Convolution Intuitive Explanations Mathematics

What is an intuitive explanation for convolution?

16 Answers

Mark Eichenlaub, graduate student in physics
12.5k Views • Upvoted by Hadayat Seddiqi, engineering @ biotech startup
Mark has 40+ answers in Mathematics.

I think Sridhar and David have good answers. I'll talk about the contexts in which I've personally run into convolutions of the form $A^*B(x) \equiv \int_{-\infty}^{\infty} \mathrm{d}y A(y)B(x-y)$

First let's just look at the formula. A and B are just functions. Think of them as some sort of lump.

In the integral, y is just a dummy variable. So we're going to run all along the real line and at each point, look at the value of A.

$B(x-y)$ is the function B translated over to y. So essentially what the integral says to do is look at each point along the real line, make a copy of B, multiply its height by A at the point, and slap the copy down centered on that point. Then add up all those copies of B across the entire number line, and you have the convolution of A and B.

Here are some places you'd want to do that:

Probability

Suppose you roll a 20-sided die (die A) and a 6-sided die (die B) and add the results. What is the probability distribution for this sum?

To figure this out, consider some particular result - say 14. You could get this by rolling 13 and 1, 12 and 2, 11 and 3, etc. In general, to get a result of n, you could roll m and $n-m$ for any number m. You need to take the probability of each of those events and add them. The formula is

$$P_{A+B}(n) = \sum_{m=1}^{20} P_A(m)P_B(n-m)$$

this is the convolution of P_A and P_B.

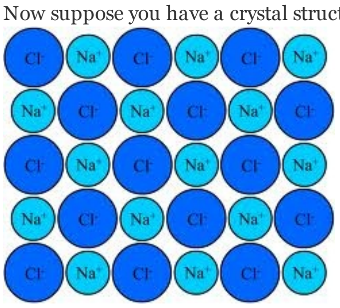
In general, if P_A and P_B are arbitrary probability distributions, the distribution if you sample randomly from each and add the results is

$$P_{A+B}(x) = \int_{y=-\infty}^{\infty} P_A(y)P_B(x-y)$$

This is the convolution formula. It says that we run across all the possibilities for the first variable, take that as a shift for the distribution of the second variable, and add up all the shifted possibilities.

Crystal Structure

Now suppose you have a crystal structure like this:



The locations of all the atoms is a convolution of a single Na-Cl pair with a grid of dots, one at the center of each Cl. The pattern Na-Cl is our B, and the grid of dots is our A, and we slap down a copy of B at each point in A. Note that we are now dealing with two dimensions, so x has become a vector.

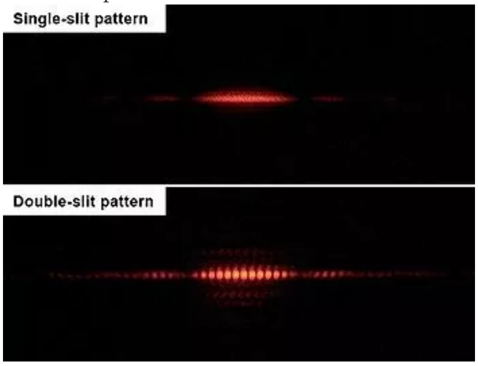
This is a useful way to describe the locations of the atoms because if we want to calculate

Related Questions

What's an intuitive explanation for "Congruence class"?
Convolution: What are the best possible functions to use when intuitively describing convolution to someone?
What is an intuitive explanation of umpolung?
What is an intuitive explanation of what a two-dimensional convolution is?
What are convolution and deconvolution, intuitively?
What is an intuitive explanation of a scheme?
What is an intuitive explanation for Codata?
What is an intuitive explanation of Morita equivalence?
What is an intuitive explanation for eigenface?
What is an intuitive explanation of Groebner bases?

the diffraction pattern from scattering electrons off this crystal, the Convolution theorem lets us say that the diffraction pattern is just the multiple of the diffraction pattern for a single Na-Cl cell with the pattern for the grid of dots.

Similarly, if we shine a laser on a pair of slits for a double slit experiment, the slits are a convolution of a single slit with two dots, one at the center of each slit. The diffraction pattern of two dots is just a series of vertical bars (in a $\sin^2$ intensity). We can just multiply these vertical bars to a single-slit pattern (which is a sinc^2 function) to get the final diffraction pattern of our double slits.



Astronomical observation
Because optics are never perfect, telescopes can't focus light from a single point in the sky to a single point on your detector. Instead, that single point source of light will be spread out in some sort of characteristic pattern called a "point-spread function".

If you have two points of light in the sky, their images on your detector will be two point spread functions added on top each other. If you're looking at some sort of extended object like a nebula, the light on your detector is a convolution of the point spread function with the actual source of light. The point spread function is our function B and the light from the nebula is function A.

Detecting Radiation
Suppose a sample is undergoing random radioactive decay. Each time an ion is emitted, your detector causes a characteristic voltage signal to run through your electronics. Then the voltage as a function of time is the convolution of the function that describes when the decays occur with the characteristic voltage signal.

Exercise
As a silly example to further illustrate the idea, suppose that each day you measure the temperature at 9am. Then for every degree it is outside, you do one pushup and two crunches. Then the exercise you do is the convolution of the temperature with the function (one pushup + 2 crunches).

Here are some examples to test yourself on:

- In 2D, the convolution of a horizontal line segment with a vertical line segment is a rectangle
- In 1D, the convolution of two single square bumps with each other is a symmetric triangular bump as wide as it is high
- $A^*B = B^*A$
- The convolution of an even function with an odd function is odd. The convolution of two odd functions or two even functions is even.
- various examples in the wikipedia article Convolution

And remember, if you're exploring the base of a river and run into the founder of quantum electrodynamics, don't ask him to convert something to a different language. It'll come out garbled because translation is just convolution with a Dirac delta.

Updated Jan 9, 2014 • View Upvotes • Answer requested by Michael Woods

Related Questions
More Answers Below

What's an intuitive explanation for "Congruence class"?

Convolution: What are the best possible functions to use when intuitively describing convolution to someone?

What is an intuitive explanation of umpolung?

What is an intuitive explanation of what a two-dimensional convolution is?

What are convolution and deconvolution, intuitively?

Sridhar Ramesh, Former Ph.D. student in mathematical logic, UC Berkeley
12.7k Views • Upvoted by Vladimir Novakovski, silver medals, IOI 2001 and IPhO 2001
Sridhar has 15 endorsements in Mathematics.

Get out a pen and paper and tell me, what's $(5x^2 + 3x - 9) \times (4x + 1)$?

Very good: it's $20x^3 + 17x^2 - 33x - 9$.

You've just performed a convolution! Convolution is just a fancy word for the operation turning the coefficients of polynomials into the coefficients of their product. The basic rule is that $x^i \times x^j = x^{i+j}$, and $\times$ is linear in each argument. Everything else follows from that.

[We can generalize to the case where polynomials are allowed to have terms with negative or fractional or what have you exponents, but everything still works in just the same way. The output polynomial's coefficient of degree d is given by the sum of the products of all the input polynomials' coefficients over degrees adding up to d.]

Written Jan 4, 2014 • View Upvotes

Máté Kovács, math aficionado
3.1k Views

TL;DR: Imagine that you have two shapes on the plane. You can slide them around freely but you can't rotate them. Once you arrange them somehow, you can measure the area of overlap. If we create a function that tells us how big the overlap is for any possible arrangement, we're basically computing a "toy" convolution of the two shapes. By generalizing this to work not only with clear-cut but also with blurry things (and by flipping one of the shapes), we get "real" convolution. Although it's usually introduced in one dimension, I think it's easier to imagine in two.

Details

Let's build some mathematics to model what we just talked about. We're going to represent a shape F as a function f that assigns a value to each location x in a reference frame attached to the shape. If x is on the shape, then f(x) = 1, otherwise f(x) = 0. In other words, f is the indicator function of the point set F.

So we have two shapes F and G, with representing functions f and g. If we pick any point p on the plane, it has a location in both reference frames. Let's call its location in F's reference frame x, and its location in G's reference frame y. Note that the displacement D = x - y depends only on the arrangement of F and G, and is independent of the choice of p. From now on we can write x - D instead of y.

Now p is in the overlapping part of F and G precisely when both f(x) and g(y) are 1, which can be expressed more compactly by saying that f(x)*g(y) = 1, or equivalently, f(x)*g(x - D) = 1. To measure the area of the overlap for the arrangement given by the displacement D, we just need to integrate that product over all x locations:

$$\text{overlap}(D) = \int_{x \in \mathbb{R}^2} f(x) \cdot g(x - D)\, dx$$

"Real" convolution flips one of the shapes before computing the overlap-function. Moreover, it doesn't restrict the values of the functions f and g to the set {0, 1}, so they should be thought of as representing blurry shapes that don't have clear-cut boundaries.

Updated Oct 16 • View Upvotes

Hadayat Seddiqi, engineering @ biotech startup
4.2k Views • Hadayat has 3 endorsements in Mathematics.

Some great answers here. I thought that showing a simple example for using it in a signal processing context would be easiest to understand. I've included the code and some plots, in case you want to fiddle with it.

I won't explain any of the definitions or details of convolution, since Sridhar gave a really super-intuitive explanation of that, so did David. Mark's answer gives some really interesting application areas that are far more interesting than my example, but I think starting from a simple signal is easiest.

First, define some function:

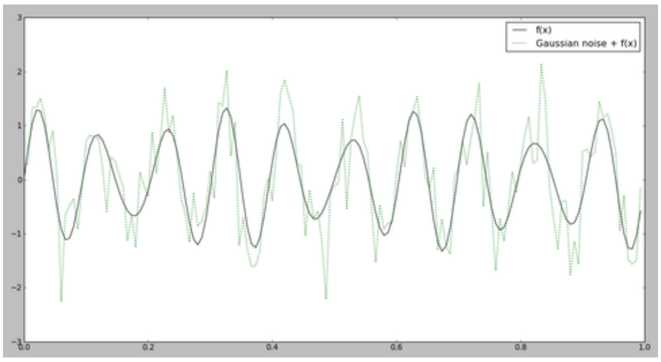
$$F(x) = \sin(2\pi f_1 x) + \sin(2\pi f_2 x)/3$$

```
ff1 = 10
ff2 = 13
f = lambda f1, f2, x: np.sin(2*np.pi*f1*x) + np.sin(2*np.pi*f2*x)/3.0
x = np.arange(0,1,1./150)
y = f(ff1, ff2, x)
```

And then add some Gaussian noise,

```
dy = 0.5 + 1.e-1 * np.random.random(y.shape)
noise = np.random.normal(0, dy)
ynoisy = y + noise
```

Plot those two real quick:



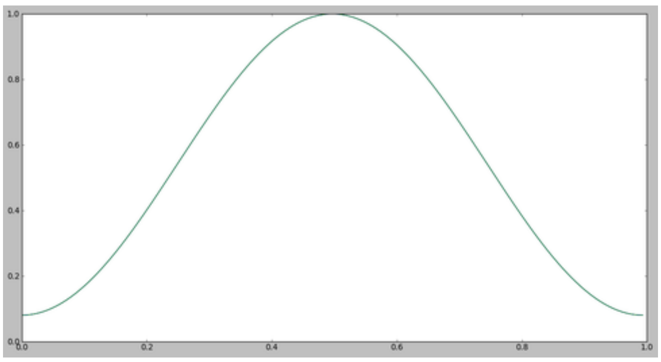
OK cool. Now define a smoothing function that takes the convolution between our noisy function and some "window" (see: Window function):

```
def smooth(x,window_len=11,window='hanning'):
    if x.size < window_len:
        raise ValueError, "Input vector needs to be bigger than window size."
    if window_len<3:
        return x
    s = np.r_[x[window_len-1:0:-1],x,x[-1:-window_len:-1]]
    if window == 'flat': #moving average
        w = np.ones(window_len,'d')
    else:
        w = eval('np.'+window+'(window_len)')
    y = np.convolve(w/w.sum(),s,mode='valid')
    return y[(window_len/2):-(window_len/2)]
```

And then try a couple of window types, like the Hamming window

$W_{hamming}(x) = 0.54 - 0.46\cos\left(\frac{2\pi x}{M-1}\right)$

where M is the window size in the x-direction (which is the second argument to smooth()), which looks like:



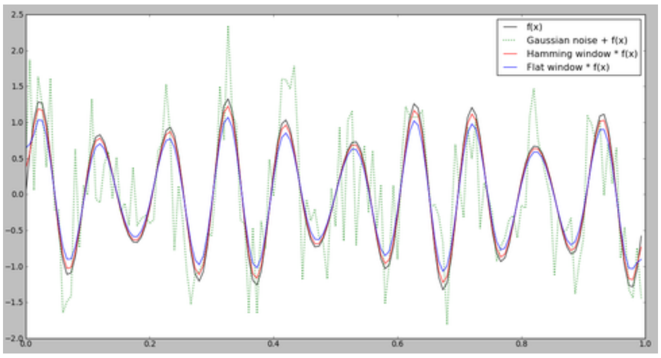
and the flat window

$W_{flat}(x) = 1$

and you know what that looks like (btw, the real geeks might notice that the Hamming window isn't exactly appropriate to use here, but it's just a random example).

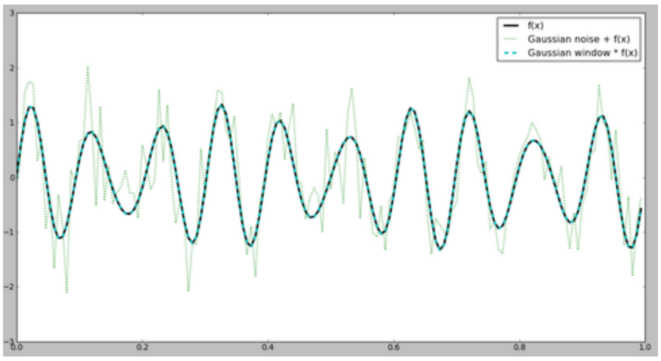
```
# Get a smoothed function
ys_ham = smooth(y, window_len=5, window='hamming')
ys_flat = smooth(y, window_len=5 window='flat')
```

And now plot everything:



Cool! The flat window doesn't do as well because it's just a box, but the Hamming window looks kinda like a Gaussian, so we expect it to do decently well and it does. If you look at the original noisy function by itself, it's almost amazing that such a simple mathematical tool can extract an almost perfect representation of the underlying generating function.

But for general signal smoothing we like to use a Gaussian function for our window. Of course, when I use a Gaussian window, the result exactly overlaps with $F(x)$:

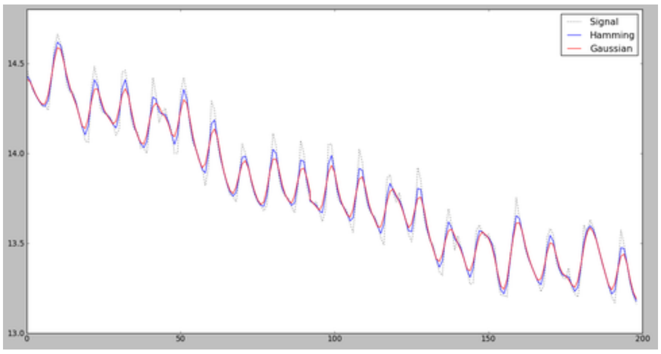


FYI, you can't input 'gaussian' as an argument to smooth(), I had to add an elif to smooth(), here's the line:

```
w = sp.signal.gaussian(window_len, 0.1)
```

and you might say I'm cheating here, and I am, because I know exactly what the variance of the noise was, and the fact that it was Gaussian (to clarify, the noise is in the y-direction, so I'm not saying that it's reversing this). If you make the variance larger, you'll see that it starts to deviate (in fact, if you make it large enough, you'll see that it misses one of the terms in the original function).

And finally, here's an example with real-world data (it's water temperature over a week or so):



It's harder to say here which does better because we don't have the ground truth. Looking at the frequency spectrum of your data can help to decide, but of course it all depends on what you're trying to do.

Anyway, here's the code all in one place (note: you need numpy, scipy, and matplotlib/pylab):

```
import numpy as np
import scipy as sp
import pylab as plt
from scipy import signal
# First define a function
ff1 = 10
ff2 = 13
f = lambda f1, f2, x: np.sin(2*np.pi*f1*x) + np.sin(2*np.pi*f2*x)/3.0
x = np.arange(0,1,1./150)
y = f(ff1, ff2, x)
# Add some noise
dy = 0.5 + 1.e-1 * np.random.random(y.shape)
noise = np.random.normal(0, dy)
ynoisy = y + noise
# Define the smoothing function
def smooth(x,window_len=11,window='hanning', var=0.1):
    """
    Taken from the scipy cookbook:
    http://wiki.scipy.org/Cookbook/SignalSmooth
    """
    if x.size < window_len:
        raise ValueError, "Input vector needs to be bigger than window size."
    if window_len<3:
        return x
    s = np.r_[x[window_len-1:0:-1],x,x[-1:-window_len:-1]]
    if window == 'flat': #moving average
        w = np.ones(window_len,'d')
    elif window == 'gaussian':
        w = sp.signal.gaussian(window_len, var)
    else:
        w = eval('np.'+window+'(window_len)')
    y = np.convolve(w/w.sum(),s,mode='valid')
    return y[(window_len/2):-(window_len/2)]
# Get a smoothed function
wlen = 5
ys_ham = smooth(y, window_len=wlen, window='hamming')
ys_flat = smooth(y, window_len=wlen, window='flat')
ys_gauss = smooth(y, window_len=wlen, window='gaussian', var=0.1)
# Show some plots
plt.plot(x, y, 'k', linewidth=3, label="f(x)")
plt.plot(x, ynoisy, 'g:', linewidth=2, label="Gaussian noise + f(x)")
plt.plot(x, ys_ham, 'r', linewidth=1, label="Hamming window * f(x)")
plt.plot(x, ys_flat, 'b', linewidth=1, label="Flat window * f(x)")
plt.plot(x, ys_gauss, 'c--', linewidth=3, label="Gaussian window * f(x)")
plt.legend()
plt.show()
```

Usually looking at how to implement the math algorithmically helps me to understand it better, so if you're curious about how the convolution itself is implemented, see the Numpy source code on Github: https://github.com/numpy/numpy/b... , although admittedly this one looks pretty cryptic. I would try to help more with this but I'll leave the detective work up to you because my internet is being a real shithead right now. The real thing to remember is that you're working in a discrete domain, so just extend your domain to the (bounded) real line and replace the sum with an integral.

Updated Jan 10, 2014 • View Upvotes

David Greenspan, Meteor dev, Etherpad creator
4.5k Views • Upvoted by Hadayat Seddiqi, engineering @ biotech startup

Suppose you have a special laser pointer that makes a star shape on the wall. You tape together a bunch of these laser pointers in the shape of a square. The pattern on the wall now is the convolution of a star with a square.

Convolution has a lot of applications in undergrad physics, electrical engineering, and signal processing. Instead of the two dimensions of a wall, take the one dimension of time. You have an object on a spring that, when you give it a kick, starts swinging back and forth a little. Under certain assumptions, the system is "linear," meaning that if you give it a second kick, the resulting behavior of the object can be found by summing the response to the first kick alone and the response to the second kick alone. If you are tracking the x-coordinate of the object as a function of time as it swings back and forth, you can graph the response to a kick (an instantaneous force) as a function $x = f(t)$. In fact, there is a single function $f(t)$ representing the "impulse response" of the system, and you can calculate the response of the system to any series of kicks by adding up scaled copies of it. If you express the "kick force" as a function of time, $k(t)$, then the behavior of the object is the convolution of $f(t)$ with $k(t)$. The kick function k could be series of impulses, or it could itself be a continuous function. The object could be in a magnetic field, for example, or connected to something that pulls on it with varying force.

It wasn't until the final exam of my Differential Equations course that I finally "got" what was going on with these convolutions (which I had been working through as calculus problems all term). I began to imagine an infinite number of copies of a function, scaled by different amounts, all being added together, as each instant kicked off a new response in the system that would be added to its behavior for the rest of time.

Voltages and currents in circuits with resistors, capacitors, etc. are linear, so there's a large universe of applicability of this sort of analysis.

Written Jan 4, 2014 • View Upvotes

Related Questions

What is an intuitive explanation of a scheme?
What is an intuitive explanation for Codata?
What is an intuitive explanation of Morita equivalence?
What is an intuitive explanation for eigenface?
What is an intuitive explanation of Groebner bases?
What is an intuitive explanation of the Grothendieck topology?
What is an intuitive explanation of ultrafilters?
What is an intuitive explanation for the Bessel functions?
What is an intuitive explanation of Kaluza-Klein theory?
What is an intuitive explanation to the bicycle pulling puzzle?
Mathematical Expression: What is an intuitive explanation of a closed-form expression?
What's an intuitive explanation of Meusnier's theorem?
What is an intuitive explanation for independence?
What is an intuitive explanation of Bootstrap resampling?
What is an intuitive explanation of pseudoconvexity?

decrease in gun accidents. When people are familiar with something, they are less likely

Read More

superstitious." The con artistry comes into play when psychics don't tell the truth about

Read More

We don't agree on politics. You won't convince me and I won't convince you and

Read More