Questions

Today's exercises are mostly math problems to get you familiar with the convolution operation. This is a key operation that we'll be using for image processing in the first half of the course. Convolution can be challenging to understand, but you will get the hang of it with some practice. (Indeed the word "convoluted" in English means "extremely complex and difficult to follow" and "intricately folded, twisted, or coiled.")

1. Define a *local difference* filter $D(x)$ such that

$$D(x) * I(x) = \frac{1}{2}(I(x+1) - I(x-1))$$

   and a *local average* filter $B(x)$ such that

$$B(x) * I(x) = \frac{1}{4}I(x+1) + \frac{1}{2}I(x) + \frac{1}{4}I(x-1).$$

   (a) Write $D(x)$ and $B(x)$ as a sum of impulse functions.
   (b) What is $D(x) * B(x)$?
   (c) What is $D(x) * D(x)$?
   (d) What is $B(x) * B(x)$?

   Verify your answers in Matlab using `conv`. You'll need to be careful with the indexing of x when you interpret the result. We'll deal with this in a later exercise.

2. Let

$$I(x) * h(x) \;=\; -3\ I(x+2) \;+\; 4\ I(x+1) \;+\; 2\ I(x-2)$$

   What is the function $h(x)$?

3. Let

$$I(t) * f(t) \;=\; 2I(t) - 3I(t-1) + I(t-2).$$

   What is $f(t)$?

4. What is the result of convolving a filter $f(x)$ with a shifted delta function:

$$f(x) * \delta(x - x_0) = \quad ?$$

5. Convolution is easy to compute in Matlab, namely using the functions `conv` (1D) and `conv2` (2D). But the indexing can be a bit tricky to interpret. One issue is that Matlab indexes vectors starting at 1, rather than 0. So, whereas the usual mathematical definition of convolution is:

$$I(x) * f(x) \equiv \sum I(u) f(x-u)$$

   where the u range depends on where $I()$ is defined, Matlab's definition of `conv` needs to be slightly different. Your task in this question is to figure out what the Matlab definition of `conv` is. Doing so successfully will help you understand how convolution works, especially in Matlab.

We will suppose $I(x)$ is defined on 1 to n and $f(x)$ is defined from 1 to m, so the x values of the functions correspond to the Matlab indices.

(a) What would be the range of values of x of the result $I(x) * f(x)$ if we were to use the standard convolution definition above?

(b) How can we tweak the above convolution definition such that the result will be a vector defined on indices x from 1 to $n + m - 1$ (which is what Matlab produces) ?

6. Another issue is that we often want to define functions on negative values of x, for example, the $D(x)$ and $B(x)$ functions mentioned earlier, as well as a Gaussian function.

   Suppose we have a function $g(x)$ that is defined from $x = -m, \ldots, 0, \ldots, m$ and represented by a Matlab vector `g` of size $2m + 1$. Suppose we convolve $g(x)$ with a function $I(x)$ that is defined on $x = 1, \ldots, n$ and is represented by a Matlab vector `I`.

   (a) What is the length of `conv(I,g)`.

   (b) What are the x position values *represented* by the result ? (The subtlety here is that g is defined on both positive and negative x.)

7. Prove the associative law for 1-D convolution,

$$(f * g) * h \;=\; f * (g * h)$$

   You may assume that each of the three functions are defined over all the integers, but only have non-zero values over a finite range.

8. Let $I(x)$ be an a 1D image (signal) and let $n(x)$ be a noise function added to the image. If we blur $I(x) + n(x)$ by convolving with some Gaussian $G(x)$, then the resulting function is equal to the sum of the blurred $I(x)$ and the blurred noise. Why? Which algebraic property of convolution makes this true?

Solutions

1. (a)

$$D(x) = \begin{cases} \frac{1}{2}, & x = -1 \\ -\frac{1}{2}, & x = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$B(x) = \begin{cases} \frac{1}{4}, & x = -1 \\ \frac{1}{2}, & x = 0 \\ \frac{1}{4}, & x = 1 \\ 0, & \text{otherwise} \end{cases}$$

which we can write as

$$D(x) = -\frac{1}{2}\delta(x-1) + \frac{1}{2}\delta(x+1)$$

$$B(x) = \frac{1}{4}\delta(x-1) + \frac{1}{2}\delta(x) + \frac{1}{2}\delta(x+1)$$

(b) By the definition of convolution

$$D(x) * B(x) = \sum D(x - x')B(x')$$

So, we have

$$D(x) * B(x) = \frac{D(x-1)}{4} + \frac{D(x)}{2} + \frac{D(x+1)}{4}$$

You can compute this by hand by drawing a picture of shifted versions of $D()$ and summing:

x	-2	-1	0	1	2
$\frac{1}{2}D(x)$		$\frac{1}{4}$	0	$-\frac{1}{4}$	
$\frac{1}{4}D(x-1)$			$\frac{1}{8}$	0	$-\frac{1}{8}$
$\frac{1}{4}D(x+1)$	$\frac{1}{8}$	0	$-\frac{1}{8}$		
	$\frac{1}{8}$	$\frac{1}{4}$	0	$-\frac{1}{4}$	$-\frac{1}{8}$

$$D(x) * B(x) = \begin{cases} \frac{1}{8}, & x = -2 \\ \frac{1}{4}, & x = -1, \\ -\frac{1}{4}, & x = 1, \\ -\frac{1}{8}, & x = 2 \\ 0, & \text{otherwise} \end{cases}$$

Verify in Matlab by running `conv( [.5 0 -.5], [.25, .5, .25] )`

(c)

$$D(x) * D(x) = \begin{cases} \frac{1}{4}, & x = 2, -2 \\ -\frac{1}{2}, & x = 0, \\ 0, & \text{otherwise} \end{cases}$$

To verify, in Matlab run `conv([.5 0 -.5], [.5 0 -.5])`

(d)

$$B(x) * B(x) = \begin{cases} \frac{1}{16}, & x = 2, -2 \\ \frac{1}{4}, & x = -1, 1 \\ \frac{3}{8}, & x = 0 \\ 0, & \text{otherwise} \end{cases}$$

To verify, in Matlab run `conv([.25 .5 .25], [.25 .5 .25])` Also, notice that the result sums to 1, just as $B(x)$ sums to 1.

2. The function $h(x)$ is

$$h(x) = \begin{cases} -3, & x = -2 \\ 4, & x = -1 \\ 2, & x = 2 \\ 0, & \text{otherwise} \end{cases}$$

We could alternatively write in terms of impulse response functions:

$$h(x) = -3\delta(x+2) + 4\delta(x+1) + 2\delta(x-2)$$

3. $f(0) = 2, f(1) = -3, f(2) = 1$. We could alternatively write in terms of impulse response functions:

$$f(t) = 2\delta(t) - 3\delta(t-1) + \delta(t-2)$$

4. Let $g(x) = \delta(x - x_0)$.

$$\begin{aligned} f(x) * g(x) &= \sum_u f(x-u)g(u) \\ &= \sum_u f(x-u)\delta(u - x_0) \\ &= f(x - x_0)\ . \end{aligned}$$

since the delta function has value 1 when $u = x_0$ and 0 everywhere else. Thus, convolving $f(x)$ with a shifted delta function just shifts $f(x)$ by the same amount as the delta function.

5. (a) Since the u values in the definition of convolution are from 1 to n, the functions $f(x-u)$ are shifted to the right by 1 to n. Moreover, since $f(x)$ itself is defined from $x = 1, \ldots, m$, the resulting weighted sum of these shifted functions $f(x-u)$ will contribute to indices $2 \leq x \leq m+n$, which is a range of length $m+n-1$.

   (b) Instead, we want the convolution to contribution to index positions $1 \leq x \leq m+n-1$. This can be achieved by tweaking the shift positions of the $f()$ functions so that they are all shifted by $u-1$ instead of by u, namely
$$I(x) * f(x) = \sum_{u=1}^{n} I(u) f(x-(u-1))$$
   This is indeed the Matlab definition given in "`doc conv`".

6. (a) The length of `conv(I,g)` is $n + (2m+1) - 1$, since the length of `I` is n and the length of `g` is $2m+1$. (See previous question.)

   (b) The first element in the result vector would represent the smallest x value. This smallest x value would come from the first term $g(x-1)I(1)$ in the mathematical definition of convolution:
$$g(x) * I(x) = \sum_{u=1}^{n} g(x-u) I(u)$$
   But $g(x-1)$ is $g(x)$ shifted to the right by 1, so it would be defined on x values going from $-m+1, \ldots, m+1$. Thus its leftmost value would be at $x = -m+1$. This would be the smallest x value in the result since all other g functions would be shifted by more to the right.

   Similarly, the largest x value would come from the last term $g(x-n)I(1)$ in the mathematical definition of convolution. But $g(x-n)$ is $g(x)$ shifted to the right by n, so it would be defined on x values going from $n-m, \ldots, n+m$. Thus its rightmost value would be at $x = n+m$.

   Thus, the values of `conv(I,g)` would represent the $n+2m+1$ values in $x = -m+1, \ldots, n+m$.

7. All summations in the following are over $\infty, \ldots, \infty$.
$$\begin{aligned}
(f*g)(x) * h(x) &= \sum_{x'} (f*g)(x') \;\; h(x-x') \\
&= \sum_{x'} (\sum_{x''} f(x'') g(x'-x'') \;) \; h(x-x') \\
&= \sum_{x''} f(x'') \sum_{x'} g(x'-x'') \;\; h(x-x') \\
& \quad \text{let } x' - x'' = v \\
&= \sum_{x''} f(x'') \sum_{v} g(v) \;\; h((x-x'')-v) \\
&= \sum_{x''} f(x'') (g*h)(x-x'') \\
&= (f*(g*h)) \; (x)
\end{aligned}$$