Computational Foundations of Cognitive Science

Lecture 15: Convolutions and Kernels

Frank Keller

School of Informatics
University of Edinburgh
keller@inf.ed.ac.uk

February 23, 2010

Definition

Definition: Convolution

If f and g are discrete functions, then $f * g$ is the convolution of f and g and is defined as:

$$(f * g)(x) = \sum_{u=-\infty}^{+\infty} f(u)g(x-u)$$

Intuitively, the convolution of two functions represents the amount of overlap between the two functions. The function g is the *input*, f the *kernel* of the convolution.

Convolutions are often used for filtering, both in the temporal or frequency domain (one dimensional) and in the spatial domain (two dimensional).

Definition

Theorem: Properties of Convolution

If f, g, and h are functions and a is a constant, then:

- $f * g = g * f$ (commutativity)
- $f * (g * h) = (f * g) * h$ (associativity)
- $f * (g + h) = (f * g) + (f * h)$ (distributivity)
- $a(f * g) = (af) * g = f * (ag)$ (associativity with scalar multiplication)

Note that it doesn't matter if g or f is the kernel, due to commutativity.

Convolution of Vectors

If a function f ranges over a finite set of values $\mathbf{a} = a_1, a_2, \dots, a_n$, then it can be represented as vector $\begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}$.

Definition: Convolution of Vectors

If the functions f and g are represented as vectors $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & \dots & a_m \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} b_1 & b_2 & \dots & b_n \end{bmatrix}$, then $f * g$ is a vector $\mathbf{c} = \begin{bmatrix} c_1 & c_2 & \dots & c_{m+n-1} \end{bmatrix}$ as follows:

$$c_x = \sum_u a_u b_{x-u+1}$$

where u ranges over all legal subscripts for a_u and b_{x-u+1}, specifically $u = \max(1, x-n+1) \dots \min(x, m)$.

Convolution of Vectors

If we assume that the two vectors $\mathbf{a}$ and $\mathbf{b}$ have the same dimensionality, then the convolution $\mathbf{c}$ is:

$c_1 = a_1 b_1$
$c_2 = a_1 b_2 + a_2 b_1$
$c_3 = a_1 b_3 + a_2 b_2 + a_3 b_1$
...
$c_n = a_1 b_n + a_2 b_{n-1} + \dots + a_n b_1$
...
$c_{2n-1} = a_n b_n$

Note that the sum for each component only includes those products for which the subscripts are valid.

Convolution of Vectors

Example

Assume $\mathbf{a} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$.

Then

$$\mathbf{a} * \mathbf{b} = \begin{bmatrix} a_1 b_1 \\ a_1 b_2 + a_2 b_1 \\ a_1 b_3 + a_2 b_2 + a_3 b_1 \\ a_2 b_3 + a_3 b_2 \\ a_3 b_3 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 \\ 1 \cdot 1 + 0 \cdot 0 \\ 1 \cdot 2 + 0 \cdot 1 + (-1)0 \\ 0 \cdot 2 + (-1)1 \\ (-1)2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ -1 \\ -2 \end{bmatrix}.$$

Mid-lecture Problem

What happens if the kernel is smaller than the input vector (this is actually the typical case)?

Assume $\mathbf{a} = \begin{bmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$. Compute $\mathbf{a} * \mathbf{b}$.

What is the purpose of the kernel $\mathbf{a}$?

Convolution of Matrices

We can extend convolution to functions of two variables $f(x,y)$ and $g(x,y)$.

Definition: Convolution for Functions of two Variables

If f and g are discrete functions of two variables, then $f * g$ is the convolution of f and g and is defined as:

$$(f*g)(x,y) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} f(u,v)g(x-u,y-v)$$

Convolution of Matrices

We can regard functions of two variables as matrices with $A_{xy} = f(x,y)$, and obtain a matrix definition of convolution.

Definition: Convolution of Matrices

If the functions f and g are represented as the $n \times m$ matrix A and the $k \times l$ matrix B, then $f * g$ is an $(n+k-1) \times (m+l-1)$ matrix C:

$$c_{xy} = \sum_u \sum_v a_{uv} b_{x-u+1,y-v+1}$$

where u and v range over all legal subscripts for a_{uv} and $b_{x-u+1,y-v+1}$.

Note: the treatment of subscripts can vary from implementation to implementation, and affects the size of C (this is parameterizable in Matlab, see documentation of `conv2` function).

Convolution of Matrices

Example

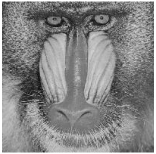
Let $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ and $B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} \end{bmatrix}$.

Then for $C = A * B$, the entry $c_{33} = a_{11}b_{33} + a_{12}b_{32} + a_{13}b_{31} + a_{21}b_{23} + a_{22}b_{22} + a_{23}b_{21} + a_{31}b_{13} + a_{32}b_{12} + a_{33}b_{11}$.

Here, B could represent an image, and A could represent a kernel performing an image operation, for instance.

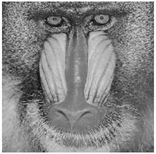
Example: Image Processing

Convolving and image with a kernel (typically a 3×3 matrix) is a powerful tool for image processing.

$B =$  $K * B =$ 

$K = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$ implements a *mean filter* which smooths an image by replacing each pixel value with the mean of its neighbors.

Example: Image Processing

$B =$  $K * B =$ 

The kernel $K = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$ implements the *Sobel edge detector.*

It detects gradients in the pixel values (sudden changes in brightness), which correspond to edges. The example is for vertical edges.

Definition

We can also define convolution for continuous functions. In this case, we replace the sums by integrals in the definition.

Definition: Convolution

If f and g are continuous functions, then $f * g$ is the convolution of f and g and is defined as:

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(u)g(x-u)du$$

Convolutions of continuous functions are widely used in signal processing for filtering continuous signals, e.g., speech.

Definition

Example

Assume the following step functions:

$$g(x) = \begin{cases} 3 & \text{if } 0 \leq x \leq 4 \\ 0 & \text{otherwise} \end{cases} \qquad f(x) = \begin{cases} \frac{1}{2} & \text{if } -1 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

If we integrate $g(x)$, we get: $G(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 3x & \text{if } 0 \leq x \leq 4 \\ 12 & \text{if } x > 4 \end{cases}$.

Then the convolution $f * g$ is:

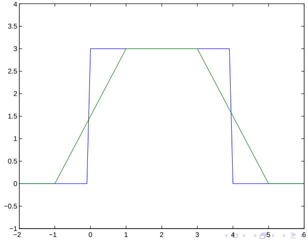
$$(f * g)(x) = \int_{-\infty}^{+\infty} f(u)g(x-u)du = \frac{1}{2}\int_{-1}^{1} g(x-u)du =$$

$$-\frac{1}{2}\int_{x+1}^{x-1} g(u)du = -\frac{1}{2}(G(x-1) - G(x+1)) =$$

$$\begin{cases} \frac{3}{2}(x+1) & \text{if } -1 \leq x < 1 \\ 3 & \text{if } 1 \leq x \leq 3 \\ -\frac{3}{2}(x-1)+6 & \text{if } 3 < x \leq 5 \\ 0 & \text{otherwise} \end{cases}$$

Definition

Function $g(x)$ (blue) and convolution $f * g(x)$ (green):



Example: Signal Processing

Assume we have a function $I(x)$ that represents the intensity of a signal over time. This can be a very spiky function.

We can make this function less spiky by convolving it with a *Gaussian kernel*. This is a kernel given by the Gaussian function:

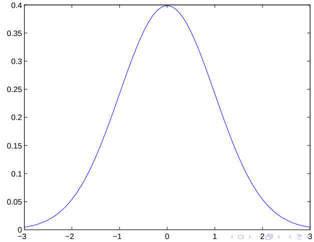
$$G(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

The convolution $G * I$ is a smoothed version of the original intensity function.

We will learn more about the Gaussian function (aka normal distribution) in the second half of this course.

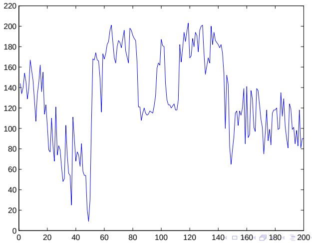
Example: Signal Processing

Gaussian function $G(x)$:



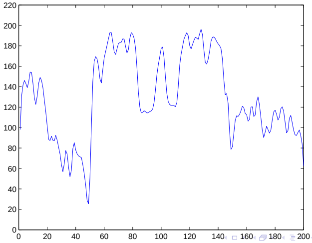
Example: Signal Processing

Original intensity function $I(x)$:



Example: Signal Processing

Smoothed function obtained by convolution with a Gaussian kernel:



Summary

- The convolution $(f * g)(x) = \sum f(u)g(x-u)$ represents the overlap between a discrete function g and a kernel f;
- convolutions in one dimension can be represented as vectors, convolutions in two dimensions as matrices;
- in image processing, two dimensional convolution can be used to filter an image or for edge detection;
- for continuous functions, convolution is defined as $(f * g)(x) = \int f(u)g(x-u)du$;
- this can be used in signal processing, e.g., to smooth a signal.