Lecture 3: Introduction to Convolutional Neural Networks

Rafael Martínez-Galarza, Pavlos Protopapas
Harvard-Smithsonian Center for Astrophysics

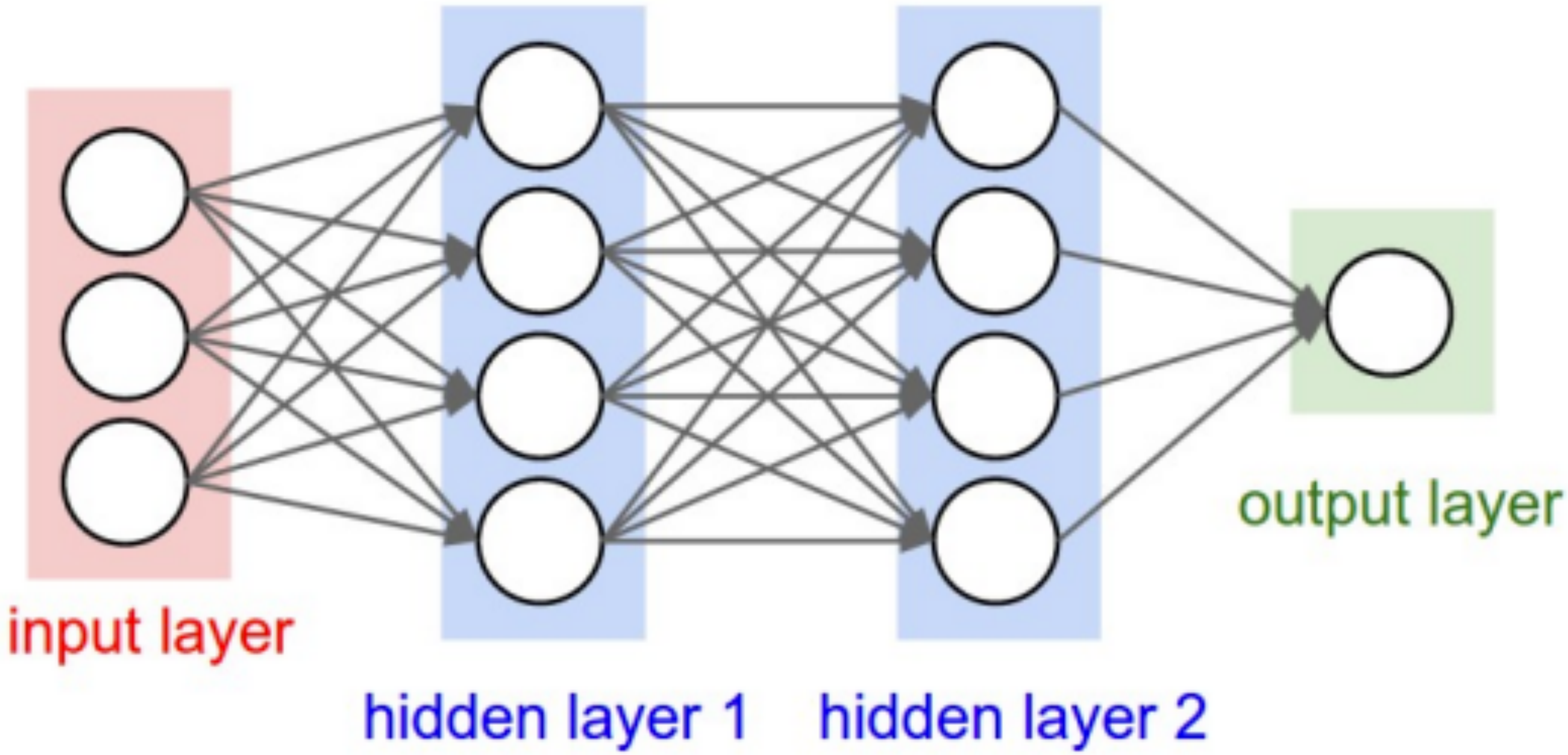
CENTER FOR ASTROPHYSICS
HARVARD & SMITHSONIAN

This lecture:

- The need for computer vision
- The architecture of a CNN
- Key features of CNNs
- ReLU activation function

Convolutional Neural Networks

A fully connected neural net



Each neuron in hidden layer 1 is connected to all neurons in the previous (input) layer and to all neurons in the following (hidden2) layer.

Very good for regression, for learning complex functions, but…

Finding structure in images

Nearby pixels are more strongly related than distant ones.

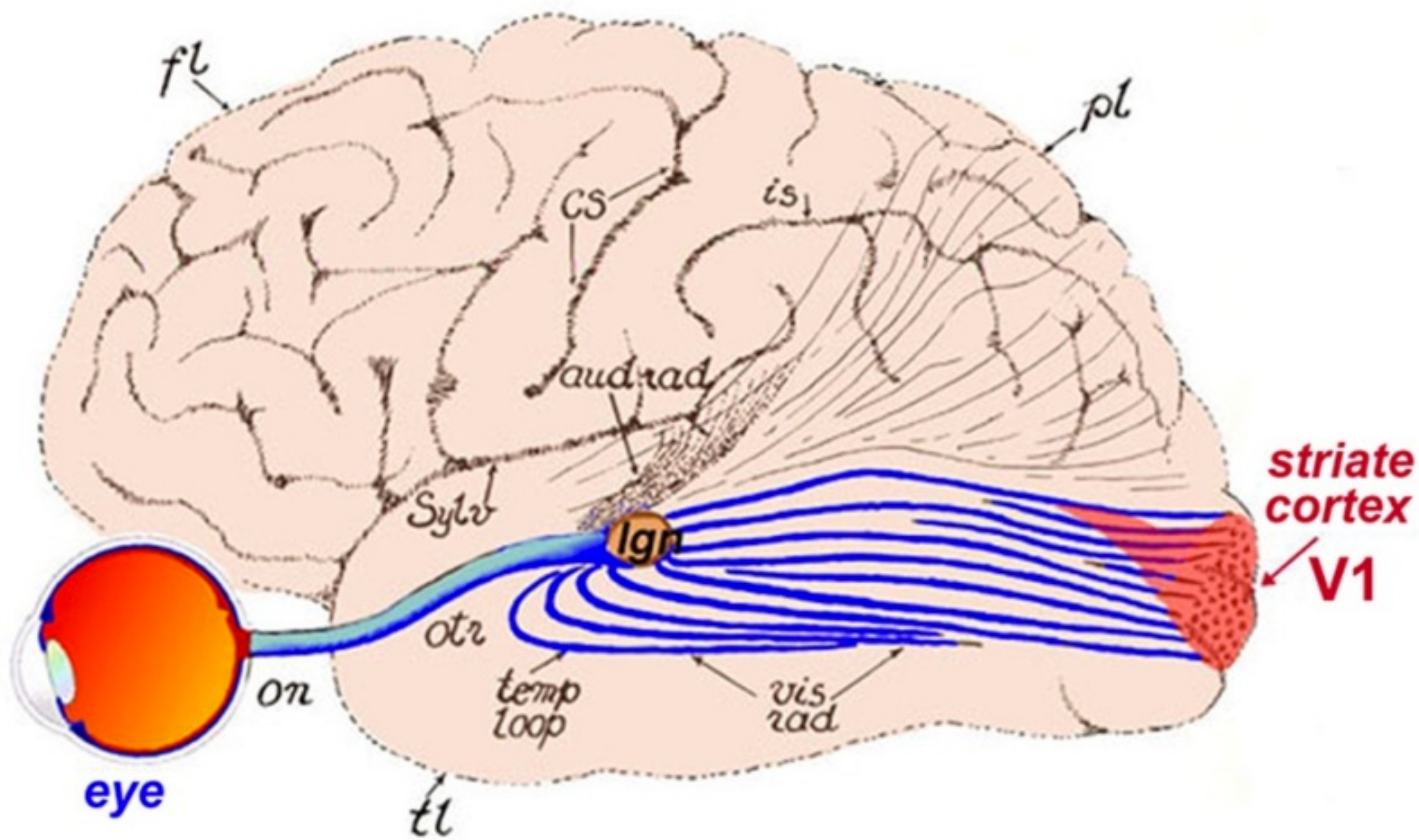
Objects are built up out of smaller parts.

A fully connected network would be very inefficient at learning patterns here.

Instead, we can learn from nature.

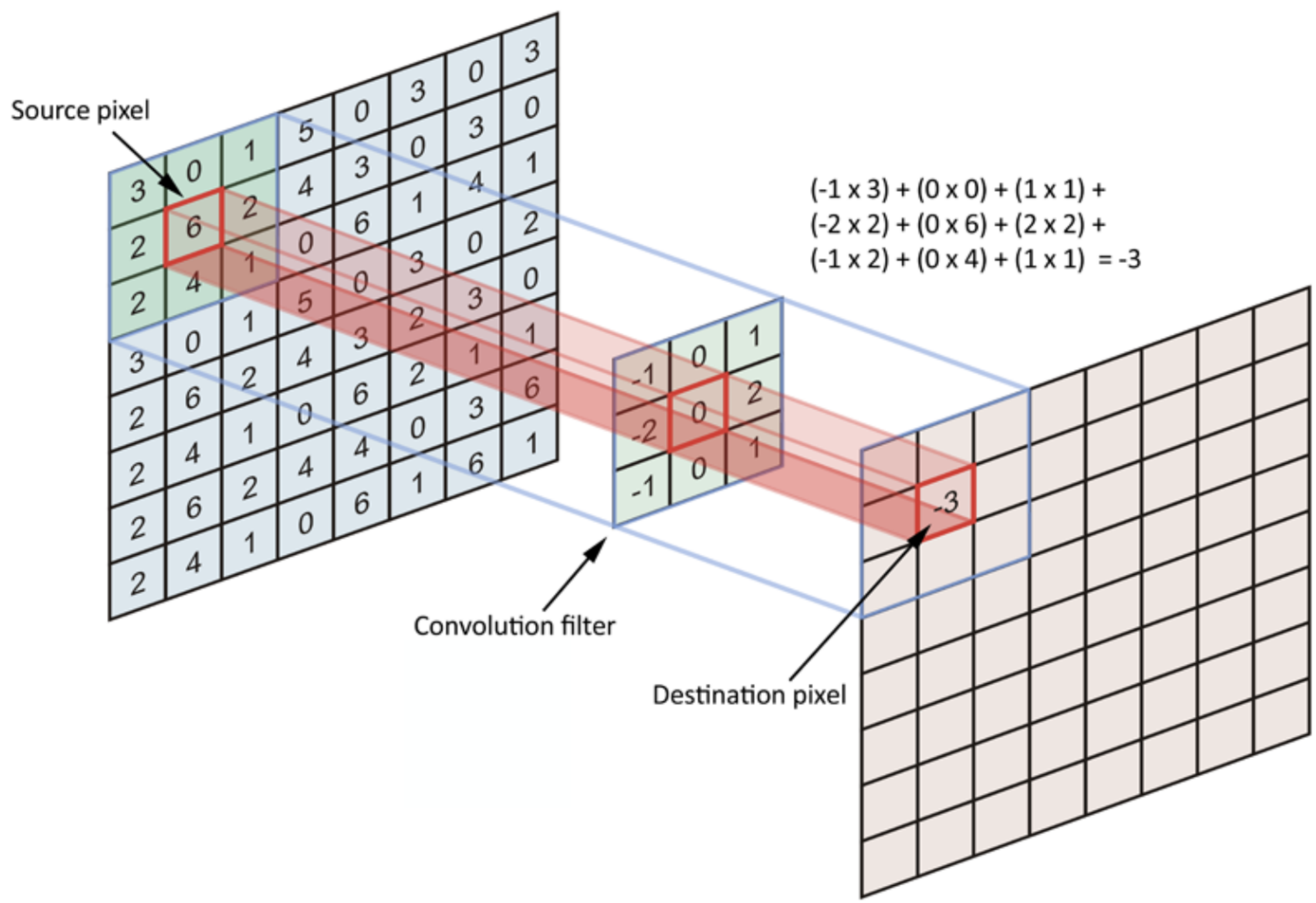
Orientation invariance

Vision in nature



Cell in the visual cortex are not sensitive to the entire visual field
First neurons connected to the retina specialize in detecting edited
Neurons in further layers identify other types of patterns

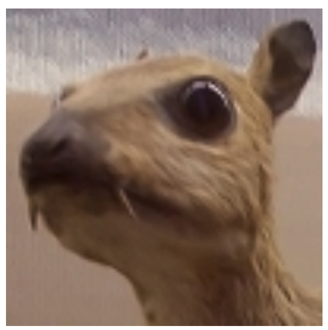
“Convolution” operation



Feature extraction

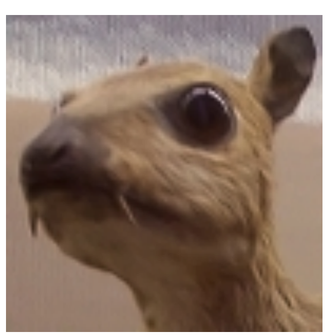
Kernel

Edge detection

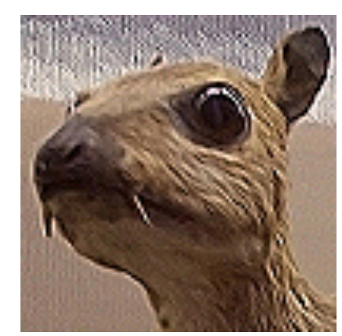


$$* \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$

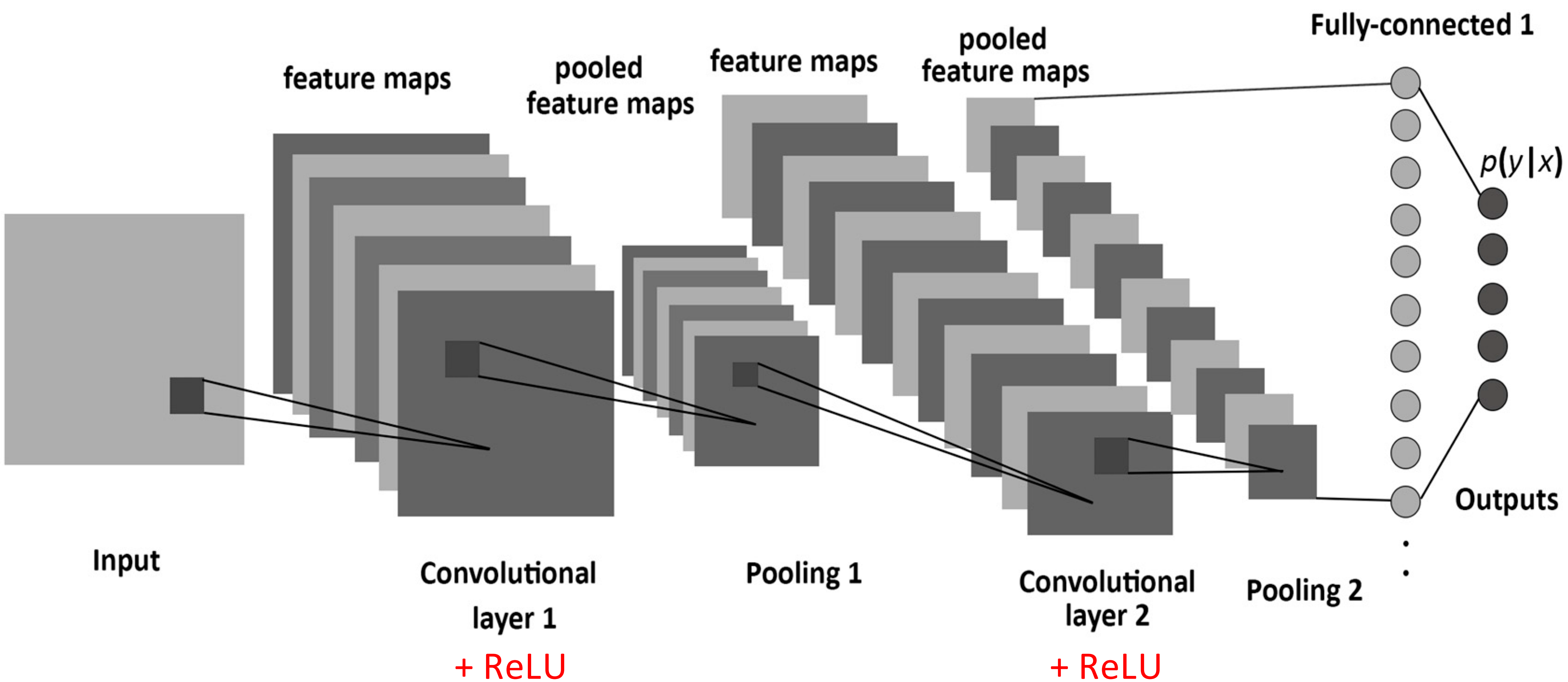
Sharpen



$$* \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} =$$



A convolutional network



Some key features

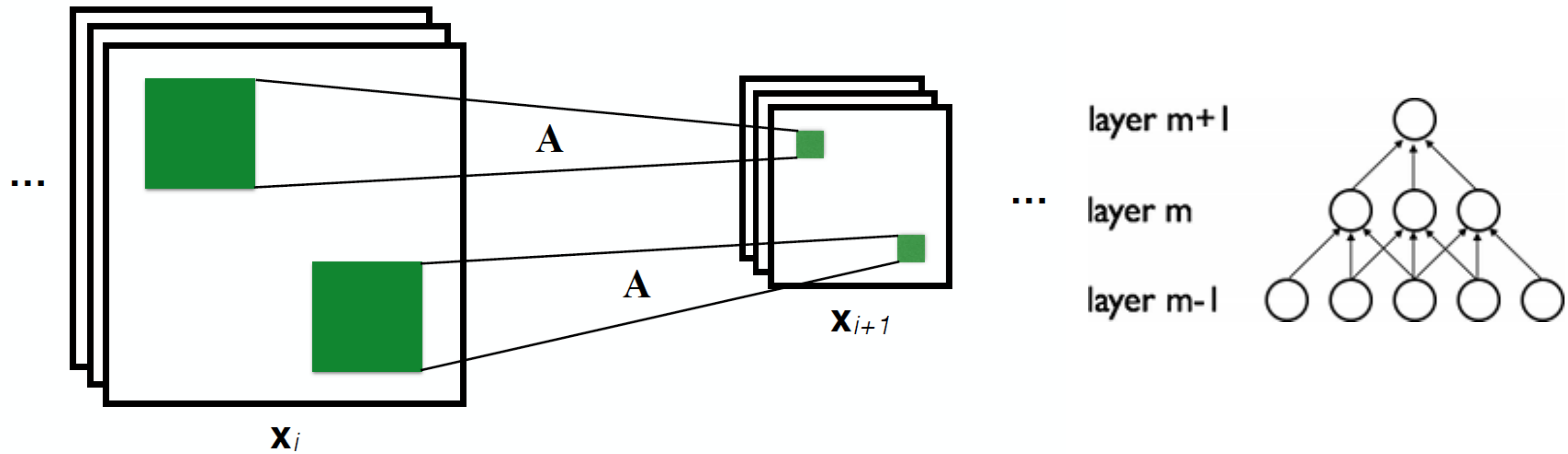
- **Sparse interactions**: neurons in a given layer are sensitive only to a sub-field of nearby neurons in the previous layer.

- **Parameter sharing**: same kernel applied to different regions of the input image. Learn kernel only.

- **Pooling**: reduce spatial dimensions of input volume, by averaging over small square windows of the field.

- **Padding**: add additional layer to the border of the image to avoid loss of information.

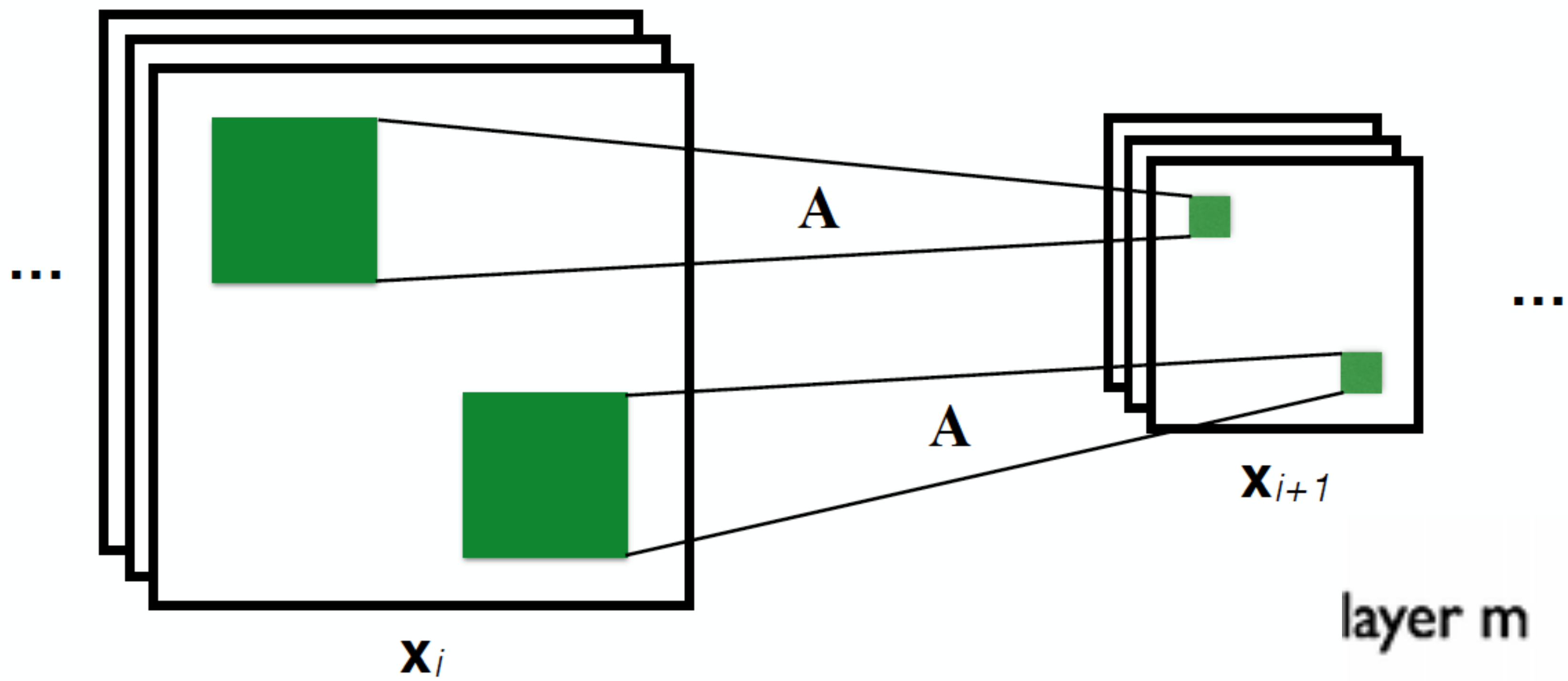
Sparse interactions



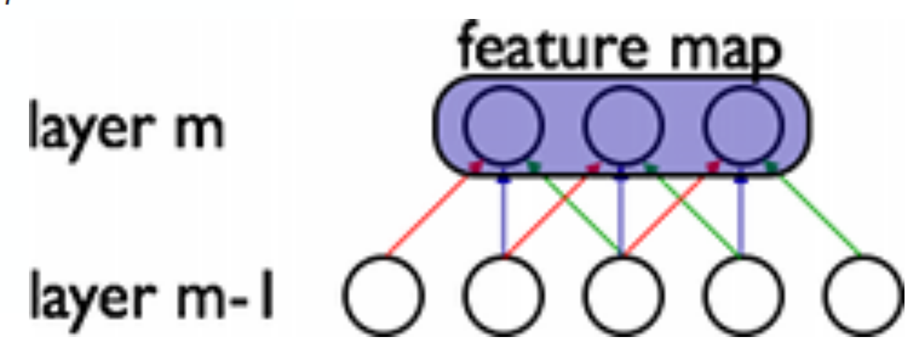
$$\mathbf{x}_{i+1} = \sigma_i(\mathbf{W}_i \mathbf{x}_i + \mathbf{b}_i)$$

- Exploits local spatial correlations.
- Receptive field is limited, but increases with depth.
- Reduces memory and run time, as the kerne size is a couple of order of magnitudes smaller than the input image

Parameter sharing

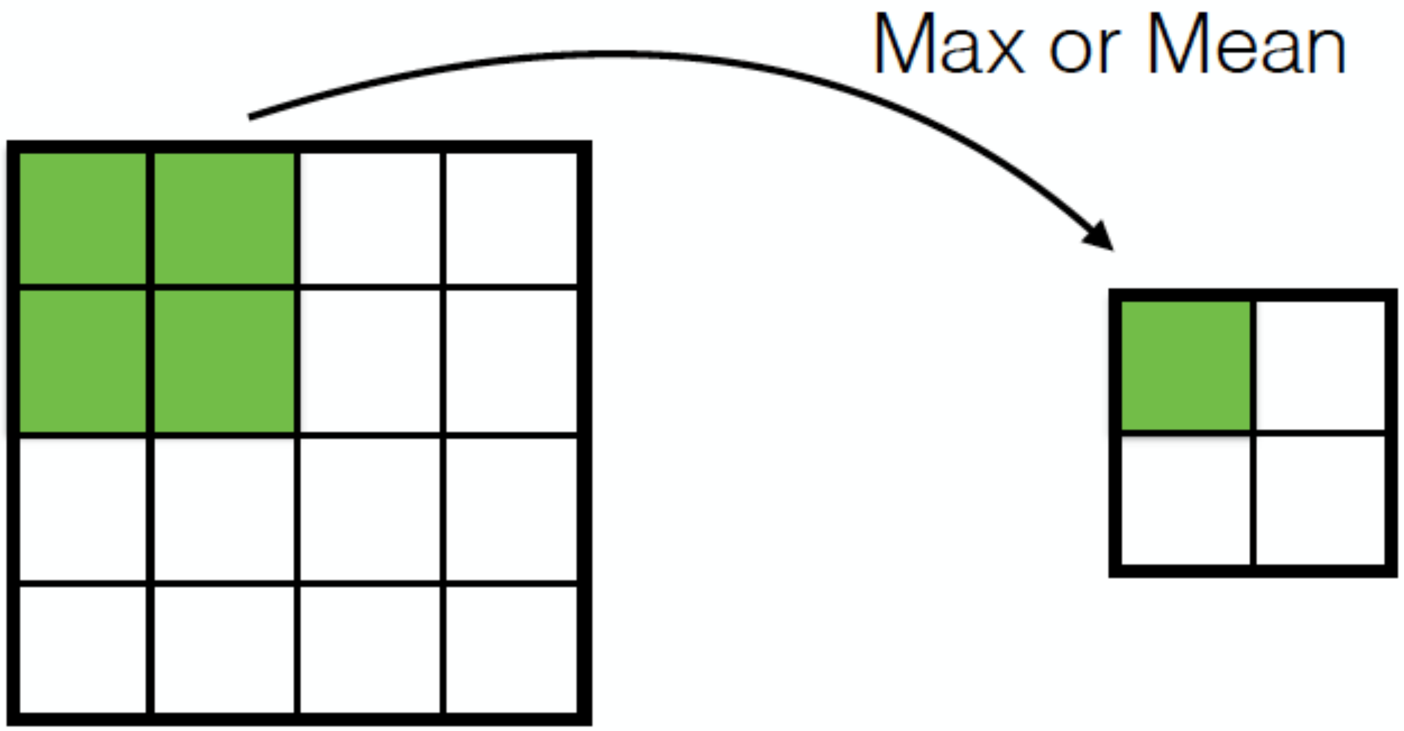


$$\mathbf{x}_{i+1} = \sigma_i(\mathbf{W}_i \mathbf{x}_i + \mathbf{b}_i)$$



- Sharing weights helps in translational invariance and also reduces computation time, as there are less free parameters

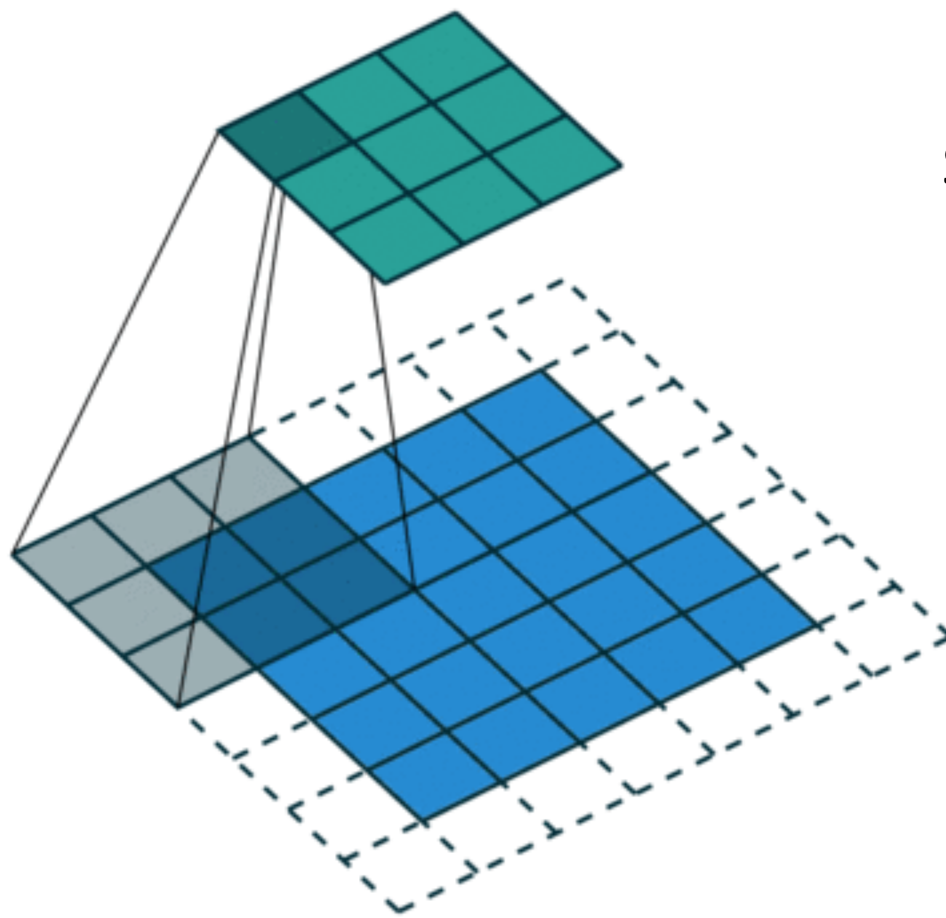
Pooling



- Reduced input size without loosing capability to generalize.
- Invariant to small local transitions.

Padding

- Shift the kernel not by 1 pixel, but by k pixels (stride)



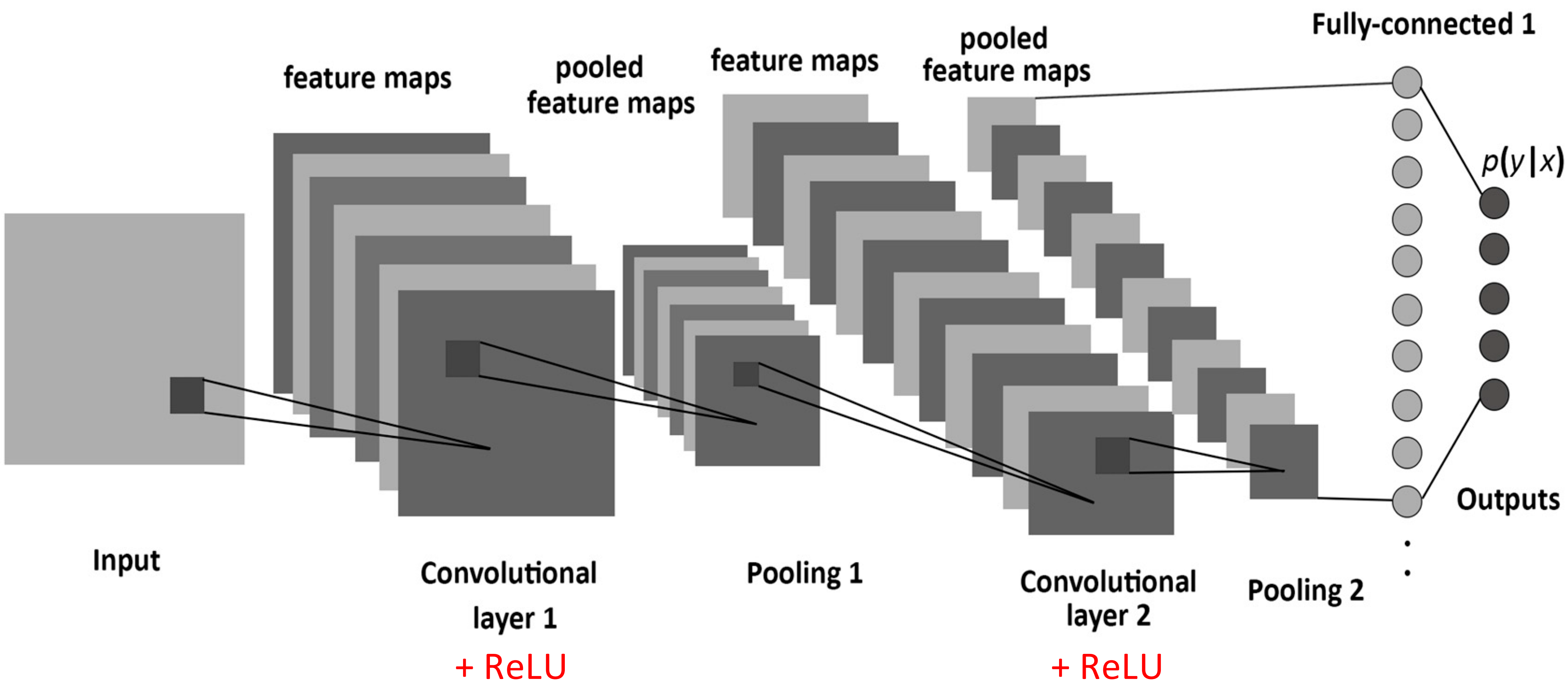
Stride = 2

Sees entire image, but low computational and storage costs

Prevents loss of information near the edges

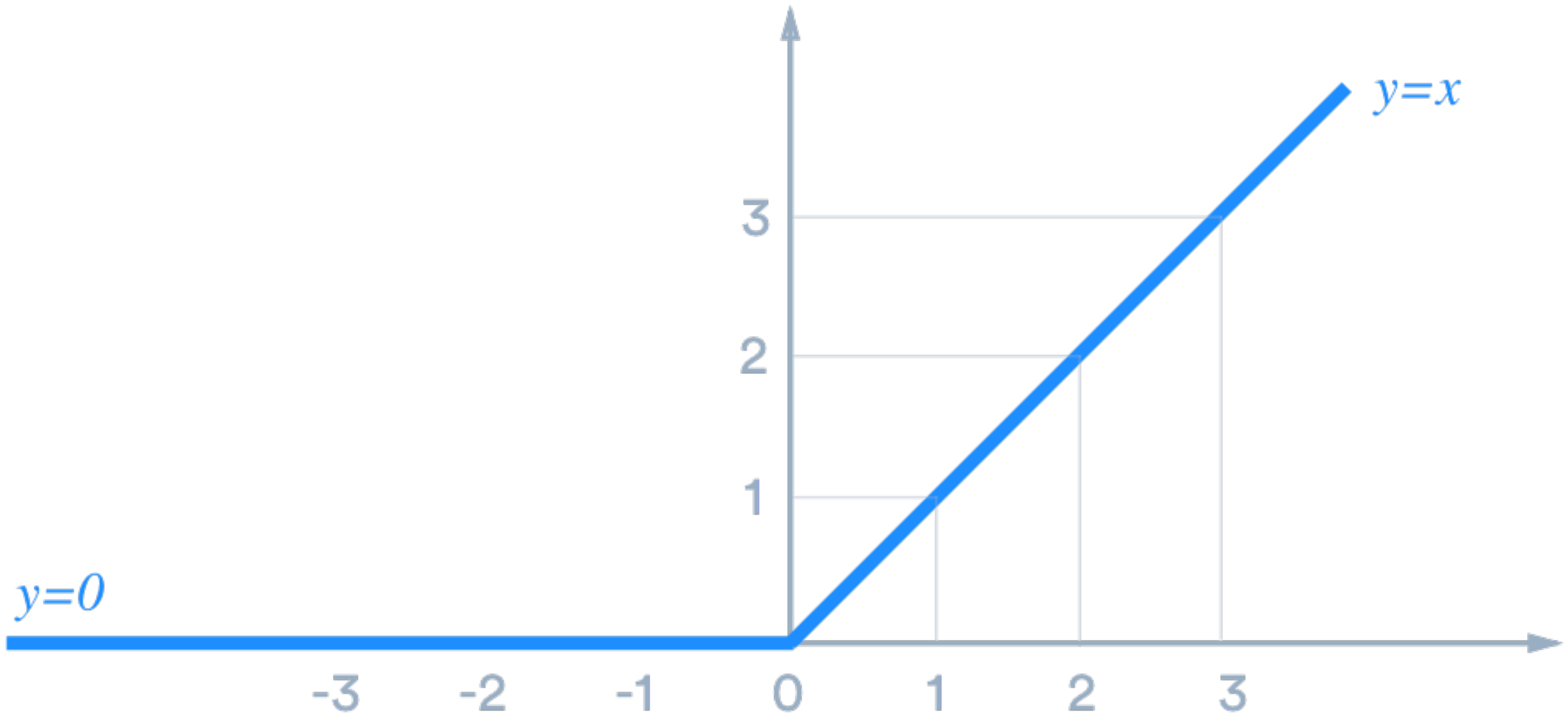
ReLU

A common activation function: ReLU



- Activation functions don't always need to be interpreted as probabilities. They can be anything that produces new representations when applied to the input.
- ReLU: Rectified Linear Unit is a very common activation function used in NNs.
- Defined as: y = max(0,x)

A common activation function: ReLU



- Easy to compute: no complicated math.
- Converges fast (gradient always well defined, except for one point).
- Sparsely activated