Convolutional Neural Networks

Instructor: Revendranath T

Convolutional Neural Networks (CNNs)

Components:

1. Kernels and filters
2. Convolution
3. Strides
4. Pooling
5. Padding
6. Dropout

Kernels & Filters

1. Filters are square matrices with $n_k x n_k$ dimensions
2. n_k is usually 3 or 5

The following kernel will allow the detection of horizontal edges

$$\Im_H = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

Kernels & Filters

The following kernel will allow the detection of vertical edges

$$\mathfrak{I}_V = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$$

The following kernel will allow the detection of edges when luminosity changes drastically

$$\mathfrak{I}_L = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Kernels & Filters

The following kernel will blur edges in an image

$$\mathfrak{I}_B = -\frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Convolution

1. Convolution is multiplying corresponding positional values in two tensors and summing them up.
2. Convolution is not tensor or matrix multiplication

$$\begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{pmatrix} * \begin{pmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \\ k_7 & k_8 & k_9 \end{pmatrix} = \sum_{i=1}^{9} a_i k_i$$

Convolution: Illustration

- Assume matrix A is 4x4 & Kernel K is 3x3 matrix
- Convolution of A*K results in matrix B (2x2 matrix)

$$A = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ a_9 & a_{10} & a_{11} & a_{12} \\ a_{13} & a_{14} & a_{15} & a_{16} \end{pmatrix} \quad K = \begin{pmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \\ k_7 & k_8 & k_9 \end{pmatrix} \quad B = \begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix}$$

Convolution: Illustration

- First Convolution of A*K

$$A=\begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ a_9 & a_{10} & a_{11} & a_{12} \\ a_{13} & a_{14} & a_{15} & a_{16} \end{pmatrix} \quad A_1=\begin{pmatrix} a_1 & a_2 & a_3 \\ a_5 & a_6 & a_7 \\ a_9 & a_{10} & a_{11} \end{pmatrix} \quad K=\begin{pmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \\ k_7 & k_8 & k_9 \end{pmatrix} \quad B=\begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix}$$

$$B_1 = A_1 * K = a_1k_1 + a_2k_2 + a_3k_3 + k_4a_5 + k_5a_5 + k_6a_7 + k_7a_9 + k_8a_{10} + k_9a_{11}$$

Convolution: Illustration

- Second Convolution of A*K

$$A=\begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ a_9 & a_{10} & a_{11} & a_{12} \\ a_{13} & a_{14} & a_{15} & a_{16} \end{pmatrix} \qquad K=\begin{pmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \\ k_7 & k_8 & k_9 \end{pmatrix} \qquad B=\begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix}$$

$$B_2 = A_2 * K = a_2k_1 + a_3k_2 + a_4k_3 + a_6k_4 + a_7k_5 + a_8k_6 + a_{10}k_7 + a_{11}k_8 + a_{12}k_9$$

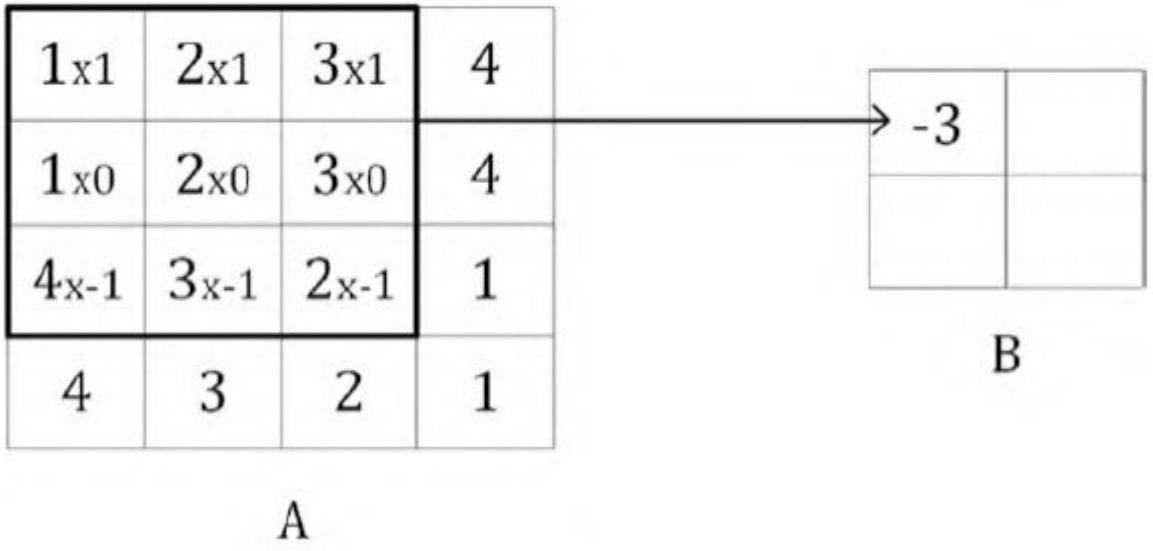
Stride

- **Stride**: The number of rows to move down and columns to move to right when performing convolution between a tensor and the kernel
- In the previous example between A*K, the 3 × 3 kernel region was moved one column to the right and one row down.
  - In this example , the stride is 1 (s=1).
- Depending on S, the new matrix created after convolution will have the following shape:

$$n_B = \frac{n_A - n_K}{s} + 1$$

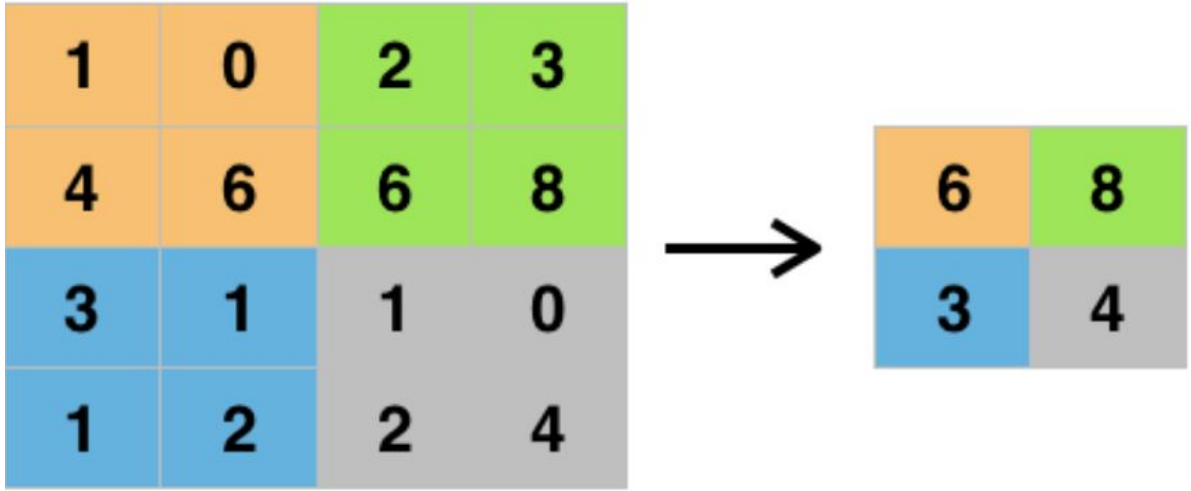
Convolution: Visual Illustration

$$B_{11} = 1\times1+2\times1+3\times1+1\times0+2\times0+3\times0+4\times(-1)+3\times(-1)+2\times(-1)=-3$$



Pooling

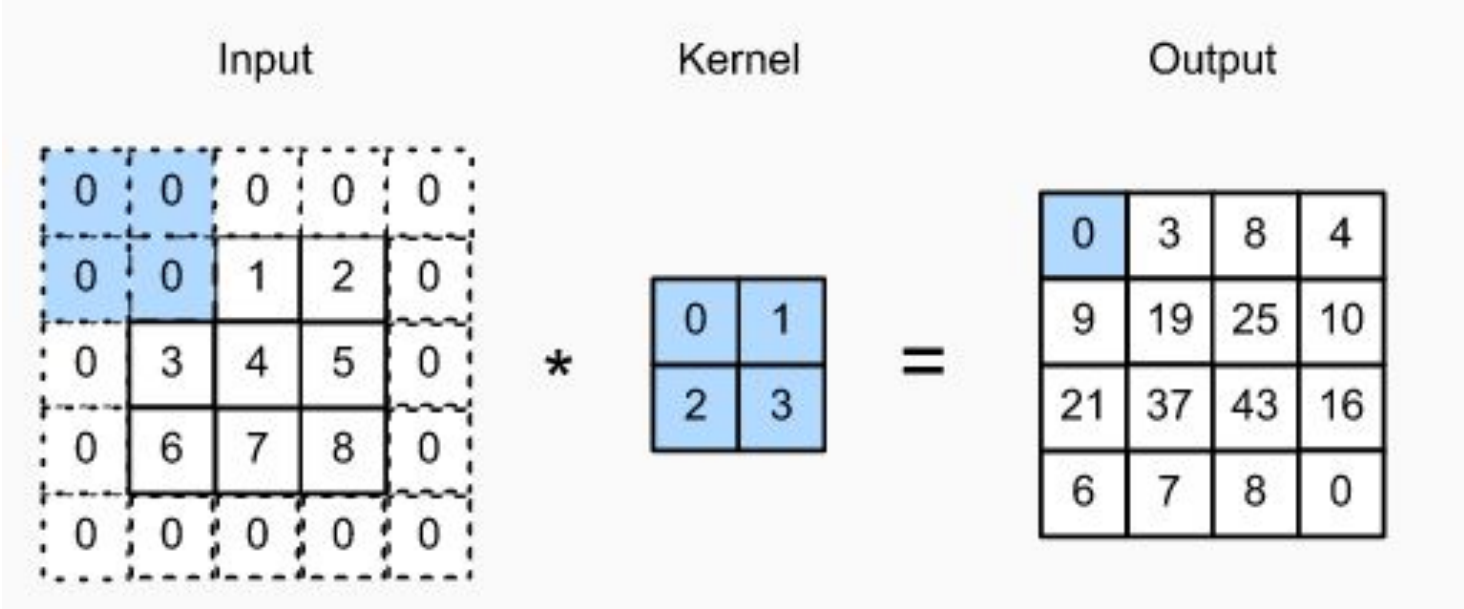
- **Pooling**: From a given matrix, stride and kernel, identify the maximum or average in that block
- Most often used pooling technique is **max pooling**



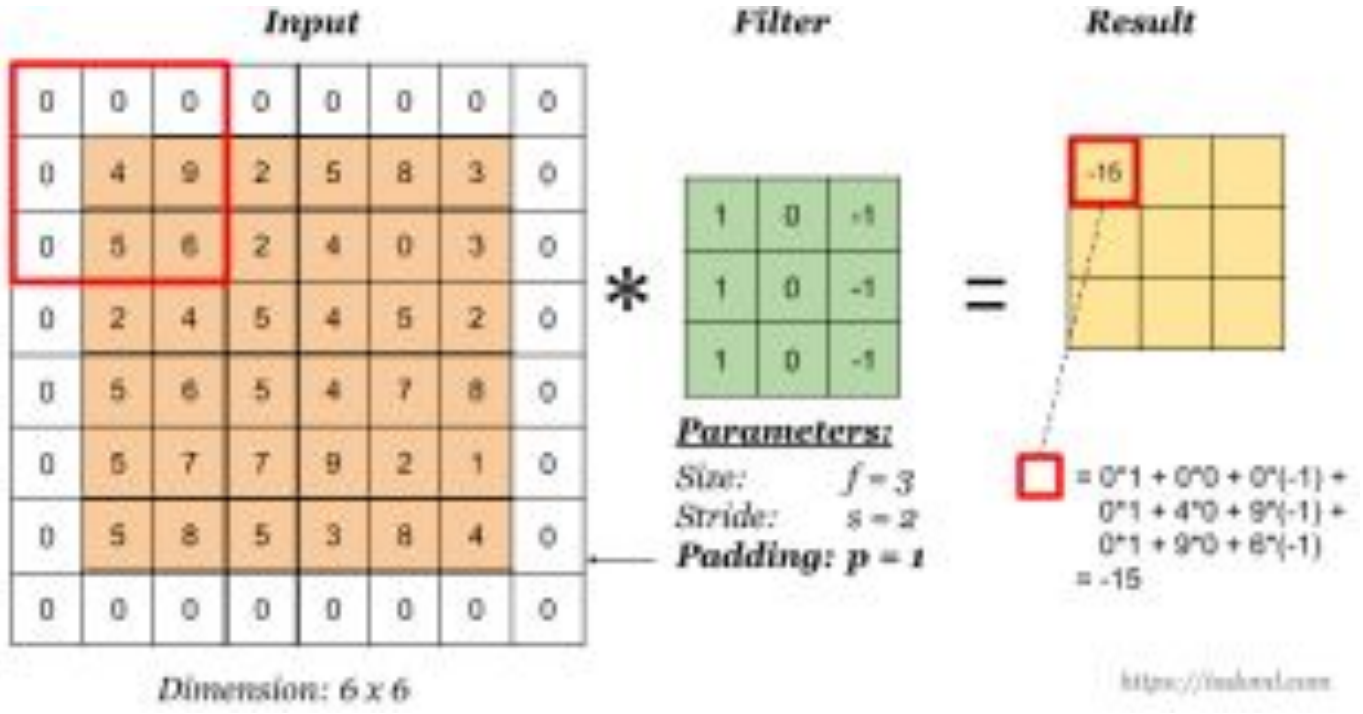
Padding

- **Padding**
- Add rows of pixels to the top and bottom and to the columns of pixels on the right and on the left of the final images so that resulting matrix after convolution with kernel is same as original image matrix
- Some strategies for padding are to fill the added pixels with zeros, with the values of the closest pixels, and so on.

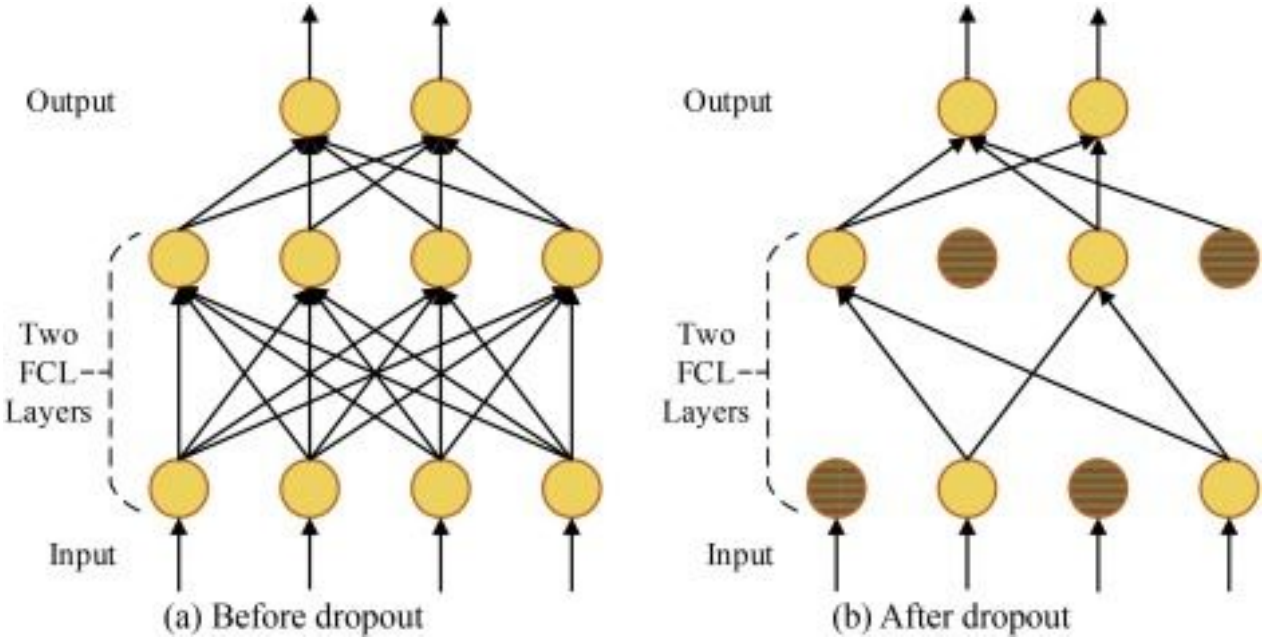
Padding: Illustration



Padding: Illustration



Dropout: Illustration



Convolutional Neural Network: Illustration