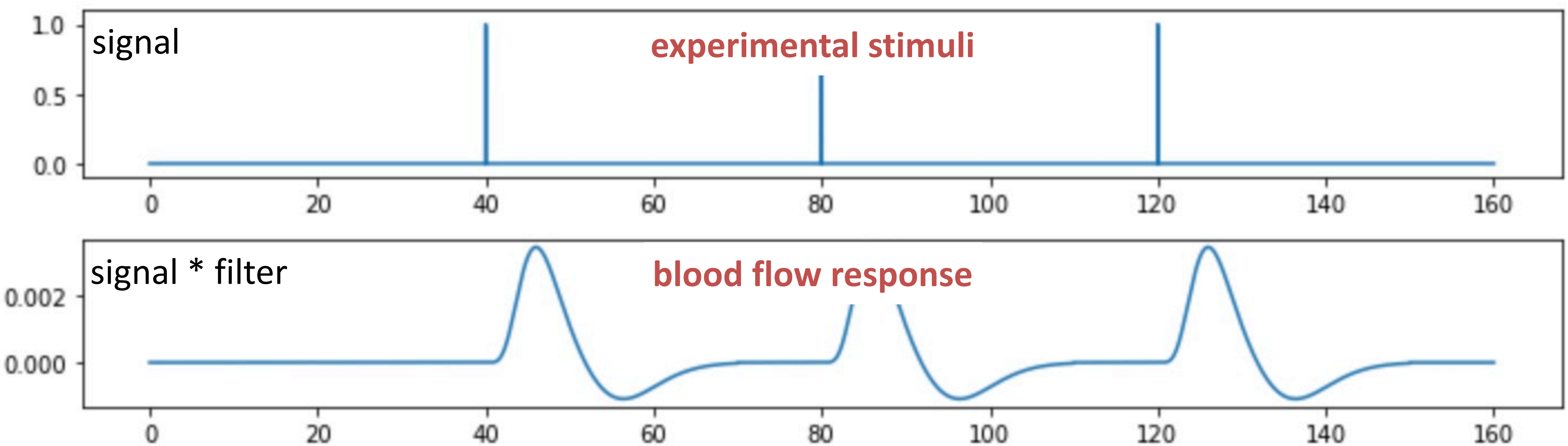
```
filtered = signal.convolve2(sig, fil, mode='same')
```

hemodynamic response function example



hemodynamic response function example

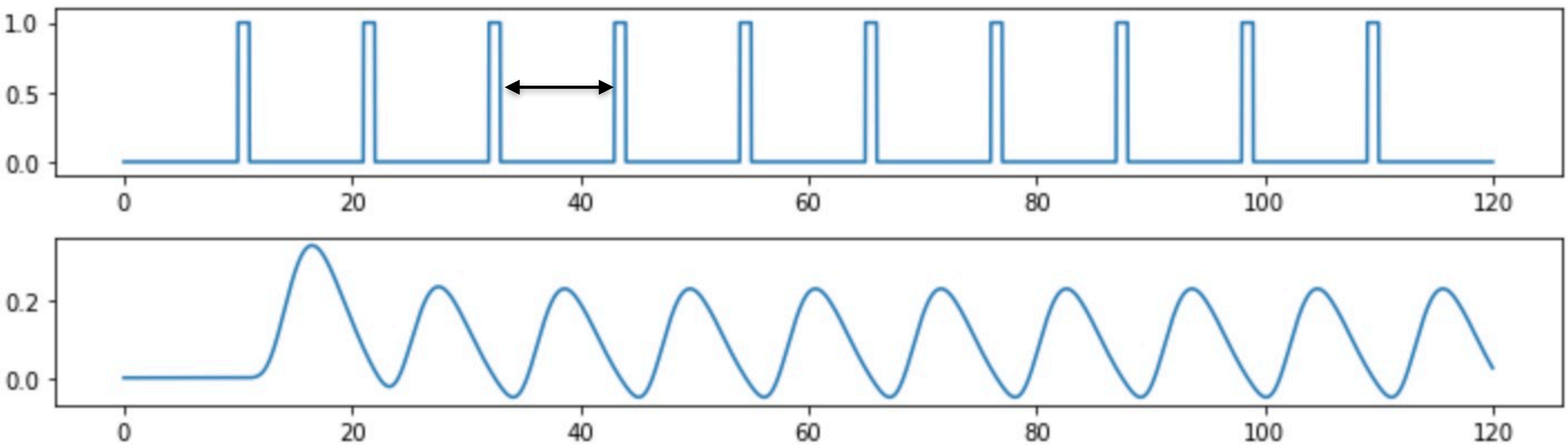




hemodynamic response function example

1sec
stimuli

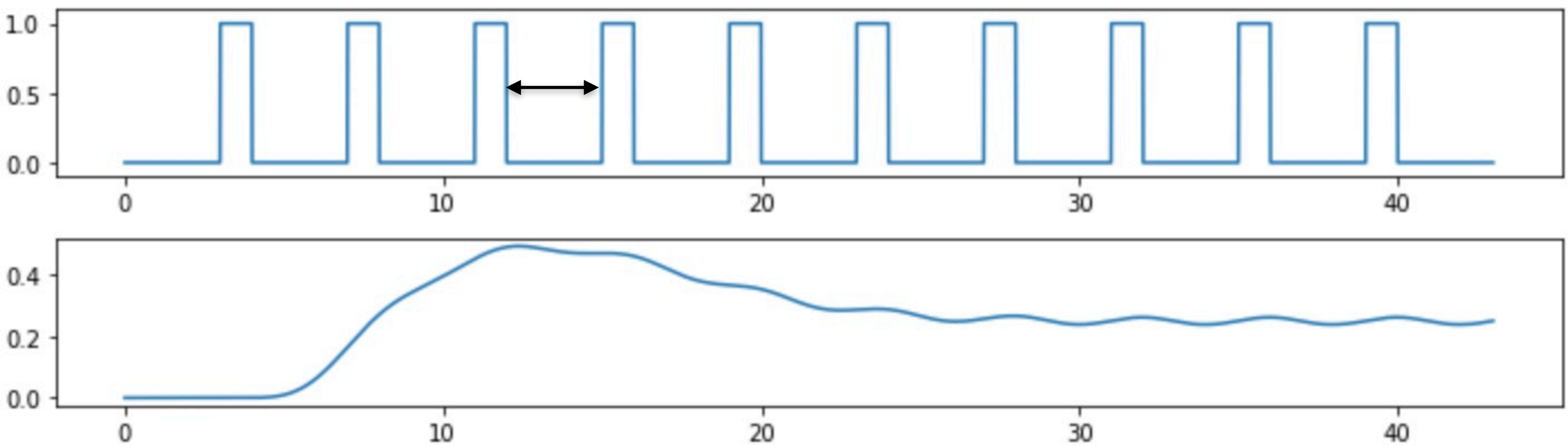
separate by
10secs



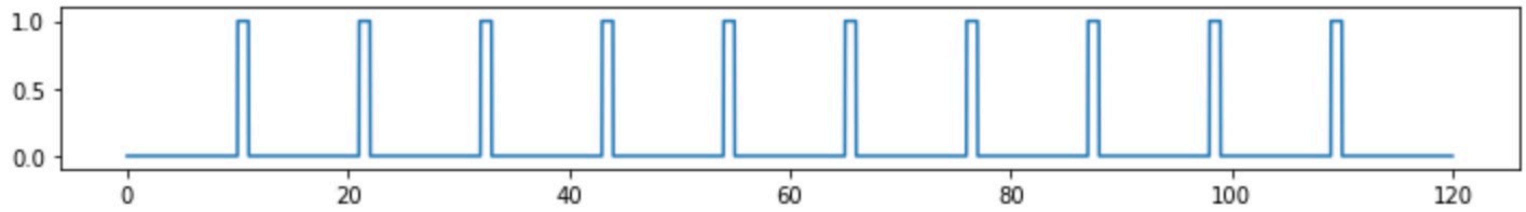
hemodynamic response function example

1sec
stimuli

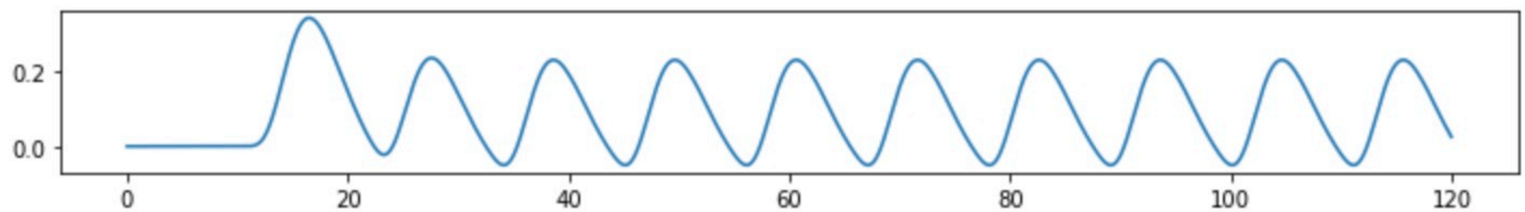
separate by
3secs



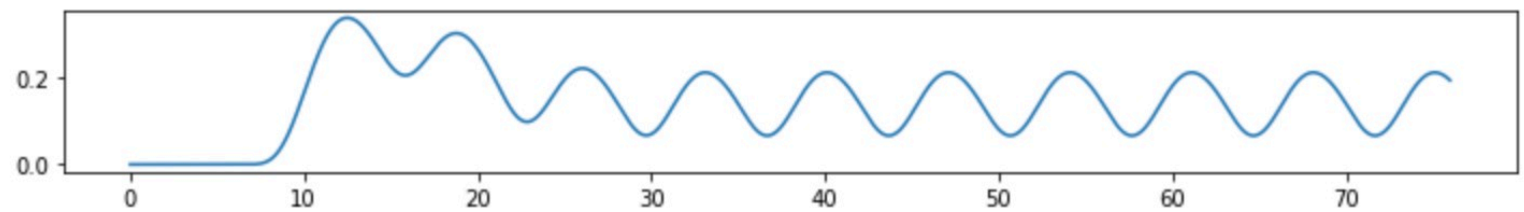
hemodynamic response function example



separate by 10secs



separate by 6secs



separate by 4secs

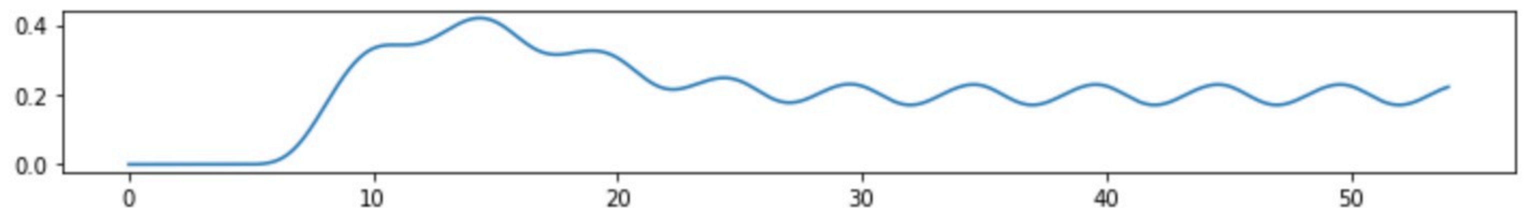


image processing example

Gaussian Filter

```
[[1, 2, 1]
 [2, 4, 2]
 [1, 2, 1]]
```

(normalized)

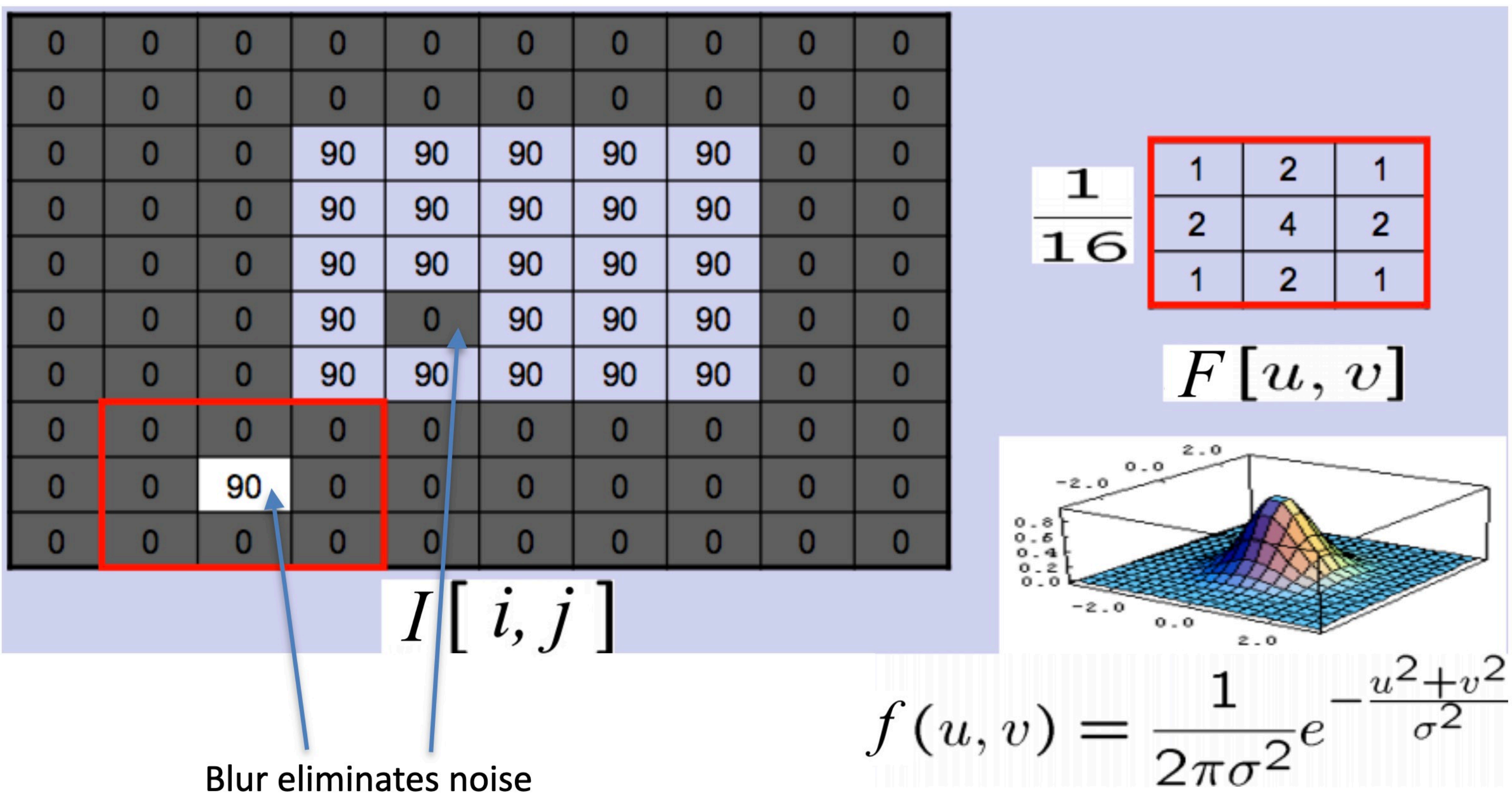


image processing example

Gaussian Filter

```
[[1, 2, 1]
 [2, 4, 2]
 [1, 2, 1]]
```

original

gaussian filtered

image processing example

<u>Sharpen Mask</u>

```
[[ 0, -1,  0]
 [-1,  5, -1]
 [ 0, -1,  0]]
```

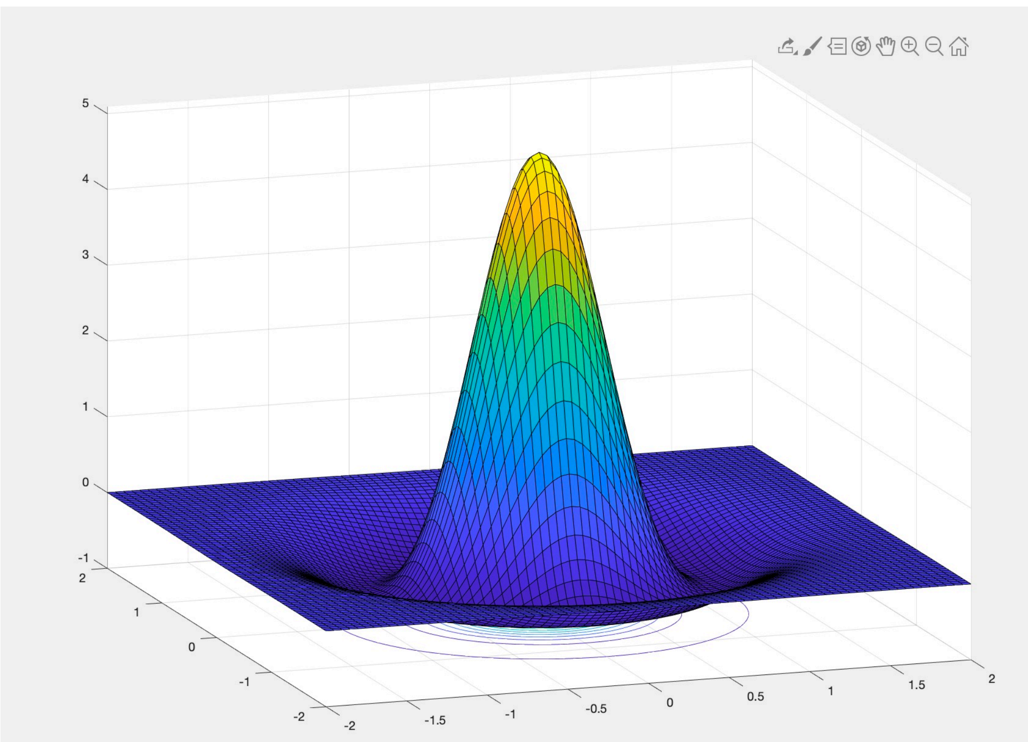
original

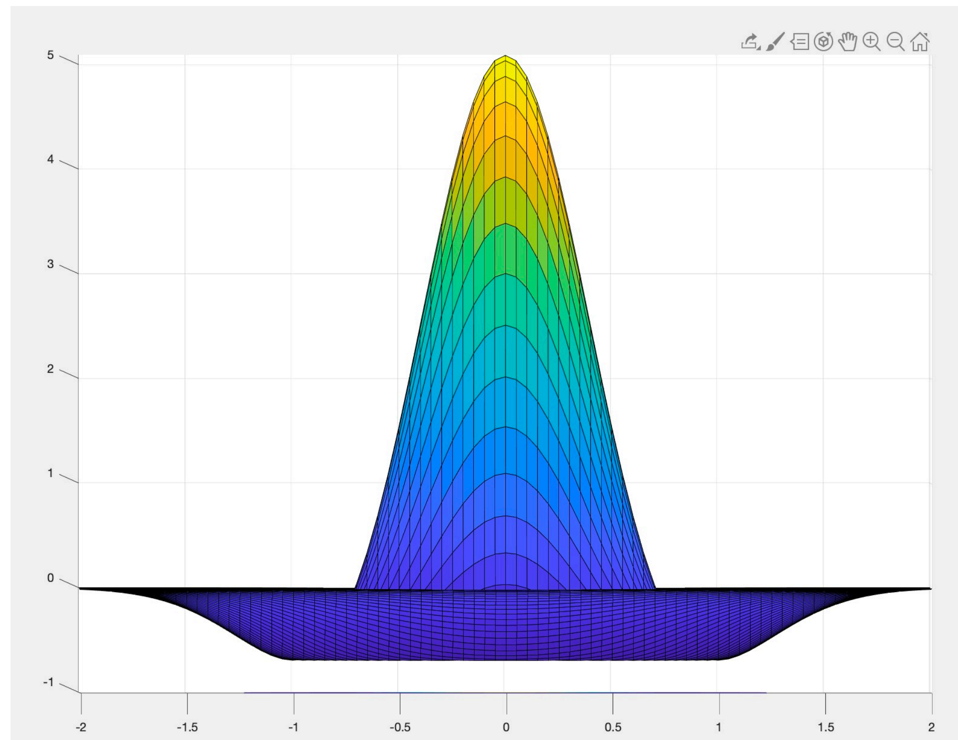
gaussian filtered

sharpen filtered

laplacian of gaussians

$$LoG(x, y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2 + y^2}{2\sigma^2}\right]\exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$





laplacian of gaussians

original

LoG filtered

laplacian of gaussians

$$\underbrace{L(x,y)}_{\text{Laplacian gives 2nd derivative (indicates edges)}} * \left[G(x,y) * I(x,y)\right]$$

laplacian of gaussians

$$L(x, y) * \underbrace{\left[G(x, y) * I(x, y)\right]}_{\text{apply a Gaussian blur to image } I}$$

apply a Gaussian blur
to image *I*

the second derivative is highly sensitive to noise …
Gaussian blur reduced noise in the image

laplacian of gaussians

$$L(x,y) * \left[G(x,y) * I(x,y)\right]$$

$$\underbrace{\left[L(x,y) * G(x,y)\right]}_{\substack{\text{convolution is} \\ \text{associative}}} * I(x,y)$$

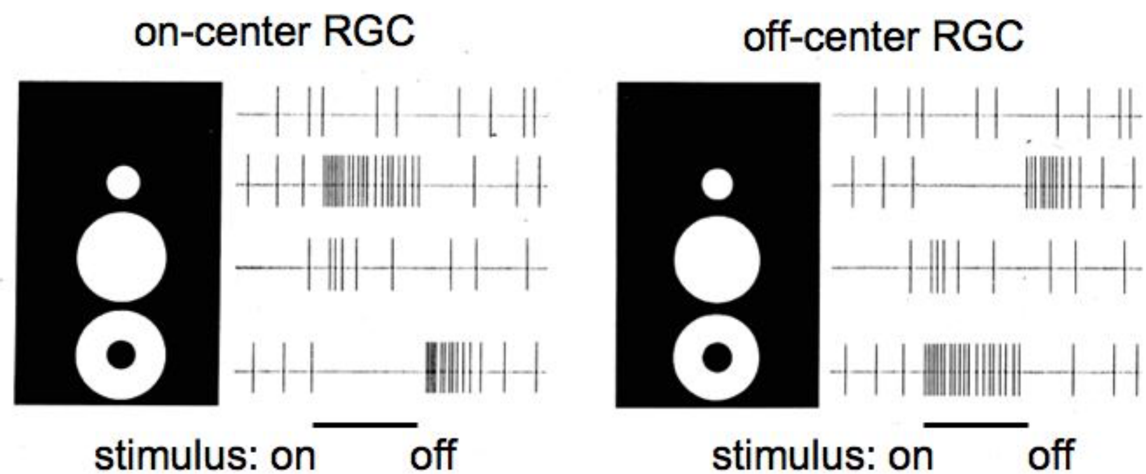
laplacian of gaussians

$$L(x,y) * \left[G(x,y) * I(x,y)\right]$$

$$\left[L(x,y) * G(x,y)\right] * I(x,y)$$

$$\underbrace{LoG(x,y)}_{\text{Laplacian of Gaussians filter}} * I(x,y)$$

approximates filtering by retinal ganglion cells



Homework 8 (due Nov 9th)

- give you Laplacian and Gaussian 2D filters
- confirm convolution is commutative
- compare an implementation of convolution with the built-in convolution function
- create a Laplacian of Gaussian (LoG) filter
- apply the Laplacian, Gaussian, and LoG to an image

convolution

$$I_{out}(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} F(u,v)\ I_{in}(i-u, j-v)$$

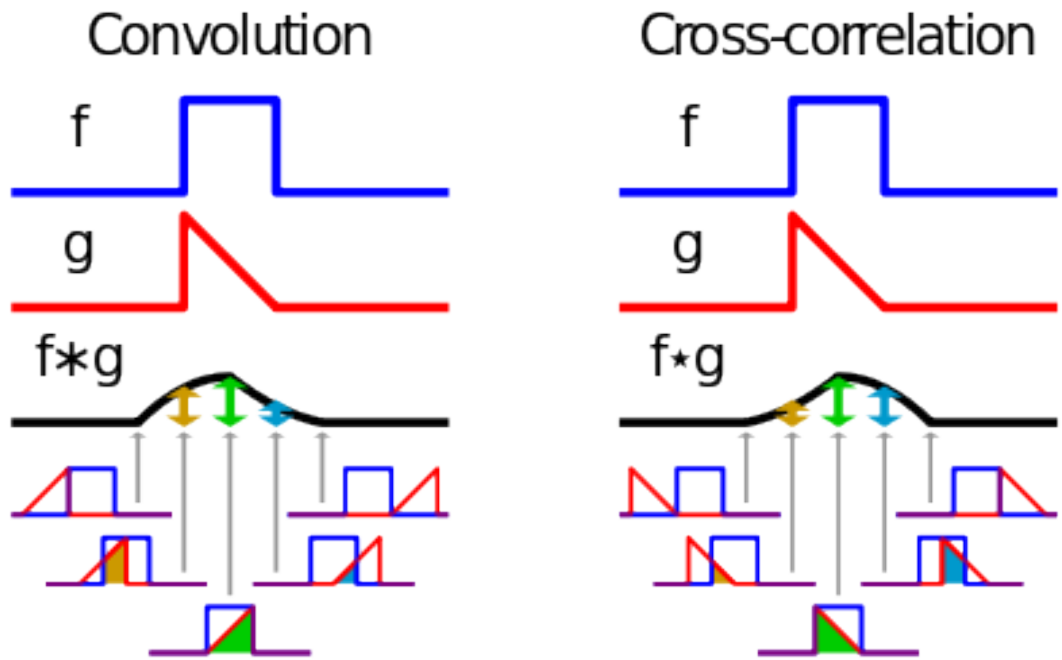
cross-correlation (also called a sliding dot product)

$$I_{out}(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} F(u,v)\ I_{in}(i+u, j+v)$$

unlike convolution, cross-correlation is not commutative
("convolutional neural networks" often implement cross-correlation

convolution

$$I_{out}(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} F(u,v)\ I_{in}(i-u, j-v)$$

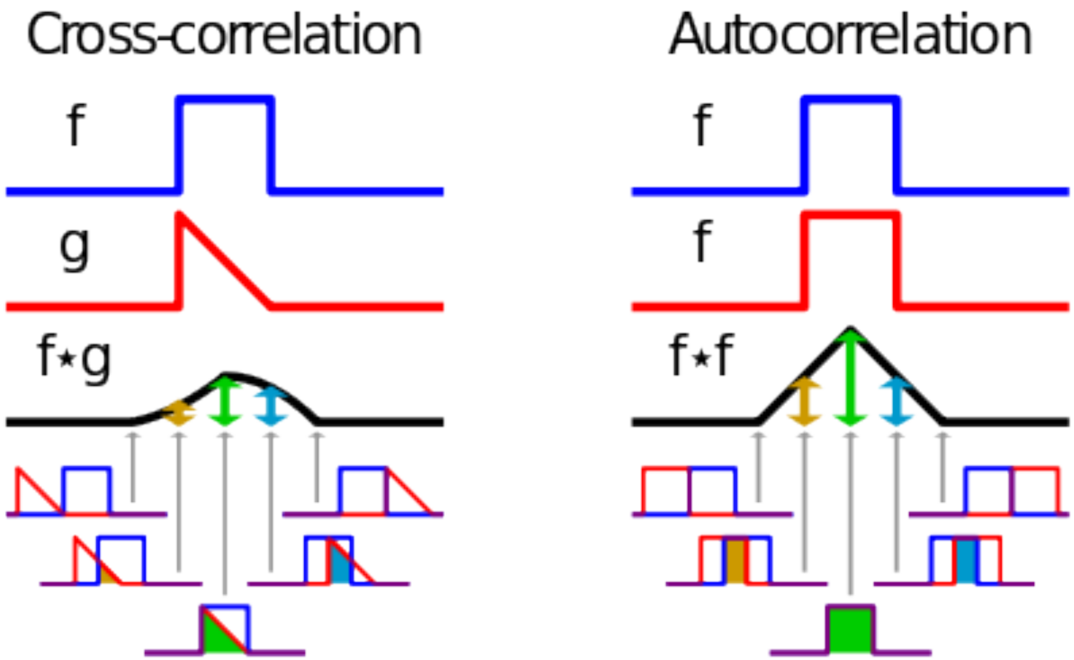


cross-correlation

$$I_{out}(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} F(u,v)\ I_{in}(i+u, j+v)$$

unlike convolution, cross-correlation is not commutative
("convolutional neural networks" often implement cross-correlation

auto-correlation

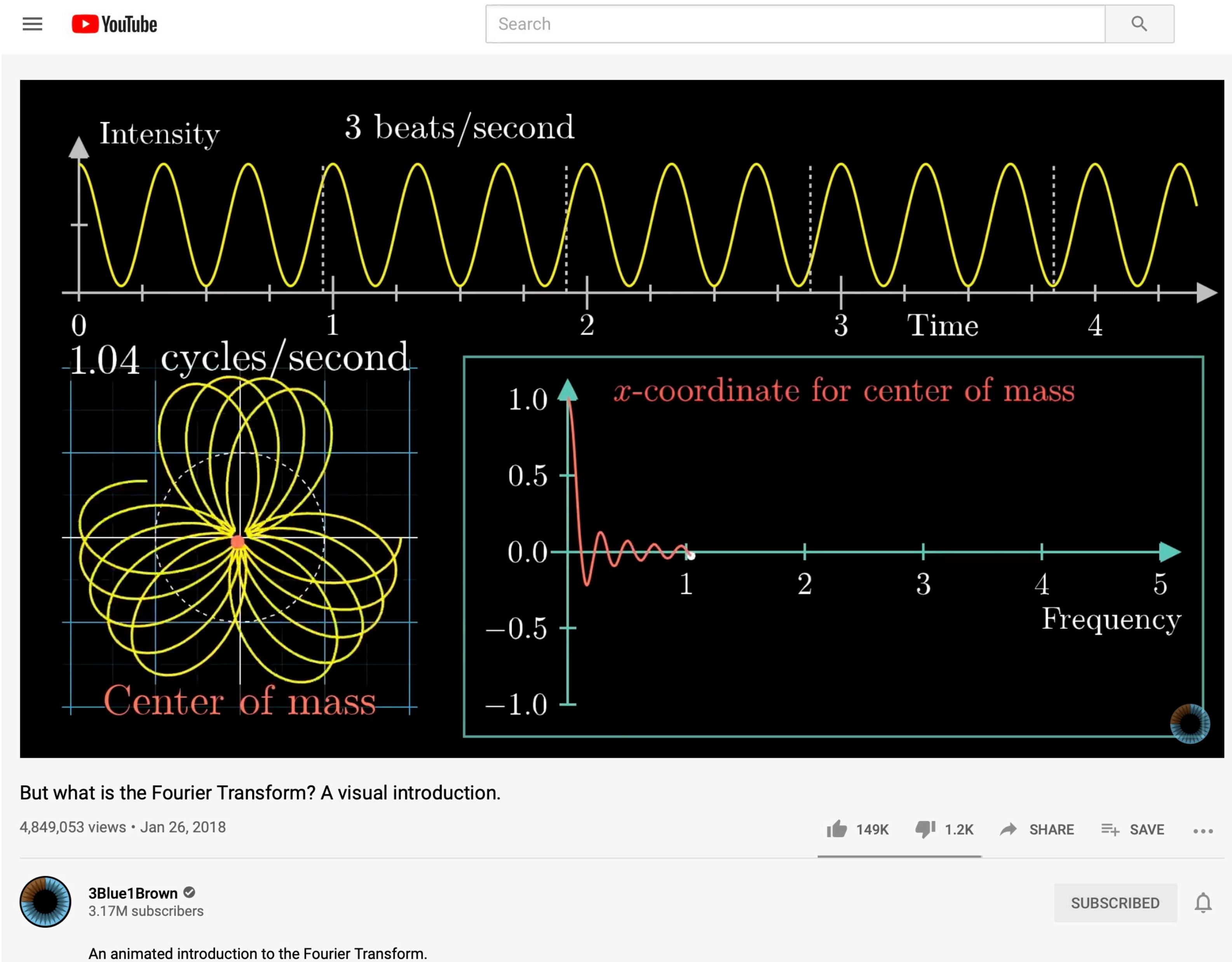


auto-correlation is a cross-correlation of a signal with itself.

if a signal is completely random, it should only autocorrelate at lag = 0 (i.e., when the two signals overlap perfectly)

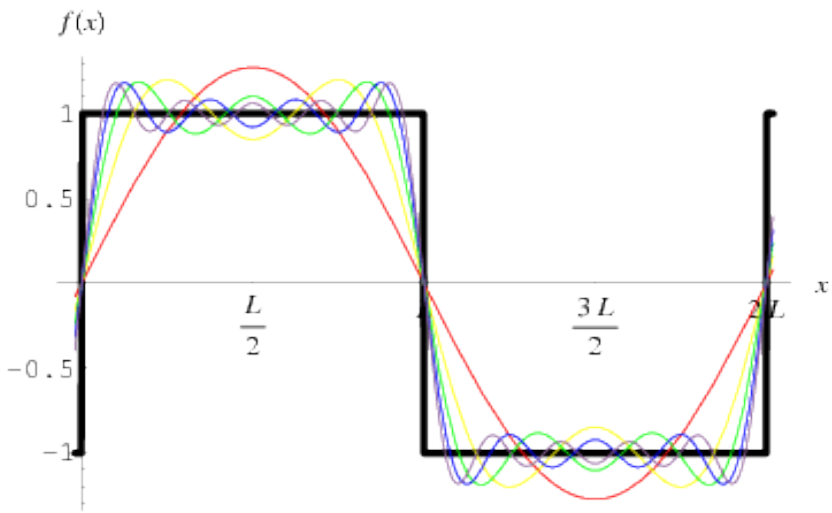
Fourier Analysis

https://en.wikipedia.org/wiki/Fourier_series https://www.youtube.com/watch?v=spUNpyF58BY

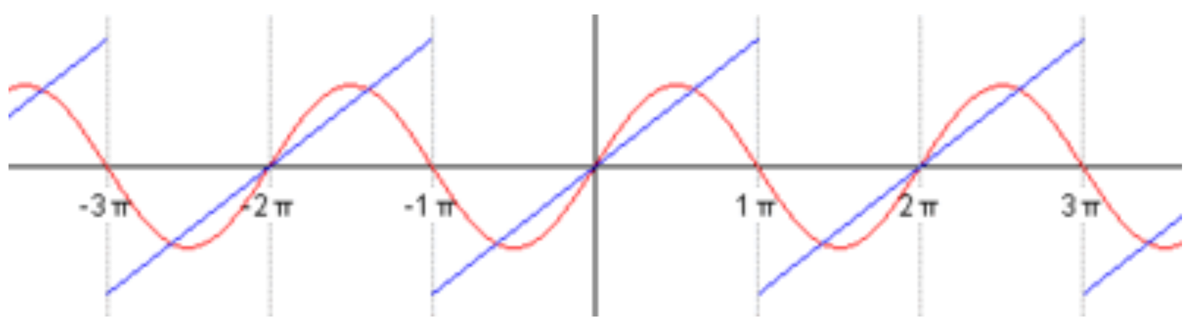


Fourier Analysis

(continuous)
any function (1D signal, 2D image, etc.) can be expressed as linear combination of sinusoids having some amplitude and phase



see Python example



courtesy of Wikipedia

Fourier Analysis

(continuous)
any function (1D signal, 2D image, etc.) can be expressed as linear combination of sinusoids having some amplitude and phase

Fourier Analysis takes some function (1D signal, 2D image, etc.) and decomposes it into a linear combination of sinusoids having some amplitude and phase

Fourier Analysis

time domain

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)\ e^{-2\pi i \omega t}\ dt$$

$$f(t) = \int_{-\infty}^{+\infty} F(\omega)\ e^{2\pi i \omega t}\ d\omega$$

frequency domain

Euler's Formula

$$e^{it} = \cos(t) + i \sin(t)$$

Fourier Analysis

space domain

$$F(\omega) = \int_{-\infty}^{+\infty} f(x)\ e^{-2\pi i \omega x}\ dx$$

$$f(x) = \int_{-\infty}^{+\infty} F(\omega)\ e^{2\pi i \omega x}\ d\omega$$

frequency domain

Euler's Formula

$$e^{ix} = \cos(x) + i\sin(x)$$

Fourier Analysis

$$F(\omega) = \int_{-\infty}^{+\infty} \underbrace{f(x)}_{} \; e^{-2\pi i \omega x} \, dx$$

f(x) is our function

Fourier Analysis

$$F(\omega) = \int_{-\infty}^{+\infty} f(x)\ \underbrace{e^{-2\pi i \omega x}}_{\substack{\text{Euler's Formula} \\ e^{ix} = \cos(x) + i\sin(x)}}\ dx$$

f(x) will be expressed as a combination of sines and cosines

http://en.wikipedia.org/wiki/Euler's_formula

Fourier Analysis

$$\underbrace{F(\omega)}_{F(\omega)\text{ is complex}} = \int_{-\infty}^{+\infty} f(x)\ e^{-2\pi i \omega x}\ dx$$

$$F(\omega) = R(\omega) + iI(\omega)$$

Fourier Analysis

$$\underbrace{F(\omega)}_{F(\omega)\text{ is complex}} = \int_{-\infty}^{+\infty} f(x)\, e^{-2\pi i \omega x}\, dx$$

$$F(\omega) = R(\omega) + iI(\omega)$$

Amplitude

Phase

Fourier Analysis

$$F(\omega) = \underbrace{\int_{-\infty}^{+\infty} f(x)\, e^{-2\pi i \omega x}\, dx}$$

F(ω) is complex

$$F(\omega) = R(\omega) + iI(\omega)$$

Amplitude

$$|F(\omega)| = \sqrt{R^2(\omega) + I^2(\omega)}$$

Phase

$$\phi(\omega) = \tan^{-1}\left(\frac{I(\omega)}{R(\omega)}\right)$$

Fourier Analysis

Amplitude

$$A(\omega) = \sqrt{R^2(\omega) + I^2(\omega)}$$

Phase

$$P(\omega) = \tan^{-1}\left(\frac{I(\omega)}{R(\omega)}\right)$$

Fourier Analysis

Amplitude

$$A(\omega) = \sqrt{R^2(\omega) + I^2(\omega)}$$

Phase

$$P(\omega) = \tan^{-1}\left(\frac{I(\omega)}{R(\omega)}\right)$$

How do you get back to the *F(ω)* given *A(ω)* and *P(ω)*?

$$F(\omega) = A(\omega)\big(\cos(P(\omega)) + i\sin(P(\omega))\big)$$

$$F(\omega) = A(\omega)\exp(iP(\omega))$$

$$F(\omega) = pol2cart(P(\omega), A(\omega))$$

A few properties of Fourier Transforms

$$\frac{df(x)}{dx} \Longleftrightarrow i\omega F(\omega)$$

$$\int f(x)\ dx \Longleftrightarrow \frac{F(\omega)}{i\omega}$$

$$f(x) * g(x) \Longleftrightarrow F(\omega)G(\omega)$$