CHAPTER 6

Convolutional Neural Networks

A *convolutional neural network* (CNN) is a deep, feed-forward artificial neural network in which the neural network preserves the hierarchical structure by learning internal feature representations and generalizing the features in the common image problems like object recognition and other computer vision problems. It is not restricted to images; it also achieves state-of-the-art results in natural language processing problems and speech recognition.

Different Layers in a CNN

A CNN consists of multiple layers, as shown in Figure 6-1.

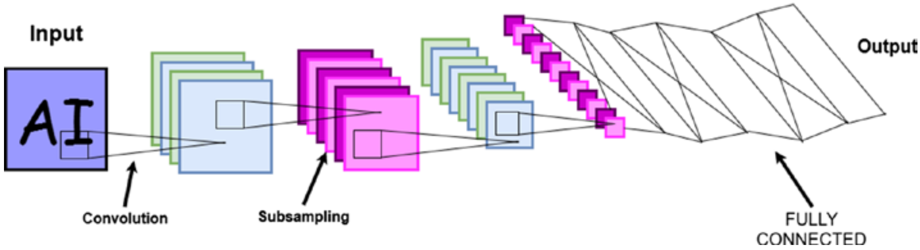


Figure 6-1. *Layers in a convolution neural network*

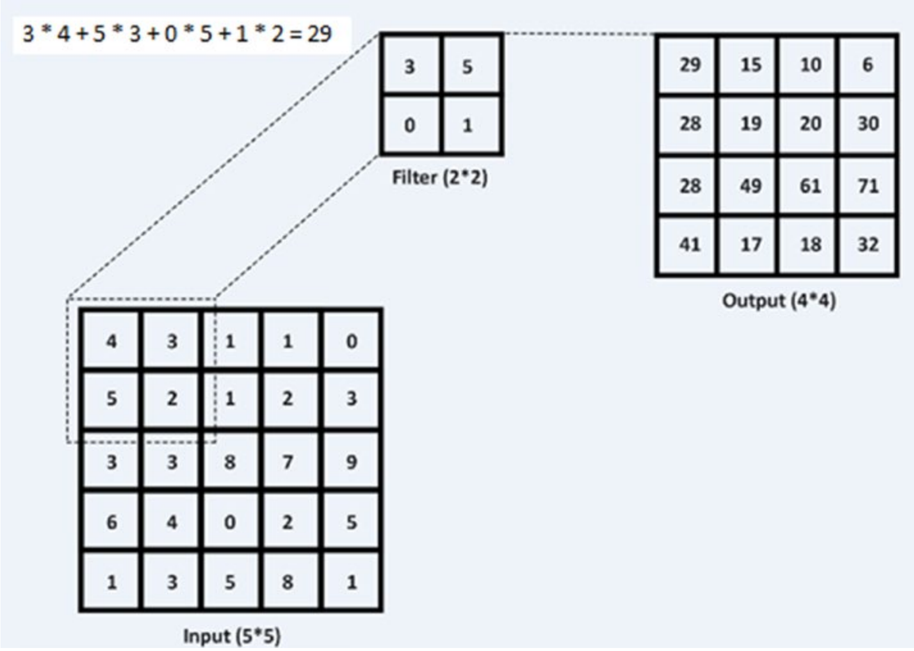
N. K. Manaswi, *Deep Learning with Applications Using Python*,
https://doi.org/10.1007/978-1-4842-3516-4_6

The *convolution layers* consist of filters and image maps. Consider the grayscale input image to have a size of 5×5, which is a matrix of 25 pixel values. The image data is expressed as a three-dimensional matrix of width × height × channels.

Note An image map is a list of coordinates relating to a specific image.

Convolution aims to extract features from the input image, and hence it preserves the spatial relationship between pixels by learning image features using small squares of input data. Rotational invariance, translation invariance, and scale invariance can be expected. For example, a rotated cat image or rescaled cat image can be easily identified by a CNN because of the convolution step. You slide the filter (square matrix) over your original image (here, 1 pixel), and at each given position, you compute element-wise multiplication (between the matrices of the filter and the original image) and add the multiplication outputs to get the final integer that forms the elements of the output matrix.

Subsampling is simply the average pooling with learnable weights per feature map, as shown in Figure 6-2.



Filter (2*2)

3	5
0	1

Output (4*4)

29	15	10	6
28	19	20	30
28	49	61	71
41	17	18	32

Input (5*5)

4	3	1	1	0
5	2	1	2	3
3	3	8	7	9
6	4	0	2	5
1	3	5	8	1

Figure 6-2. *Subsampling*

As shown in Figure 6-2, filters have input weights and generate an output neuron. Let's say you define a convolutional layer with six filters and receptive fields that are 2 pixels wide and 2 pixels high and use a default stride width of 1, and the default padding is set to 0. Each filter receives input from 2×2 pixels, section of image. In other words, that's 4 pixels at a time. Hence, you can say it will require 4 + 1 (bias) input weights.

The input volume is 5×5×3 (width × height × number of channel), there are six filters of size 2×2 with stride 1 and pad 0. Hence, the number of parameters in this layer for each filter has 2*2*3 + 1 = 13 parameters (added +1 for bias). Since there are six filters, you get 13*6 = 78 parameters.

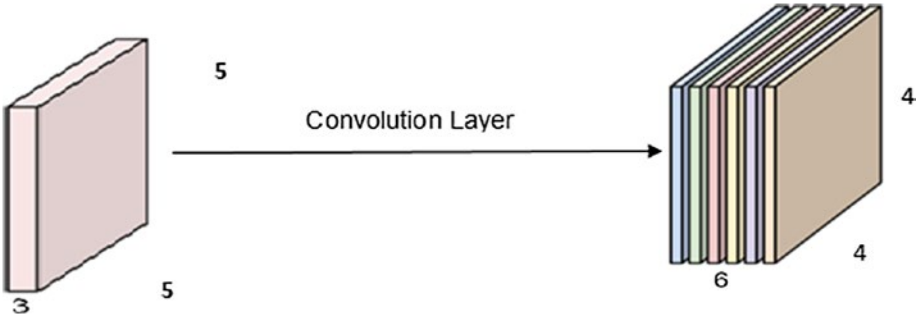
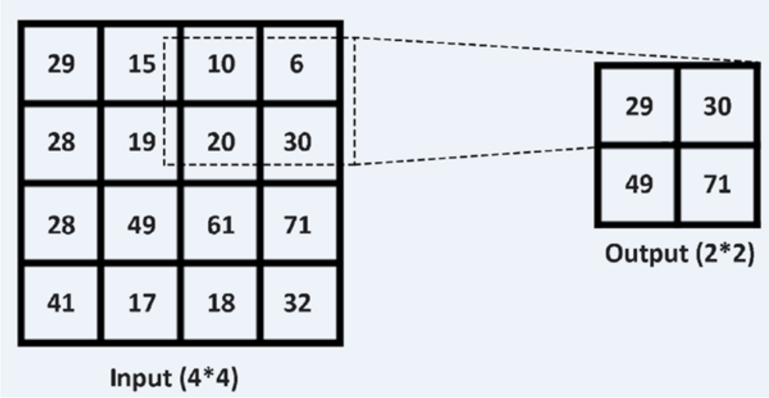


Figure 6-3. *Input volume*

Here's a summary:

- The input volume is of size W1 × H1 × D1.
- The model requires hyperparameters: number of filters (f), stride (S), amount of zero padding (P).
- This produces a volume of size W2 × H2 × D2.
- W2 = (W1-f+ 2P) /S + 1 = 4.
- H2 = (H1-f+2P)/S +1 = 4.
- D2 = Number of filters = f = 6.

The pooling layers reduce the previous layers' activation maps. It is followed by one or more convolutional layers and consolidates all the features that were learned in the previous layers' activation maps. This reduces the overfitting of the training data and generalizes the features represented by the network. The receptive field size is almost always set to 2×2 and use a stride of 1 or 2 (or higher) to ensure there is no overlap. You will use a max operation for each receptive field so that the activation is the maximum input value. Here, every four numbers map to just one number. So, the number of pixels goes down to one-fourth of the original in this step (Figure 6-4).



29	15	10	6
28	19	20	30
28	49	61	71
41	17	18	32

29	30
49	71

Figure 6-4. *Maxpooling-reducing the number of pixels*

A fully connected layer is a feed-forward artificial neural network layer. These layers have a nonlinear activation function to output class prediction probabilities. They are used toward the end after all the features are identified and extracted by convolutional layers and have been consolidated by the pooling layers in the network. Here, the hidden and output layers are the fully connected layers.

CNN Architectures

A CNN is a feed-forward deep neural network architecture comprised of a few convolutional layers, each followed by a pooling layer, activation function, and optionally batch normalization. It also comprises of the fully connected layers. As an image moves through the network, it gets smaller, mostly because of max pooling. The final layer outputs the class probabilities prediction.

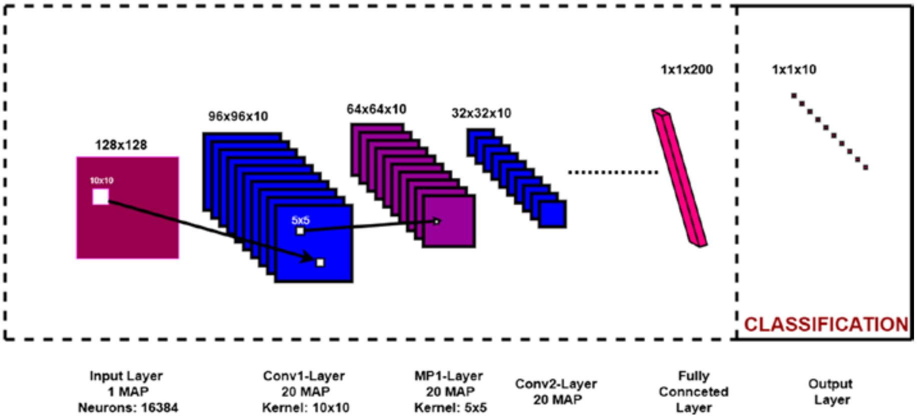


Figure 6-5. *CNN Architecture for Classification*

The past few years have seen many architectures being developed that have made tremendous progress in the field of image classification. Award-winning pretrained networks (VGG16, VGG19, ResNet50, Inception V3, and Xception) have been used for various image classification challenges including medical imaging. Transfer learning is the kind of practice where you use pretrained models in addition to a couple of layers. It can be used to solve image classification challenges in every field.