Artificial intelligence in data science

Convolutional neural networks

Janos Török

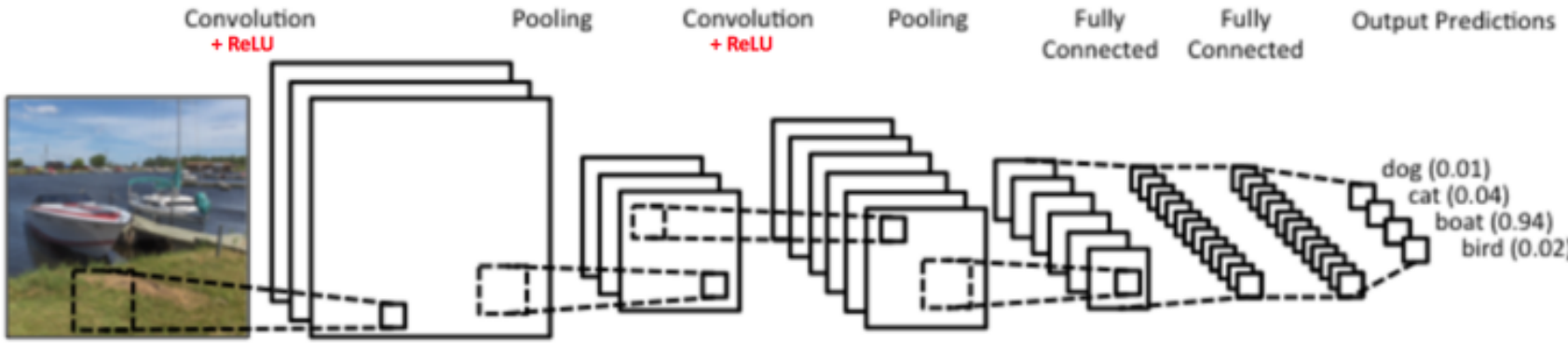
Department of Theoretical Physics

October 13, 2022

LeNet Architecture

Yann LeCun

- Convolution
- Non-linearity
- Pooling
- Classification



Author: ujjwalkarn

Convolution operator

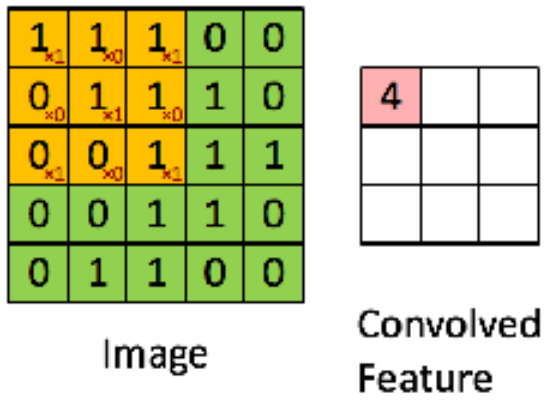
- 2d matrix
- Example:
  - Original image:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

  - Convolution matrix:

1	0	1
0	1	0
1	0	1

  - Result:



Convolution operator

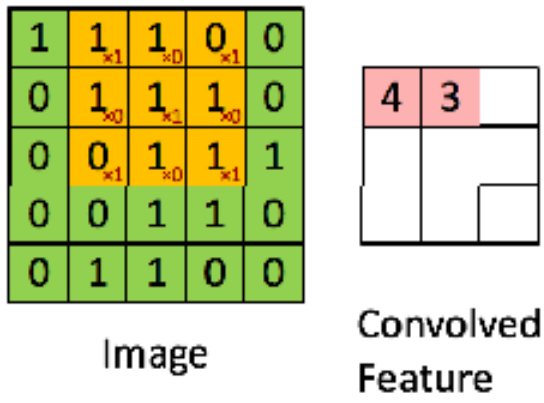
- 2d matrix
- Example:
  - Original image:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

  - Convolution matrix:

1	0	1
0	1	0
1	0	1

  - Result:



Convolution operator

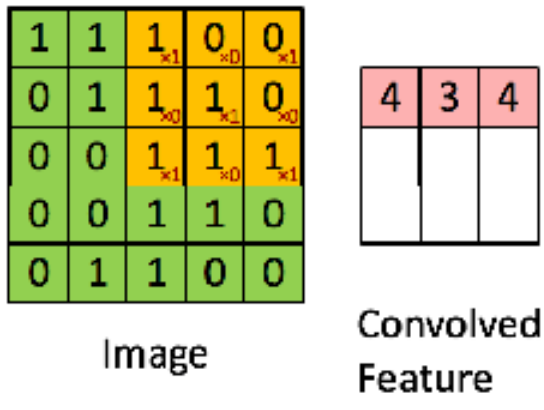
- 2d matrix
- Example:
  - Original image:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

  - Convolution matrix:

1	0	1
0	1	0
1	0	1

  - Result:



Convolution operator

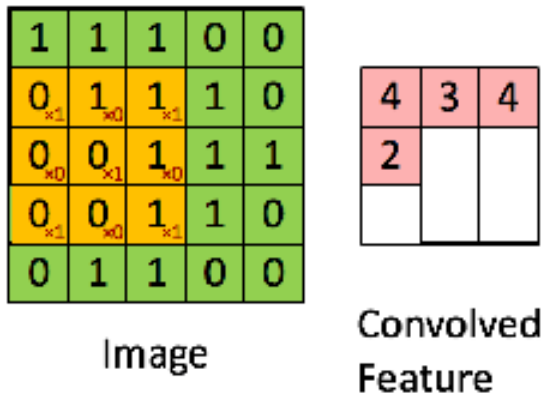
- 2d matrix
- Example:
  - Original image:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

  - Convolution matrix:

1	0	1
0	1	0
1	0	1

  - Result:



Convolution operator

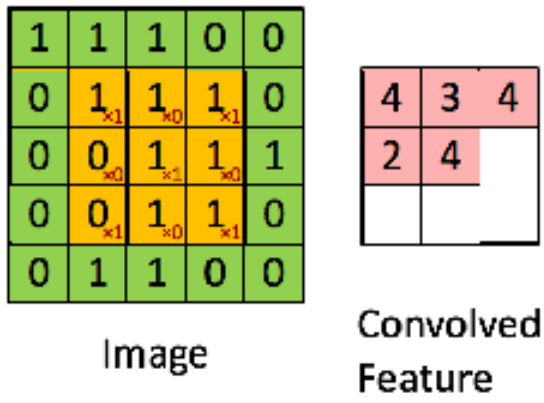
- 2d matrix
- Example:
  - Original image:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

  - Convolution matrix:

1	0	1
0	1	0
1	0	1

  - Result:



Convolution operator

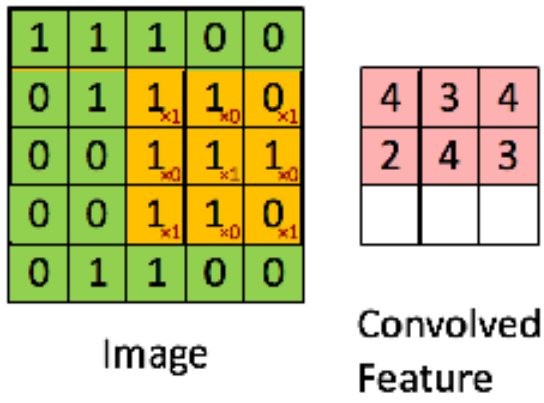
- 2d matrix
- Example:
  - Original image:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

  - Convolution matrix:

1	0	1
0	1	0
1	0	1

  - Result:



Convolution operator

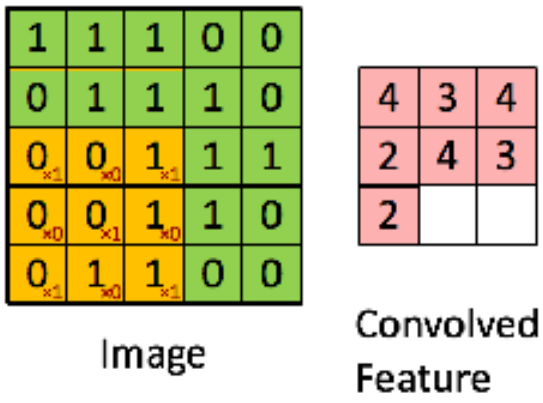
- 2d matrix
- Example:
  - Original image:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

  - Convolution matrix:

1	0	1
0	1	0
1	0	1

  - Result:



Convolution operator

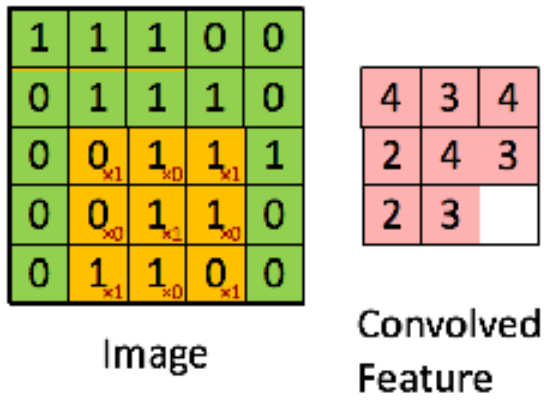
- 2d matrix
- Example:
  - Original image:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

  - Convolution matrix:

1	0	1
0	1	0
1	0	1

  - Result:



Convolution operator

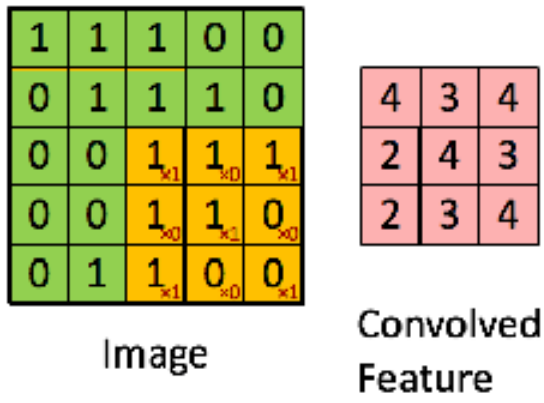
- 2d matrix
- Example:
  - Original image:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

  - Convolution matrix:

1	0	1
0	1	0
1	0	1

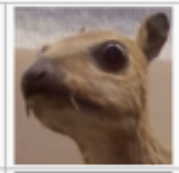
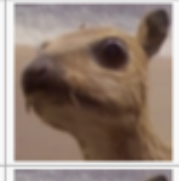
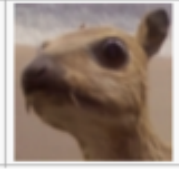
  - Result:



Convolution operator

- The convolution operator is called *filter or kernel*
- The result of the convolution is *feature map*

Convolution operator: examples

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

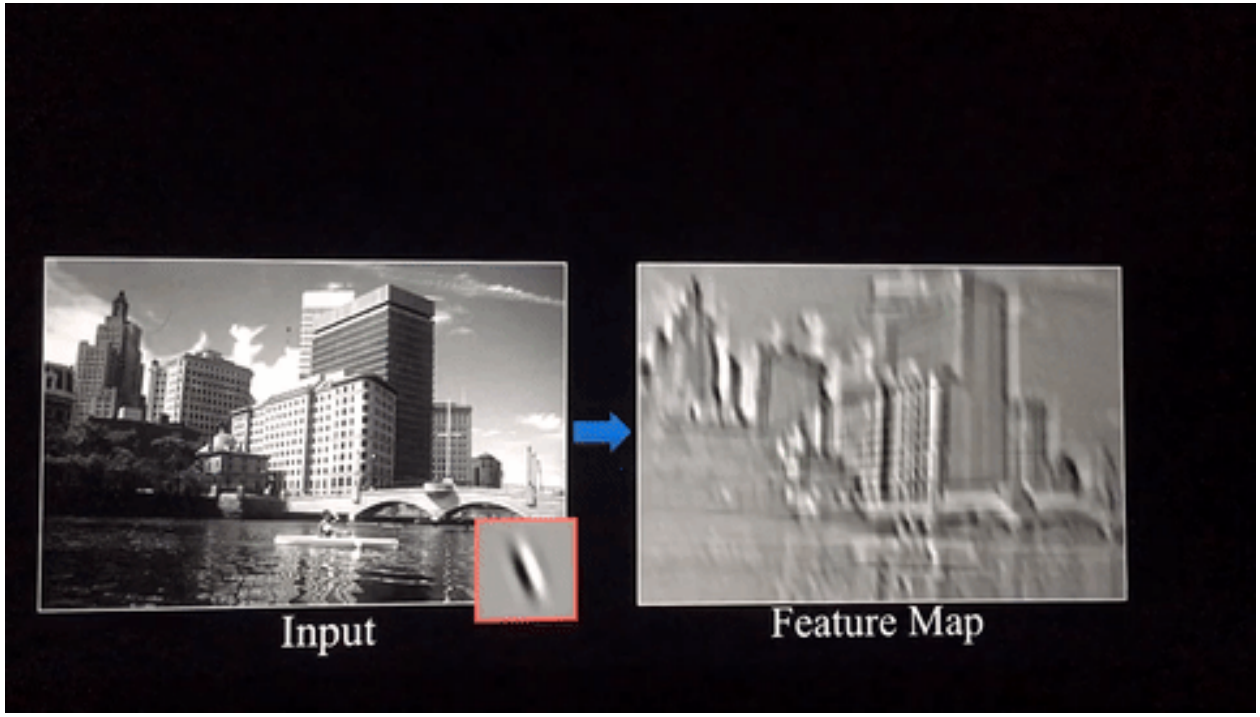
Convolution operator: examples, emboss

$$\begin{pmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

Convolution operator: examples

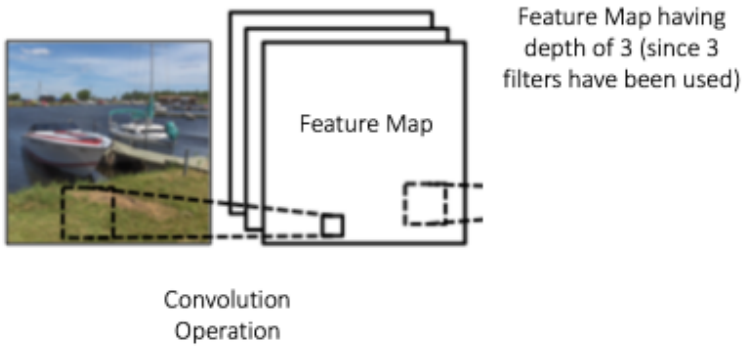
Convolution operator: examples

Convolution operator: examples



Convolution operator: application

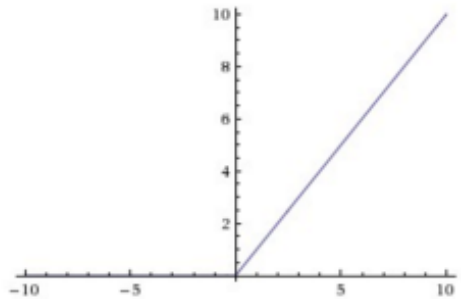
- **Depth:** number of color chanels
- **Stride:** number of pixels a filter moves each time
- **Padding:** What to do with the borders (zero-padding or no padding



Introduce non-linearity

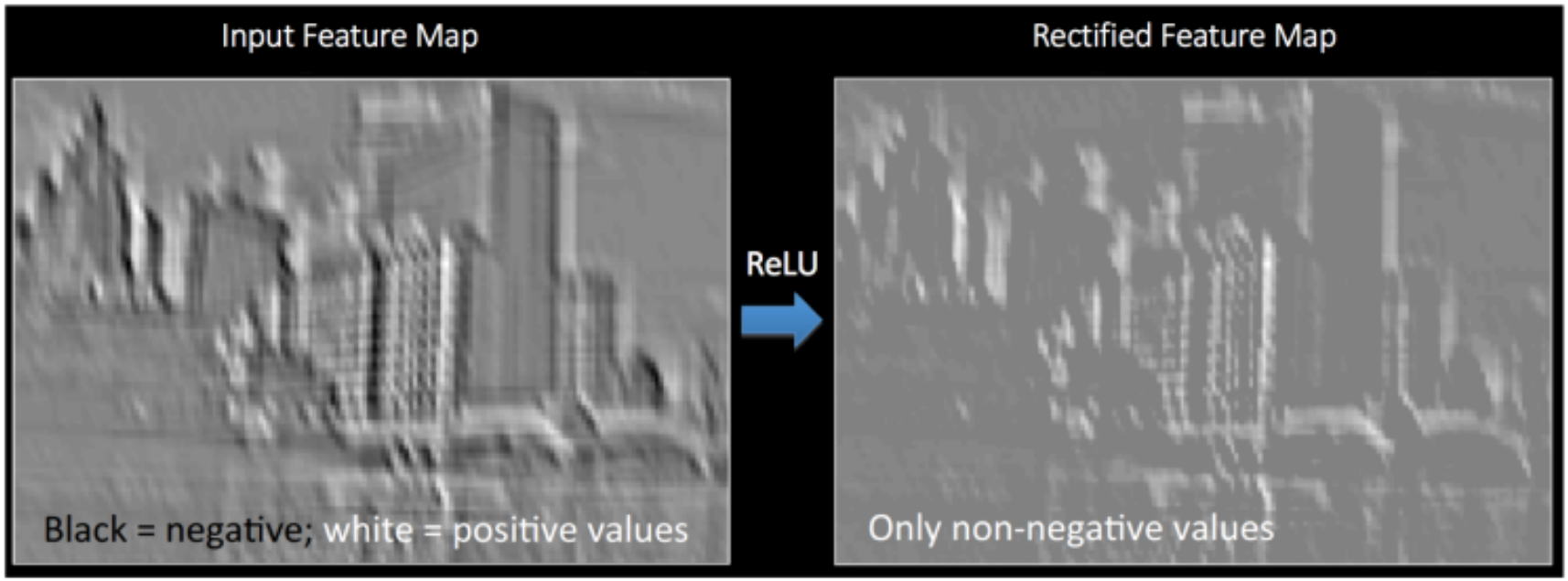
- Results of convolution can be negative
- Introduce non-linearity
- most common is `relu`

$$\text{Output} = \text{Max}(\text{zero}, \text{Input})$$



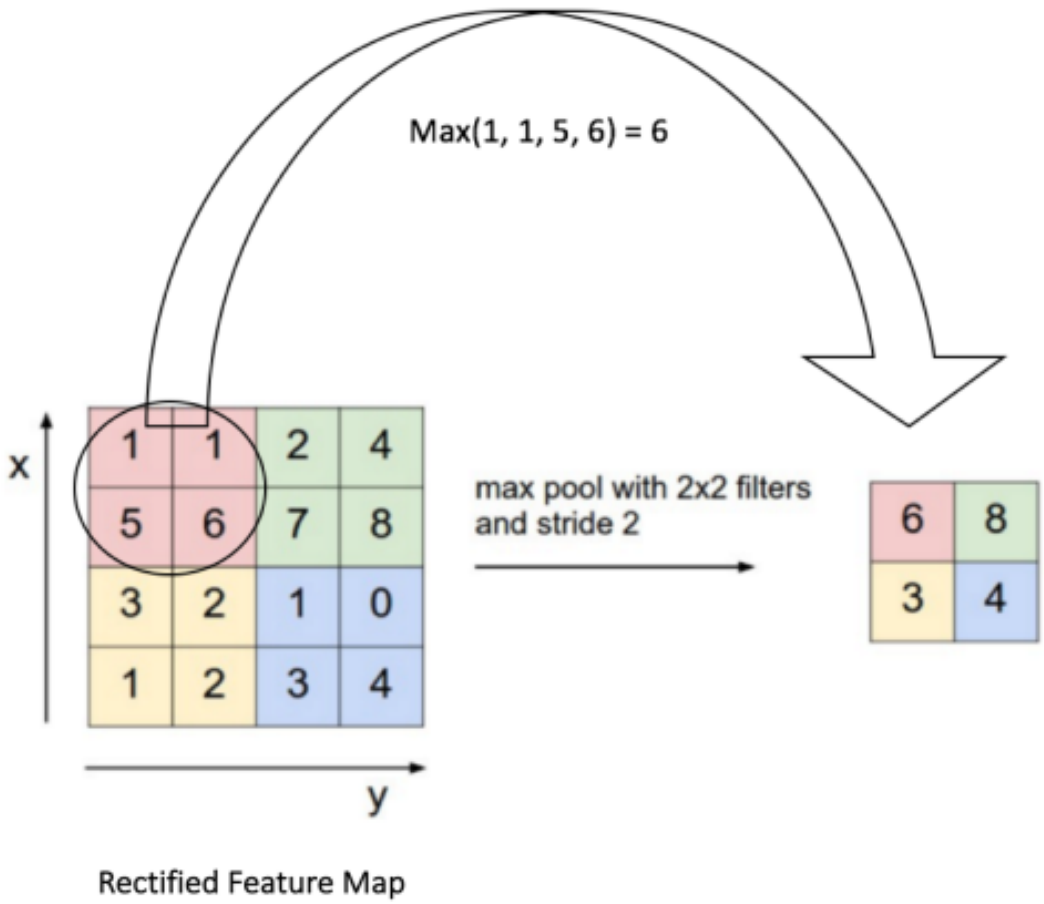
Introduce non-linearity

- Results of convolution can be negative
- Introduce non-linearity
- most common is `relu`, (sigmoid, tanh, etc.)

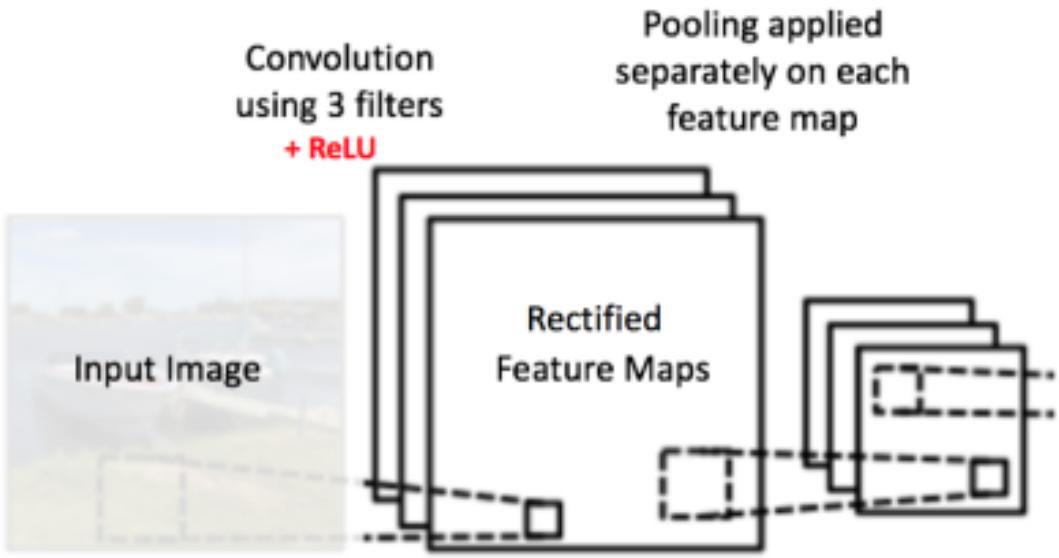


Pooling

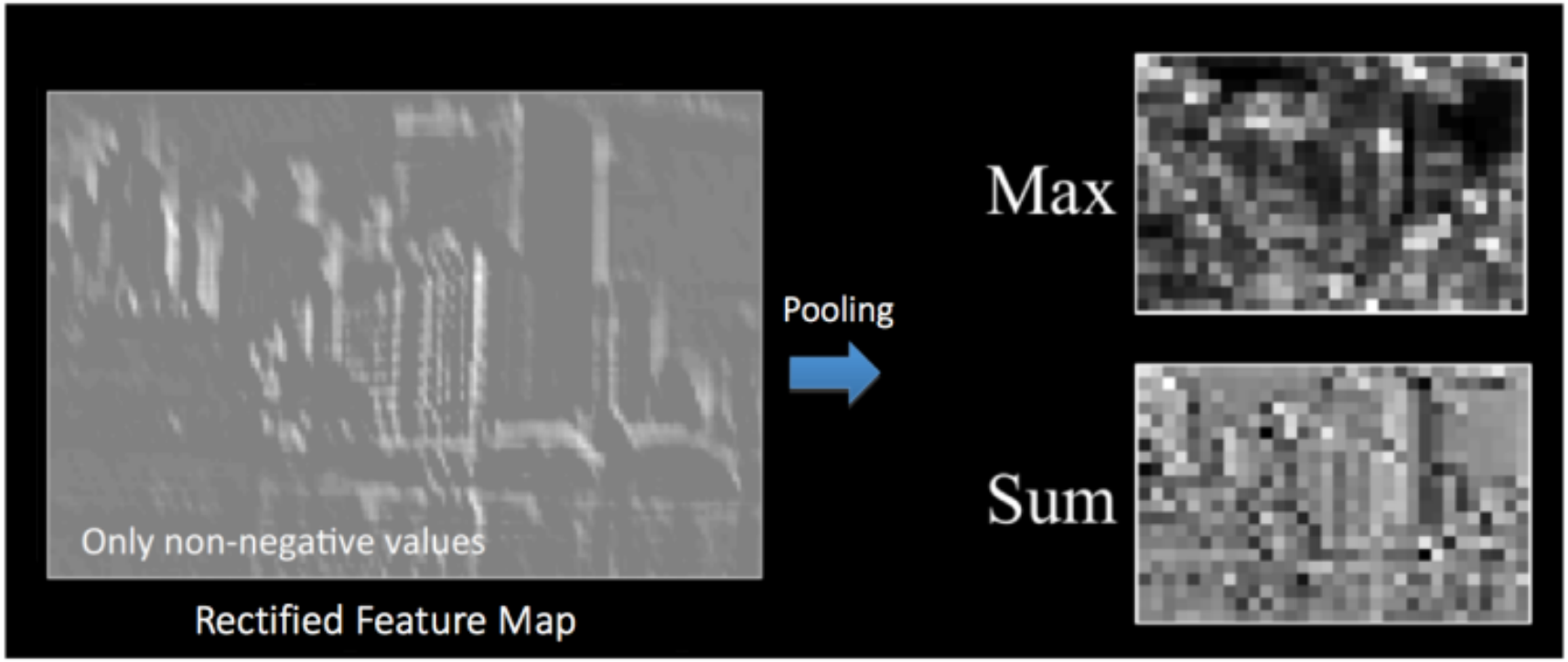
- Downsampling
- Parameter reduction
- Important for overfitting and running time
- Generally keep the maximun insode a square region



Pooling on the Rectified Feature Map

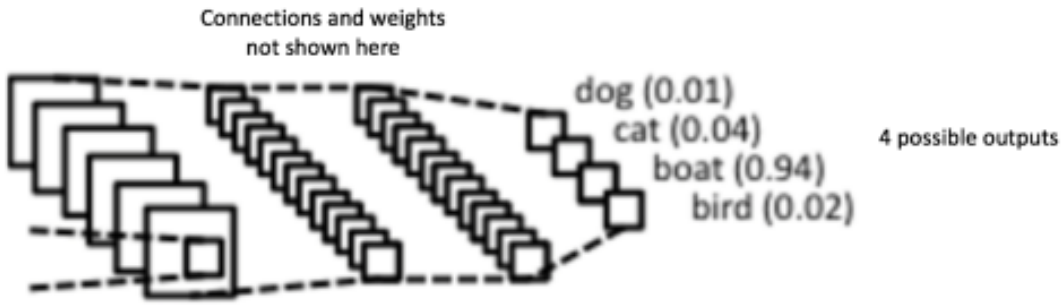


Pooling on the Rectified Feature Map

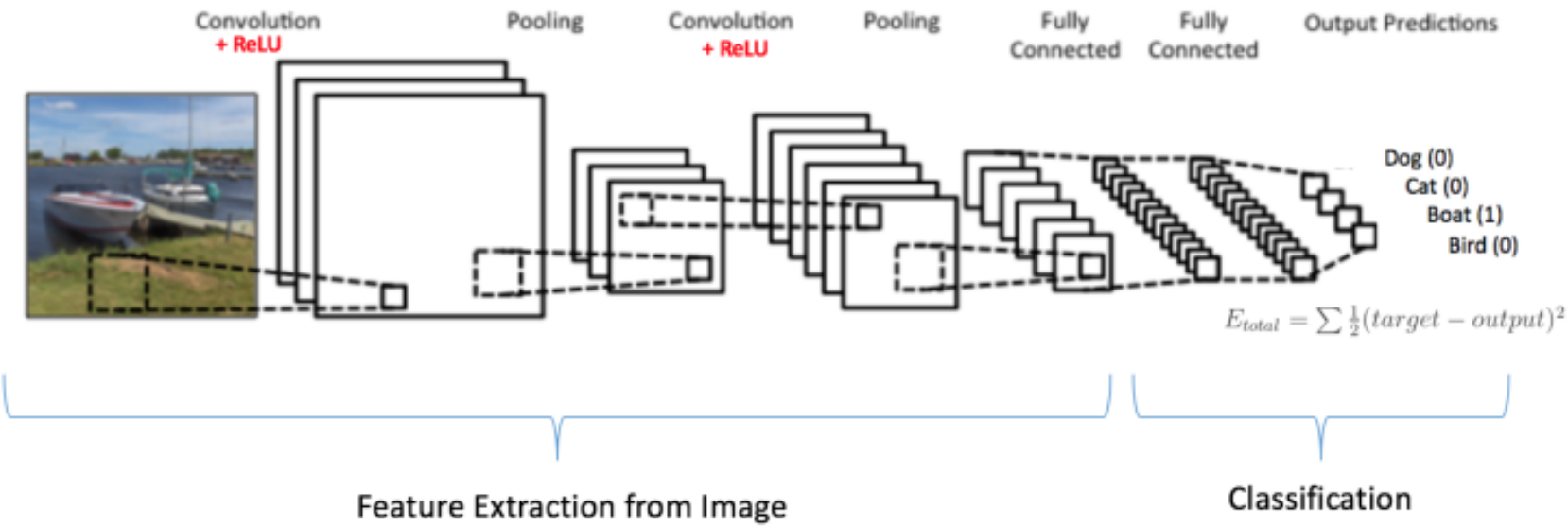


Classification layer

- Feedforward neural network
- Very often dropout: randomly setting a fraction rate of input units to 0 at each update during training time

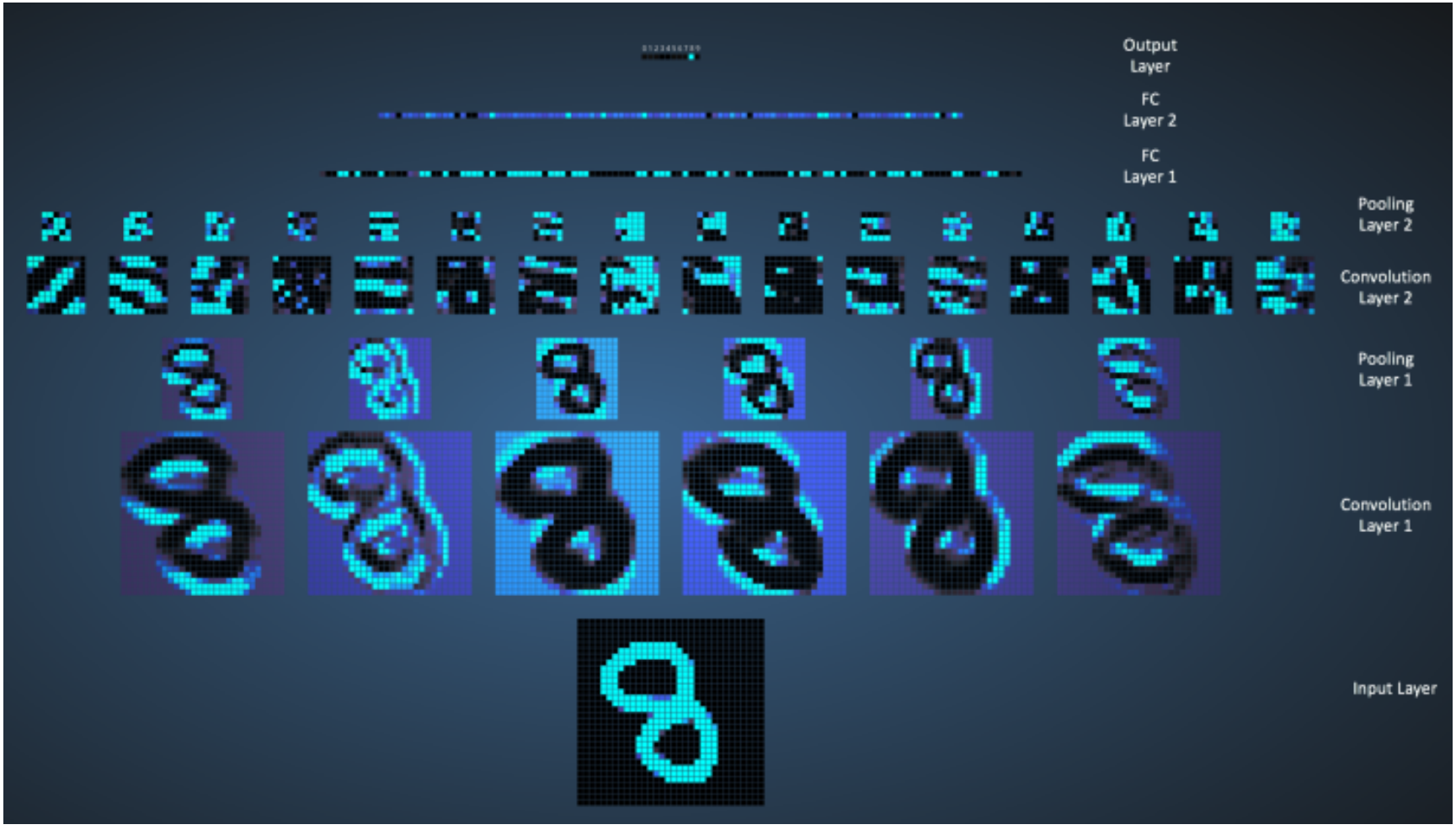


Full convolution neural network



Examples of learned features

- MNIST example
  `https://adamharley.com/nn_vis/cnn/2d.html`



Convolutional layers

- Number of parameters (200x200 RGB image):
  - Fully connected layer to a layer of 300 nodes:
    - Weights: $(200 \cdot 200 \cdot 3) \cdot 300$
    - Biases: 300
    - Total: $36000300 \simeq 3.6 \cdot 10^7$
  - Convolutional layer
    - Weights per filter $w \cdot w \cdot 3$, where w is the width of the filters
    - One bias
    - Number of weights per filter $w^2 + 1$
    - For 300 filter (usually people use only a few dozens)
    - Total: $300 \cdot 10 = 3000$