MATLAB Deep Learning Notes X

Key Words : Convolutional Neural Network; Convolutional Layer; Pooling Layer

1. Basic Concept of Convolutional Neural Network

1. **ConvNet** is not just a deep neural network that has many hidden layers. It is a deep network that imitates how the visual cortex of the brain processes and recognizes images. We have to clear that directly using the original images for image recognition leads to poor results, regardless of the recognition method; the images should be processed to contrast the features. So, before ConvNet, the feature extractor has been designed. This kind of feature extractor is composed of special kinds of neural networks, of which the weights are determined via the training process. See Fig.1 as follows

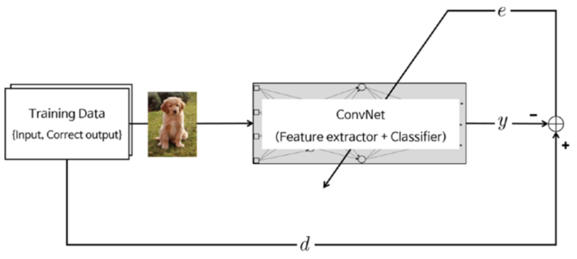

Figure 1: ConvNet's feature extractor is composed of special kinds of neural networks.

2. **ConvNet** consists of a neural network that extracts features of the input image and another neural network that classifies the feature image.

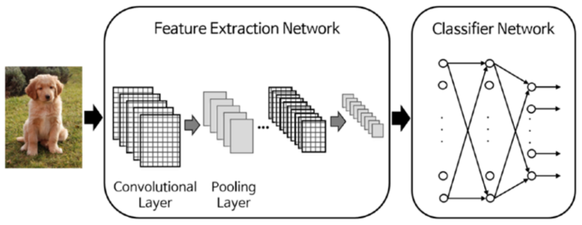

Figure 2: Typical architecture of ConvNet.

2. Convolutional Layer

3. The feature extraction neural network consists of piles of the convolutional layer and pooling layer pairs. The convolution layer, as its name implies, converts the image using the convolution operation. It can be thought of as a collection of digital filters, and could generates new images called *feature maps*.

4. It is noteworthy that the convolutional layer does not employ connection weights and a weighted sum. Instead, it contains filters that convert images, and we called these filters *convolution filters*.

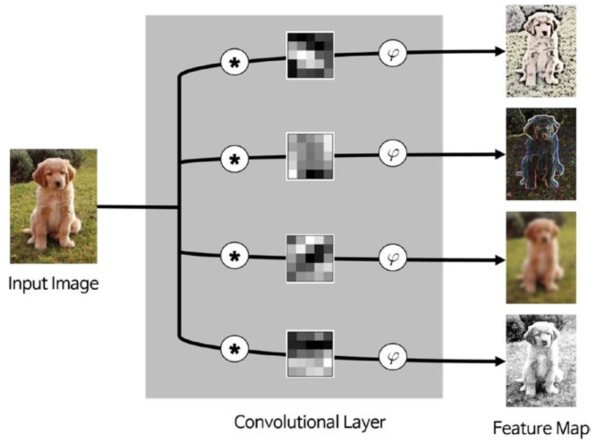

Figure 3: The convolution layer generates the same number of feature maps as the convolution filters.

5. The filters of the convolution layer are two-dimensional matrices. As addressed in the previous section, the values of the filter matrix are determined through the training process. Therefore, these values are continuously trained throughout the training process. This aspect is similar to the updating process of the connection weights of the ordinary neural network.

6. So what is the convolution? This gif shows the simple concept.For example, suppose

$$\mathbf{x} = \begin{bmatrix} 1 & 1 & 1 & 3 \\ 4 & 6 & 4 & 8 \\ 3 & 0 & 1 & 5 \\ 0 & 2 & 2 & 4 \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Then the convolution of $\mathbf{x}$ and $\mathbf{W}$ (i.e., $\mathbf{x} \otimes \mathbf{W}$) can be calculated as following steps:

7(a). Since the size of $\mathbf{x}$ and $\mathbf{W}$ is 4×4 and 2×2, respectively. So, the size of the output of the convolution is $(4-2+1) \times (4-2+1)$, i.e., 3×3.

7(b). The convolution operation begins at the upper-left corner of the submatrix that is the same size as the convolution filter.

$$\begin{bmatrix} 1 & 1 & 1 & 3 \\ 4 & 6 & 4 & 8 \\ 3 & 0 & 1 & 5 \\ 0 & 2 & 2 & 4 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \times 1 & 1 \times 0 & 1 & 3 \\ 4 \times 0 & 6 \times 1 & 4 & 8 \\ 3 & 0 & 1 & 5 \\ 0 & 2 & 2 & 4 \end{bmatrix}$$

7(c). Another convolution operation is conducted for the next submatrix

$$\begin{bmatrix} 1 & 1 \times 1 & 1 \times 0 & 3 \\ 4 & 6 \times 0 & 4 \times 1 & 8 \\ 3 & 0 & 1 & 5 \\ 0 & 2 & 2 & 4 \end{bmatrix}$$

7(d). It repeats the same process until the feature map of the given filter is produced

$$\begin{bmatrix} 7 & 5 & 9 \\ 4 & 7 & 9 \\ 32 & 2 & 5 \end{bmatrix}$$

8. In summary, the convolution layer operates the convolution filters on the input image and produces the feature maps. The features that are extracted in the convolution layer determined by the trained convolution filters. Therefore, the features that the convolution layer extracts vary depending on which convolution filter is used.

Program 24: conv

Listing 1: conv.m

```
function [y] = conv(x, W)

[wrow, wcol, numFilters] = size(W);
[xrow, xcol, ~] = size(x);

yrow = xrow — wrow + 1;
ycol = xcol — wcol + 1;

y = zeros(yrow, ycol, numFilters);

for k = 1 : numFilters
    filter = W(:, :, k);
    filter = rot90(squeeze(filter), 2);
    y(:, :, k) = conv2(x, filter, 'valid');
end

end
```

Output 24:

```
>> x = [1, 1, 1, 3; 4, 6, 4, 8; 30, 0, 1, 5; 0, 2, 2, 4];
>> W = [1, 0; 0, 1];
>> y = conv(x, W)

y =

     7     5     9
     4     7     9
    32     2     5
```

3. Pooling Layer

9. The pooling layer reduces the size of the image, as it combines neighboring pixels of a certain area of the image into a single representative value.

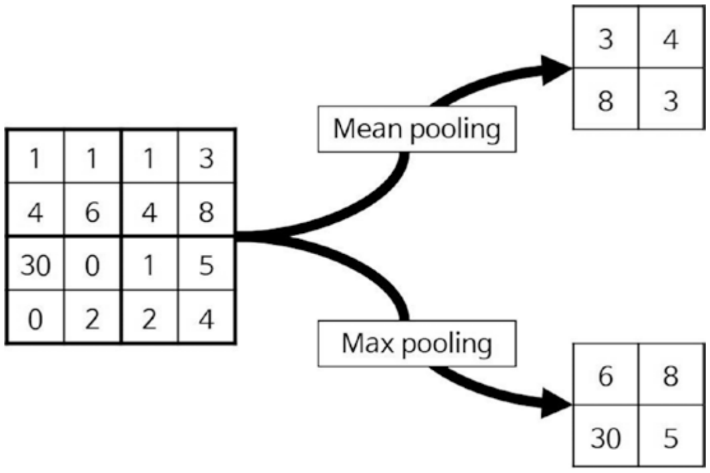

Figure 4: The resultant cases of pooling using two different methods.

10. Actually, in a mathematical sense, the pooling process is a type of convolution operation. The difference from the convolution layer is that the convolution filter is stationary, and the convolution areas do not overlap. The pooling layer compensates for eccentric and tilted objects to some extent. In addition, as the pooling process reduces the image size, it is highly beneficial for relieving the computational load and preventing overfitting.

Program 25: pool

Listing 2: pool.m

```matlab
function y = pool(x)
% 2x2 mean pooling

[xrow, xcol, numFilters] = size(x)

y = zeros(xrow/2, xcol/2, numFilters)
for k = 1 : numFilters
    filter = ones(2) / (2*2)
    image = conv2(x(:, :, k), filter, 'valid')
    y(:, :, k) = image(1 : 2 : end, 1 : 2 : end);
end

end
```

Output 25:

```
>> x = [1, 1, 1, 3; 4, 6, 4, 8; 30, 0, 1, 5; 0, 2, 2, 4];
>> pool(x)

xrow =

     4

xcol =

     4

numFilters =

     1

y =

     0     0
     0     0
```

```

filter =

    0.2500    0.2500
    0.2500    0.2500

image =

    3.0000    3.0000    4.0000
   10.0000    2.7500    4.5000
    8.0000    1.2500    3.0000

ans =

     3     4
     8     3
```