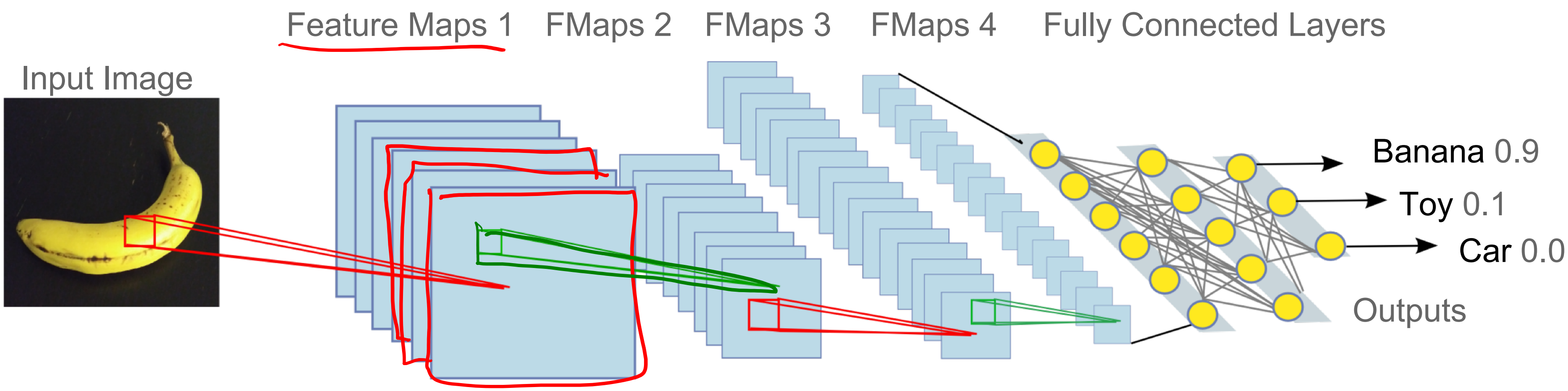
Where Do Kernels Come From?

- Kernel is critical
- In the past, human-engineered
- Tricky to come up with, depending on the task to be performed
- No analytical approach to derive numbers
- CNNs solve this problem

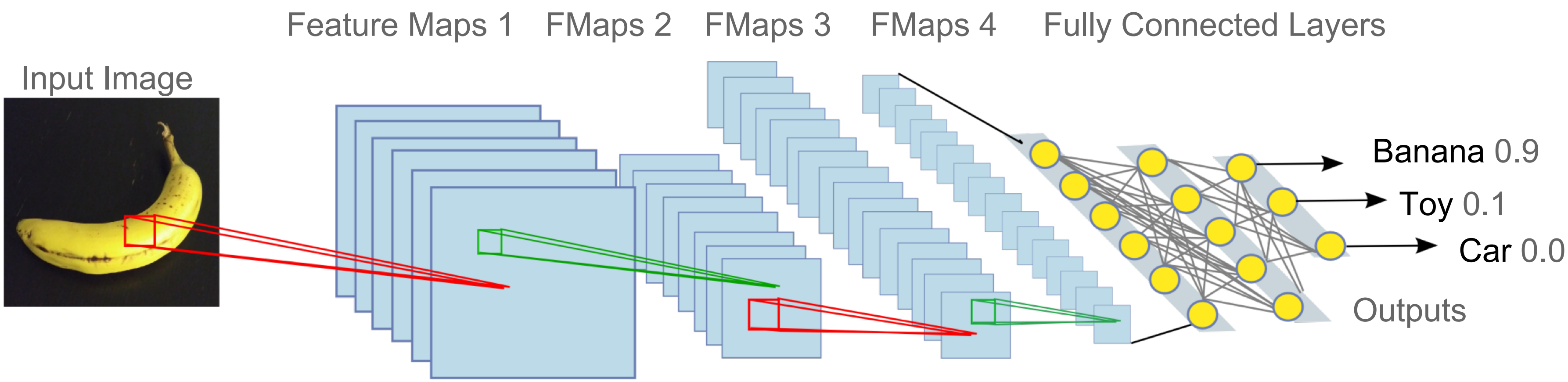
Convolution Kernel

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Convolutional Neural Networks



Convolutional Neural Networks

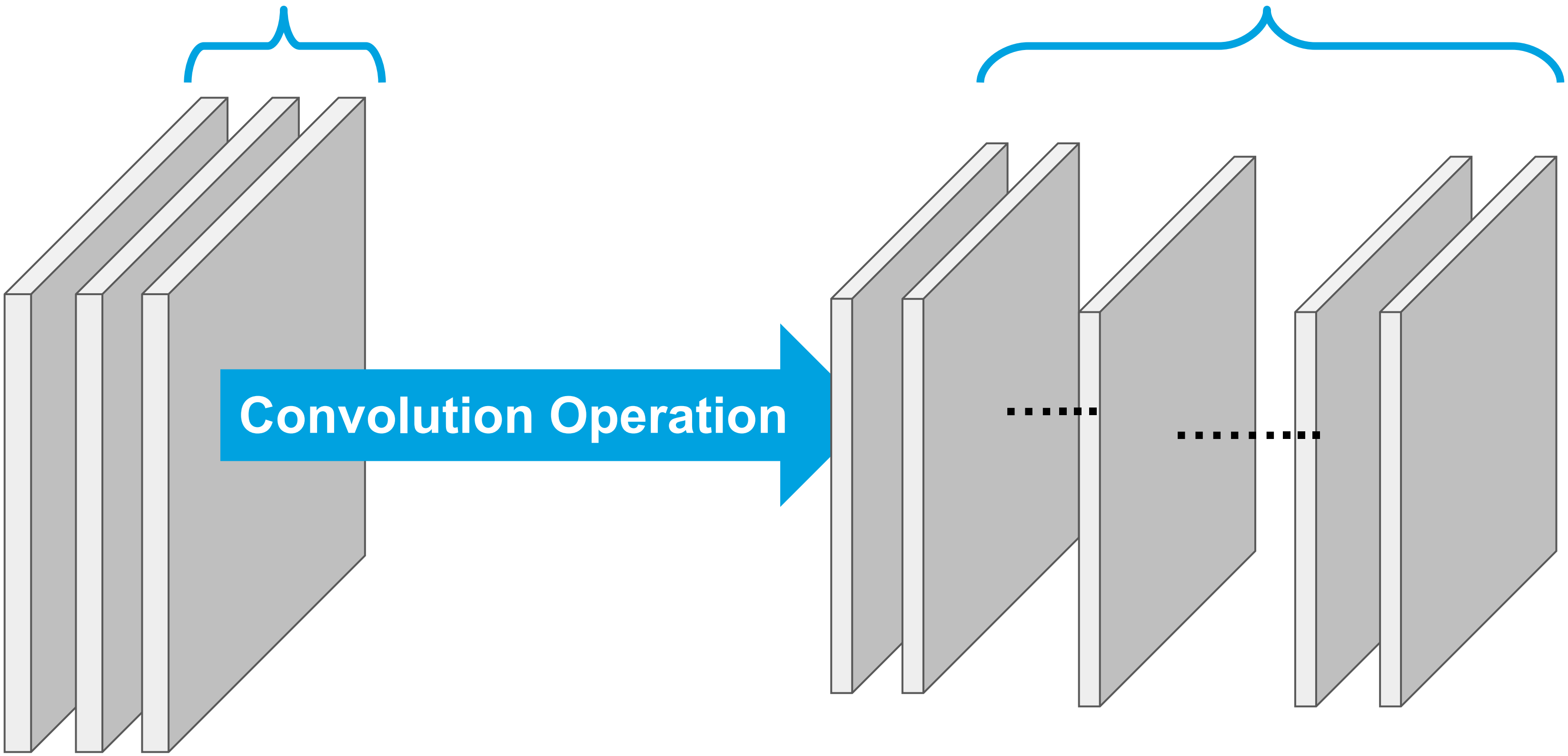


Convolution (+ReLU)

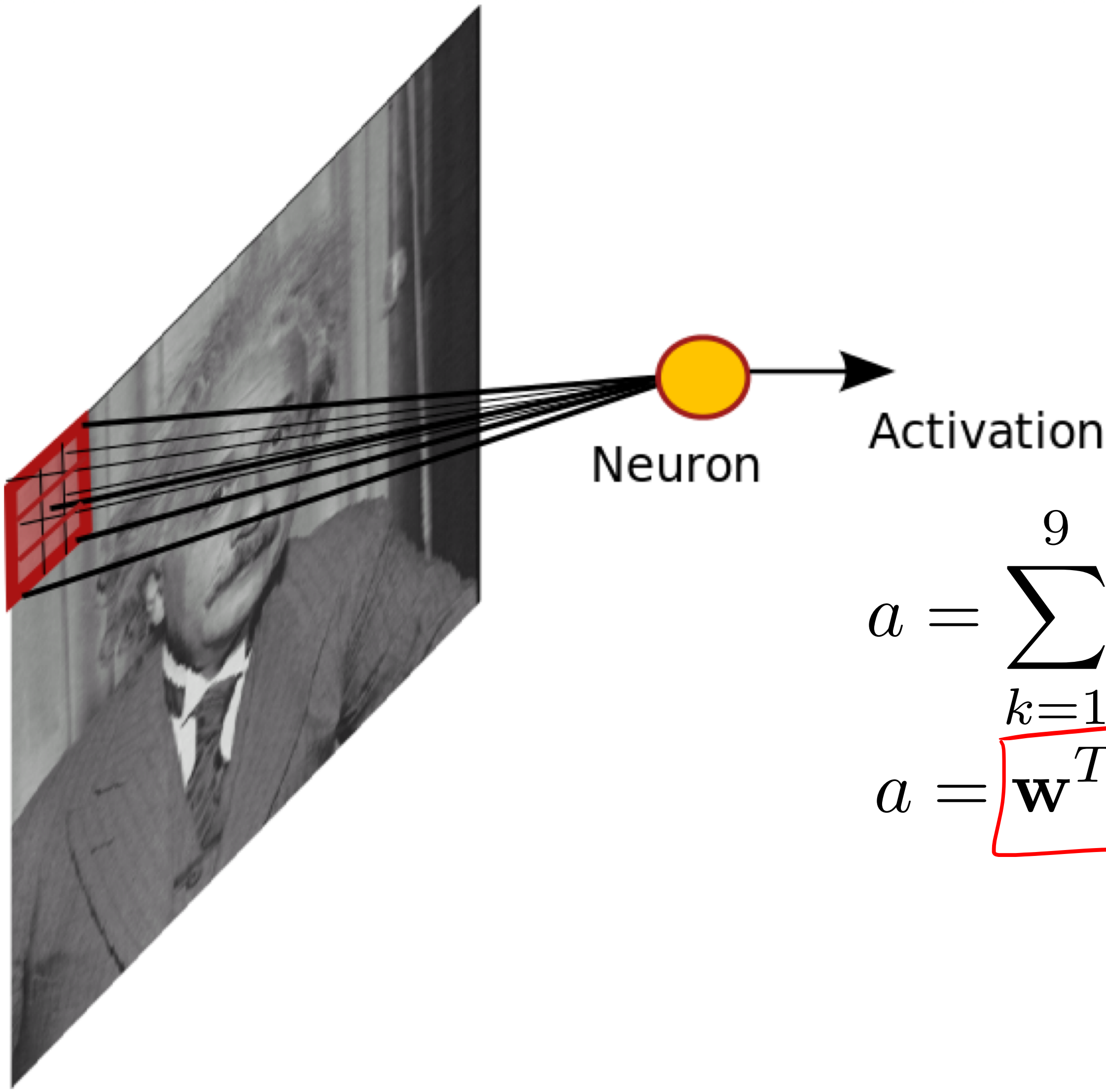
Convolutions in CNN

Input:
Set of activation maps, often an RGB image

Output:
Set of activation maps or feature maps where each individual map represents a spatial similarity matrix between the input and its corresponding filter or neuron



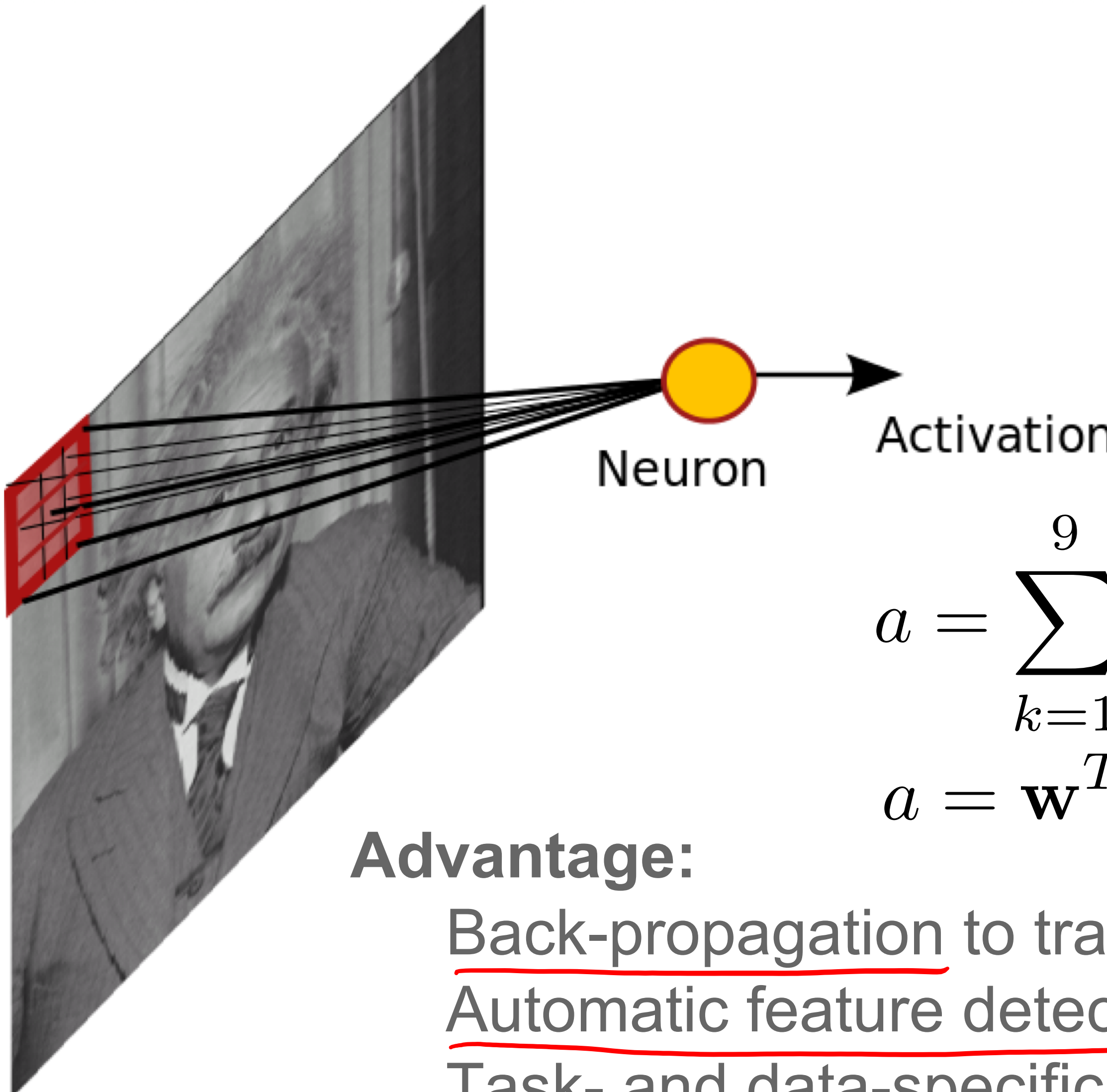
Convolutions as Neurons?



$$a = \sum_{k=1}^{9} w_k x_k$$

$$a = \boxed{\mathbf{w}^T \mathbf{x}}$$

Convolutions as Neurons?



$$a = \sum_{k=1}^{9} w_k x_k$$

$$a = \mathbf{w}^T \mathbf{x}$$

Advantage:

- Back-propagation to train the weights
- Automatic feature detection
- Task- and data-specific

Example Convolution Calculation

$$20 = \underline{3 \times 0 + 6 \times 1 + 7 \times 2 + 9 \times 0}$$

input image

$x_t \in \mathbb{R}^{5\times 5}$

$$\begin{bmatrix} 3 & 6 & 1 & 4 & 2 \\ 7 & 9 & 1 & 1 & 4 \\ 1 & 0 & 0 & 1 & 3 \\ 0 & 2 & 4 & 5 & 2 \\ 1 & 1 & 7 & 7 & 1 \end{bmatrix}$$

Kernel

$k \in \mathbb{R}^{2\times 2}$

$$\begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}$$

Convolve *x* with *k*

activation map

$h_t \in \mathbb{R}^{4\times 4}$

$$\begin{bmatrix} \underline{20} & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

Example Convolution Calculation

$$20 = 3 \times 0 + 6 \times 1 + 7 \times 2 + 9 \times 0$$

$$19 = 6 \times 0 + 1 \times 1 + 9 \times 2 + 1 \times 0$$

$$x_t \in \mathbb{R}^{5 \times 5}$$

$$\begin{bmatrix} 3 & 6 & 1 & 4 & 2 \\ 7 & 9 & 1 & 1 & 4 \\ 1 & 0 & 0 & 1 & 3 \\ 0 & 2 & 4 & 5 & 2 \\ 1 & 1 & 7 & 7 & 1 \end{bmatrix}$$

$$k \in \mathbb{R}^{2 \times 2}$$

$$\begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}$$

Convolve ***x*** with ***k***

$$h_t \in \mathbb{R}^{4 \times 4}$$

$$\begin{bmatrix} 20 & 19 & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

Example Convolution Calculation

$$20 = 3 \times 0 + 6 \times 1 + 7 \times 2 + 9 \times 0$$
$$19 = 6 \times 0 + 1 \times 1 + 9 \times 2 + 1 \times 0$$
$$\vdots$$
$$11 = 7 \times 0 + 9 \times 1 + 1 \times 2 + 0 \times 0$$

$$x_t \in \mathbb{R}^{5\times5}$$
$$\begin{bmatrix} 3 & 6 & 1 & 4 & 2 \\ 7 & 9 & 1 & 1 & 4 \\ 1 & 0 & 0 & 1 & 3 \\ 0 & 2 & 4 & 5 & 2 \\ 1 & 1 & 7 & 7 & 1 \end{bmatrix}$$

$$k \in \mathbb{R}^{2\times2}$$
$$\begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}$$

Convolve *x* with *k*

$$h_t \in \mathbb{R}^{4\times4}$$
$$\begin{bmatrix} 20 & 19 & 6 & 4 \\ 11 & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

Example Convolution Calculation

$$20 = 3 \times 0 + 6 \times 1 + 7 \times 2 + 9 \times 0$$

$$19 = 6 \times 0 + 1 \times 1 + 9 \times 2 + 1 \times 0$$

$$\vdots$$

$$11 = 7 \times 0 + 9 \times 1 + 1 \times 2 + 0 \times 0$$

$$\vdots$$

$$16 = 5 \times 0 + 2 \times 1 + 7 \times 2 + 1 \times 0$$

$x_t \in \mathbb{R}^{5\times5}$

$$\begin{bmatrix} 3 & 6 & 1 & 4 & 2 \\ 7 & 9 & 1 & 1 & 4 \\ 1 & 0 & 0 & 1 & 3 \\ 0 & 2 & 4 & 5 & 2 \\ 1 & 1 & 7 & 7 & 1 \end{bmatrix}$$

$k \in \mathbb{R}^{2\times2}$

$$\begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}$$

Convolve *x* with *k*

$h_t \in \mathbb{R}^{4\times4}$

$$\begin{bmatrix} 20 & 19 & 6 & 4 \\ 11 & 1 & 1 & 6 \\ 0 & 4 & 9 & 13 \\ 4 & 6 & 19 & 16 \end{bmatrix}$$

Stride

- Stride determines the step size of the discrete convolution across the input activation map in both the height and width dimensions
- Assume stride = 2

$$x_t \in \mathbb{R}^{5\times 5}$$

$$\begin{bmatrix} 3 & 6 & 1 & 4 & 2 \\ 7 & 9 & 1 & 1 & 4 \\ 1 & 0 & 0 & 1 & 3 \\ 0 & 2 & 4 & 5 & 2 \\ 1 & 1 & 7 & 7 & 1 \end{bmatrix}$$

$$k \in \mathbb{R}^{2\times 2}$$

$$\begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}$$

Convolve *x* with *k*

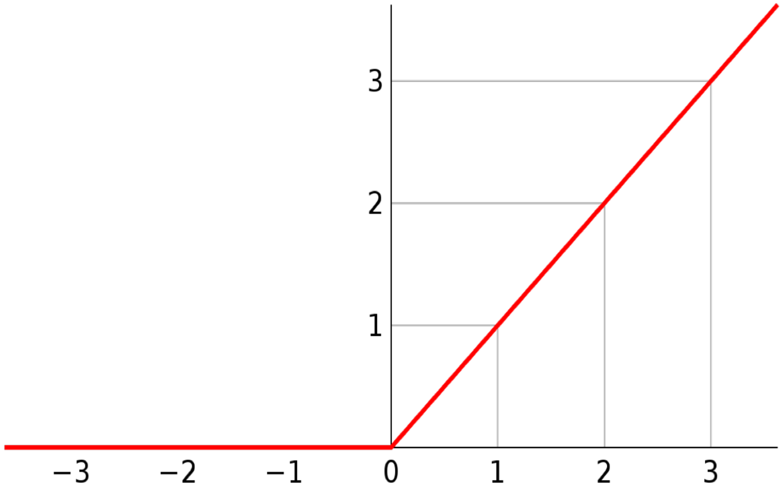
activation map

$$h_t \in \mathbb{R}^{2\times 2}$$

$$\begin{bmatrix} 20 & 6 \\ 0 & 9 \end{bmatrix}$$

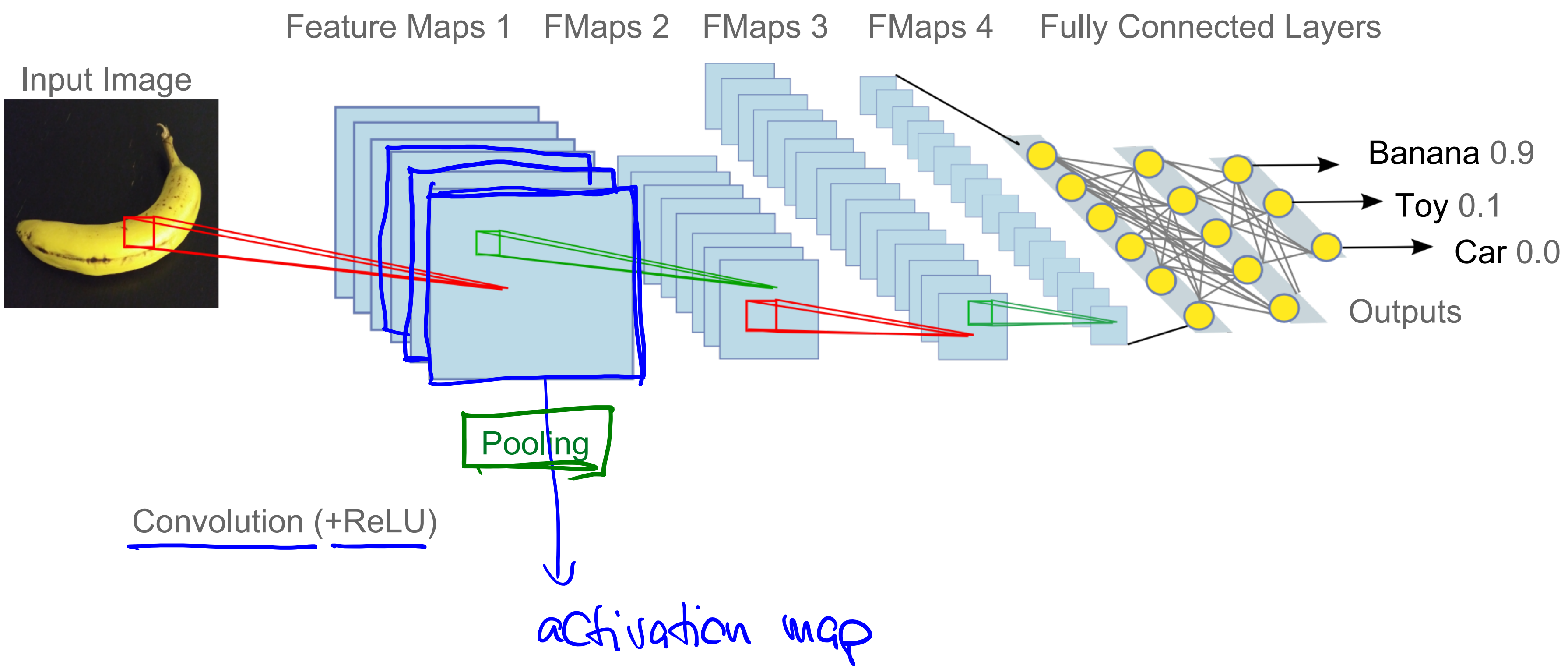
ReLu Activation

- ReLU is applied to feature maps
- Generates a response in positive part only



$$\phi(z) = \max(0, z)$$

Pooling



Feature Maps 1
FMaps 2
FMaps 3
FMaps 4
Fully Connected Layers
Input Image
Banana 0.9
Toy 0.1
Car 0.0
Outputs
Pooling
Convolution (+ReLU)
activation map

Pooling

- Pooling layer is applied with a stride in both the x and y directions by the width and height, respectively, of the pooling filter.

Pooling: Max Pooling

- Pooling provides a way to spatially down sample a set of activation maps
- Various pooling operations can be used; one of the most popular is called max pooling

activation map

$$x_t \in \mathbb{R}^{4\times 4} \quad \left[\begin{array}{cc|cc} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ \hline a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{array}\right]$$

$$maxpool(x_t) = \begin{bmatrix} max(a_{00}...a_{11}) & max(a_{02}...a_{13}) \\ max(a_{20}...a_{31}) & max(a_{22}...a_{33}) \end{bmatrix}$$

Frequent Operations in CNN