Convolutional Neural Networks

Abstract

Keywords:

There are no related labels.

Motivation for Convolutional Neural Networks

Finding good internal representations of images objects and features has been the main goal since the beginning of computer vision. Therefore many tools have been invented to deal with images. Many of these are based on a mathematical operation, called convolution. Even when Neural Networks are used to process images, convolution remains the core operation.

Convolutional Neural Networks finally take the advantages of **Neural Networks (link to Neural Networks)** in general and goes even further to deal with two-dimensional data. Thus, the training parameters are elements of two-dimensional filters. As a result of applying a filter to an image a feature map is created which contains information about how well the patch corresponds to the related position in the image.

Additionally, convolution connects perceptrons locally. Because features always belong to their spatial position in the image, there is no need to fully connect each stage with each other. Convolving preserves information about the surrounding perceptrons and processes them according to their corresponding weights. In each stage, the data is additionally processed by a non-linearity and a rectification. In the end, pooling subsamples each layer.

Deep learning finally leads to multiple trainable stages, so that the internal representation is structured hierarchically. Especially for images, it turned out that such a representation is very powerful. Low-level stages are used to detected primary edges. High-level stages lastly connect information on where and how objects are positioned regarding the scene.

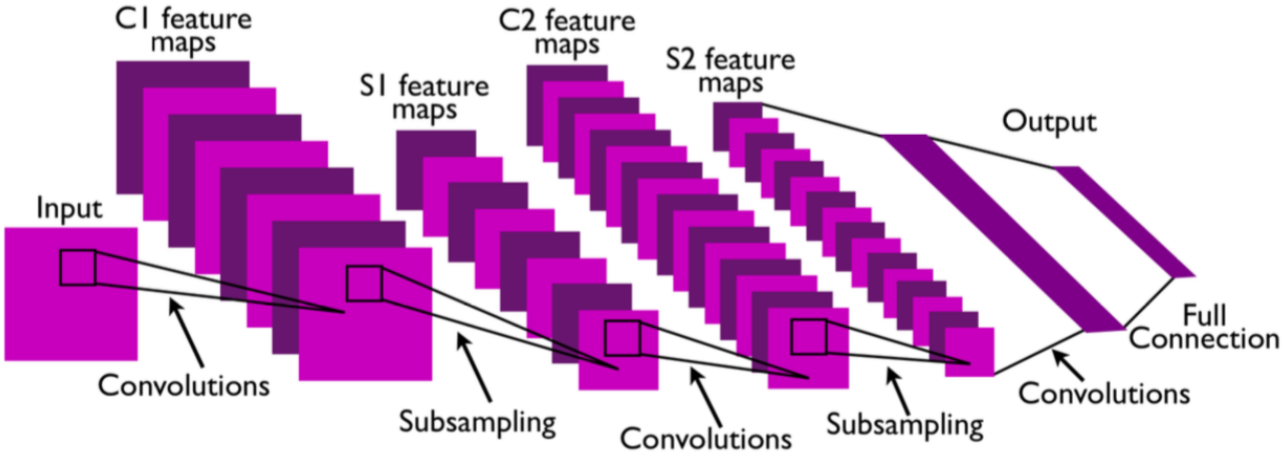
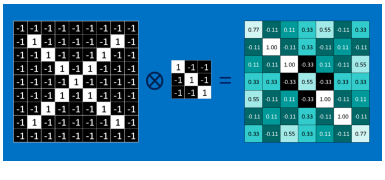


Figure 1: Typical convolutional neural network with two feature stages [2].

After introducing relevant basics in image processing and discrete convolution, the typical layers of convolutional neural networks are regarded more precisely.

Image Processing and Convolution



An image, as far as computer vision is concerned, is a two-dimensional brightness array of intensity values from 0 to 255. Thus, for a multicolor image, three intensity matrices are necessary. They are related to the color channels red, green and blue, short RGB. One channel is a map I, defined on a compact region Ω of two-dimensional surface, taking values in the positive real numbers. So I is a function

(1) $$I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}_+; (i,j) \mapsto I_{i,j}$$

such that one channel of the image I can be represented by a matrix of size $n_1 \times n_2$ [1, p. 46].

Fig. 2: Image convolved with filter (diagonal from top-left to bottom-right) to show how the resulted feature map is calculated. For reasons of simplification the image values are chosen with values of {-1; 1} [2].

Discrete two-dimensional convolution

Out of two functions, convolution produces a third one, by putting information of both input functions together. The result, in convolutional neural networks, called feature map and it describes how patterns of the filter are connected to the image.

Mathematically, discrete two-dimensional convolution can be described as followed. Given the filter $K \in \mathbb{R}^{2h_1+1 \times 2h_2+1}$, the discrete convolution of the image I with filter K is given by

(2) $$(I * K)_{r,s} := \sum_{u=-h_1}^{h_1} \sum_{u=-h_2}^{h_2} K_{u,v} I_{r-u,s-v}$$

where the filter K is given by

$$K = \begin{bmatrix} K_{-h_1,-h_2} & \dots & K_{-h_1,h_2} \\ \vdots & K_{0,0} & \vdots \\ K_{h_1,-h_2} & \dots & K_{h_1,h_2} \end{bmatrix}.$$

Basically, in convolutional neural networks, the operation is used to match a filter with a patch of the image. This fact is represented in *figure 2.* Therefore, the resulting feature map provides information on how well the local filter fits the patch. In contrast to correlation, the filter mask is flipped horizontally and vertically. Hence, one single filter can correlate with several features at once. That's based on the associative property of convolution.

Layers

A typical convolutional neural network is composed of multiple stages. Each of them takes a volume of feature maps as an input and provides a new feature map, henceforth called activation volume. The stages are consecutive separated in three layers: A convolutional layer, a ReLU layer and a pooling layer. The fully-connected layer finally maps the last activation volume onto a class of probability distributions at the output.

The following chapters will provide an overview regarding the structure and the tasks of each layer.

Convolutional Layer

The main task of the convolutional layer is to detect local conjunctions of features from the previous layer and mapping their appearance to a feature map [3]. As a result of convolution in neuronal networks, the image is split into perceptrons, creating local receptive fields and finally compressing the perceptrons in feature maps of size $m_2 \times m_3$. Thus, this map stores the information where the feature occurs in the image and how well it corresponds to the filter. Hence, each filter is trained spatial in regard to the position in the volume it is applied to.

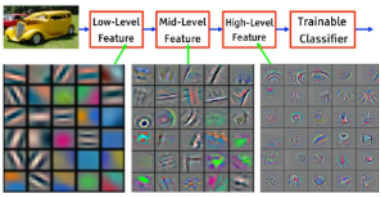


Fig. 3: Different appearance of the filters depending on the stage in the convolutional neural network.

In each layer, there is a bank of m_1 filters. The number of how many filters are applied in one stage is equivalent to the depth of the volume of output feature maps. Each filter detects a particular feature at every location on the input. The output $Y_i^{(l)}$ of layer l consists of $m_1^{(l)}$ feature maps of size $m_2^{(l)} \times m_3^{(l)}$. The i^{th} featur e map, denoted $Y_i^{(l)}$, is computed as

$$Y_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{i,j}^{(l)} * Y_j^{(l-1)} \tag{3}$$

where $B_i^{(l)}$ is a bias matrix and $K_{i,j}^{(l)}$ is the filter of size $2h_1^{(l)} + 1 \times 2h_2^{(l)} + 1$ connecti ng the j^{th} feature map in layer $(l-1)$ with i^{th} feature map in layer.

The result of staging these convolutional layers in conjunction with the following layers is that the information of the image is classified like in vision. That means that the pixels are assembled into edglets, edglets into motifs, motifs into parts, parts into objects, and objects into scenes [2] . This effect is observable in the appearance of the filters and shown in *figure 3*.

Literature

[1] Y. Ma, S. Soatt, J. Kosecka, S.S. Sastry. An Invitation to 3-D Vision: From Images to Geometric Models. Springer New York, 2005.

[2] Y. LeCun, K. Kavukvuoglu, and C. Farabet. Convolutional networks and applications in vision. In Circuits and Systems, International Symposium on, pages 253–256, 2010.

[3] Y. LeCun, Y. Bengio, G. Hilton. Deep Learning. Nature 251, pages 436-444, May 2015.

Weblinks

[1] Understanding Convolution in Deep Learning (March, 2015, Tim Dettmers)

[2] How do Convolutional Neural Networks work? (August, 2016, Brandon Rohrer)

[3] Deep Learning Tutorial (2015, Yoshua Bengio)