Convolutions for neural networks

Arnaud Bergeron

August 10, 2015

Some vocabulary:

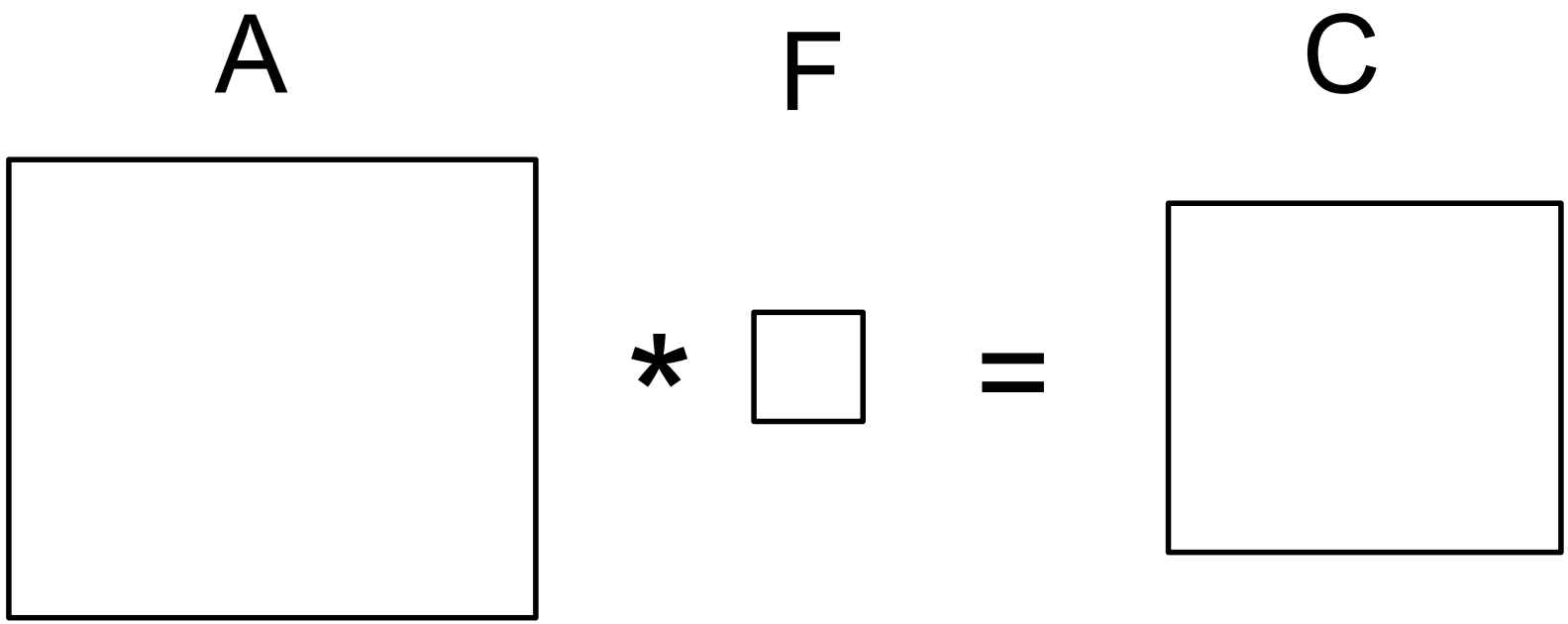
feature map analogue to layer (the activations, not the weights) for a neural network.

channel a single 2D "image" which composes a feature map.

filter weight (as in layer weights).

stack ???

The basic convolution operation (2D, discrete)



where A is the input, F are the filters and C is the convolution result.

The basic convolution operation (2D, discrete)

$$C(i,j) = \sum_{m-1}^{x=0} \sum_{n-1}^{y=0} A(x+i, y+j) F(x,y)$$

where A is the input, F are the filters and C is the convolution result.

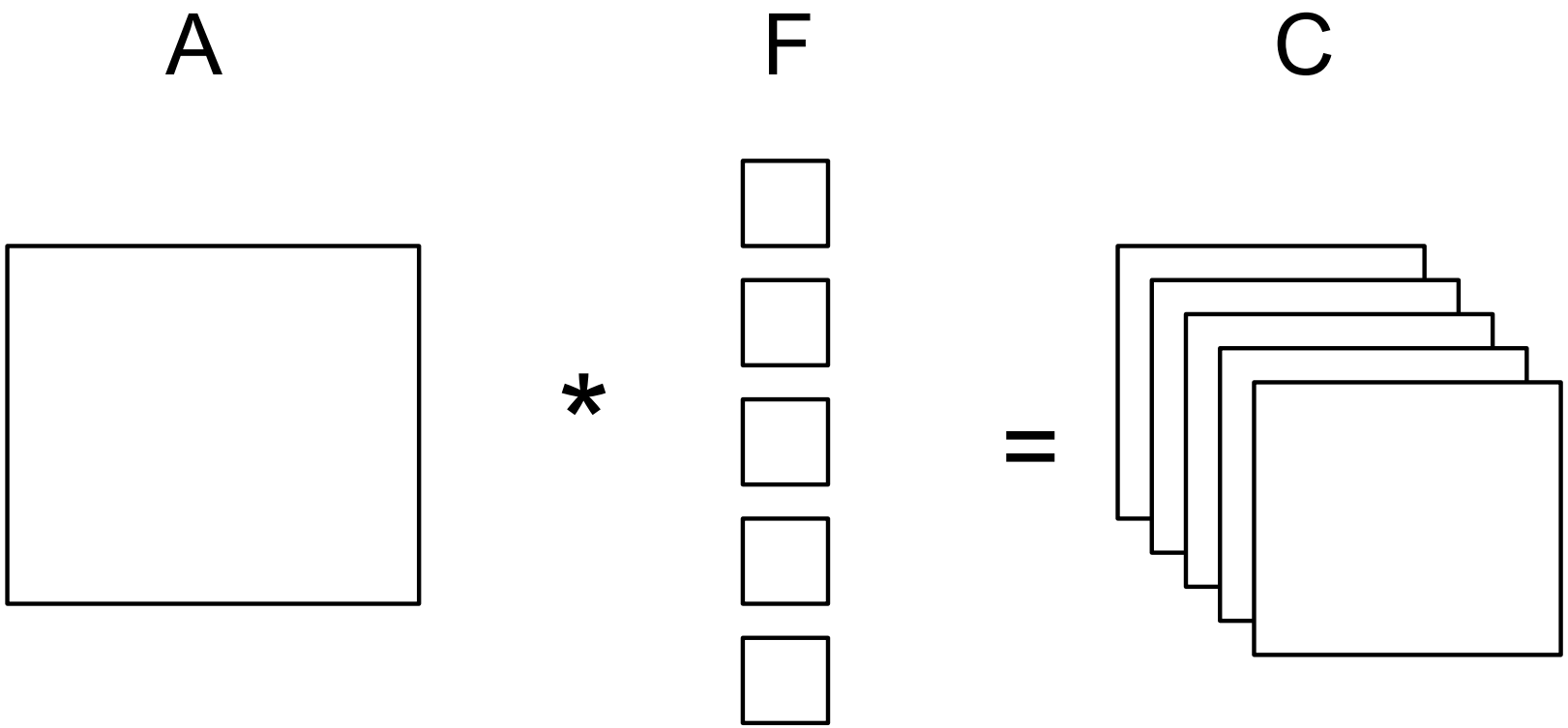
The basic convolution operation (2D, discrete)

$$C(i,j) = \sum_{m-1}^{x=0} \sum_{n-1}^{y=0} A(x+i, y+j) F(x,y)$$

- (m, n) is the size of the filter.

where A is the input, F are the filters and C is the convolution result.

The basic convolution operation (2D, discrete) for a single example



where A is the input, F are the filters and C is the convolution result.

The basic convolution operation (2D, discrete) for a single example

$$C_o(i,j) = \sum_{m-1}^{x=0} \sum_{n-1}^{y=0} A(x+i, y+j) F_o(x,y)$$

- (m, n) is the size of the filters.

where A is the input, F are the filters and C is the convolution result.

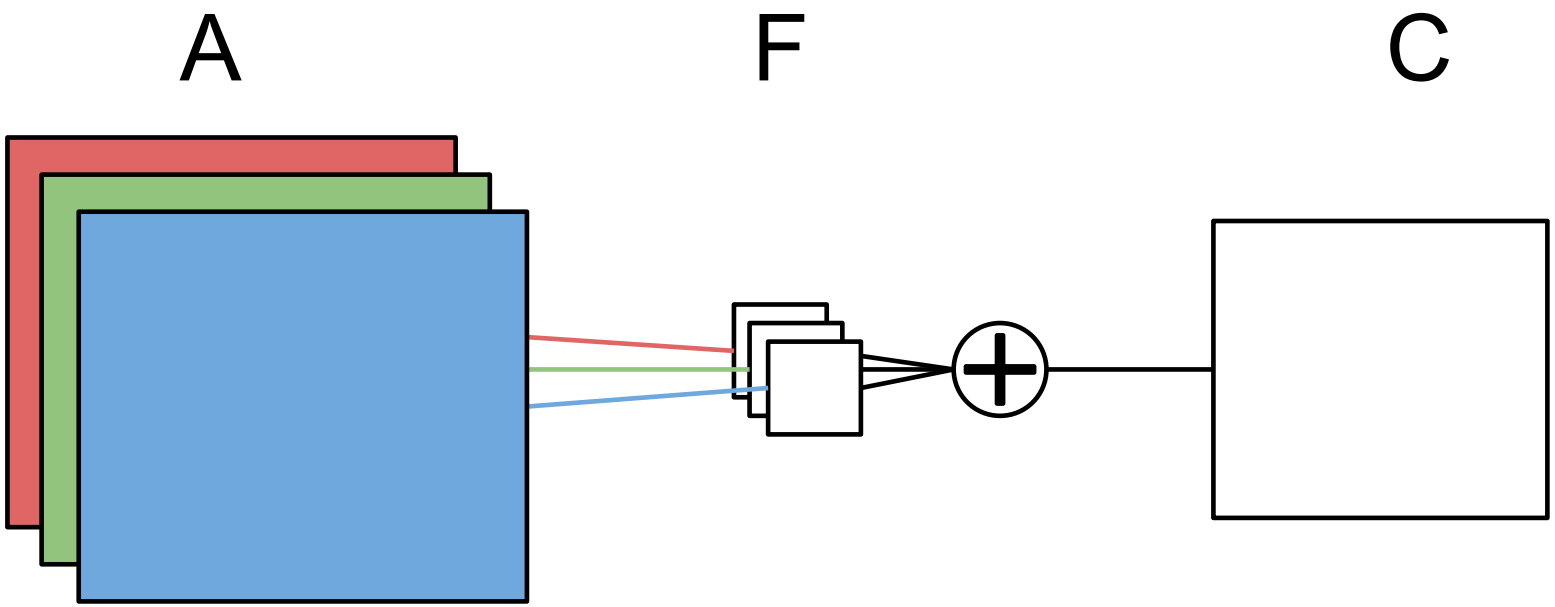
The basic convolution operation (2D, discrete) for a single example

$$C_o(i,j) = \sum_{m-1}^{x=0} \sum_{n-1}^{y=0} A(x+i, y+j) F_o(x,y)$$

- (m, n) is the size of the filters.
- o is the output channel.

where A is the input, F are the filters and C is the convolution result.

The basic convolution operation (2D, discrete) for a single example



where A is the input, F are the filters and C is the convolution result.

The basic convolution operation (2D, discrete) for a single example

$$C(i,j) = \sum_{l}^{k=0} \sum_{m-1}^{x=0} \sum_{n-1}^{y=0} A_k(x+i, y+j) F_k(x,y)$$

- (m, n) is the size of the filters.

where A is the input, F are the filters and C is the convolution result.

The basic convolution operation (2D, discrete) for a single example

$$C(i,j) = \sum_{l}^{k=0} \sum_{m-1}^{x=0} \sum_{n-1}^{y=0} A_k(x+i, y+j) F_k(x,y)$$

- (m, n) is the size of the filters.
- k is the input channel.

where A is the input, F are the filters and C is the convolution result.

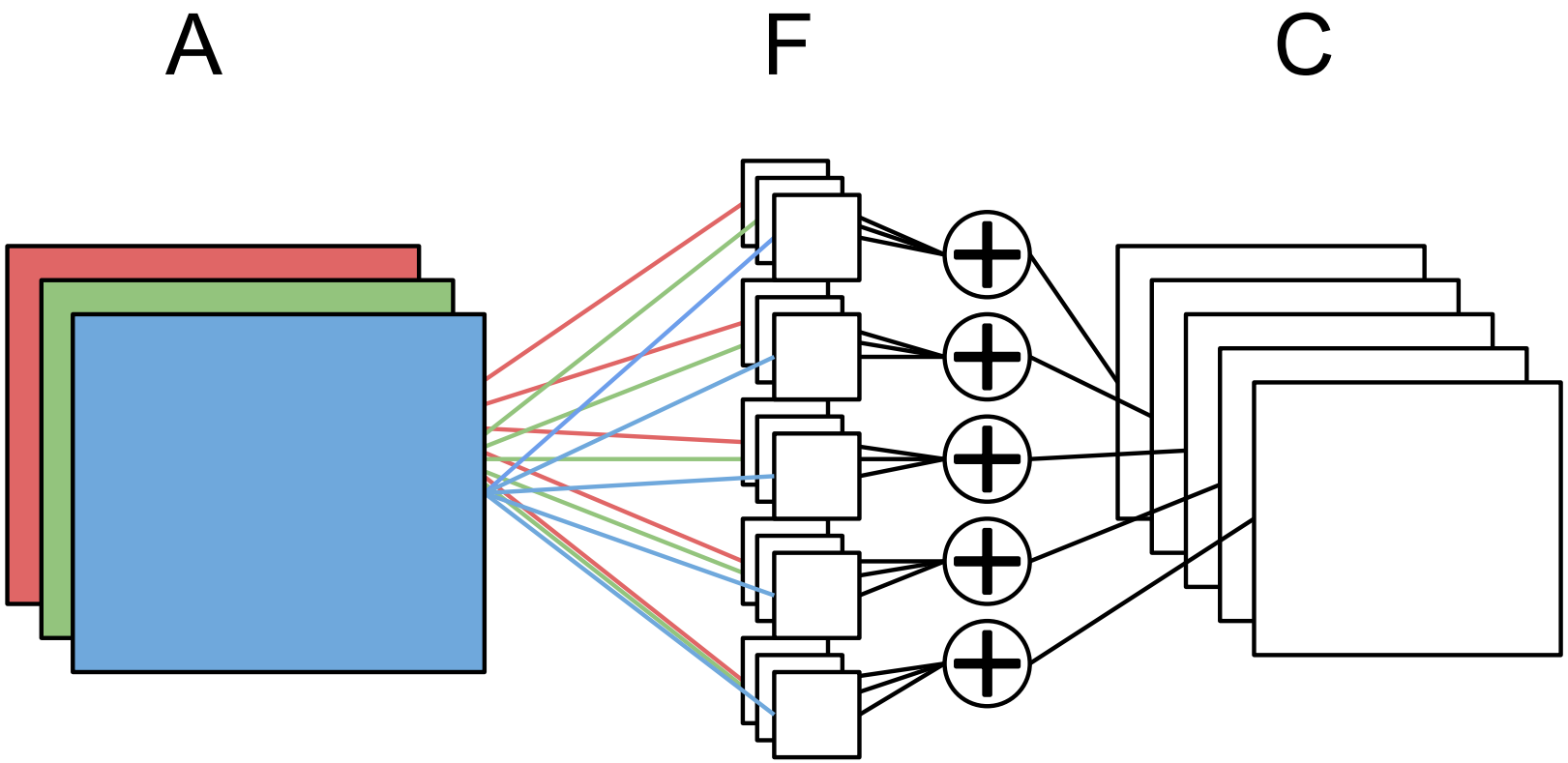
The basic convolution operation (2D, discrete) for a single example

$$C(i,j) = \sum_{l}^{k=0} \sum_{m-1}^{x=0} \sum_{n-1}^{y=0} A_k(x+i, y+j) F_k(x,y)$$

- (m, n) is the size of the filters.
- k is the input channel.
- l is the number of input channels.

where A is the input, F are the filters and C is the convolution result.

The basic convolution operation (2D, discrete)



where A is the input, F are the filters and C is the convolution result.

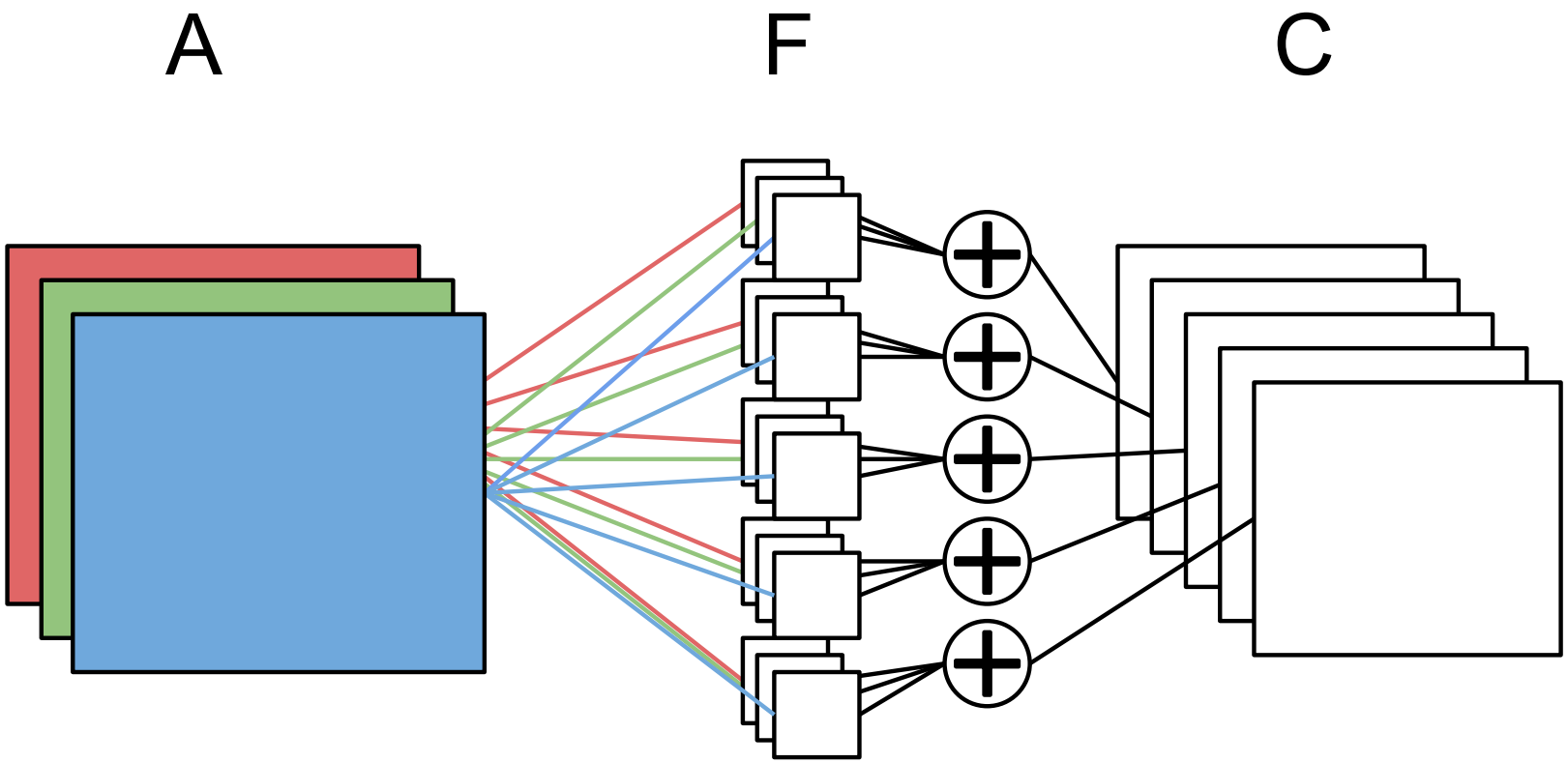
The basic convolution operation (2D, discrete)

$$C_o(i,j) = \sum_{l-1}^{k=0} \sum_{m-1}^{x=0} \sum_{n-1}^{y=0} A_k(x+i, y+j) F_{ko}(x,y)$$

- (m, n) is the size of the filters.
- o is the output channel.
- k is the input channel.
- l is the number of input channels.

where A is the input, F are the filters and C is the convolution result.

The basic convolution operation (2D, discrete)



where A is the input, F are the filters and C is the convolution result.

The basic convolution operation (2D, discrete)

$$C_{bo}(i,j) = \sum_{l-1}^{k=0} \sum_{m-1}^{x=0} \sum_{n-1}^{y=0} A_{bk}(x+i, y+j) F_{ko}(x,y)$$

- (m, n) is the size of the filters.
- o is the output channel.
- k is the input channel.
- l is the number of input channels.

where A is the input, F are the filters and C is the convolution result.

The basic convolution operation (2D, discrete)

$$C_{bo}(i,j) = \sum_{l-1}^{k=0} \sum_{m-1}^{x=0} \sum_{n-1}^{y=0} A_{bk}(x+i, y+j) F_{ko}(x,y)$$

- (m, n) is the size of the filters.
- o is the output channel.
- k is the input channel.
- l is the number of input channels.
- b is the batch.

where A is the input, F are the filters and C is the convolution result.

Memory layout for images: 'bc01'

- first dimension is the batch ('b')
- second dimension is the channel ('c')
- last two dimensions are the data ('0', '1')

Memory layout for images: 'bc01'

- first dimension is the batch ('b')
- second dimension is the channel ('c')
- last two dimensions are the data ('0', '1')

Memory layout for filters: 'nc01'

- first dimension is the output channel ('n')
- second dimension is the input channel ('c')
- last two dimensions are the data ('0', '1')

Memory layout for images: 'bc01'

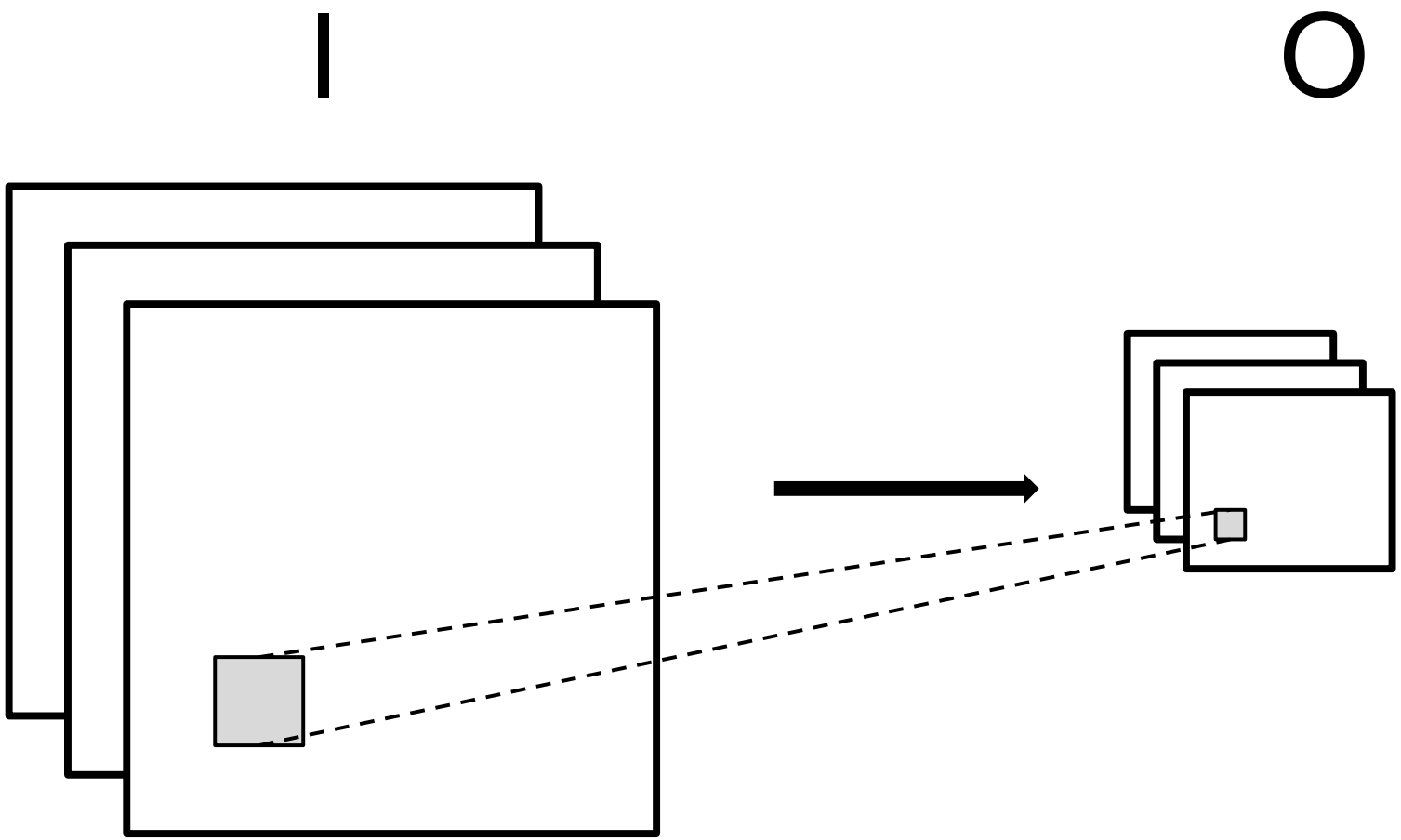
- first dimension is the batch ('b')
- second dimension is the channel ('c')
- last two dimensions are the data ('0', '1')

Memory layout for filters: 'nc01'

- first dimension is the output channel ('n')
- second dimension is the input channel ('c')
- last two dimensions are the data ('0', '1')

Some other packages may use different conventions.

Basic pooling operation



Basic pooling operation (max)

$$O_k(i,j) = \max_{\substack{0 \le x < m \\ 0 \le y < n}} I_k(x+i, y+j)$$

Basic pooling operation (max)

$$O_k(i,j) = \max_{\substack{0 \le x < m \\ 0 \le y < n}} I_k(x+i, y+j)$$

- k is the channel.

Basic pooling operation (max)

$$O_k(i,j) = \max_{\substack{0 \leq x < m \\ 0 \leq y < n}} I_k(x+i, y+j)$$

- k is the channel.
- (m, n) is the size of the filters.

- One thing to watch out for is the border handling
- Other types of pooling exist such as average pooling