2. Convolutional Neural Networks

Introduction to Convolution Neural Networks

0:00 / 0:00 1.50x

Video
Download video file

Transcripts
Download SubRip (.srt) file
Download Text (.txt) file

Motivation for CNN

2/2 points (graded)

Let's suppose that we wish to classify images of 1000×1000 dimensions.

We wish to pass the above input through a feed-forward neural network with a single hidden layer made up of 1000×1000 hidden units each of which is fully connected to the full image.

If the number of connections that exist between the first hidden layer and the input image is given by x, then enter below the value of $log_{10}(x)$, i.e. the logarithm of x to the base 10:

12 ✔ **Answer:** 12

Instead of a fully-connected layer, now suppose that we use a convolutional layer with 1 filter of shape 11×11 instead. Enter below the number of parameters in the first layer (ignoring the bias terms):

121

✔ **Answer:** 121

Solution:

Each of the hidden unit is connected to all the pixels from the input image.
So, there are a total of $1000 * 1000 = 10^6$ connections between each of the hidden layers and the input.
Since there are $1000 * 1000 = 10^6$ hidden units in the first hidden layer, the total number of connections x amounts to $x = 10^6 * 10^6 = 10^{12}$
The first convolutional layer with a 11×11 filter will have $11 * 11 = 121$ parameters that operate on the entire image.

Submit You have used 2 of 3 attempts

Answers are displayed within the problem

Second Motivation for CNN

1/1 point (graded)
Suppose a feed-forward, non-convolutional neural network is learning how to classify images. Then, it can classify images even if the relevant object is in a different part of the image.

- true
- false

✔

Solution:

The lecture explains this with a mushroom example. If the mushroom is in a different place, the weight matrix parameters at that location need to learn to recognize the mushroom anew. With convolutional layers, we have translational invariance as the same filter is passed over the entire image. Therefore, it will detect the mushroom regardless of location

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

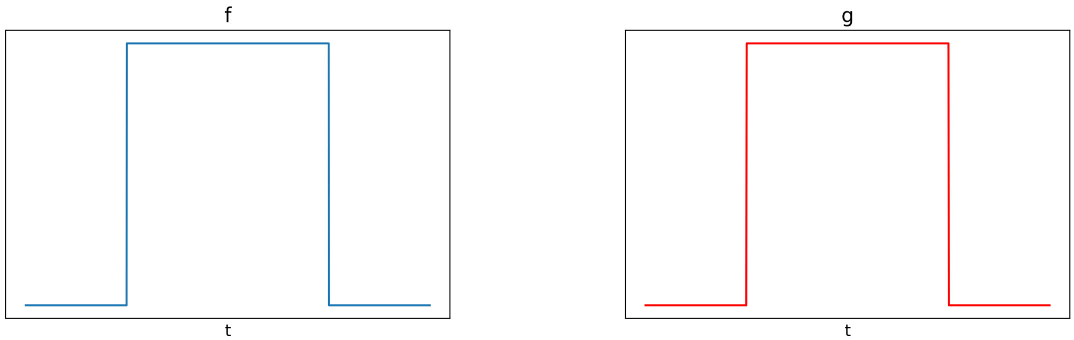
Convolution: Continuous Case

2/2 points (graded)
In the lecture we saw the example of using the convolution operation to create a feature map. Here we formally define the convolution as an operation between 2 functions f and g:

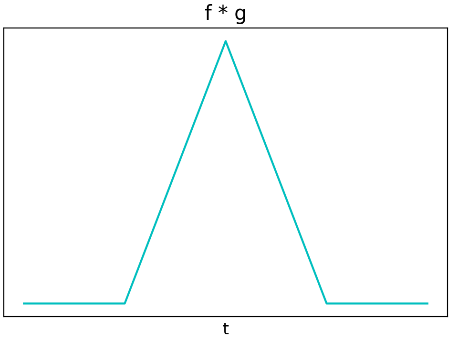
$$(f * g)(t) \equiv \int_{-\infty}^{+\infty} f(\tau)\, g(t-\tau)\, d\tau$$

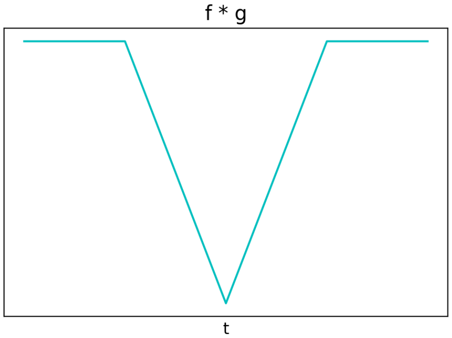
In this integral, τ is the dummy variable for integration and t is the parameter. Intuitively, convolution 'blends' the two function f and g by expressing the amount of overlap of one function as it is shifted over another function.

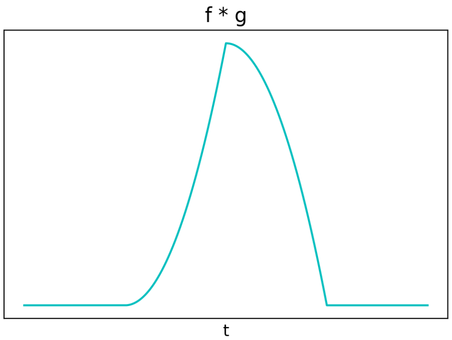
Now, suppose we are given two rectangular function f and g as shown in the figures below.

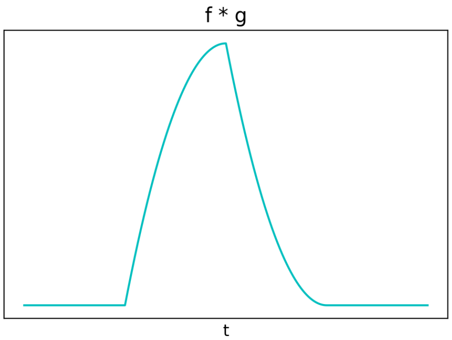


What is the shape of of $f * g$?









What is the area under the convolution: $\int_{-\infty}^{+\infty} (f * g)\, dt$

- ◯ The area under f
- ◯ The area under g
- ◉ The product of the areas under f and g
- ◯ The sum of the areas under f and g

✔

Solution:

We can flip g and shift it over f. $f * g$ stays at 0 when there's no overlap. It increse linearly and reach the peak when f and g fully overlap with each other.
The area under the convolution:

$$\begin{aligned}\int_{-\infty}^{+\infty} (f * g)\, dt &= \int_{-\infty}^{+\infty} [\int_{-\infty}^{+\infty} f(\tau)\, g(t-\tau)\, d\tau]\, dt \\ &= \int_{-\infty}^{+\infty} f(\tau) [\int_{-\infty}^{+\infty} g(t-\tau)\, dt]\, d\tau \\ &= [\int_{-\infty}^{+\infty} f(\tau)\, d\tau] [\int_{-\infty}^{+\infty} g(t)\, dt]\end{aligned}$$

This is the product of the areas under f and g.

Submit You have used 2 of 3 attempts

Answers are displayed within the problem

Convolution: 1D Discrete Case

2.0/2 points (graded)
Similarly, for discrete functions, we can define the convolution as:

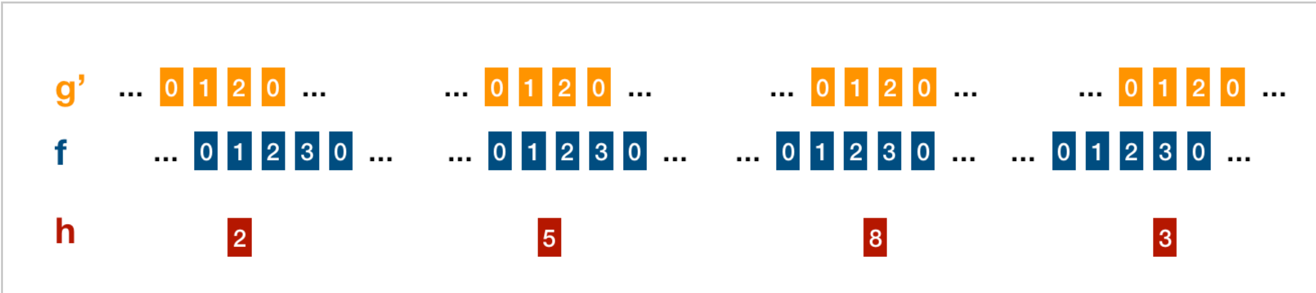
$$(f * g)[n] \equiv \sum_{m=-\infty}^{m=+\infty} f[m]\, g[n-m]$$

Here, we give an example of convolution on 1D discrete signal.
Let $f[n] = [1, 2, 3]$, $g[n] = [2, 1]$ and suppose n starts from 0. We are computing $h[n] = f[n] * g[n]$.
As f and g are finite signals, we just put 0 to where f and g are not defined. This is usually called zero padding. Now, let's compute $h[n]$ step by step:

$$\begin{aligned}h[0] &= f[0] \cdot g[0-0] + f[1] \cdot g[0-1] + \cdots = f[0] \cdot g[0] = 2 \\ h[1] &= f[0] \cdot g[1-0] + f[1] \cdot g[1-1] + f[2] \cdot g[1-2] + \cdots = f[0] \cdot g[1] + f[1] \cdot g[0] = 5 \\ h[2] &= f[0] \cdot g[2-0] + f[1] \cdot g[2-1] + f[2] \cdot g[2-2] + f[3] \cdot g[2-3] + \cdots = f[1] \cdot g[1] + f[2] \cdot g[0] = 8 \\ h[3] &= f[0] \cdot g[3-0] + f[1] \cdot g[3-1] + f[2] \cdot g[3-2] + f[3] \cdot g[3-3] + f[4] \cdot g[3-4] + \cdots = f[2] \cdot g[1] = 3\end{aligned}$$

The other parts of h are all 0.

Intuitively, we can get this result by first flipping g[n] and shift it over f[n] and compute the inner product at each step, as shown in the figures below:



In practice, it is common to call the flipped g' as filter or kernel, for the input signal or image f.

As we forced to pad zeros to where the input are not defined, the result on the edge of the input may not be accurate. To avoid this, we can just keep the convolution result where the input f is actually defined. That is $h\,[n] = [5, 8]$.

Now suppose the input $f = [1, 3, -1, 1, -3]$, and the filter $g' = [1, 0, -1]$, what is the convolutional output of $f * g$ without zero padding on f? Enter your answer as a list below (e.g. [0,0,0])

[2, 2, 2]

✔ **Answer:** [2,2,2]

What is the convolutional output of $f * g$ if we pad a 0 on both edges of f so that the output dimension is the same as the input? Enter your answer as a list below (e.g. [0,0,0,0,0])

[-3, 2, 2, 2, 1]

✔ **Answer:** [-3,2,2,2,1]

Solution:

Without zero padding, we have

$$\begin{aligned} f*g\,(0) &= 1\times 1 + 3\times 0 + (-1)\times(-1) = 2 \\ f*g\,(1) &= 3\times 1 + (-1)\times 0 + 1\times(-1) = 2 \\ f*g\,(2) &= (-1)\times 1 + 1\times 0 + (-3)\times(-1) = 2 \end{aligned}$$

With zero padding, we add 0 to f such that $f = [0, 1, 3, -1, 1, -3, 0]$,

$$\begin{aligned} f*g\,(0) &= 0\times 1 + 1\times 0 + 3\times(-1) = -3 \\ f*g\,(1) &= 1\times 1 + 3\times 0 + (-1)\times(-1) = 2 \\ f*g\,(2) &= 3\times 1 + (-1)\times 0 + 1\times(-1) = 2 \\ f*g\,(3) &= (-1)\times 1 + 1\times 0 + (-3)\times(-1) = 2 \\ f*g\,(4) &= 1\times 1 + (-3)\times 0 + 0\times(-1) = 1 \end{aligned}$$

Submit

You have used 3 of 5 attempts

Answers are displayed within the problem

Convolution: 2D Discrete Case

1/1 point (graded)

Now, let's apply the same idea on images, which are 2D discrete signals. Suppose we had an image f and a filter g' as shown below. Calculate the sum of the elements in the output matrix after passing the image through the convolutional filter, without zero padding.

$$f = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$g' = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

15

Answer: 15

Solution:

We align the filter with the top left corner of the image, and take the element wise multiplication of the filter and the 2 by 2 square in the top left corner. We then shift the filter along the top row, doing the same thing. We then apply the same procedure to the next row. If we went another row down, the bottom row of the filter would not have any numbers to be multiplied with. Thus, we stop.

The result of the convolution is

$$C = \begin{bmatrix} 4 & 4 \\ 4 & 3 \end{bmatrix}$$

. The sum is therefore 15

Submit

You have used 2 of 3 attempts

Answers are displayed within the problem

Pooling Practice

1/1 point (graded)

A pooling layer's purpose is to pick up on a feature regardless of where it appears in the image.

- (•) true
- ( ) false

Solution:

A pooling layer finds the maximum value over a given area. The max value can be seen as a "signal" representing whether or not the feature exists. For example, a high max value could indicate that the feature did appear in the image.

Submit

You have used 1 of 2 attempts

Answers are displayed within the problem

Discussion

Hide Discussion

Topic: Unit 3 Neural networks (2.5 weeks):Lecture 12. Convolutional Neural Networks / 2. Convolutional Neural Networks

Add a Post

Show all posts by recent activity

Convolution confusion
I found the explanation for convolution very confusing despite having a basic understanding of the concept. [Here's a videoon the topic from the probability course... 1

Question Convolution: 1D Discrete Case. f[0]
Is f[0] = 1? meaning 0 refers to the 1st element. 2

Input format for 1D case
I have tried as single number for sum, and array with [...], with [[...]], with ([...]) and with ([[...]]). I get Matrix entry is forbidden or Expected answer to be a vector, but... 2

Correct syntax for matrix notation
What is syntax for the result of the convolution matrix in case 2D ?. 2

[Staff] the transcript is not in sync with the the video
very confusing if you're trying to follow it along with the video. Kindly fix. 1

Learn About Verified Certificates