

Deep Learning

Theoretical Exercises – Week 12 – Chapter 9

Exercises on the book "Deep Learning" written by Ian Goodfellow,
Yoshua Bengio, and Aaron Courville.

Exercises and solutions by T. Méndez and G. Schuster

FS 2024

1 Exercises on Convolutional Networks

1. Convolution and Correlation:

- (a) What is the difference between convolution and correlation?
- (b) Which of the two operations is normally used in convolutional networks?
- (c) Why does it not matter whether convolution or correlation is used in convolutional networks?

Solution:

- (a) Convolution differs from correlation only by the flipping of the kernel. Hence, a convolution is a correlation with a flipped kernel. If the kernel is symmetric, convolution is the same as correlation.
 - (b) In convolution networks a correlation is actually performed and NOT a convolution. Nevertheless, they are called convolutional networks.
 - (c) In the context of machine learning, the learning algorithm will learn the appropriate values of the kernel in the appropriate place, so an algorithm based on convolution (with kernel flipping) will learn a kernel that is flipped relative to the kernel learned by an algorithm based on correlation (without kernel flipping). Consequently, it does not matter which operation (convolution or correlation) is used as long as the operations are not mixed.
2. What are the three stages of a convolutional layer? Describe each stage and explain what it is used for.

Solution:

• **1. Stage: Convolution Stage**

This stage performs the convolution operation (or correlation respectively). Thereby, one output is only connected to a few inputs (sparse connectivity) and the weights

for calculating the outputs are identical for all outputs (tied weights). Hence, only one parameter set (per feature) is learned for every location, which is very well suited, for example, for the detection of edges in images, since the same edges can appear everywhere in the image.


- **2. Stage: Detector Stage**

This stage performs the necessary nonlinear transformation by running each pre-activation through a nonlinear activation function, such as the rectified linear activation function.

- **3. Stage: Pooling Stage**

This stage replaces the output of the detector stage at a certain location with a summary statistic (e.g. the maximum value) of the nearby outputs. This helps to make the representation approximately invariant to small translations of the input. Invariance to local translation can be a very useful property if we care more about whether some feature is present than exactly where it is.

3. Given is the image

$$X = \begin{bmatrix} 1 & 6 & 6 & 6 & 6 & 1 \\ 1 & 6 & 10 & 10 & 6 & 1 \\ 1 & 6 & 10 & 10 & 6 & 1 \\ 1 & 6 & 6 & 6 & 6 & 1 \end{bmatrix},$$


which is passed through the convolution stage with the kernel

$$K = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$


that detects vertical edges. The origin of the kernel is in the middle of the kernel.


- Calculate the output of the convolution stage for all three cases of zero-padding (*valid*, *same* and *full*).
- For the zero-padding case *same* also calculate the output of the detector stage and the output of the pooling stage. For the detector stage use the rectified linear activation function and for the pooling stage use max-pooling, which takes the maximum output within a (non-overlapping) neighborhood of 2×2 .

Solution:

- In the case *valid*, zero-padding is not used at all and the convolution kernel is only allowed to visit positions where the entire kernel is completely contained in the image. With image X and kernel K this is only possible for the following 8 inner positions:

$$X = \begin{bmatrix} 1 & 6 & 6 & 6 & 6 & 1 \\ 1 & \textcircled{6} & \textcircled{10} & \textcircled{10} & \textcircled{6} & 1 \\ 1 & \textcircled{6} & \textcircled{10} & \textcircled{10} & \textcircled{6} & 1 \\ 1 & 6 & 6 & 6 & 6 & 1 \end{bmatrix}.$$

Hence, the output of the convolution stage is

$$A_{\text{valid}} = \begin{bmatrix} 23 & 8 & -8 & -23 \\ 23 & 8 & -8 & -23 \end{bmatrix},$$


where

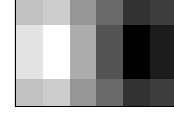
$$\begin{aligned}
23 &= -1 \cdot (1 + 1 + 1) + 0 \cdot (6 + 6 + 6) + 1 \cdot (6 + 10 + 10), \\
8 &= -1 \cdot (6 + 6 + 6) + 0 \cdot (6 + 10 + 10) + 1 \cdot (6 + 10 + 10), \\
-8 &= -1 \cdot (6 + 10 + 10) + 0 \cdot (6 + 10 + 10) + 1 \cdot (6 + 6 + 6), \\
-23 &= -1 \cdot (6 + 10 + 10) + 0 \cdot (6 + 6 + 6) + 1 \cdot (1 + 1 + 1).
\end{aligned} \tag{1.1}$$

In the case *same*, just enough zero-padding is added to keep the size of the output equal to the size of the input. For the 3×3 -Kernel K this means that one row/column of zeros has to be added on all sides:

$$\tilde{X} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 6 & 6 & 6 & 6 & 1 & 0 \\ 0 & 1 & 6 & 10 & 10 & 6 & 1 & 0 \\ 0 & 1 & 6 & 10 & 10 & 6 & 1 & 0 \\ 0 & 1 & 6 & 6 & 6 & 6 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Thus, the output of the convolution stage is

$$A_{same} = \begin{bmatrix} 12 & 14 & 4 & -4 & -14 & -12 \\ 18 & 23 & 8 & -8 & -23 & -18 \\ 18 & 23 & 8 & -8 & -23 & -18 \\ 12 & 14 & 4 & -4 & -14 & -12 \end{bmatrix}.$$

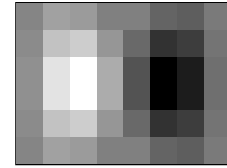


In the last case *full*, as much zero-padding is added as needed to perform a full correlation. This means that each pixel in X must be weighted once by every weight in K . For the 3×3 kernel K this means that two rows/columns of zeros have to be added on all sides:

$$\tilde{X} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 6 & 6 & 6 & 6 & 1 & 0 & 0 \\ 0 & 0 & 1 & 6 & 10 & 10 & 6 & 1 & 0 & 0 \\ 0 & 0 & 1 & 6 & 10 & 10 & 6 & 1 & 0 & 0 \\ 0 & 0 & 1 & 6 & 6 & 6 & 6 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

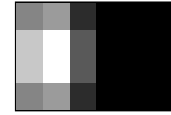
Thus, as output of the convolution stage results

$$A_{full} = \begin{bmatrix} 1 & 6 & 5 & 0 & 0 & -5 & -6 & -1 \\ 2 & 12 & 14 & 4 & -4 & -14 & -12 & -2 \\ 3 & 18 & 23 & 8 & -8 & -23 & -18 & -3 \\ 3 & 18 & 23 & 8 & -8 & -23 & -18 & -3 \\ 2 & 12 & 14 & 4 & -4 & -14 & -12 & -2 \\ 1 & 6 & 5 & 0 & 0 & -5 & -6 & -1 \end{bmatrix}.$$



- (b) For the case *same* also the output of the detector stage and the pooling stage has to be calculated. Since the rectified linear activation function is used in the detector stage, the output of the detector stage is

$$Y_{same} = \max\{A_{same}, 0\} = \begin{bmatrix} 12 & 14 & 4 & 0 & 0 & 0 \\ 18 & 23 & 8 & 0 & 0 & 0 \\ 18 & 23 & 8 & 0 & 0 & 0 \\ 12 & 14 & 4 & 0 & 0 & 0 \end{bmatrix}.$$



At the last stage, the pooling stage, four values are combined into one value

$$Y_{same} = \begin{bmatrix} \boxed{12} & \boxed{14} & \boxed{4} & \boxed{0} & \boxed{0} & \boxed{0} \\ \boxed{18} & \boxed{23} & \boxed{8} & \boxed{0} & \boxed{0} & \boxed{0} \\ \boxed{18} & \boxed{23} & \boxed{8} & \boxed{0} & \boxed{0} & \boxed{0} \\ \boxed{12} & \boxed{14} & \boxed{4} & \boxed{0} & \boxed{0} & \boxed{0} \end{bmatrix},$$

whereby the new value corresponds to the maximum of the old values. Thus the output of the pooling stage is

$$P_{same} = \begin{bmatrix} 23 & 8 & 0 \\ 23 & 8 & 0 \end{bmatrix}.$$



4. Exercise 13.3 from the book "Digital Image Processing" Rafael C. Gonzalez and Richard E. Woods:

Consider a CNN whose inputs are RGB color images of size 512×512 pixels. The network has two convolutional layers. Using this information, answer the following:

- You are told that the spatial dimensions of the feature maps in the first layer are 504×504 , and that there are 12 feature maps in the first layer. Assuming that no zero-padding is used, and that the kernels used are square, and of an odd size, what are the spatial dimensions of these kernels?
- If subsampling is done using neighborhoods of size 2×2 , what are the spatial dimensions of the pooled feature maps in the first layer?
- What is the depth (number) of the pooled feature maps in the first layer?
- The spatial dimensions of the convolution kernels in the second layer are 3×3 . Assuming no zero-padding, what are the spatial dimensions of the feature maps in the second layer?
- You are told that the number of feature maps in the second layer is 6, and that the size of the pooling neighborhoods is again 2×2 . What is the length of the vector that results from flattening the last layer of the CNN?

Solution:

- If the kernels are square ($w \times w$), of odd size, and the square dimensions of the convolution planes is 504, then

$$512 - 2 \frac{w - 1}{2} = 504$$

ans thus

$$w = 512 - 504 + 1 = 9.$$

So the spatial dimensions of the kernel are 9×9 .

- (b) Because the spatial dimensions of the subsampling kernels are 2×2 , the size of the feature planes is reduced by $\frac{1}{2}$. Therefore, the pooled feature planes are of size

$$\frac{504}{2} \times \frac{504}{2} = 252 \times 252.$$

- (c) The depth (number) of pooled feature maps in a layer is equal to the number of feature maps in that layer, thus 12.
- (d) According to (a) the spatial dimensions of the feature maps in the second layer are

$$\left(252 - 2 \frac{3-1}{2}\right) \times \left(252 - 2 \frac{3-1}{2}\right) = 250 \times 250.$$

- (e) We know from (d) that the spatial dimensions of the feature maps in the second layer are 250×250 elements. The pooling neighborhoods are of size 2×2 , so the spatial dimensions of the pooled feature maps are 125×125 . The number of pooled feature maps is the same as the number of feature maps which we are given is 6. Thus, the length of the flattened vector is $125 \times 125 \times 6 = 93\,750$.

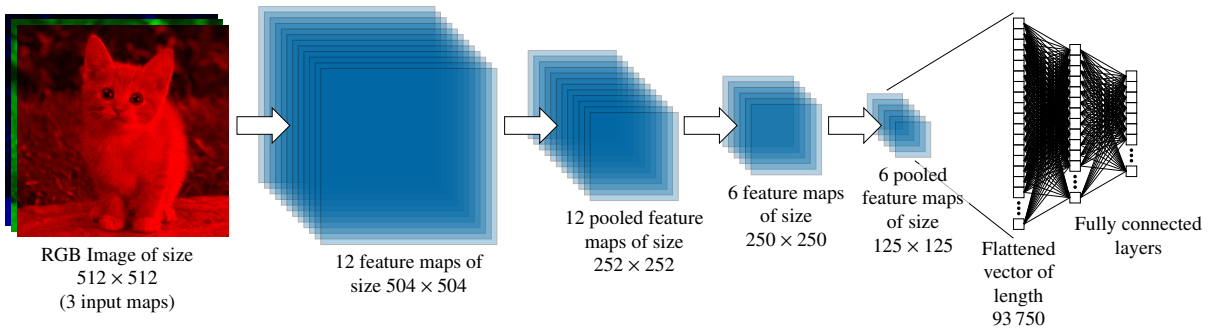


Figure 1: Visualization of the architecture of the CNN described in exercise 4.