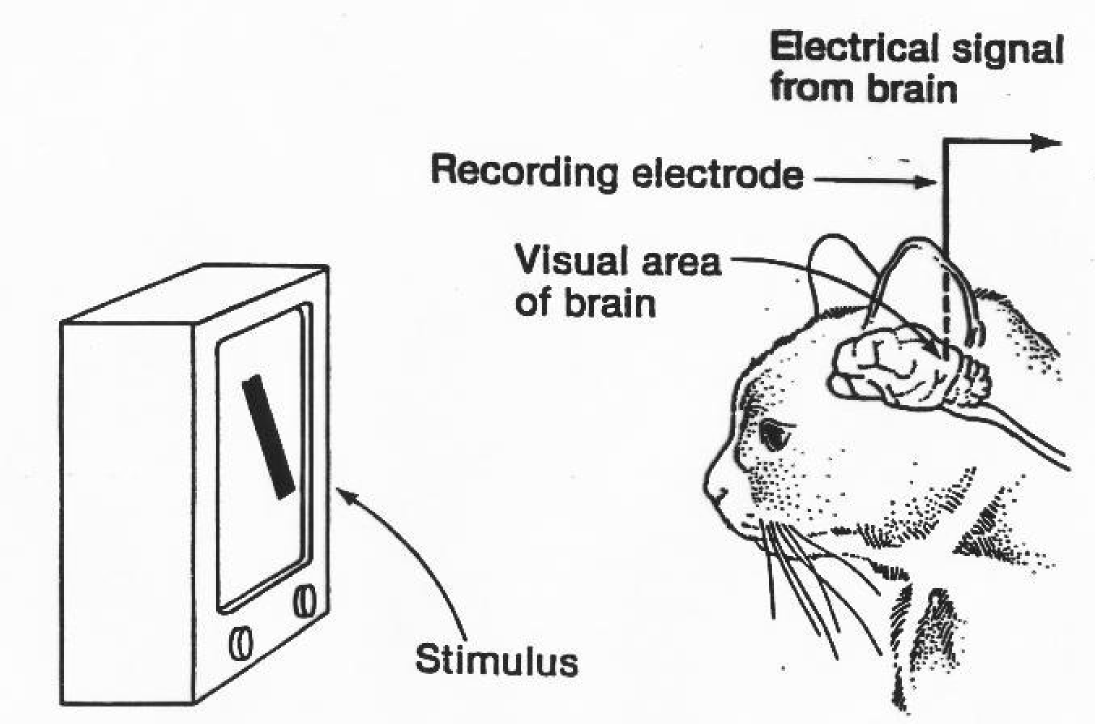
Neural networks

Convolution

The revolution is convolution

But not a new revolution

Hubel and Wiesel, 1959



The experiments showed that certain cell in the visual cortex respond to specific stimuli. Formally, the response is obtained by a convolution operation between the input signal and the cell functionality

What is convolution?

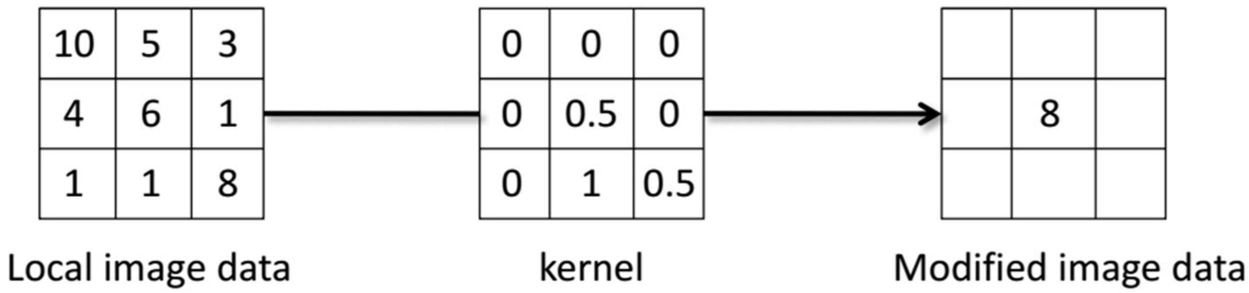
- It is a linear operation that transform the image content by structural element called *filte*r or *kernel*
- Technically, it is a linear function

Local image data

Modified image data

What is convolution?

- In general, a pixel is replaced by a combination of its neighborhood
- The kernel contains the weights to compute that combination



Convolution

- Let F be an image, H (size $2k+1, 2k+1$) be a kernel, and G be the output image

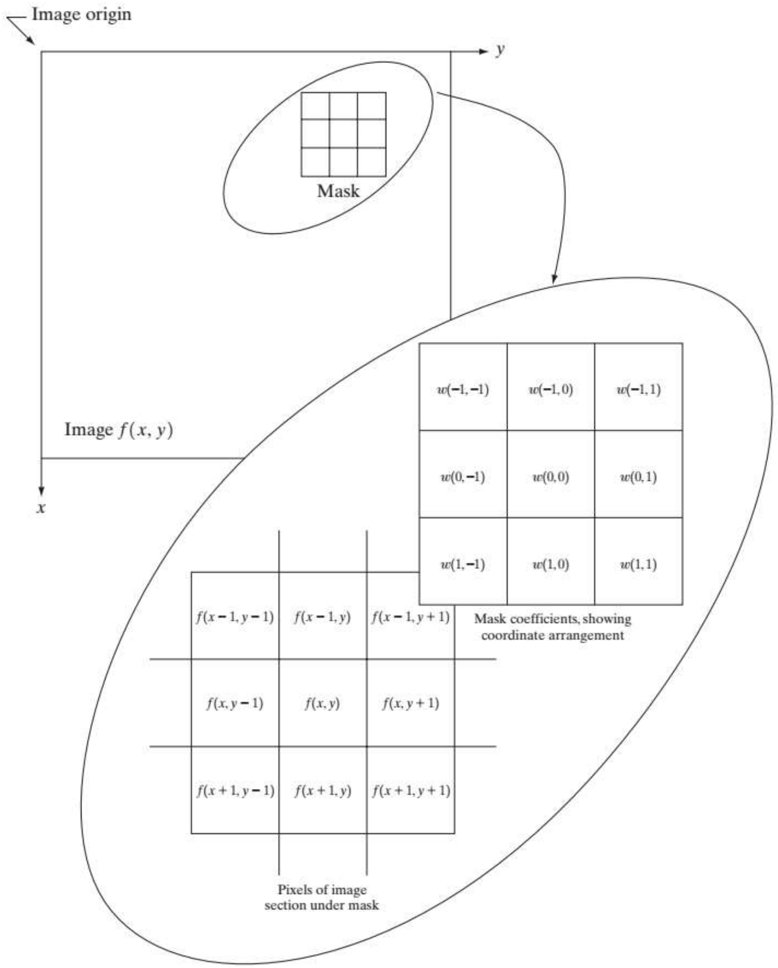
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]$$

- Convolution is associative and commutative

$$G = H * F$$

Convolution

- It is a pixel-wise operation
- The kernel slides in the image while computing the local linear transform.



Convolution: Example

$$\begin{array}{|c|c|c|c|c|}\hline 1 & 2 & 3 & 2 & 1 \\ \hline 3 & 2 & 2 & 1 & 3 \\ \hline 2 & 1 & 3 & 2 & 3 \\ \hline 2 & 3 & 3 & 1 & 2 \\ \hline 3 & 1 & 2 & 2 & 3 \\ \hline \end{array} * \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array}$$

Convolution: Example

$$
\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 3 & 2 & 2 & 1 & 3 \\ 2 & 1 & 3 & 2 & 3 \\ 2 & 3 & 3 & 1 & 2 \\ 3 & 1 & 2 & 2 & 3 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 20 & & \\ & & \\ & & \end{bmatrix}
$$

Convolution: Example

$$
\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 3 & 2 & 2 & 1 & 3 \\ 2 & 1 & 3 & 2 & 3 \\ 2 & 3 & 3 & 1 & 2 \\ 3 & 1 & 2 & 2 & 3 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 20 & 18 & \\ & & \\ & & \end{bmatrix}
$$

Convolution: Example

$$\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 3 & 2 & 2 & 1 & 3 \\ 2 & 1 & 3 & 2 & 3 \\ 2 & 3 & 3 & 1 & 2 \\ 3 & 1 & 2 & 2 & 3 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 20 & 18 & 19 \\ & & \\ & & \end{bmatrix}$$

Convolution: Example

$$
\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 3 & 2 & 2 & 1 & 3 \\ 2 & 1 & 3 & 2 & 3 \\ 2 & 3 & 3 & 1 & 2 \\ 3 & 1 & 2 & 2 & 3 \end{bmatrix}
*
\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}
=
\begin{bmatrix} 20 & 18 & 19 \\ 20 & 18 & 19 \\ 20 & 18 & 21 \end{bmatrix}
$$

Convolution: Example

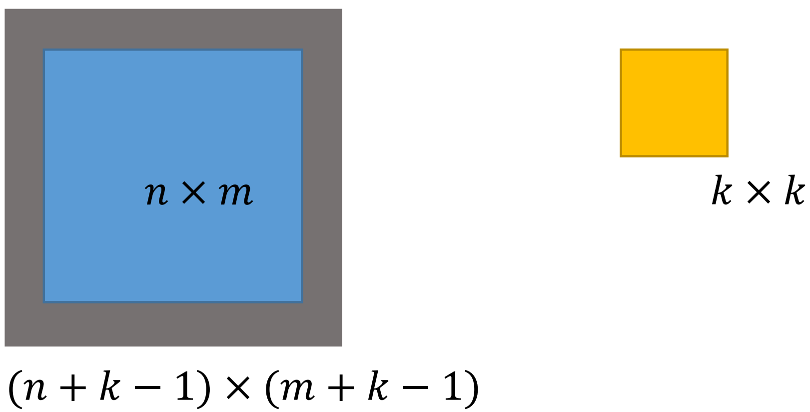
- Have you noted that we loose information?

$$\begin{array}{|c|c|c|c|c|}\hline 1 & 2 & 3 & 2 & 1 \\ \hline 3 & 2 & 2 & 1 & 3 \\ \hline 2 & 1 & 3 & 2 & 3 \\ \hline 2 & 3 & 3 & 1 & 2 \\ \hline 3 & 1 & 2 & 2 & 3 \\ \hline \end{array} * \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 20 & 18 & 19 \\ \hline 20 & 18 & 19 \\ \hline 20 & 18 & 21 \\ \hline \end{array}$$

- Input image is 5x5 and output image is 3x3
- We cannot apply the kernel in certain positions (border)

Convolution: padding

- Strategies for the boundary problem:
  - Take the output as it is. It could be not that harmful
  - Padding: fill input image with zeros, so the output is an image equal in size than the original



Convolution response

- Let's see some examples to gain insight about convolution

$$\text{Kernel} = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Which operation are we performing?

Average

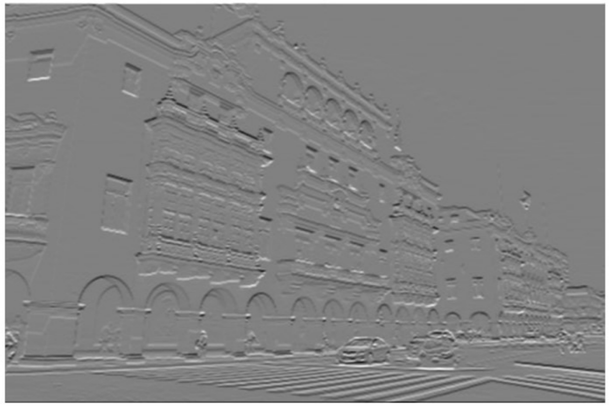
Convolution response

- Let’s see some examples to gain insight about convolution

Kernel = average kernel of size 21

Convolution response

- Let’s see some examples to gain insight about convolution



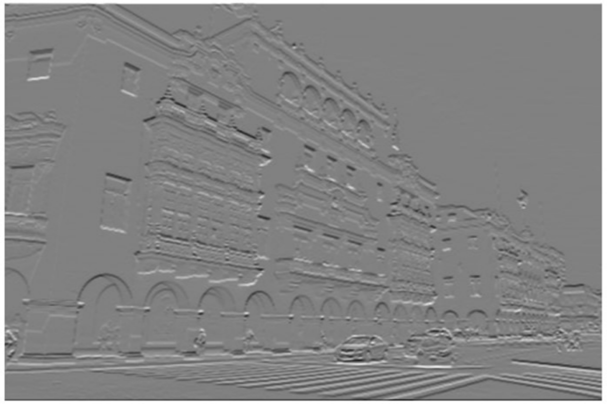
Kernel =

-1	0	1
-2	0	2
-1	0	1

Sobel, vertical edges

Convolution response

- Let’s see some examples to gain insight about convolution



Kernel =

-1	-2	-1
0	0	0
1	2	1

Sobel, horizontal edges

Summary

- Convolution is a linear operation
- It computes the response of an image against a kernel
- The kernel defines the kind of transformation to perform

Neural networks

Convolutional Layers

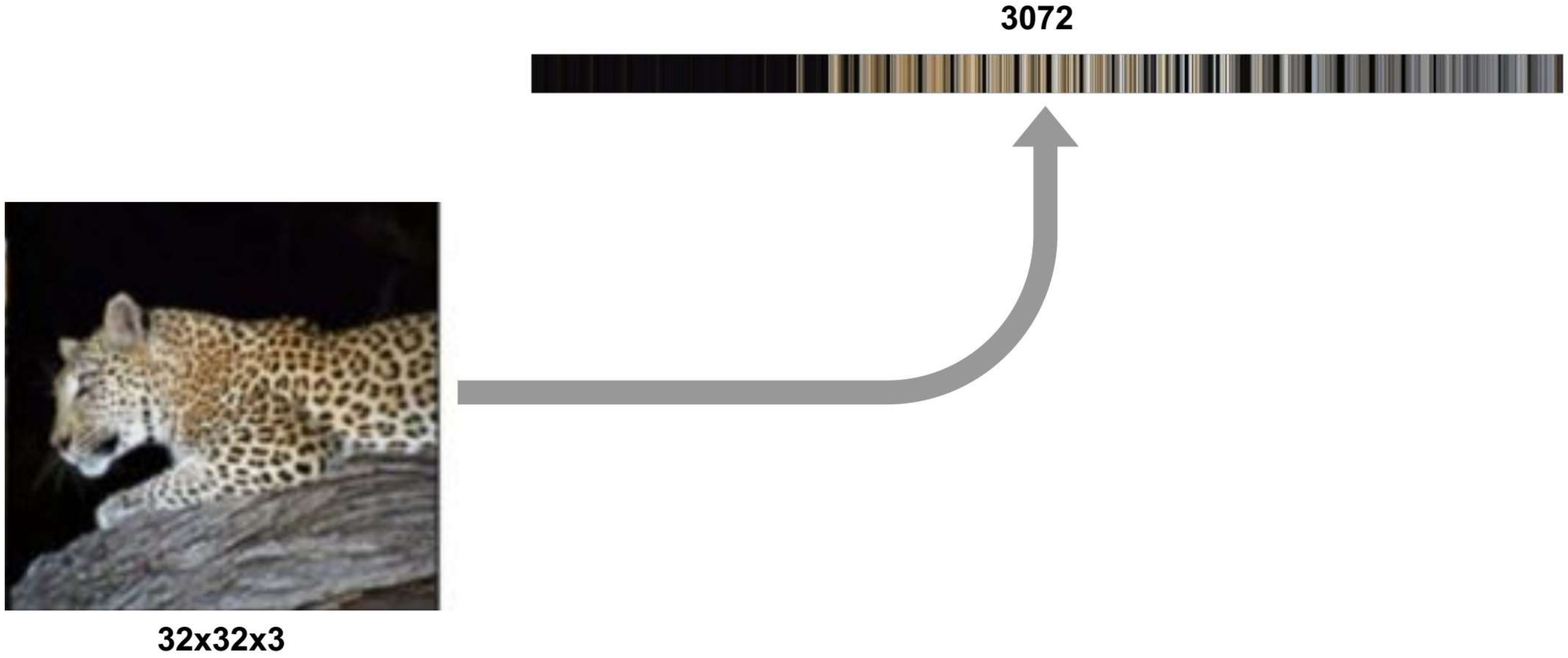
Images as data

- If we want to implement a MLP for image classification

32x32x3

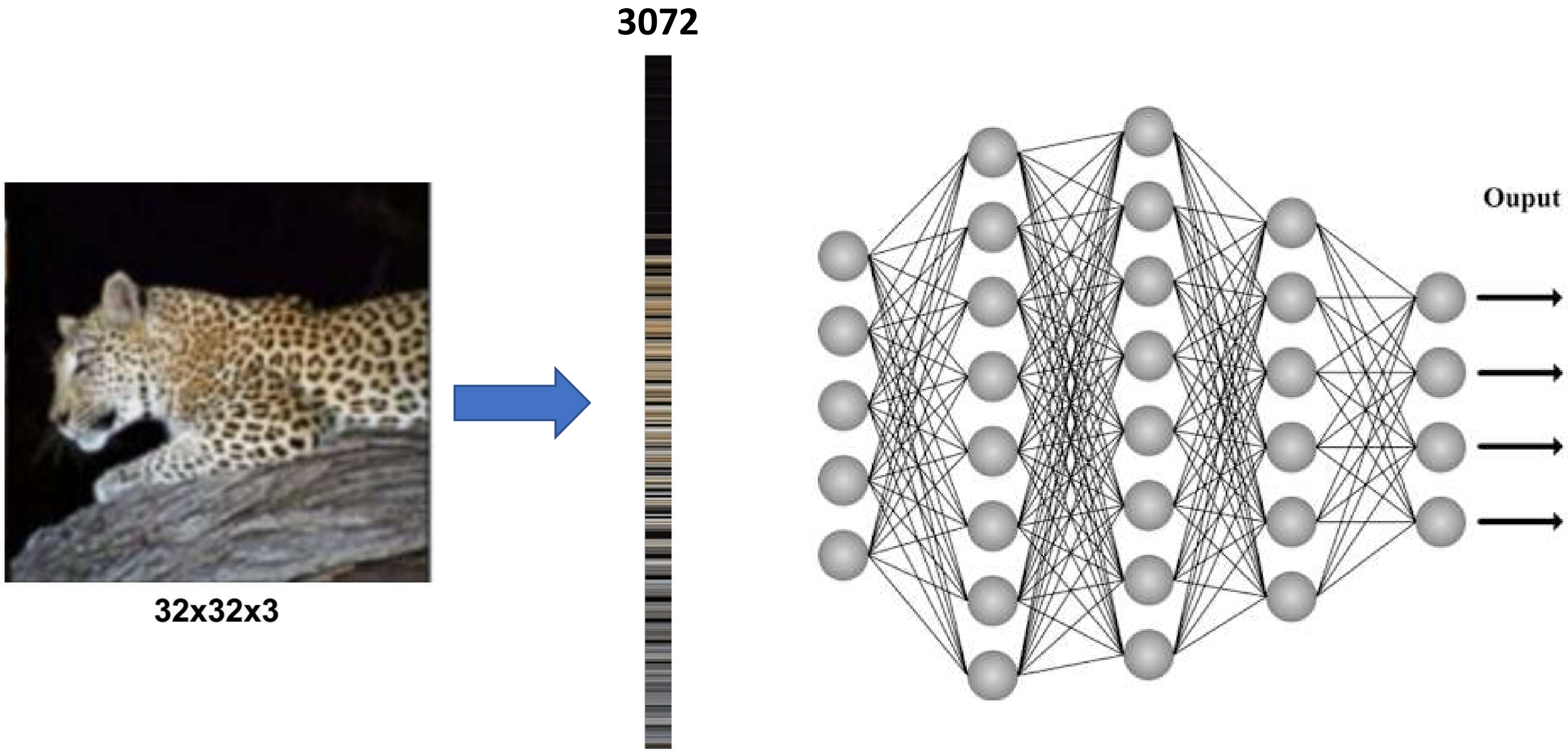
Images as data

- If we want to implement a MLP for image classification



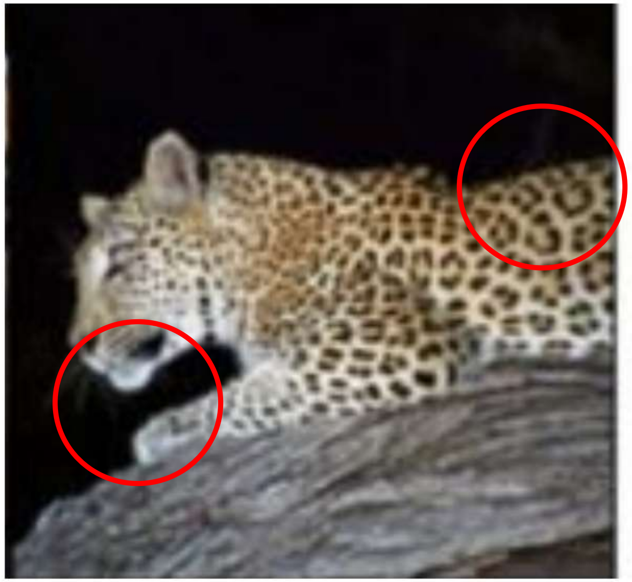
Images as data

- If we want to implement a MLP for image classification



Images as data

- If we want to implement a MLP for image classification
- The amount of parameters is huge
- A MLP tries to find all posible relations in the input data. Is that good?



Visual structure information is local, not global

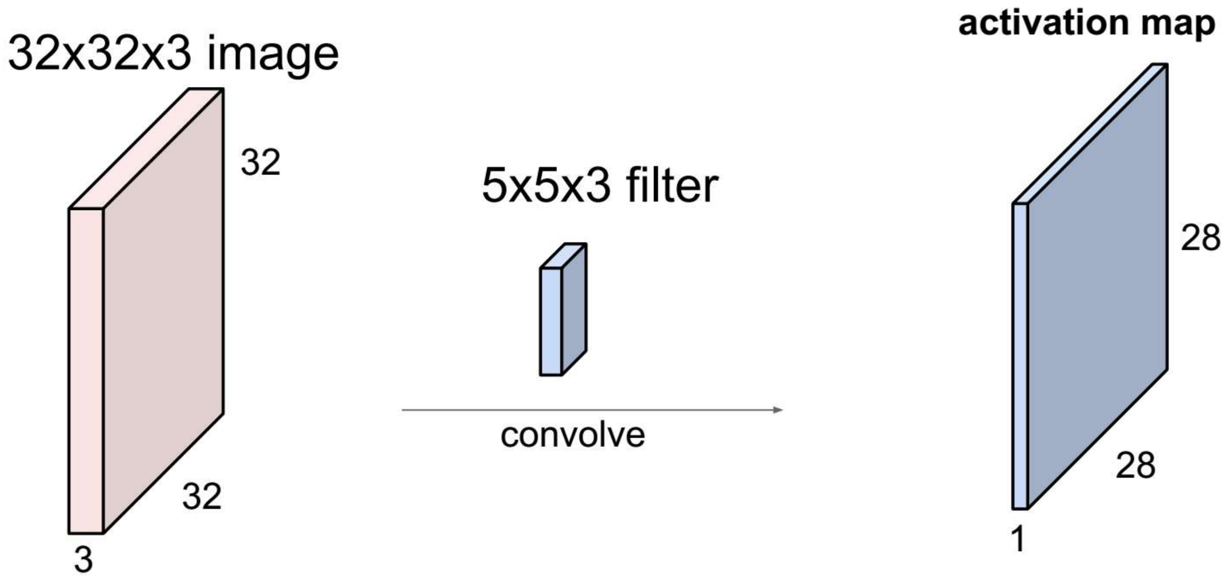
Pixels in far regions probably do not compose interesting content

We know an operation for computing local relationships

Convolution!

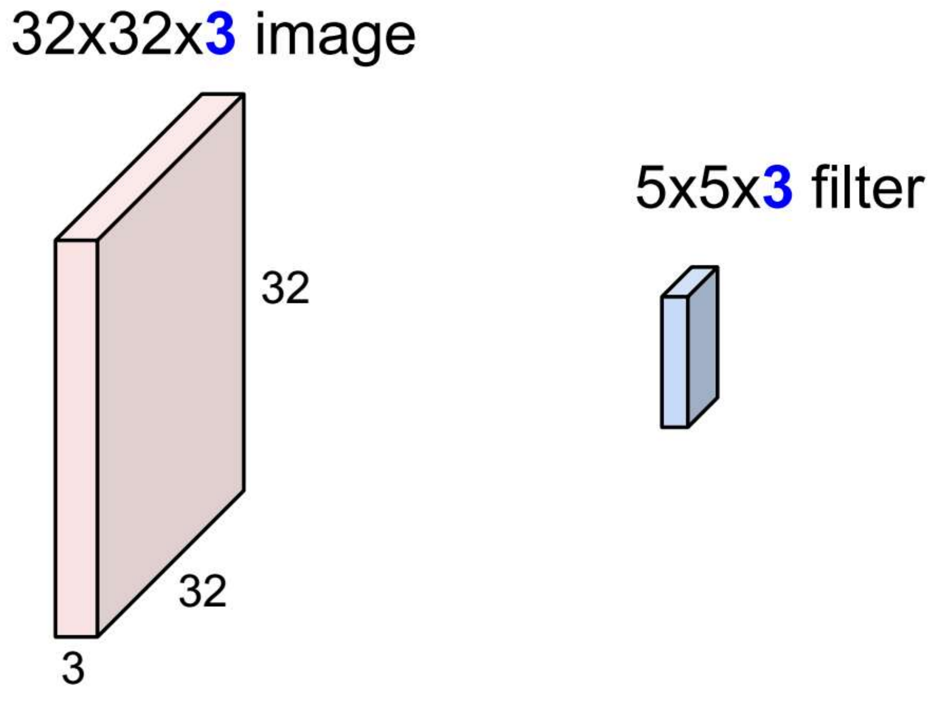
Convolutional Layers

- A convolutional layer is composed of a set of kernels
- Each kernel is convolved with the input and computes an activation map (also known as feature map)



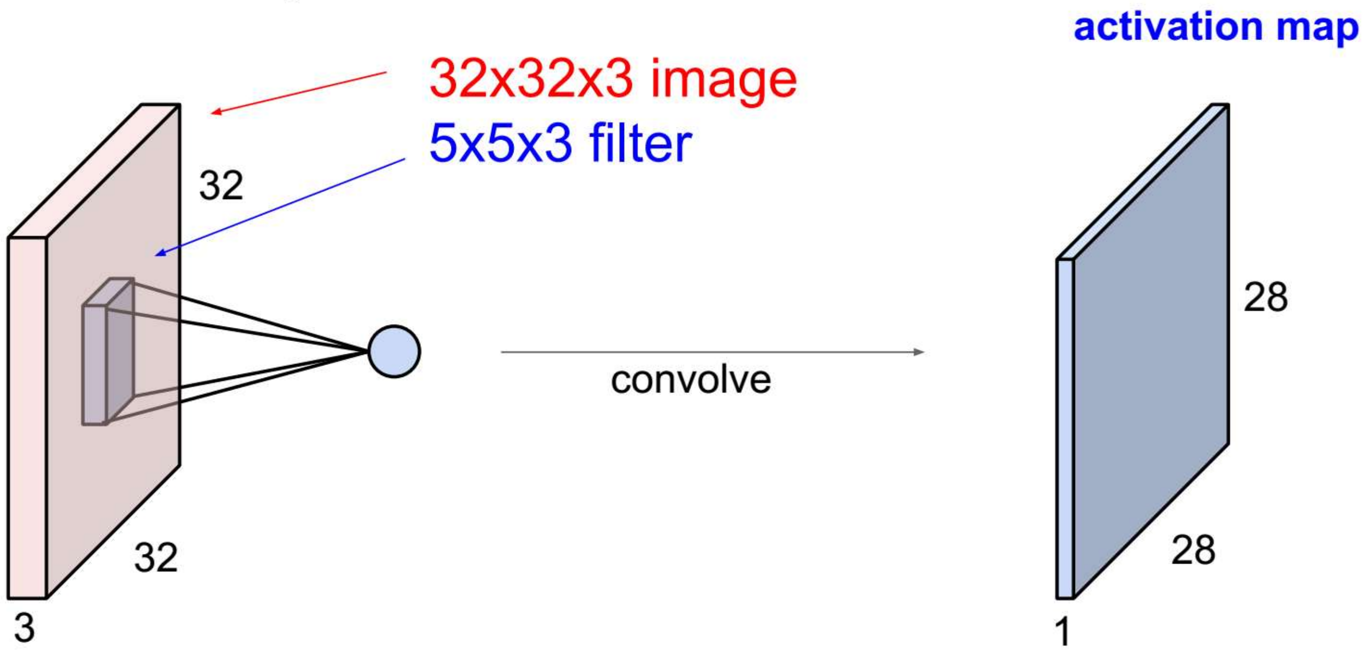
Convolutional Layers

- The depth of the input is always the depth of the kernels



Convolutional Layers

- We can have several kernels in the same layer
- Each kernel generates an activation map



Convolutional Layers

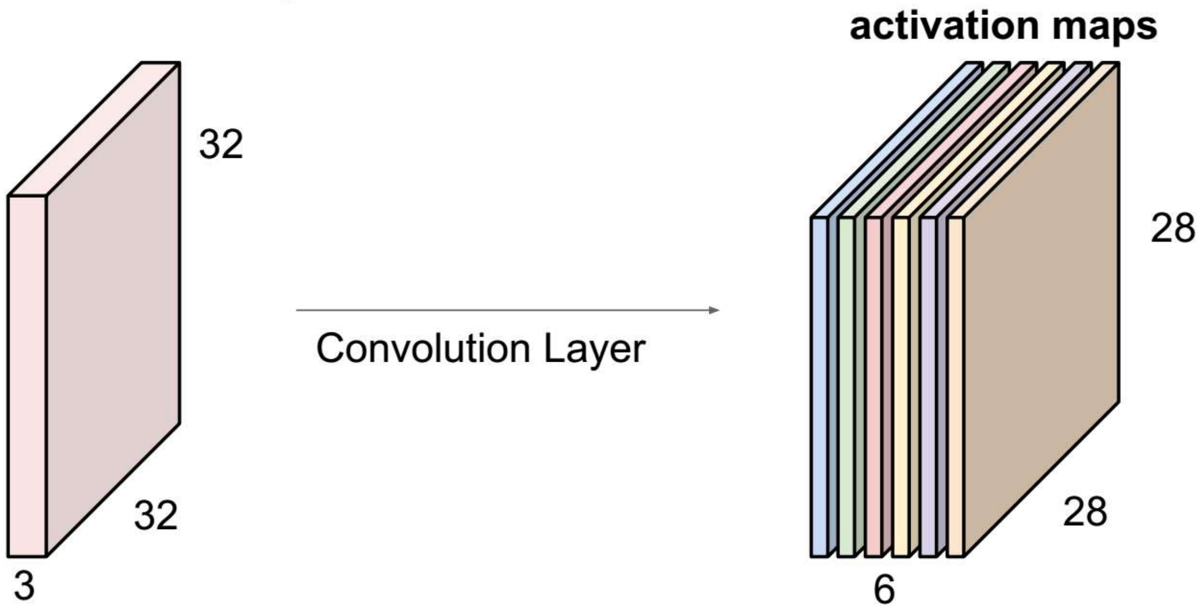
- We can have several kernels in the same layer
- Each kernel generates an activation map

For a second kernel

Convolutional Layers

- We can have several kernels in the same layer
- Each kernel generates an activation map

If we use, six kernels, the output activation map will have depth 6



Convolutional Layers

- Parameters
  - In a linear layer, the parameters are weights and the bias, and the forward operation is as follows

$$z_L = X_{L-1} \cdot W_L + \ b_L$$

  - The linear operation is the dot product between the input and the weights
  - In the convolutional layer, we have

$$z_L = X_{L-1} * W_L + \ b_L$$

    where the linear operation is the convolution, and the output is a feature map. W_L represents the kernel ans the bias is still a single value.

Convolutional Layers

- How much parameters do we have if my convolutional layer has m kernels of dimension $k \times k \times d$?

$$\underbrace{m \times k \times k \times d}_{\text{Kernel parameters}} + \underbrace{m}_{\text{Biases}}$$

Summary

- Convolutional layer is the core of the current progress in computer vision
- Convolution is a linear operation, hence we can include it in the feedforward pass with no problems
- As convolution is linear, backpropagation is straightforward

Neural networks

Hyperparameters and Pooling

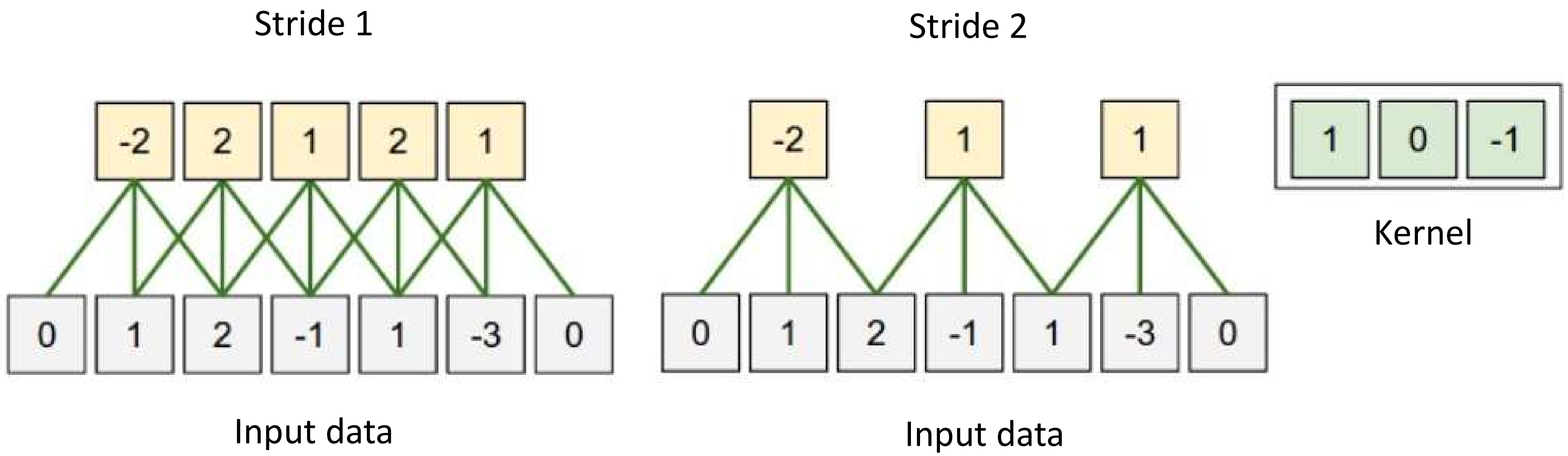
Parameters

- To build a convolutional layer, we need to set
  - K: the number of kernels in the layer
  - F: the spatial extension of the kernel
  - S: the stride (number of pixels to jump during sliding convolution)
  - P: the number f pixels for padding

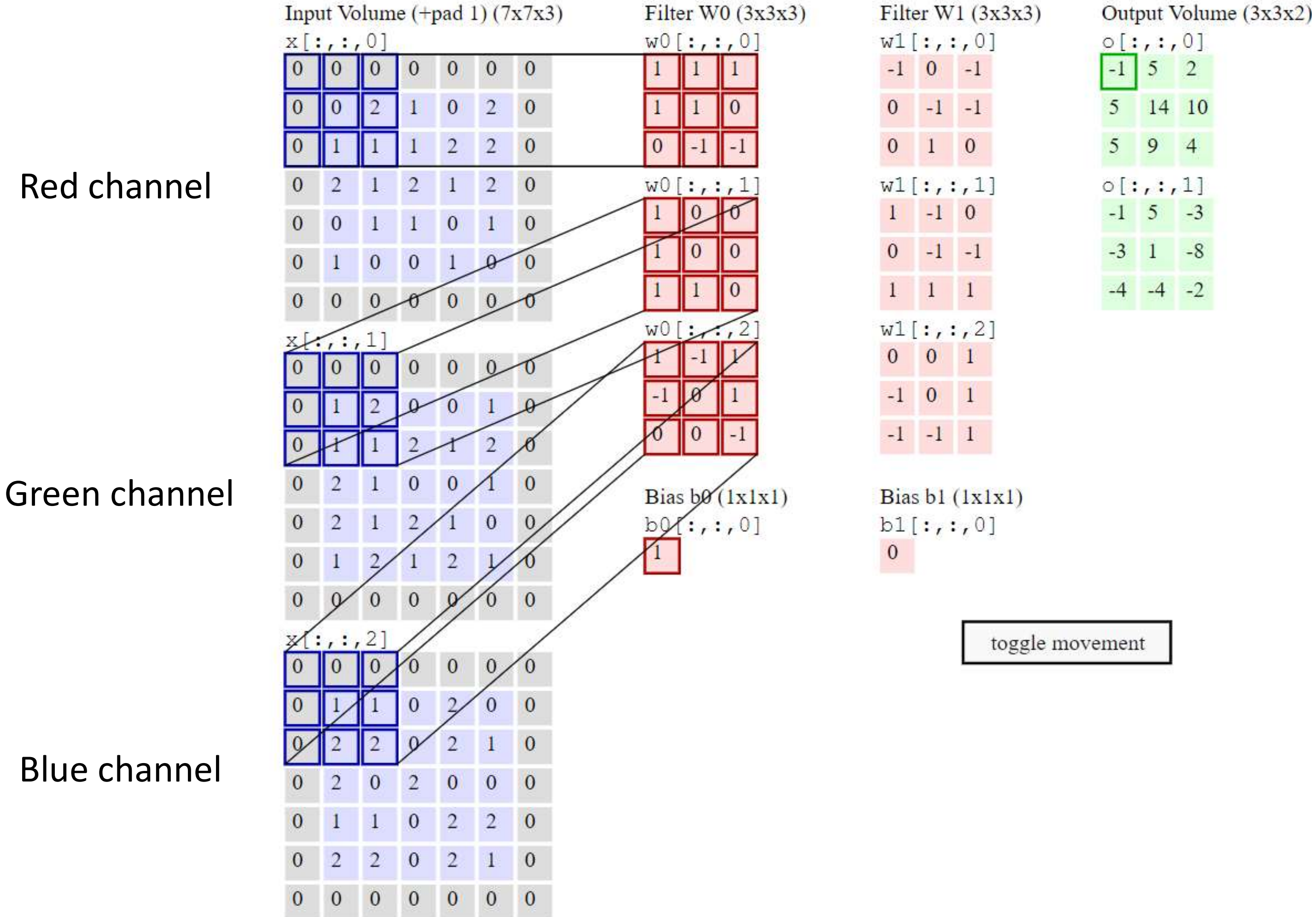
If your input volumen has size W, the size of the activation map in the convolutional layer is

$$\frac{W - F + 2P}{S} + 1$$

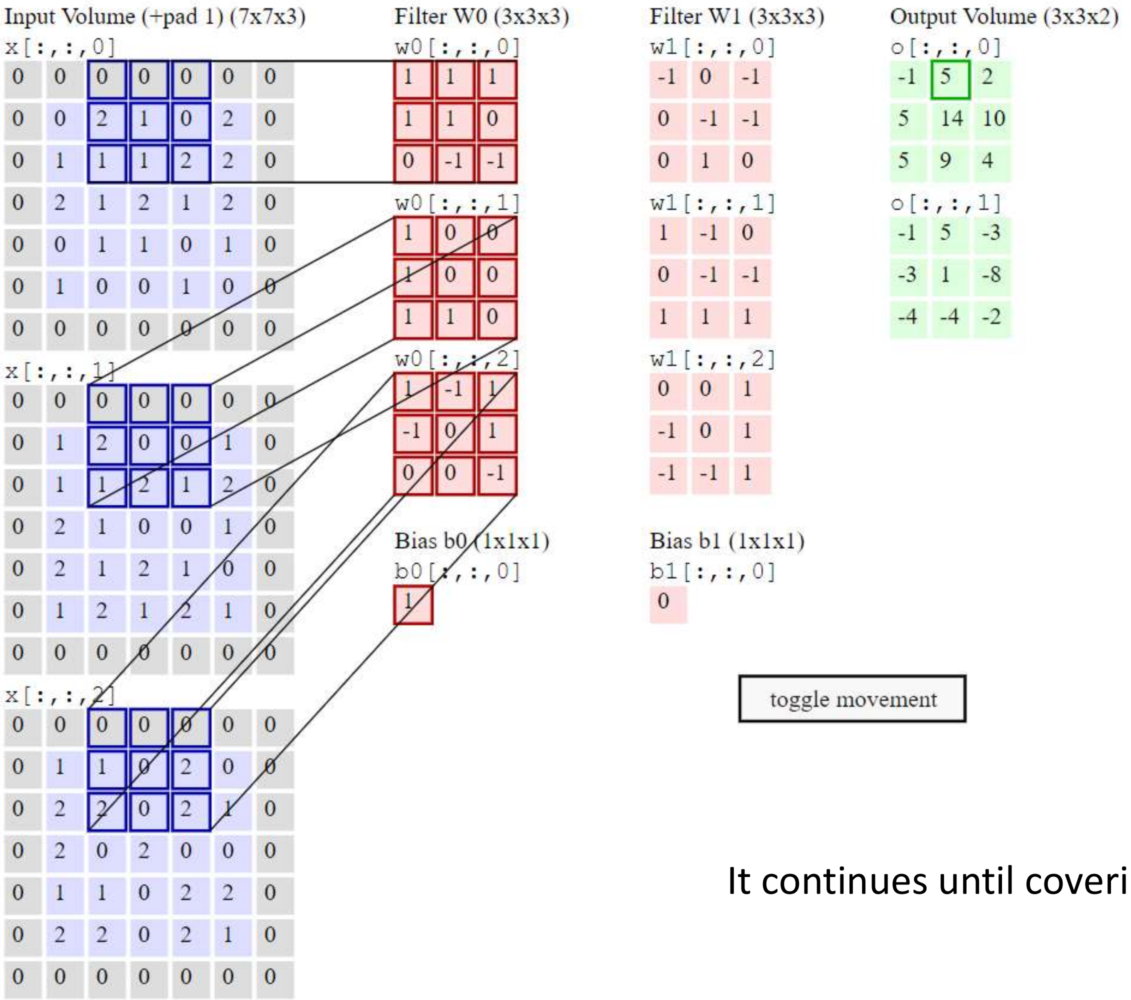
Example - Stride



Complete example

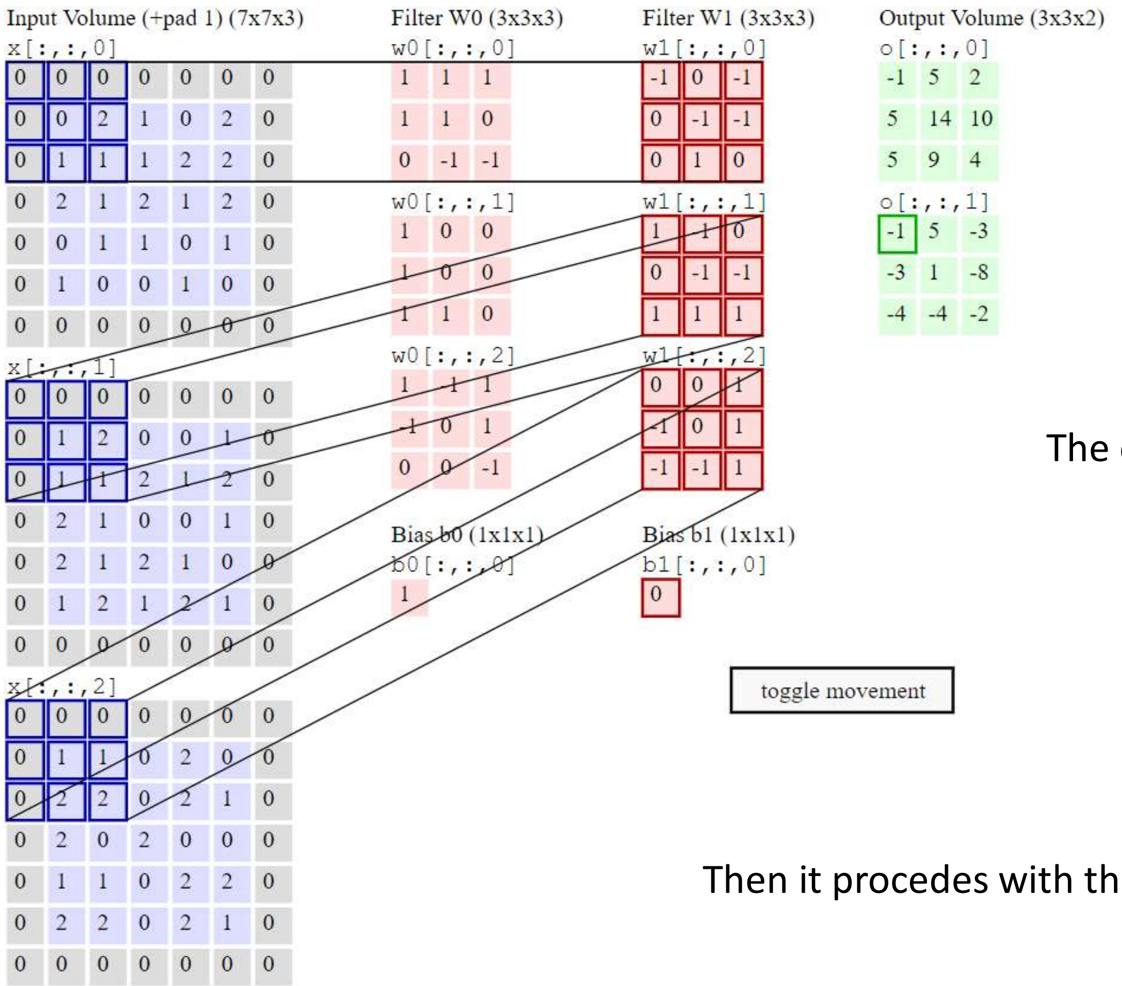


Complete example



It continues until covering the complete image

Complete example

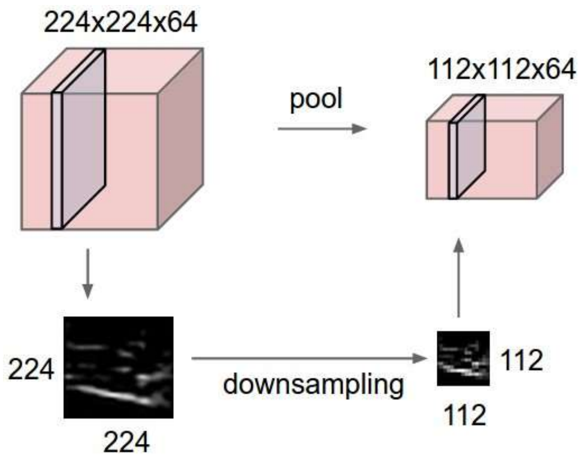


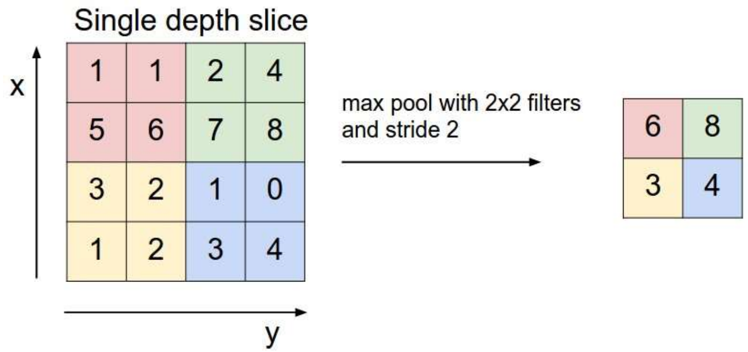
The output volumen is 3x3x2

Then it procedes with the second kernel

Pooling Layer

- The goal is to reduce the spatial size of the representation
  - Reduces the amount of parameters
  - Reduces the computation
  - Allows us to find multi-scale representation in images
- Max pooling, size 2, stride 2 is the typical choice





Summary

- Hyperparameters for ConvNets
- Volume computation
- Pooling layers