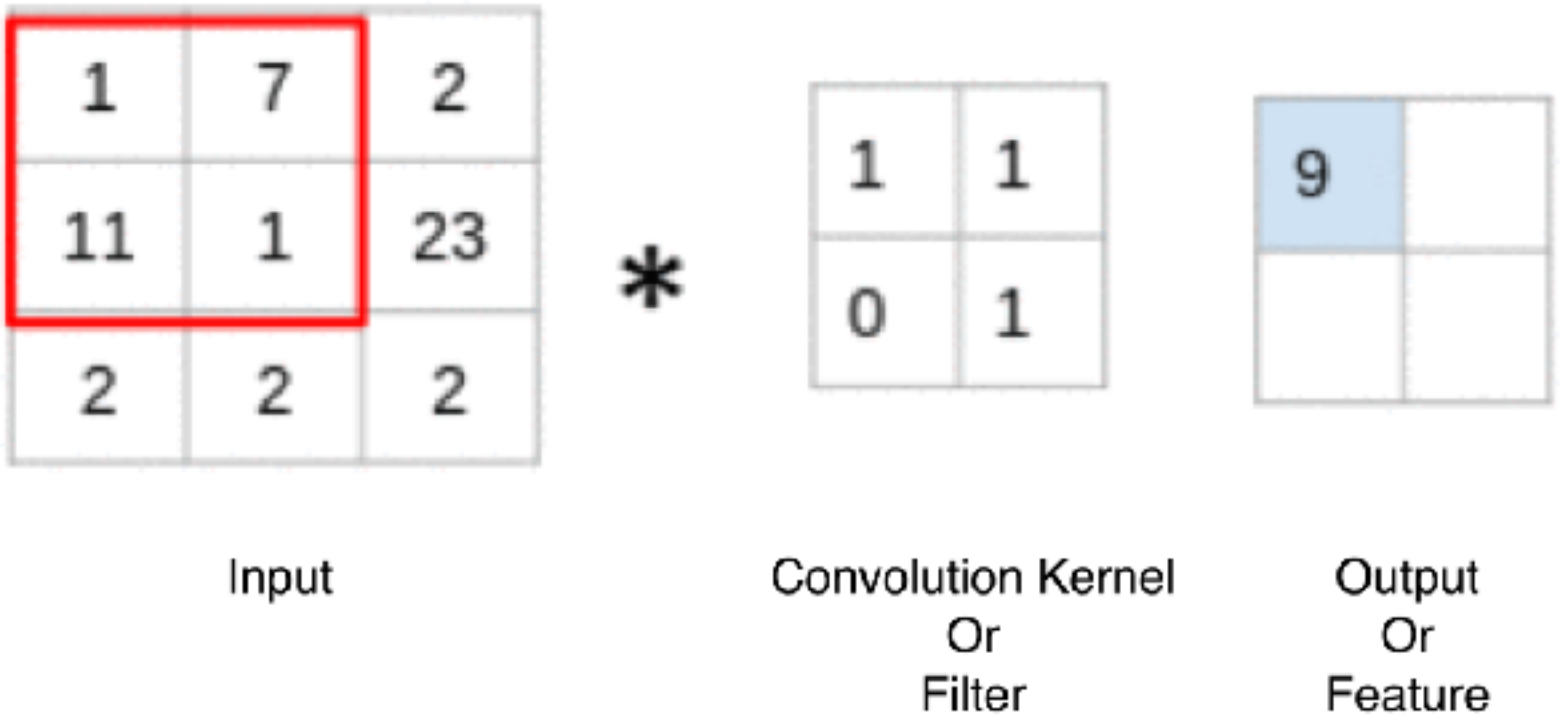
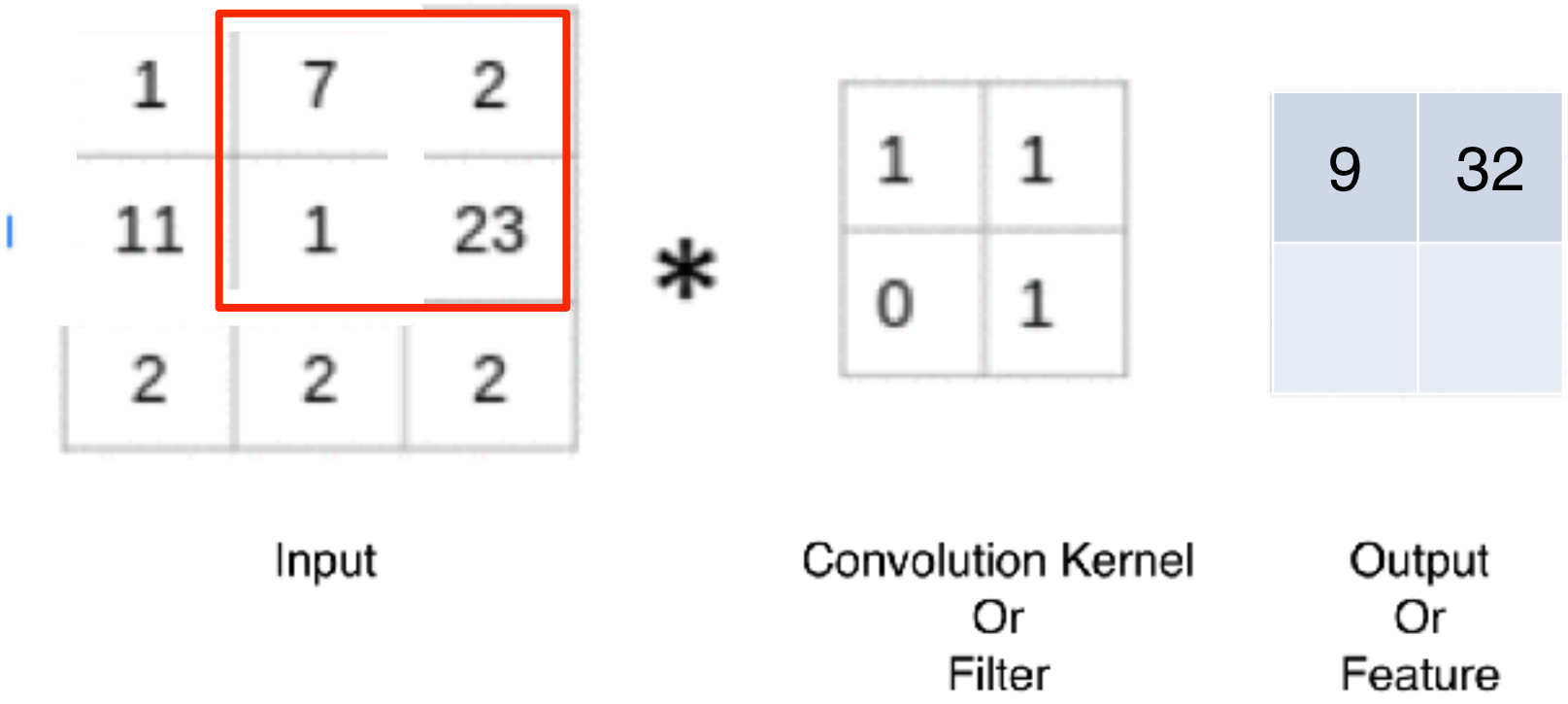
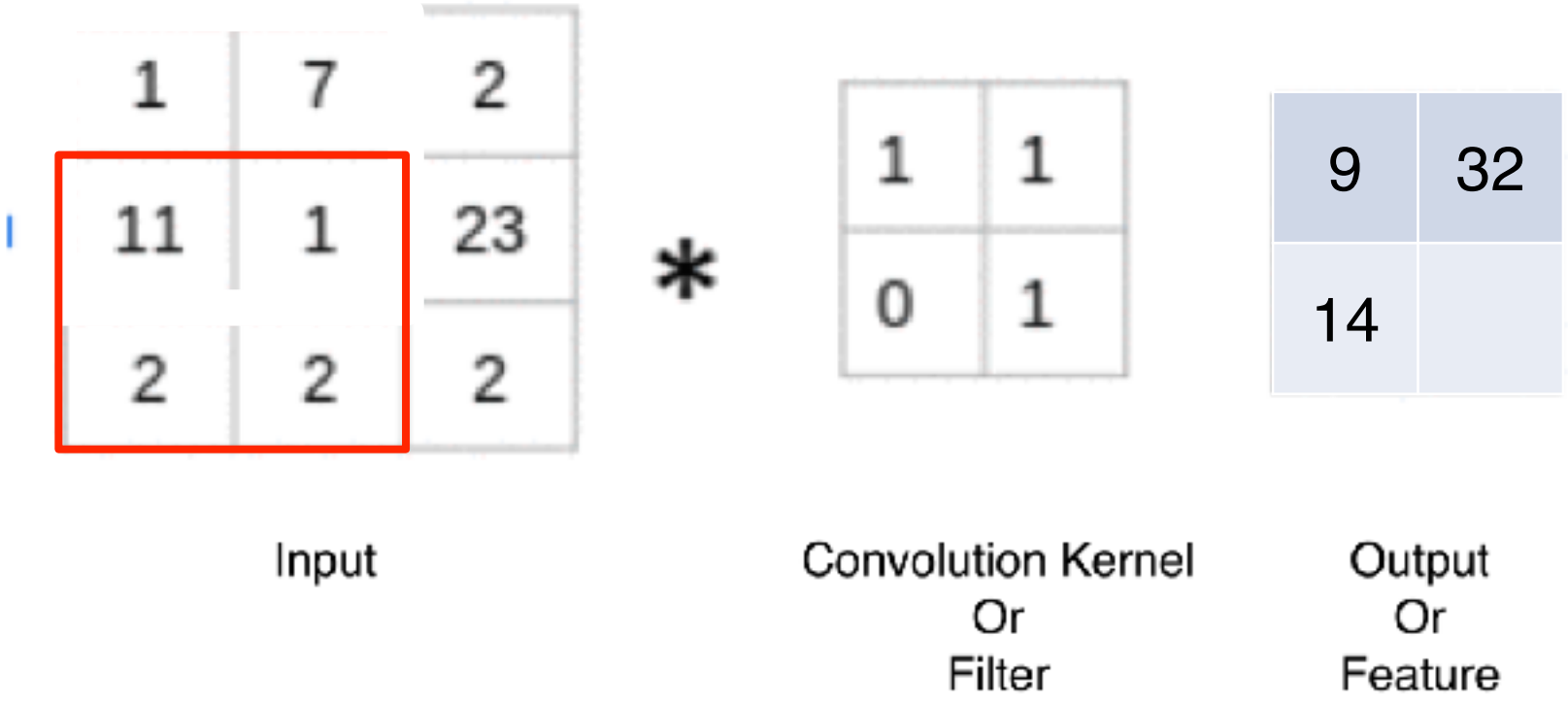
Recap: Convolution



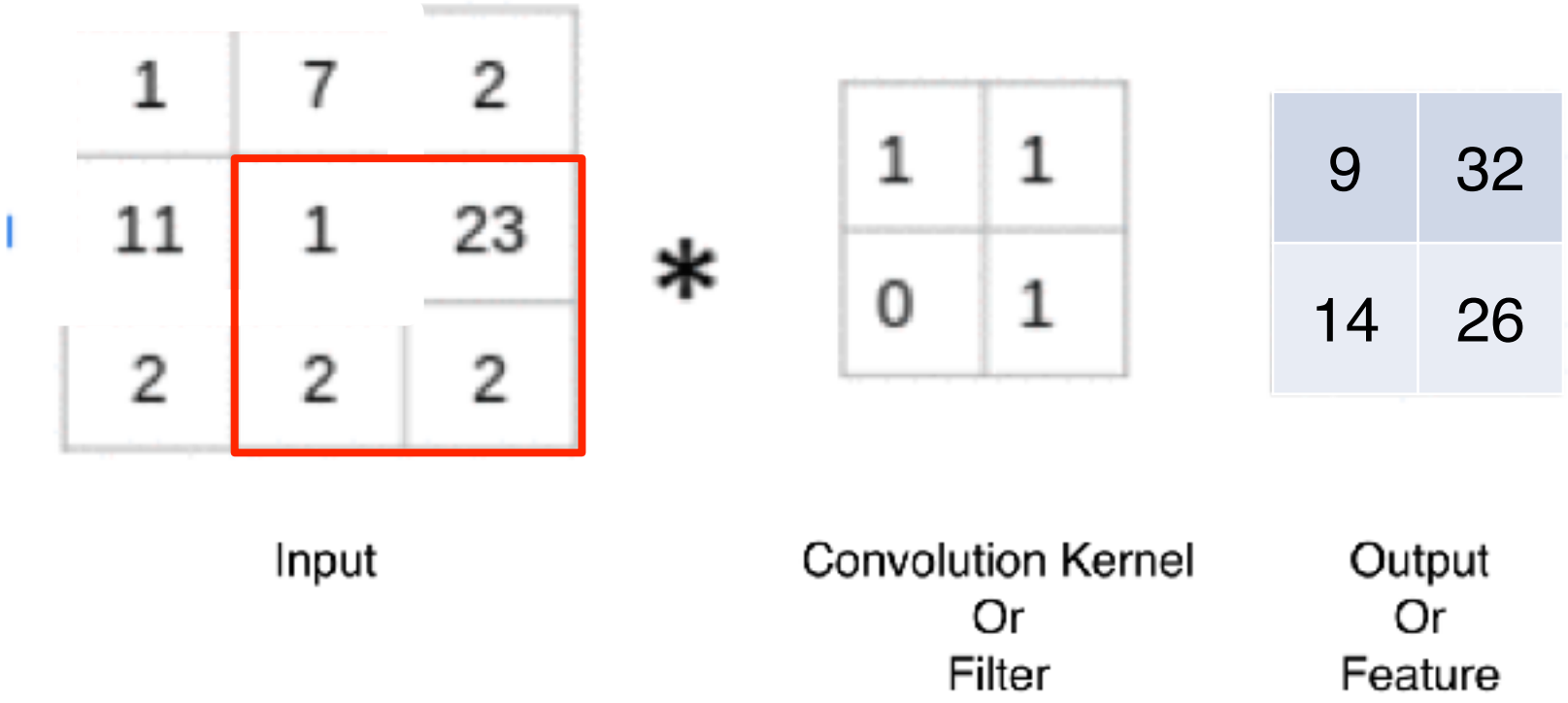
Recap: Convolution



Recap: Convolution



Recap: Convolution

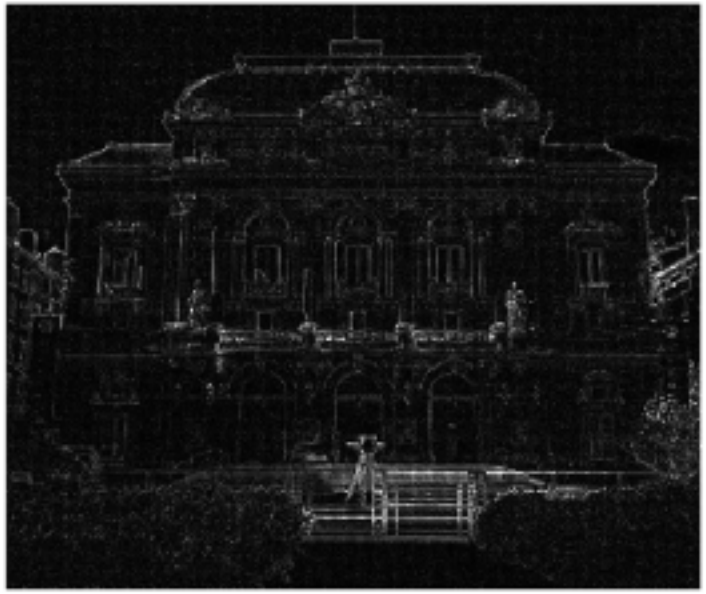


Laplacian Filter

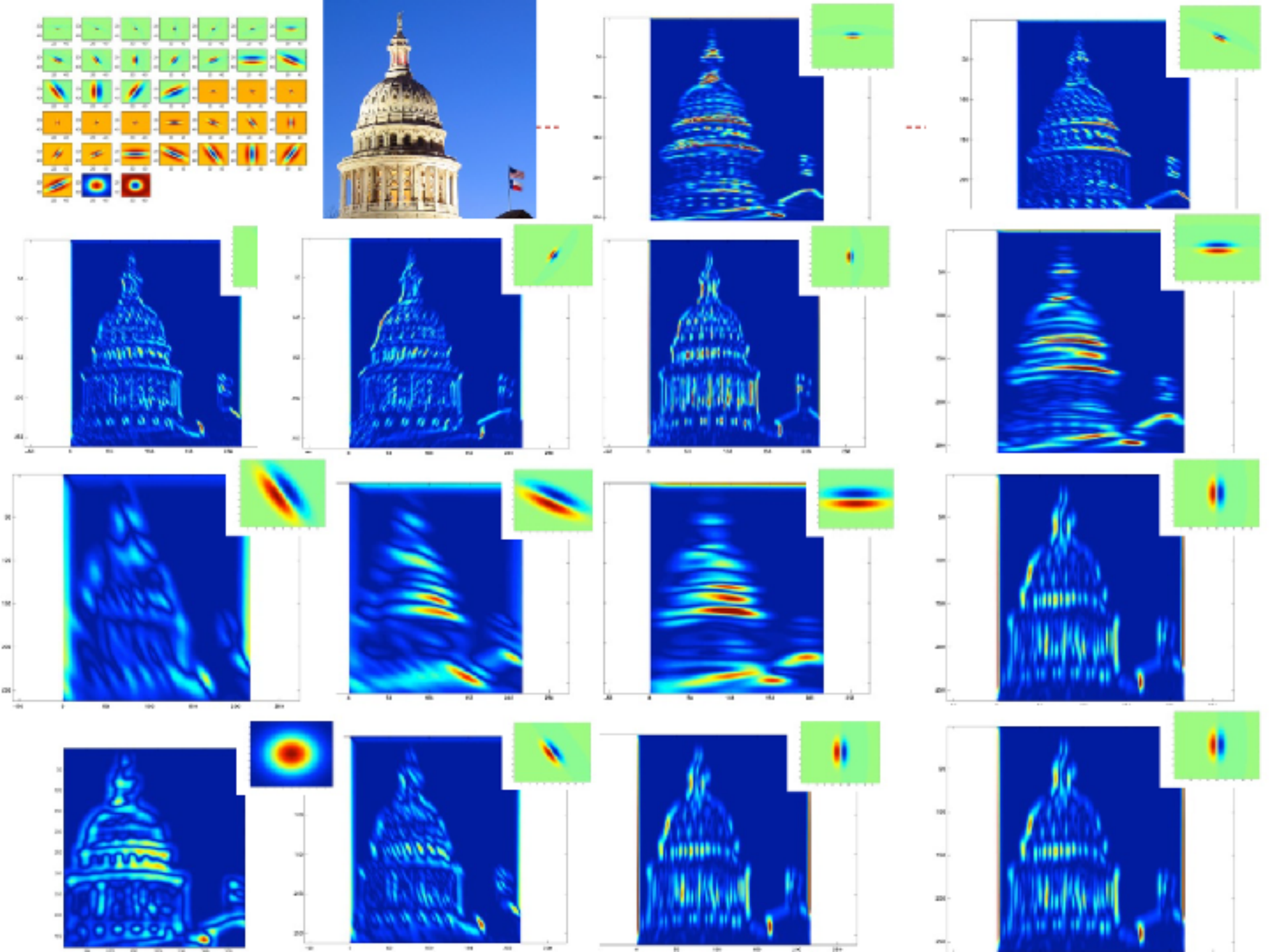
Input

$$* \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} =$$

Convolution Kernel
Or
Filter



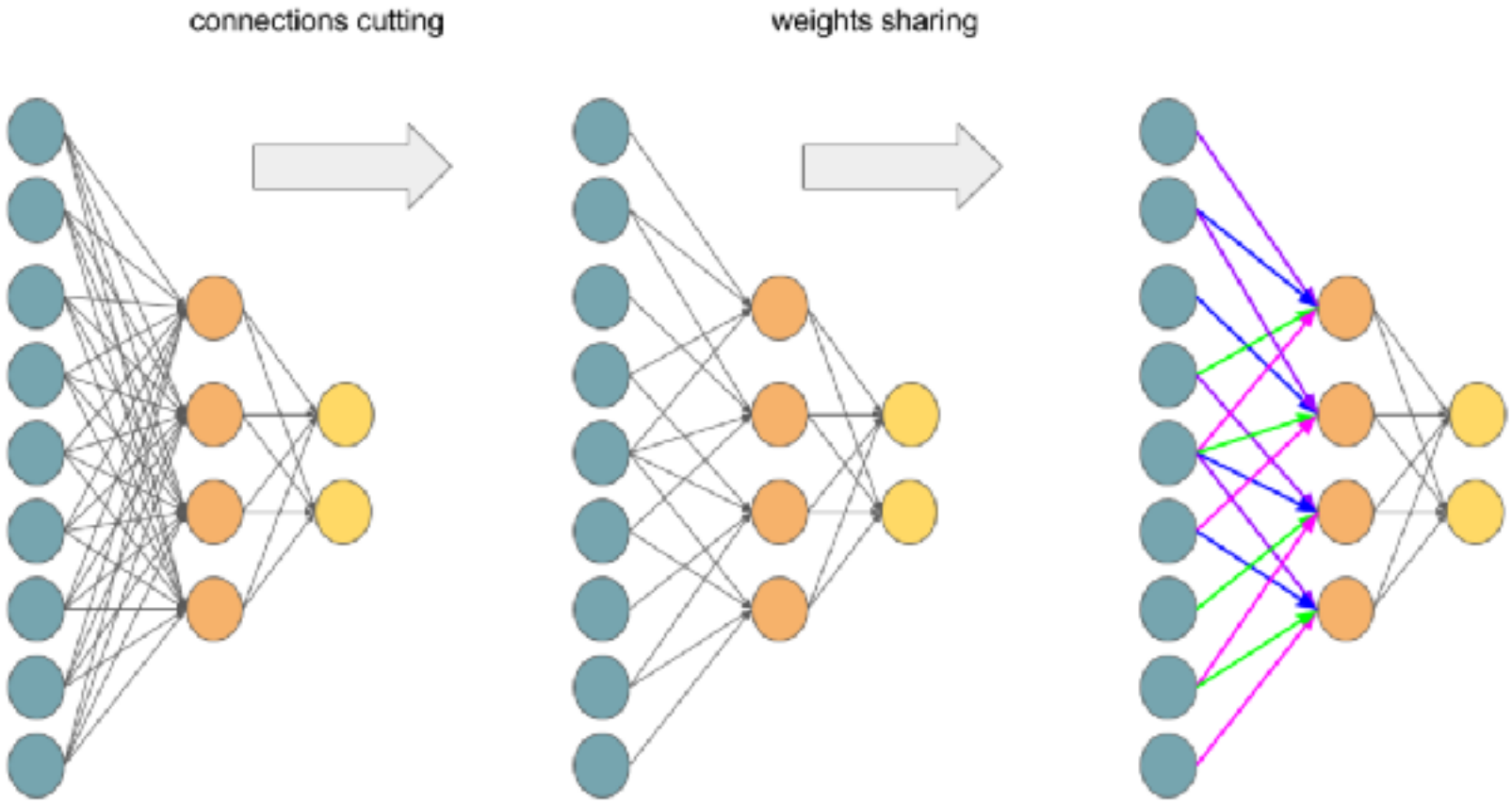
Output
Or
Feature



Any Questions?

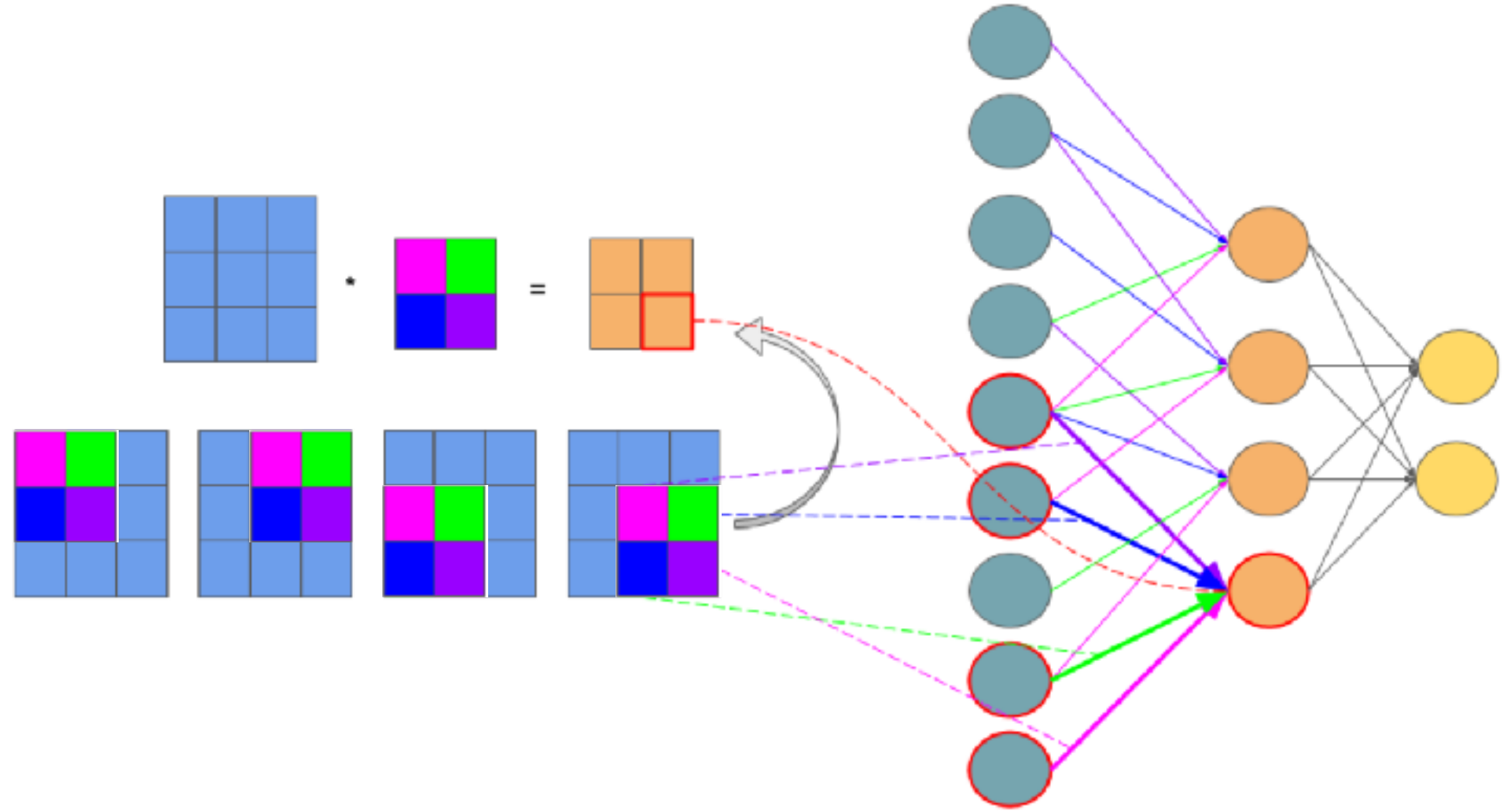
Transforming Multilayer Perceptron to Convolutional Neural Network

Transforming Multilayer Perceptron to Convolutional Neural Network



https://grzegorzgwardys.wordpress.com/2016/04/22/8/

Feedforward in CNN is identical with convolution operation



https://grzegorzgwardys.wordpress.com/2016/04/22/8/

Forward and Backward Propagation using Convolution operation

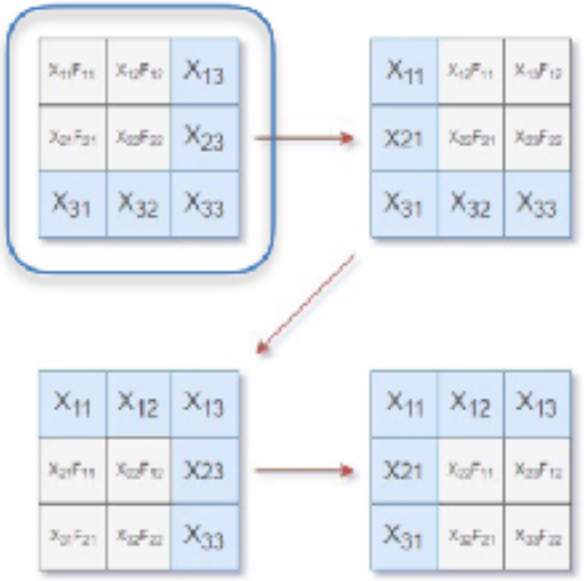
- Forward Pass
  - Ignoring 180º rotation

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

  - Note: F instead of w

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

- It can be visualized as:



https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Forward and Backward Propagation using Convolution operation

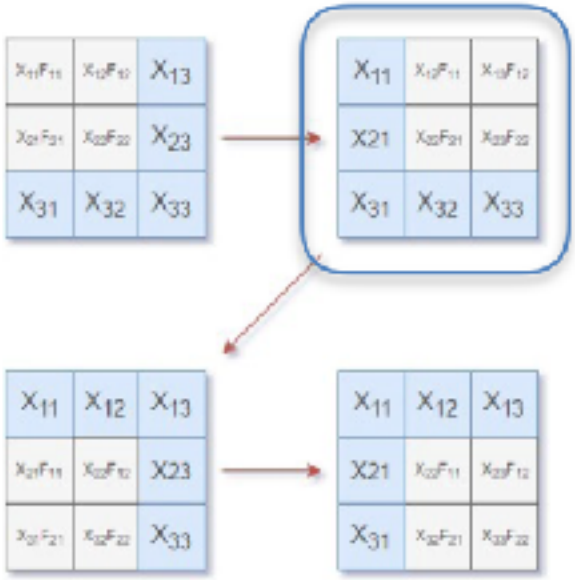
- Forward Pass
  - Ignoring 180º rotation

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

  - Note: F instead of w

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$
$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

- It can be visualized as:



https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Forward and Backward Propagation using Convolution operation

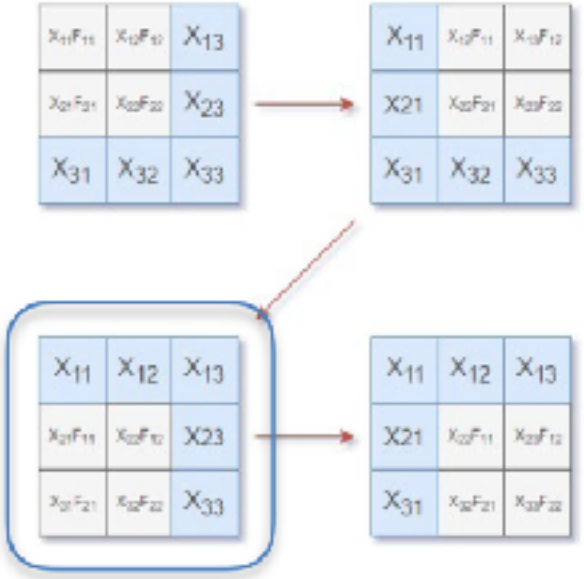
- Forward Pass
  - Ignoring 180° rotation

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

  - Note: F instead of w

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$
$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$
$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

- It can be visualized as:



https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Forward and Backward Propagation using Convolution operation

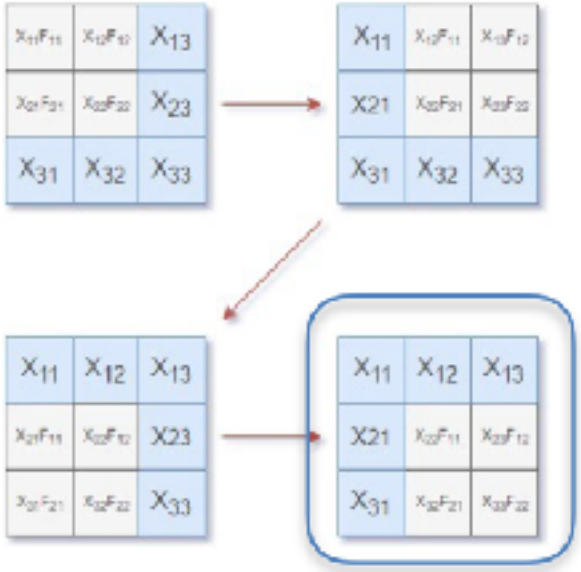
- Forward Pass
  - Ignoring 180º rotation

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

  - Note: F instead of w

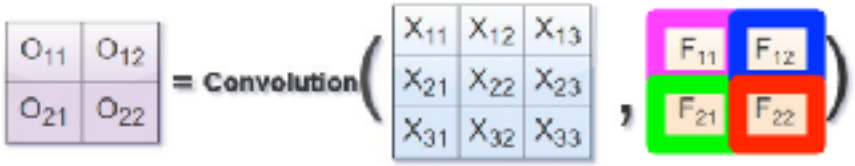
$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$
$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$
$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$
$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

- It can be visualized as:

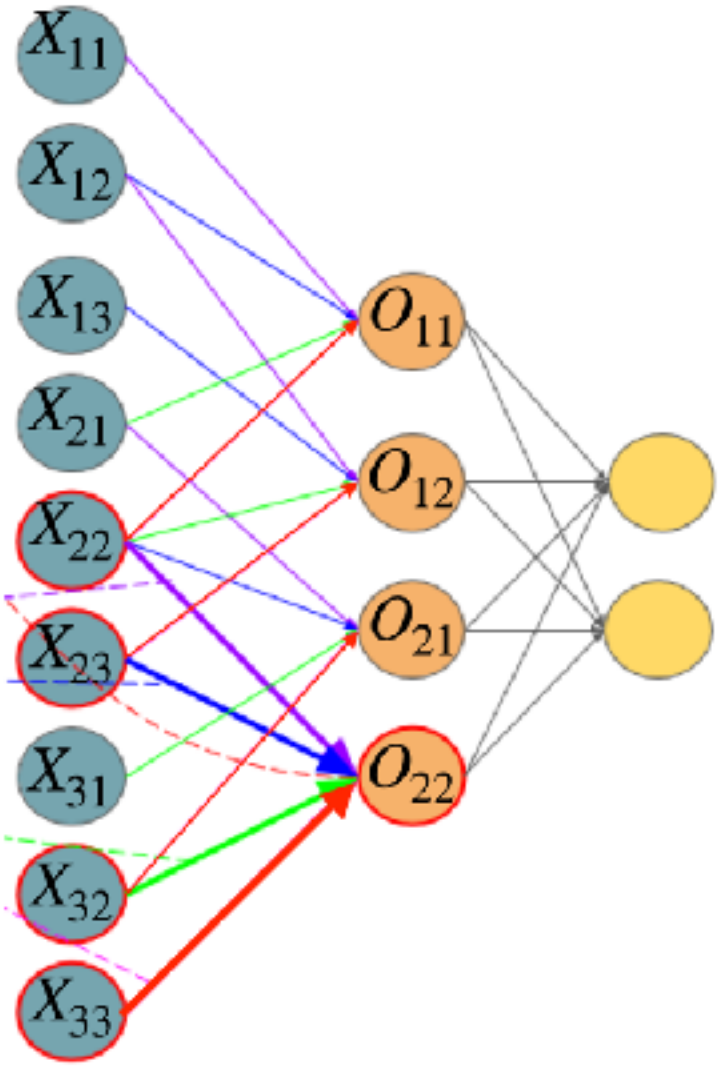


https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Feedforward in CNN is identical with convolution operation



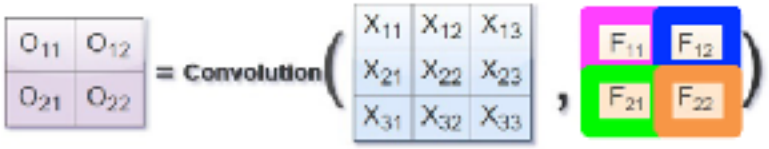
$$O_{11} = F_{11}X_{11}+F_{12}X_{12}+F_{21}X_{21}+F_{22}X_{22}$$
$$O_{12} = F_{11}X_{12}+F_{12}X_{13}+F_{21}X_{22}+F_{22}X_{23}$$
$$O_{21} = F_{11}X_{21}+F_{12}X_{22}+F_{21}X_{31}+F_{22}X_{32}$$
$$O_{22} = F_{11}X_{22}+F_{12}X_{23}+F_{21}X_{32}+F_{22}X_{33}$$



https://grzegorzgwardys.wordpress.com/2016/04/22/8/

Forward and Backward Propagation Using Convolution Operation

- Gradient of error ‘E’ with respect to the filer ‘F’



$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

Forward and Backward Propagation Using Convolution Operation

- Gradient of error ‘E’ with respect to the filer ‘F’

$$E = E_{O_{11}} + E_{O_{12}} + E_{O_{21}} + E_{O_{22}}$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{11}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{11}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{11}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{11}}$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}}X_{11} + \frac{\partial E}{\partial O_{12}}X_{12} + \frac{\partial E}{\partial O_{21}}X_{21} + \frac{\partial E}{\partial O_{22}}X_{22}$$

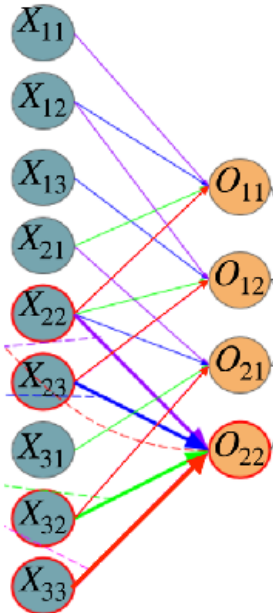
$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$



Forward and Backward Propagation Using Convolution Operation

- Gradient of error ‘E’ with respect to the filer ‘F’

$$E = E_{O_{11}} + E_{O_{12}} + E_{O_{21}} + E_{O_{22}}$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{11}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{11}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{11}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{11}}$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}}X_{11} + \frac{\partial E}{\partial O_{12}}X_{12} + \frac{\partial E}{\partial O_{21}}X_{21} + \frac{\partial E}{\partial O_{22}}X_{22}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{12}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{12}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{12}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{12}}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}}X_{12} + \frac{\partial E}{\partial O_{12}}X_{13} + \frac{\partial E}{\partial O_{21}}X_{22} + \frac{\partial E}{\partial O_{22}}X_{23}$$

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

Forward and Backward Propagation Using Convolution Operation

- Gradient of error ‘E’ with respect to the filer ‘F’

$$E = E_{O_{11}} + E_{O_{12}} + E_{O_{21}} + E_{O_{22}}$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{11}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{11}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{11}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{11}}$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}}X_{11} + \frac{\partial E}{\partial O_{12}}X_{12} + \frac{\partial E}{\partial O_{21}}X_{21} + \frac{\partial E}{\partial O_{22}}X_{22}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{12}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{12}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{12}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{12}}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}}X_{12} + \frac{\partial E}{\partial O_{12}}X_{13} + \frac{\partial E}{\partial O_{21}}X_{22} + \frac{\partial E}{\partial O_{22}}X_{23}$$

$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{21}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{21}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{21}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{21}}$$

$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}}X_{21} + \frac{\partial E}{\partial O_{12}}X_{22} + \frac{\partial E}{\partial O_{21}}X_{31} + \frac{\partial E}{\partial O_{22}}X_{32}$$

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

Forward and Backward Propagation Using Convolution Operation

- Gradient of error ‘E’ with respect to the filer ‘F’

$$E = E_{O_{11}} + E_{O_{12}} + E_{O_{21}} + E_{O_{22}}$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{11}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{11}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{11}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{11}}$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}}X_{11} + \frac{\partial E}{\partial O_{12}}X_{12} + \frac{\partial E}{\partial O_{21}}X_{21} + \frac{\partial E}{\partial O_{22}}X_{22}$$

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{12}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{12}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{12}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{12}}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}}X_{12} + \frac{\partial E}{\partial O_{12}}X_{13} + \frac{\partial E}{\partial O_{21}}X_{22} + \frac{\partial E}{\partial O_{22}}X_{23}$$

$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{21}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{21}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{21}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{21}}$$

$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}}X_{21} + \frac{\partial E}{\partial O_{12}}X_{22} + \frac{\partial E}{\partial O_{21}}X_{31} + \frac{\partial E}{\partial O_{22}}X_{32}$$

$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

$$\begin{bmatrix} \partial E/\partial F_{11} & \partial E/\partial F_{12} \\ \partial E/\partial F_{21} & \partial E/\partial F_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} \partial E/\partial O_{11} & \partial E/\partial O_{12} \\ \partial E/\partial O_{21} & \partial E/\partial O_{22} \end{bmatrix} \right)$$

$$\frac{\partial E}{\partial F_{22}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{22}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{22}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{22}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{22}}$$

$$\frac{\partial E}{\partial F_{22}} = \frac{\partial E}{\partial O_{11}}X_{22} + \frac{\partial E}{\partial O_{12}}X_{23} + \frac{\partial E}{\partial O_{21}}X_{32} + \frac{\partial E}{\partial O_{22}}X_{33}$$

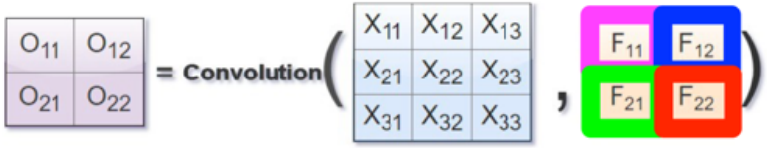
$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

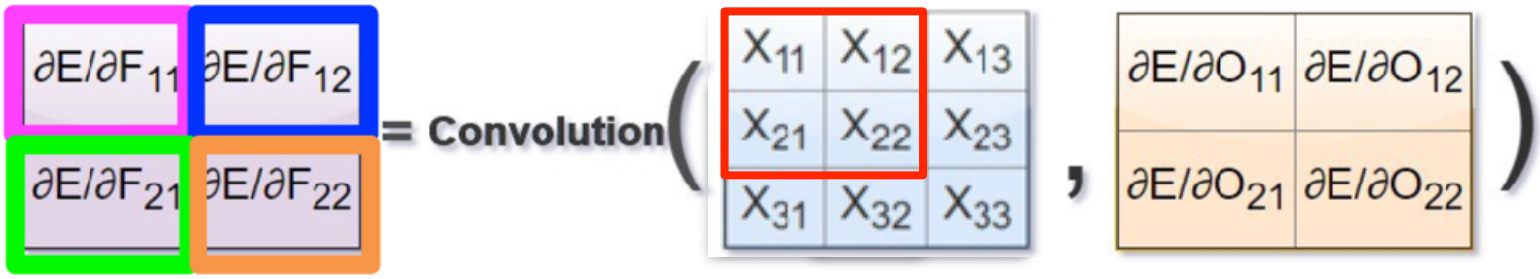
Forward and Backward Propagation using Convolution operation

- Gradients of error ‘E’ with respect to the filter ‘F’

$$E = E_{O_{11}} + E_{O_{12}} + E_{O_{21}} + E_{O_{22}}$$



$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}} X_{11} + \frac{\partial E}{\partial O_{12}} X_{12} + \frac{\partial E}{\partial O_{21}} X_{21} + \frac{\partial E}{\partial O_{22}} X_{22}$$



https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Forward and Backward Propagation using Convolution operation

- Gradients of error ‘E’ with respect to the filter ‘F’

$$E = E_{O_{11}} + E_{O_{12}} + E_{O_{21}} + E_{O_{22}}$$

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}} X_{11} + \frac{\partial E}{\partial O_{12}} X_{12} + \frac{\partial E}{\partial O_{21}} X_{21} + \frac{\partial E}{\partial O_{22}} X_{22}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}} X_{12} + \frac{\partial E}{\partial O_{12}} X_{13} + \frac{\partial E}{\partial O_{21}} X_{22} + \frac{\partial E}{\partial O_{22}} X_{23}$$

$$\begin{bmatrix} \partial E/\partial F_{11} & \partial E/\partial F_{12} \\ \partial E/\partial F_{21} & \partial E/\partial F_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} \partial E/\partial O_{11} & \partial E/\partial O_{12} \\ \partial E/\partial O_{21} & \partial E/\partial O_{22} \end{bmatrix} \right)$$

https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Forward and Backward Propagation using Convolution operation

- Gradients of error ‘E’ with respect to the filter ‘F’

$$E = E_{O_{11}} + E_{O_{12}} + E_{O_{21}} + E_{O_{22}}$$

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}} X_{11} + \frac{\partial E}{\partial O_{12}} X_{12} + \frac{\partial E}{\partial O_{21}} X_{21} + \frac{\partial E}{\partial O_{22}} X_{22}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}} X_{12} + \frac{\partial E}{\partial O_{12}} X_{13} + \frac{\partial E}{\partial O_{21}} X_{22} + \frac{\partial E}{\partial O_{22}} X_{23}$$

$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}} X_{21} + \frac{\partial E}{\partial O_{12}} X_{22} + \frac{\partial E}{\partial O_{21}} X_{31} + \frac{\partial E}{\partial O_{22}} X_{32}$$

$$\begin{bmatrix} \partial E/\partial F_{11} & \partial E/\partial F_{12} \\ \partial E/\partial F_{21} & \partial E/\partial F_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} \partial E/\partial O_{11} & \partial E/\partial O_{12} \\ \partial E/\partial O_{21} & \partial E/\partial O_{22} \end{bmatrix} \right)$$

https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Forward and Backward Propagation using Convolution operation

- Gradients of error ‘E’ with respect to the filter ‘F’

$$E = E_{O_{11}} + E_{O_{12}} + E_{O_{21}} + E_{O_{22}}$$

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}} X_{11} + \frac{\partial E}{\partial O_{12}} X_{12} + \frac{\partial E}{\partial O_{21}} X_{21} + \frac{\partial E}{\partial O_{22}} X_{22}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}} X_{12} + \frac{\partial E}{\partial O_{12}} X_{13} + \frac{\partial E}{\partial O_{21}} X_{22} + \frac{\partial E}{\partial O_{22}} X_{23}$$

$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}} X_{21} + \frac{\partial E}{\partial O_{12}} X_{22} + \frac{\partial E}{\partial O_{21}} X_{31} + \frac{\partial E}{\partial O_{22}} X_{32}$$

$$\frac{\partial E}{\partial F_{22}} = \frac{\partial E}{\partial O_{11}} X_{22} + \frac{\partial E}{\partial O_{12}} X_{23} + \frac{\partial E}{\partial O_{21}} X_{32} + \frac{\partial E}{\partial O_{22}} X_{33}$$

$$\begin{bmatrix} \partial E/\partial F_{11} & \partial E/\partial F_{12} \\ \partial E/\partial F_{21} & \partial E/\partial F_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} \partial E/\partial O_{11} & \partial E/\partial O_{12} \\ \partial E/\partial O_{21} & \partial E/\partial O_{22} \end{bmatrix} \right)$$

https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Any Questions?

Forward and Backward Propagation using Convolution operation

- Gradients of the error 'E' with respect to the input matrix 'X'

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$E = E_{O_{11}} + E_{O_{12}} + E_{O_{21}} + E_{O_{22}}$$

$$\frac{\partial E}{\partial X_{11}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial X_{11}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial X_{11}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial X_{11}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial X_{11}}$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$
$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$
$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$
$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

$$\frac{\partial E}{\partial X_{11}} = \frac{\partial E}{\partial O_{11}}F_{11} + \frac{\partial E}{\partial O_{12}}0 + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}0$$

Forward and Backward Propagation using Convolution operation

- Gradients of the error 'E' with respect to the input matrix 'X'

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$E = E_{O_{11}} + E_{O_{12}} + E_{O_{21}} + E_{O_{22}}$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$
$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$
$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$
$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

$$\frac{\partial E}{\partial X_{11}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial X_{11}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial X_{11}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial X_{11}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial X_{11}}$$

$$\frac{\partial E}{\partial X_{11}} = \frac{\partial E}{\partial O_{11}}F_{11} + \frac{\partial E}{\partial O_{12}}0 + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}0$$

$$\frac{\partial E}{\partial X_{23}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial X_{23}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial X_{23}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial X_{23}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial X_{23}}$$

$$\frac{\partial E}{\partial X_{23}} = \frac{\partial E}{\partial O_{11}}0 + \frac{\partial E}{\partial O_{12}}F_{22} + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}F_{12}$$

https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

Forward and Backward Propagation using Convolution operation

- Gradients of the error 'E' with respect to the input matrix 'X'

$$\frac{\partial E}{\partial X_{11}} = \frac{\partial E}{\partial O_{11}} F_{11} + \frac{\partial E}{\partial O_{12}} 0 + \frac{\partial E}{\partial O_{21}} 0 + \frac{\partial E}{\partial O_{22}} 0$$

$$\frac{\partial E}{\partial X_{12}} = \frac{\partial E}{\partial O_{11}} F_{12} + \frac{\partial E}{\partial O_{12}} F_{11} + \frac{\partial E}{\partial O_{21}} 0 + \frac{\partial E}{\partial O_{22}} 0$$

$$\frac{\partial E}{\partial X_{13}} = \frac{\partial E}{\partial O_{11}} 0 + \frac{\partial E}{\partial O_{12}} F_{12} + \frac{\partial E}{\partial O_{21}} 0 + \frac{\partial E}{\partial O_{22}} 0$$

$$\frac{\partial E}{\partial X_{21}} = \frac{\partial E}{\partial O_{11}} F_{21} + \frac{\partial E}{\partial O_{12}} 0 + \frac{\partial E}{\partial O_{21}} F_{11} + \frac{\partial E}{\partial O_{22}} 0$$

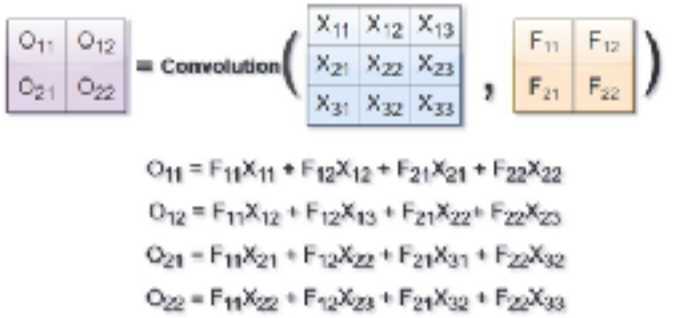
$$\frac{\partial E}{\partial X_{22}} = \frac{\partial E}{\partial O_{11}} F_{22} + \frac{\partial E}{\partial O_{12}} F_{21} + \frac{\partial E}{\partial O_{21}} f_{12} + \frac{\partial E}{\partial O_{22}} F_{11}$$

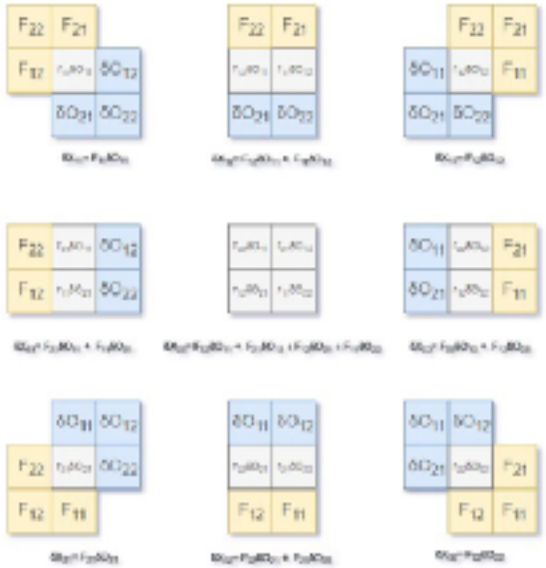
$$\frac{\partial E}{\partial X_{23}} = \frac{\partial E}{\partial O_{11}} 0 + \frac{\partial E}{\partial O_{12}} F_{22} + \frac{\partial E}{\partial O_{21}} 0 + \frac{\partial E}{\partial O_{22}} F_{12}$$

$$\frac{\partial E}{\partial X_{31}} = \frac{\partial E}{\partial O_{11}} 0 + \frac{\partial E}{\partial O_{12}} 0 + \frac{\partial E}{\partial O_{21}} F_{21} + \frac{\partial E}{\partial O_{22}} 0$$

$$\frac{\partial E}{\partial X_{32}} = \frac{\partial E}{\partial O_{11}} 0 + \frac{\partial E}{\partial O_{12}} 0 + \frac{\partial E}{\partial O_{21}} F_{22} + \frac{\partial E}{\partial O_{22}} F_{21}$$

$$\frac{\partial E}{\partial X_{33}} = \frac{\partial E}{\partial O_{11}} 0 + \frac{\partial E}{\partial O_{12}} 0 + \frac{\partial E}{\partial O_{21}} 0 + \frac{\partial E}{\partial O_{22}} F_{22}$$



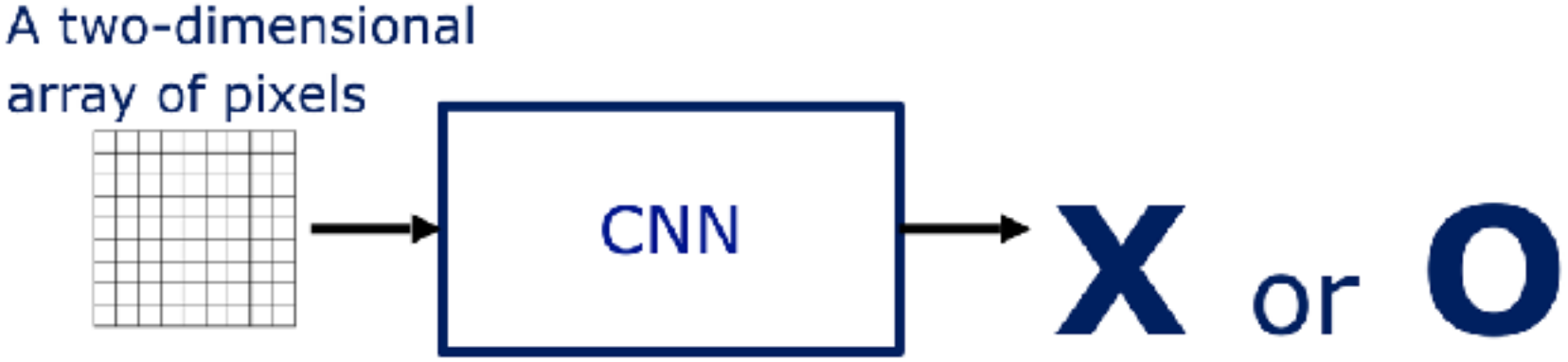


https://medium.com/@2017csm1006/forward-and-backpropagation-in-convolutional-neural-network-4dfa96d7b37e

A Toy ConvNet X's and O's

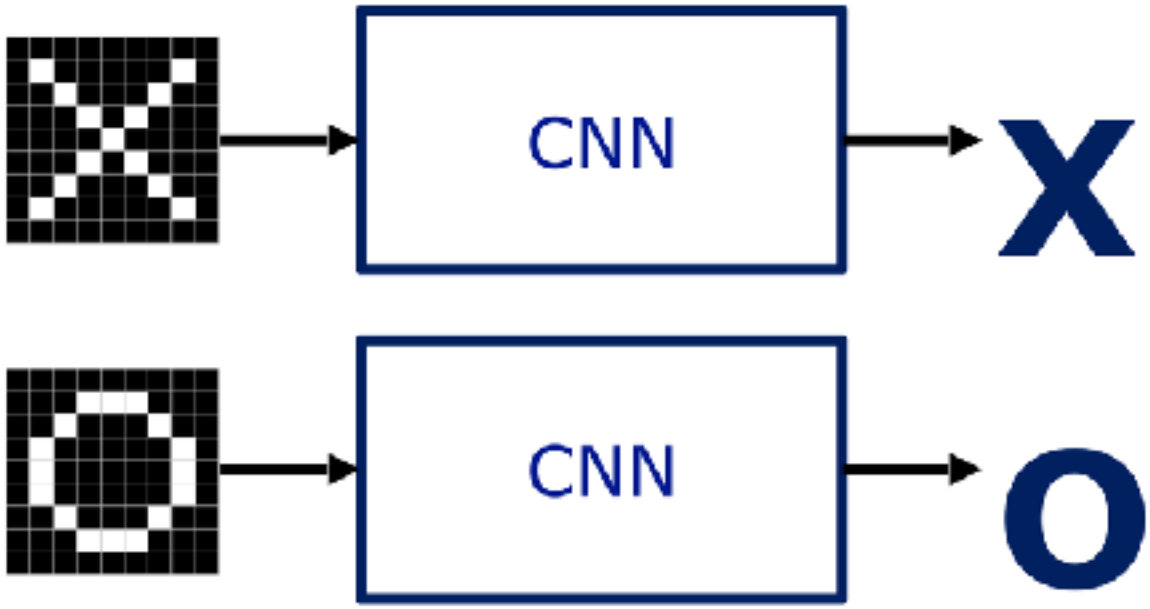
A toy ConvNet: X's and O's

Says whether a picture is of an X or an O



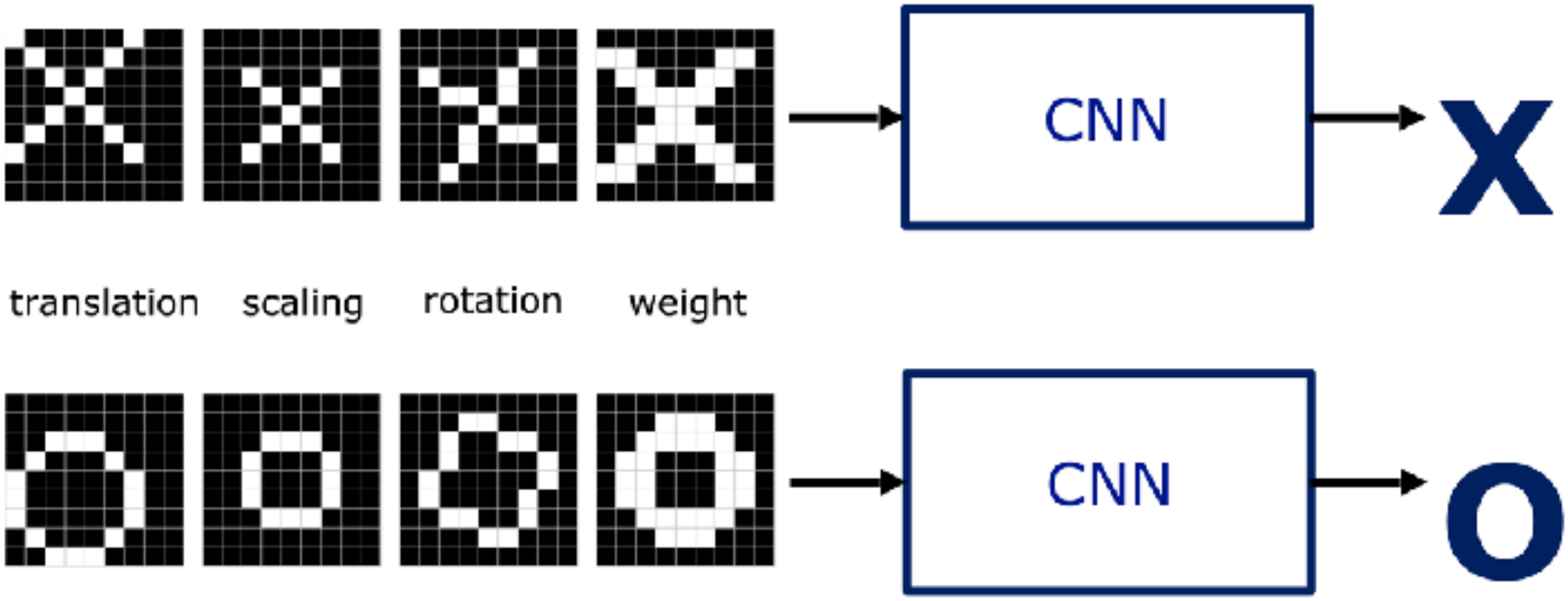
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

For example



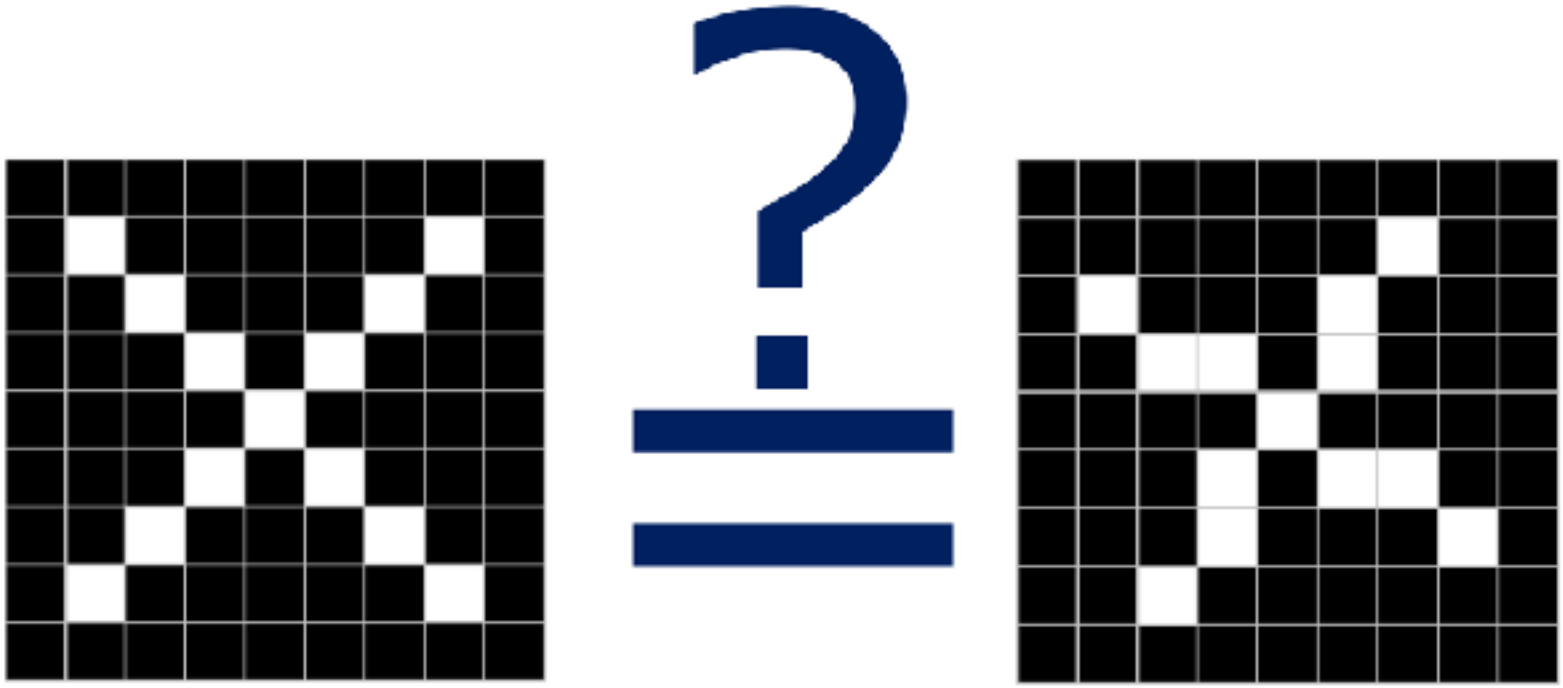
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Trickier cases



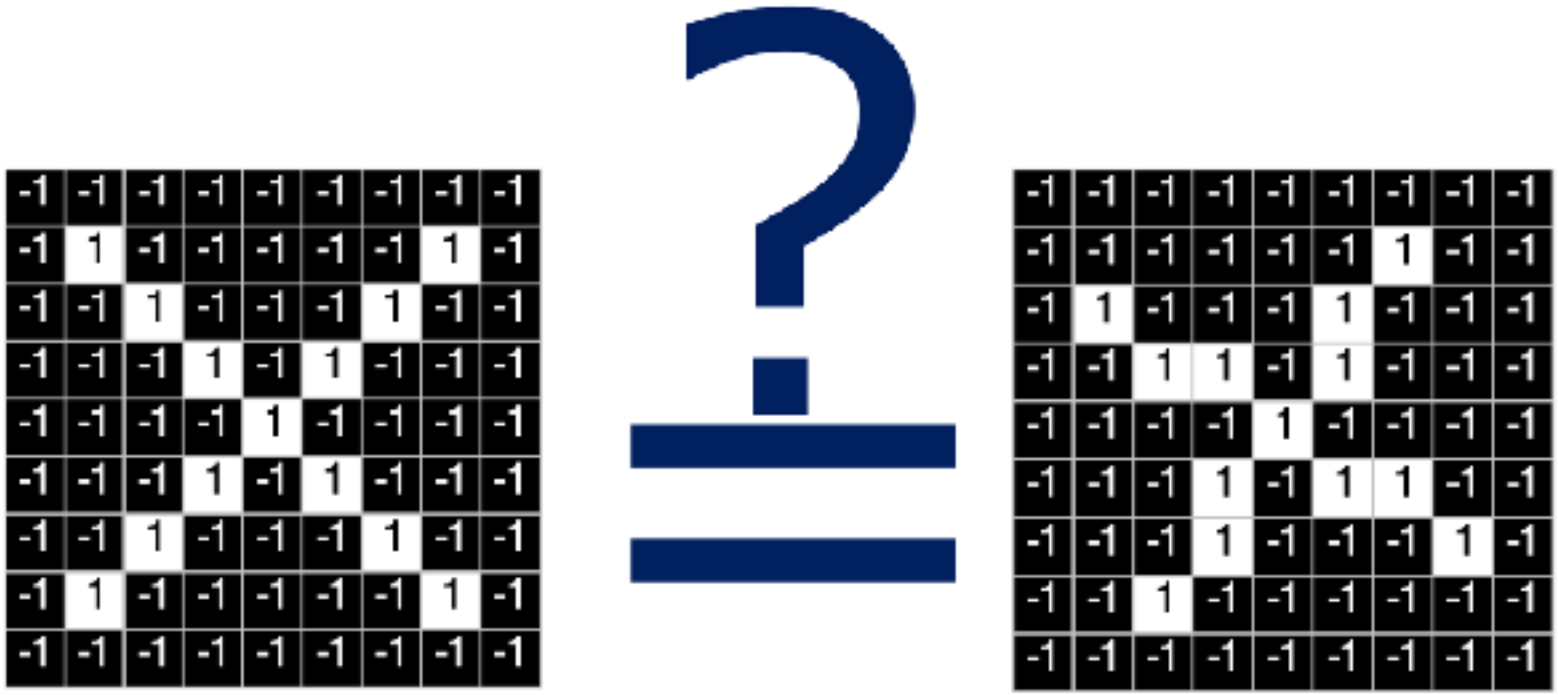
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Deciding is hard



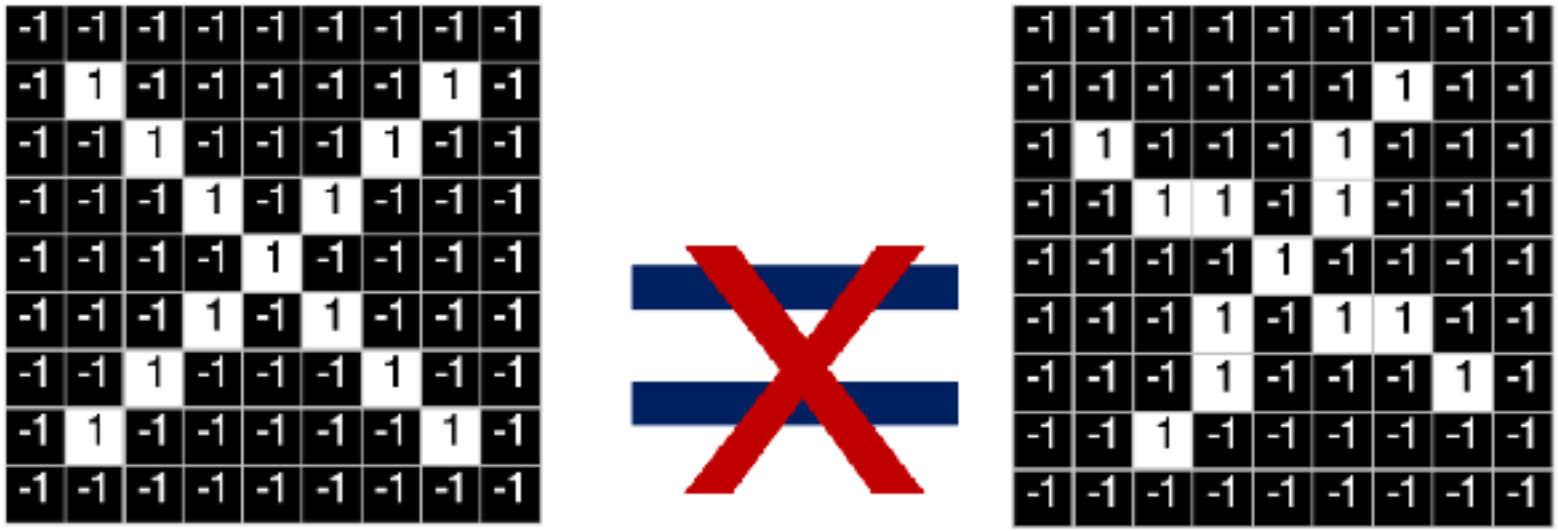
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

What computers see



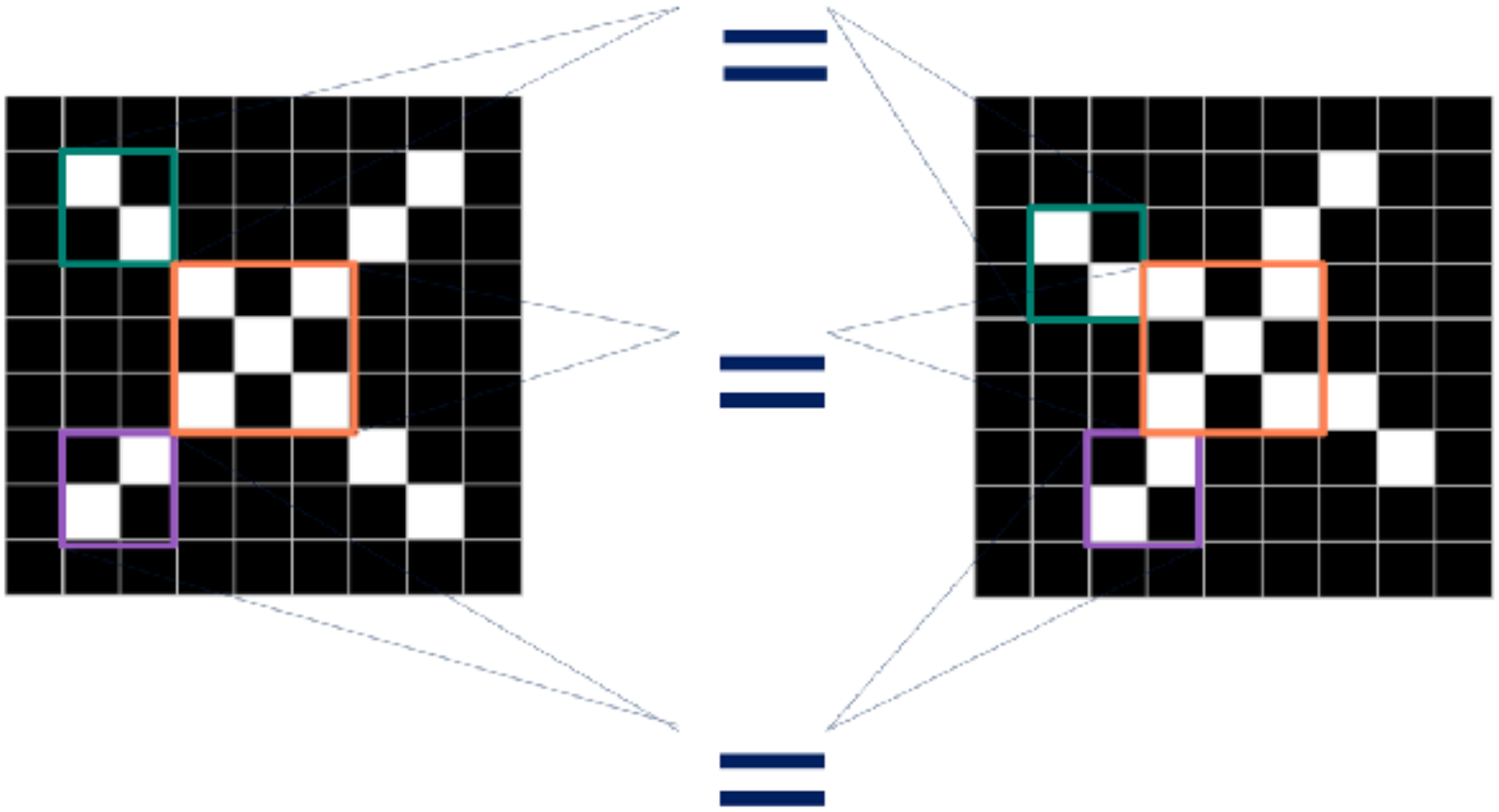
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Computers are literal



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

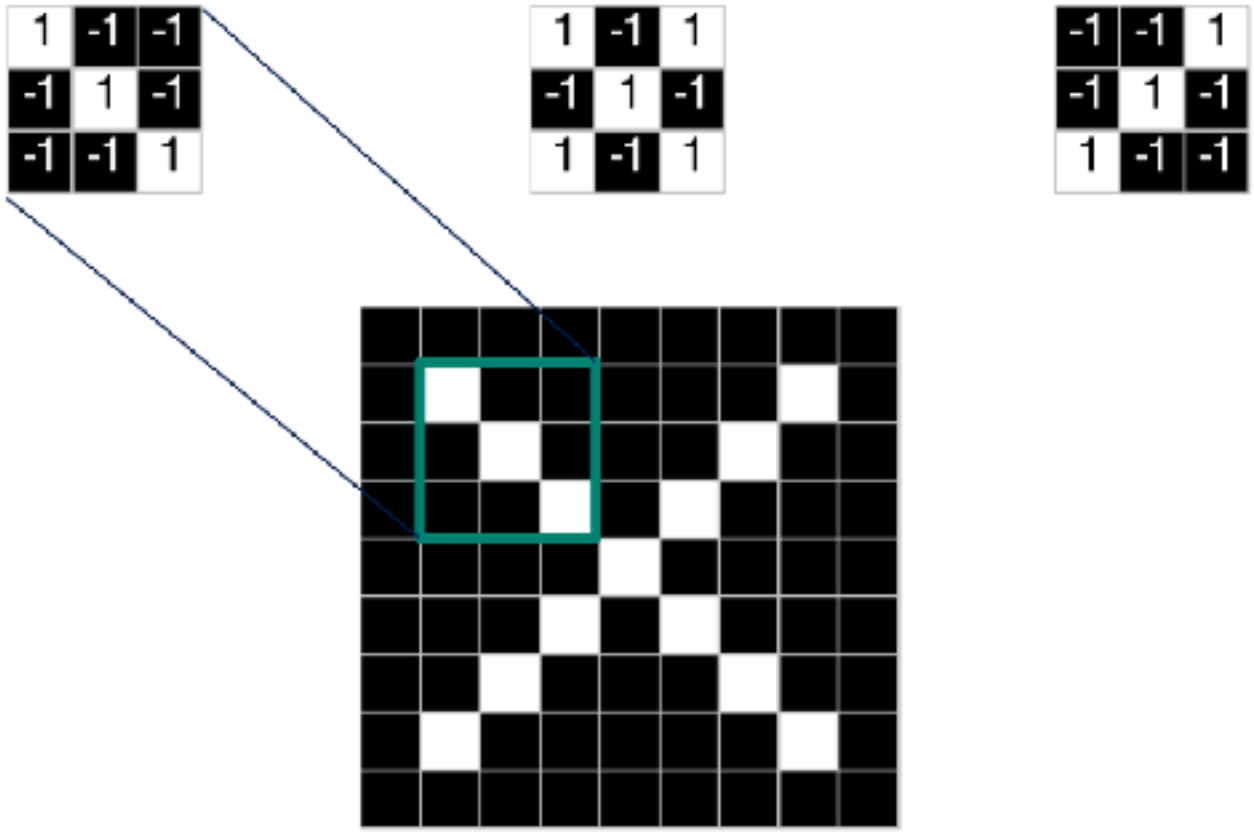
ConvNets match pieces of the image



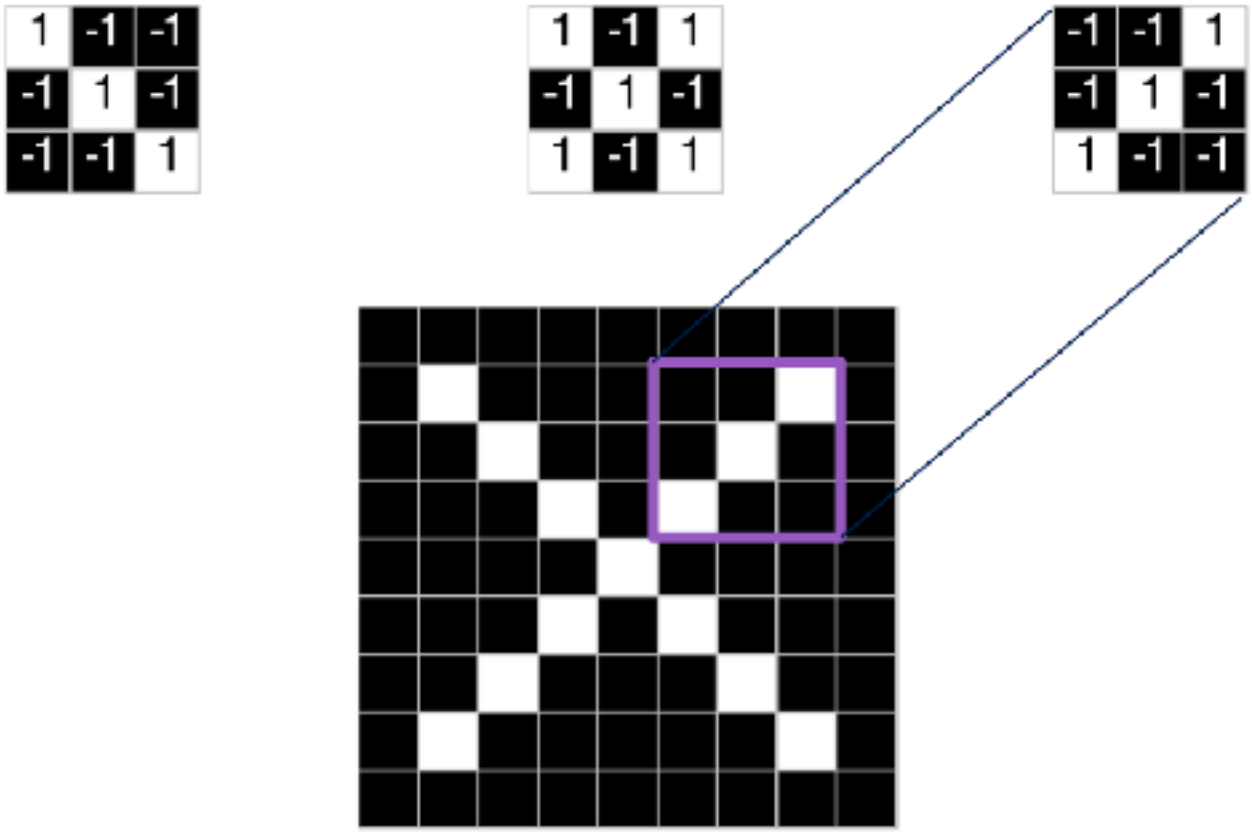
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Features match pieces of the image

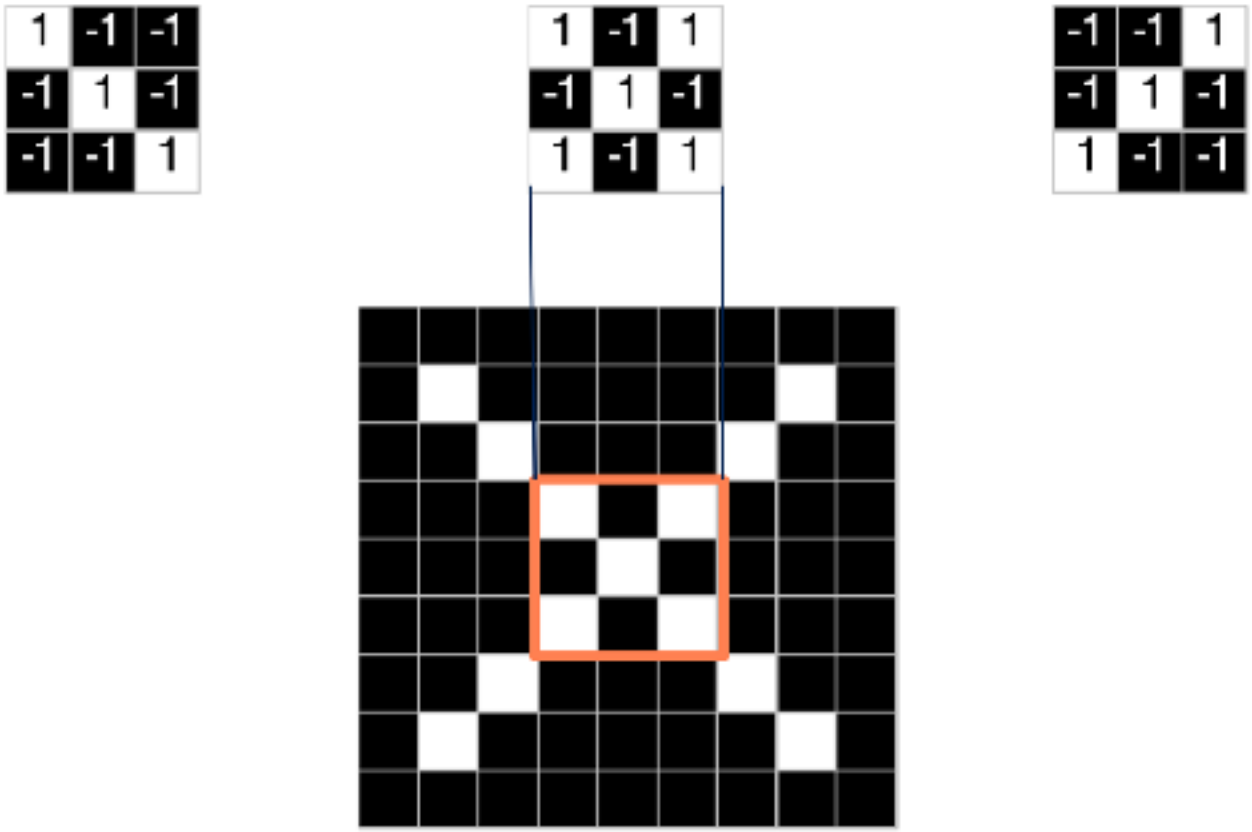
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html



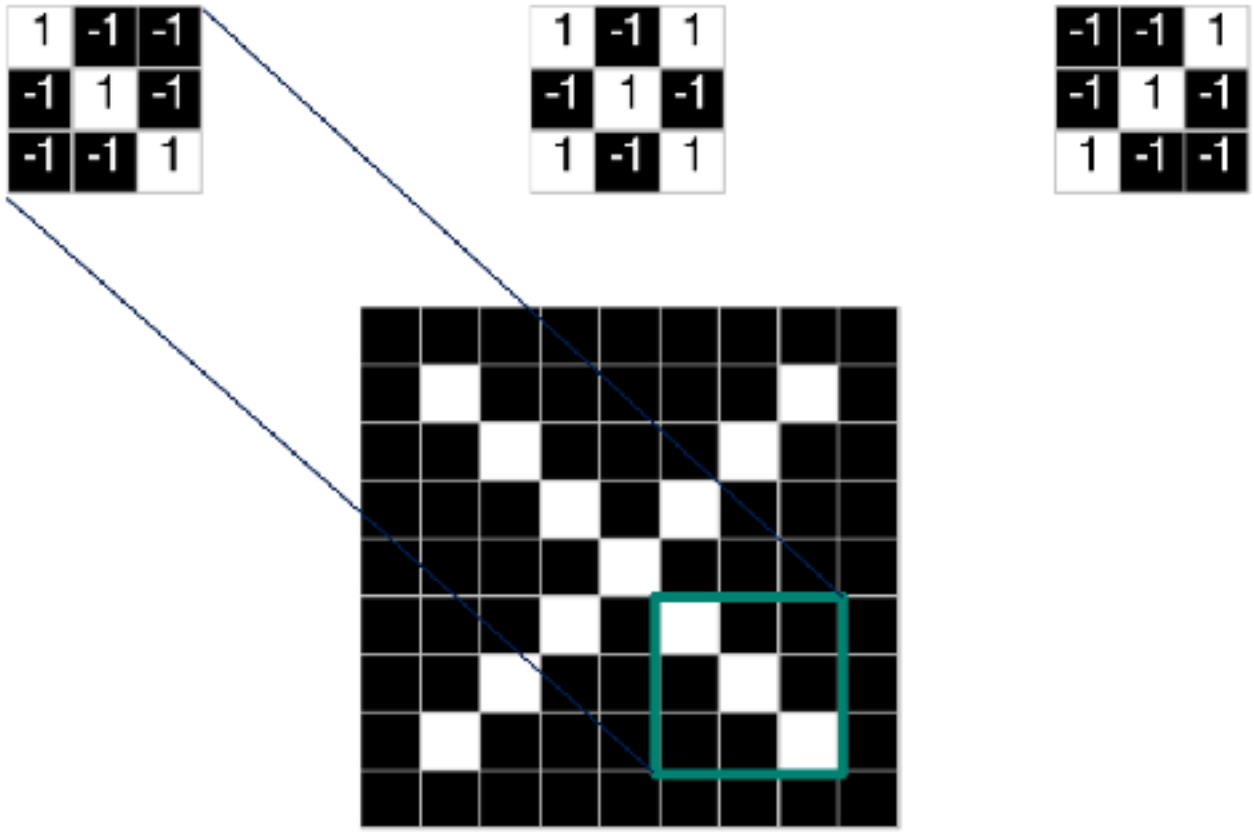
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html



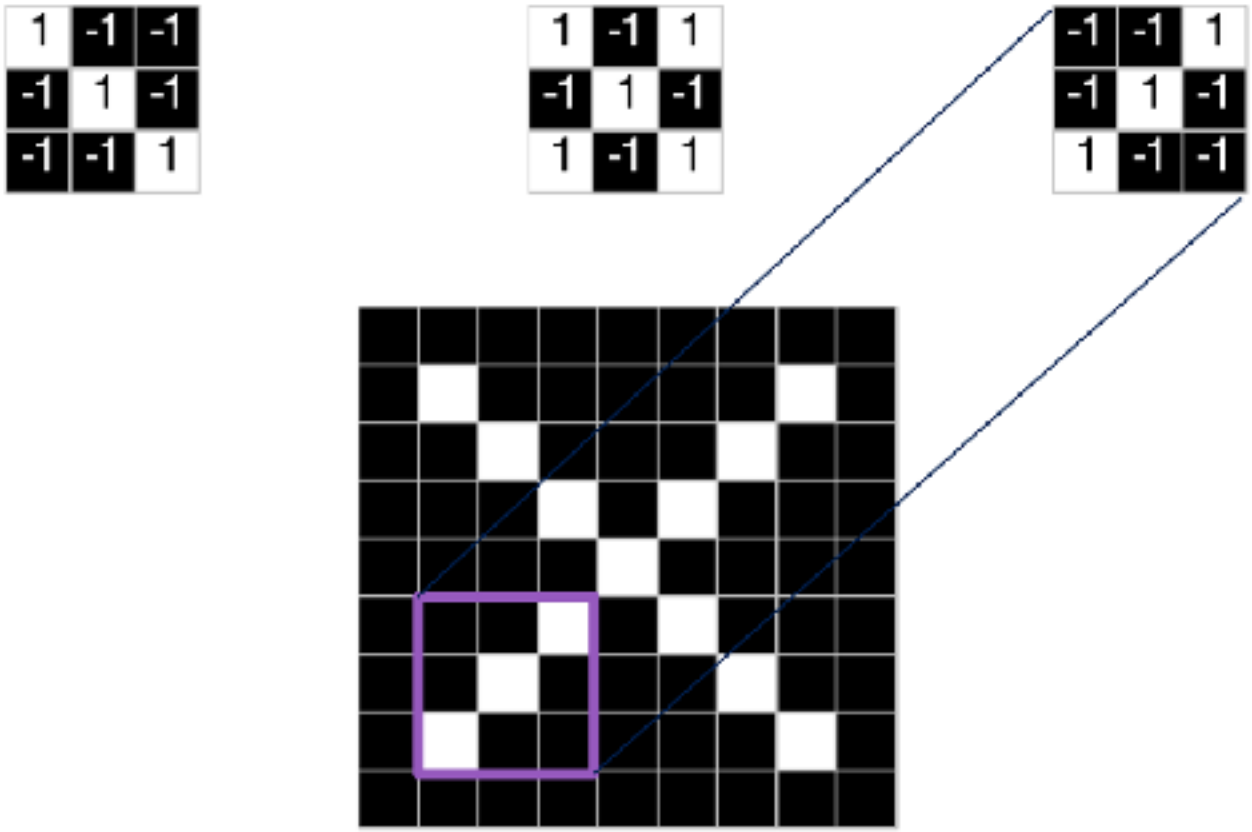
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

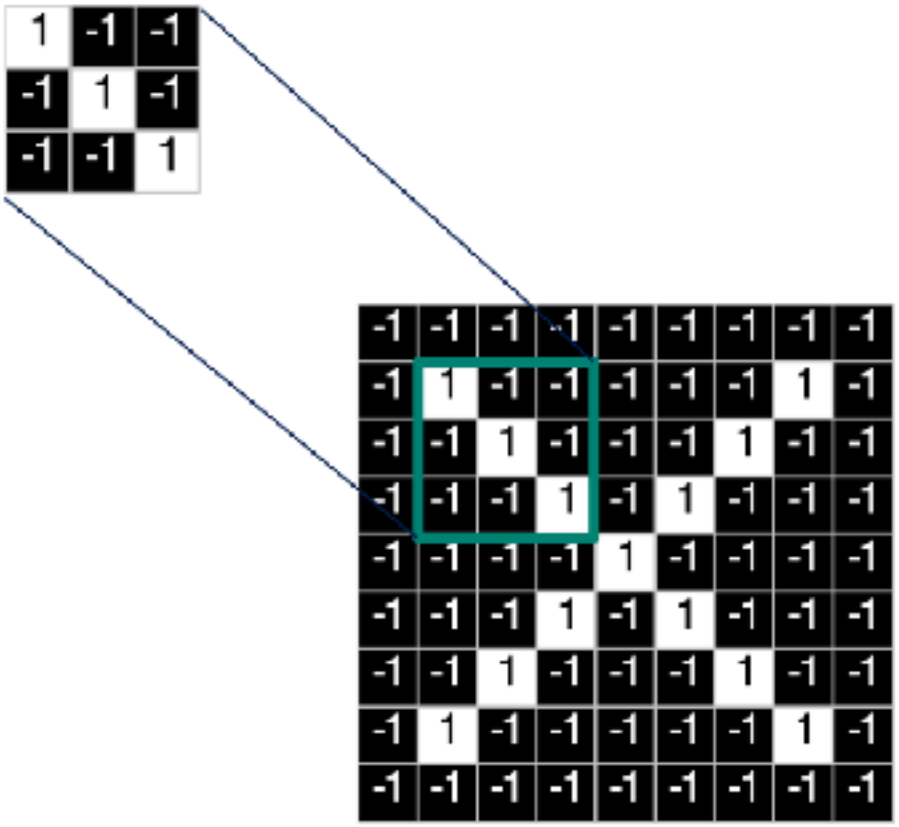


Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Filtering: The math behind the match



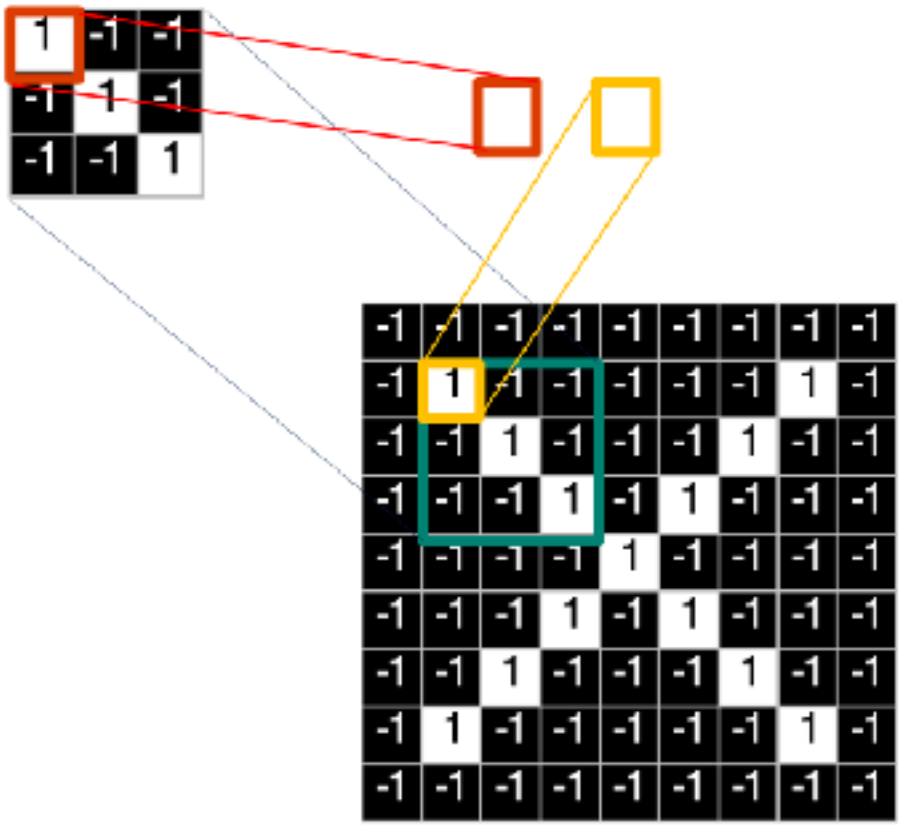
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Filtering: The math behind the match

1. Line up the feature and the image patch.
2. Multiply each image pixel by the corresponding feature pixel.
3. Add them up.
4. Divide by the total number of pixels in the feature.

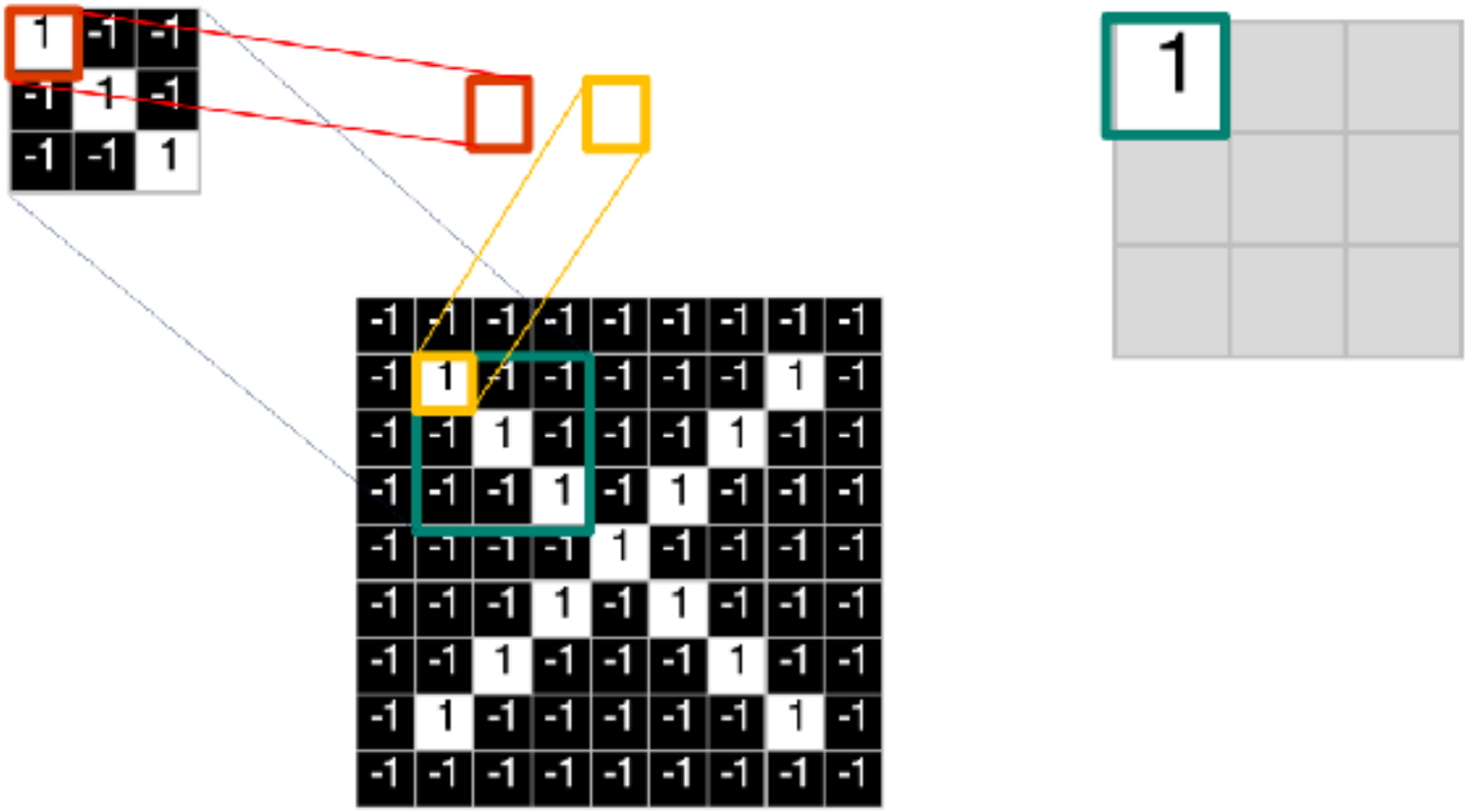
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Filtering: The math behind the match



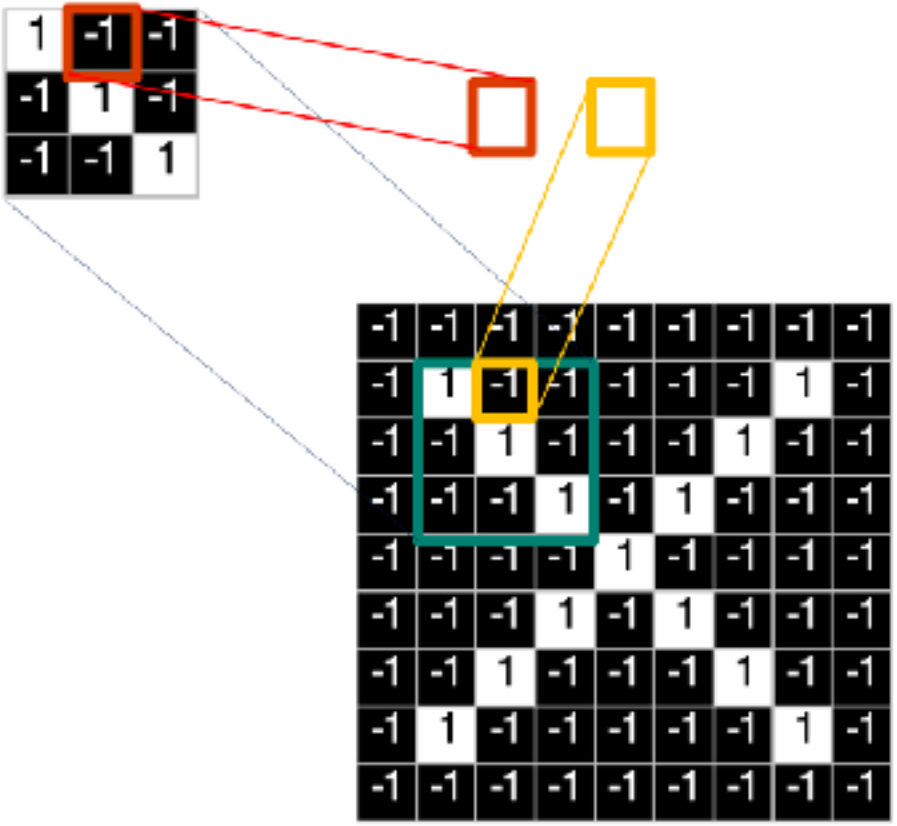
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Filtering: The math behind the match



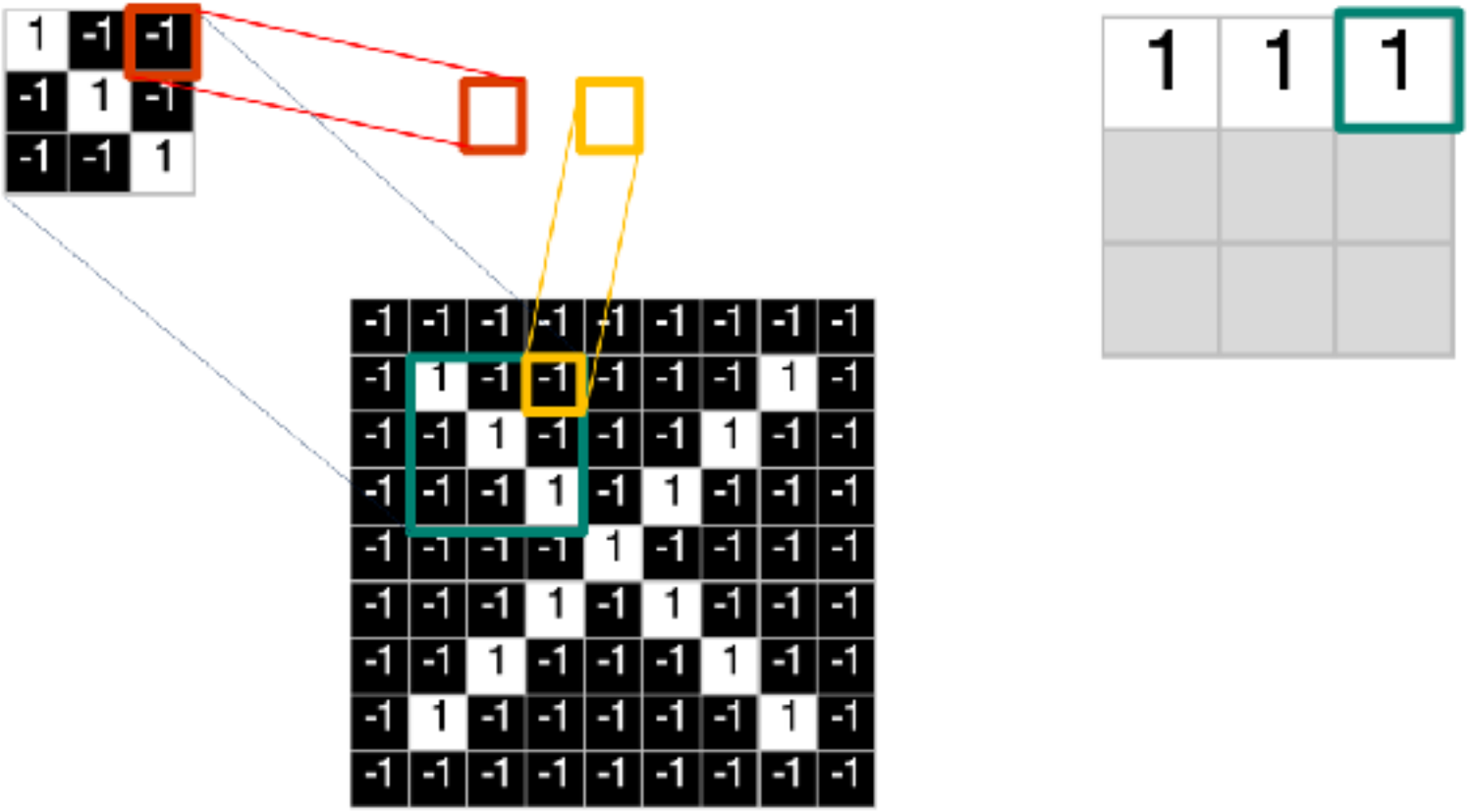
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Filtering: The math behind the match



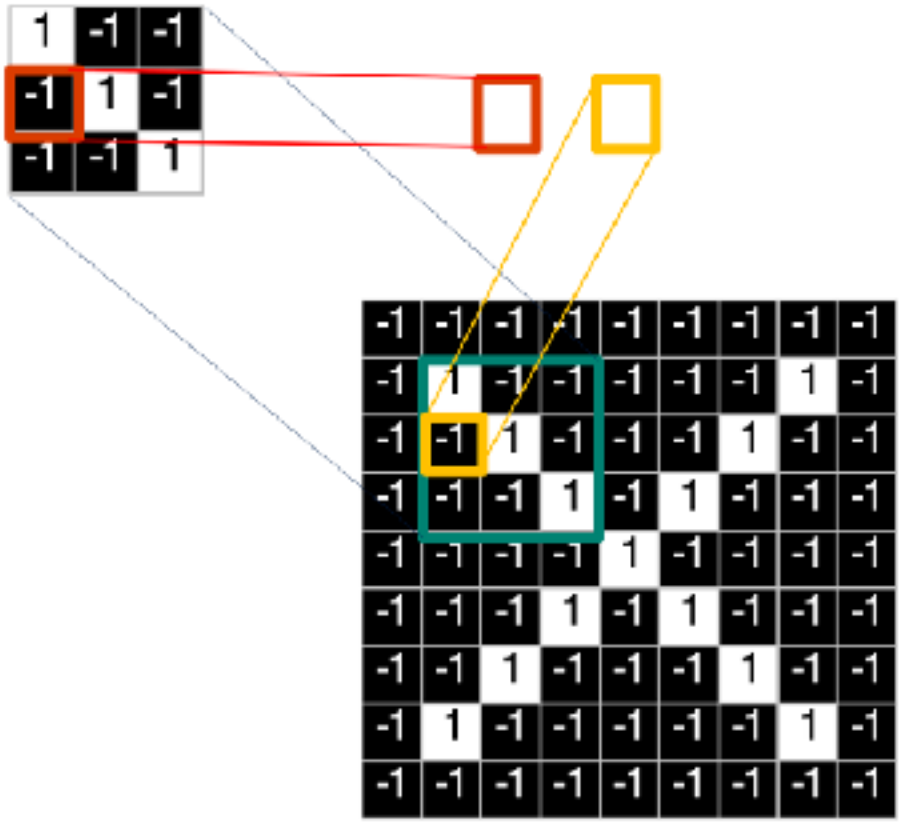
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

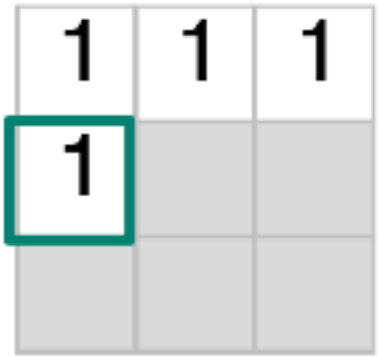
Filtering: The math behind the match



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

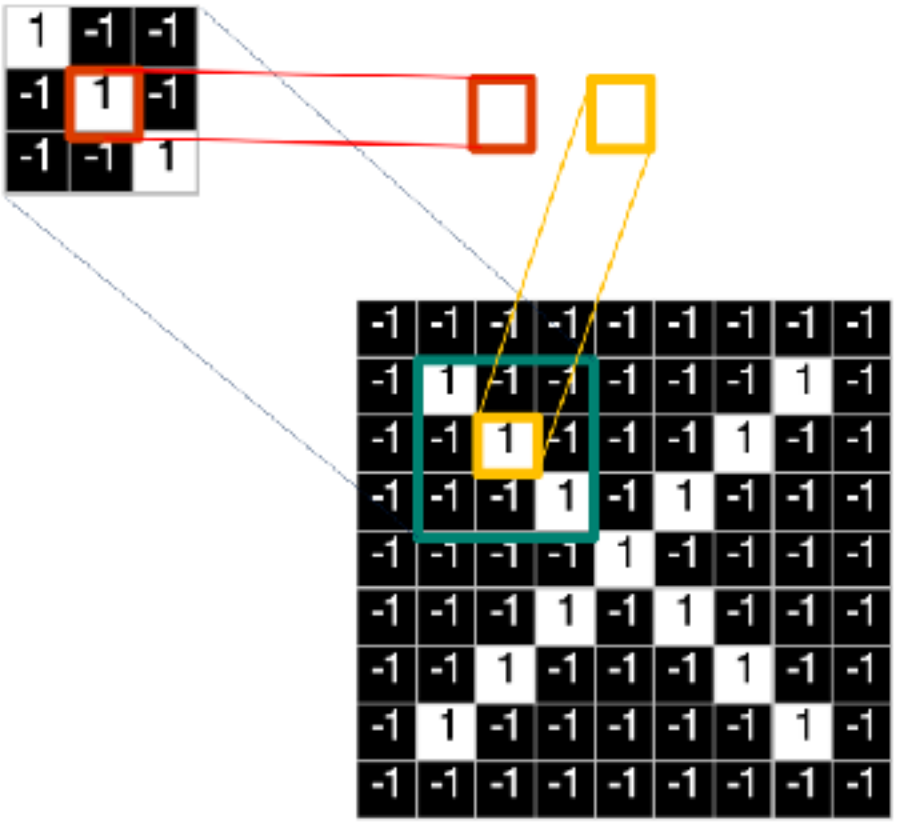
Filtering: The math behind the match

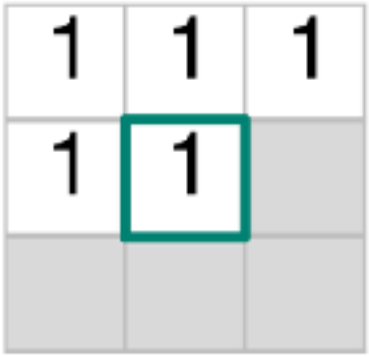




Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

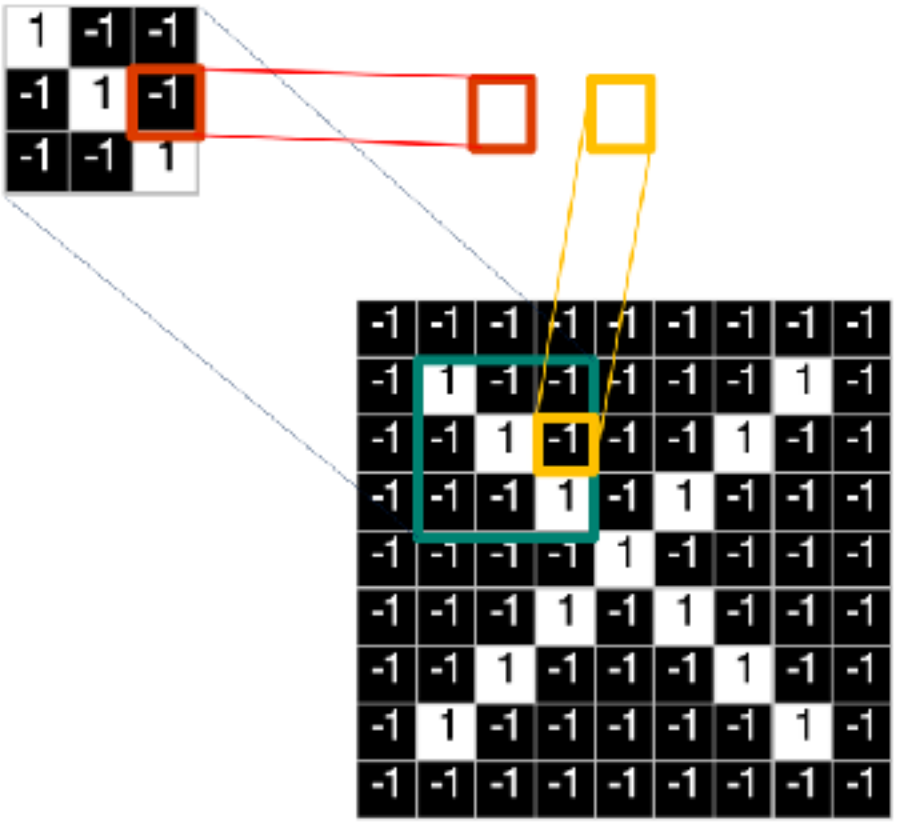
Filtering: The math behind the match





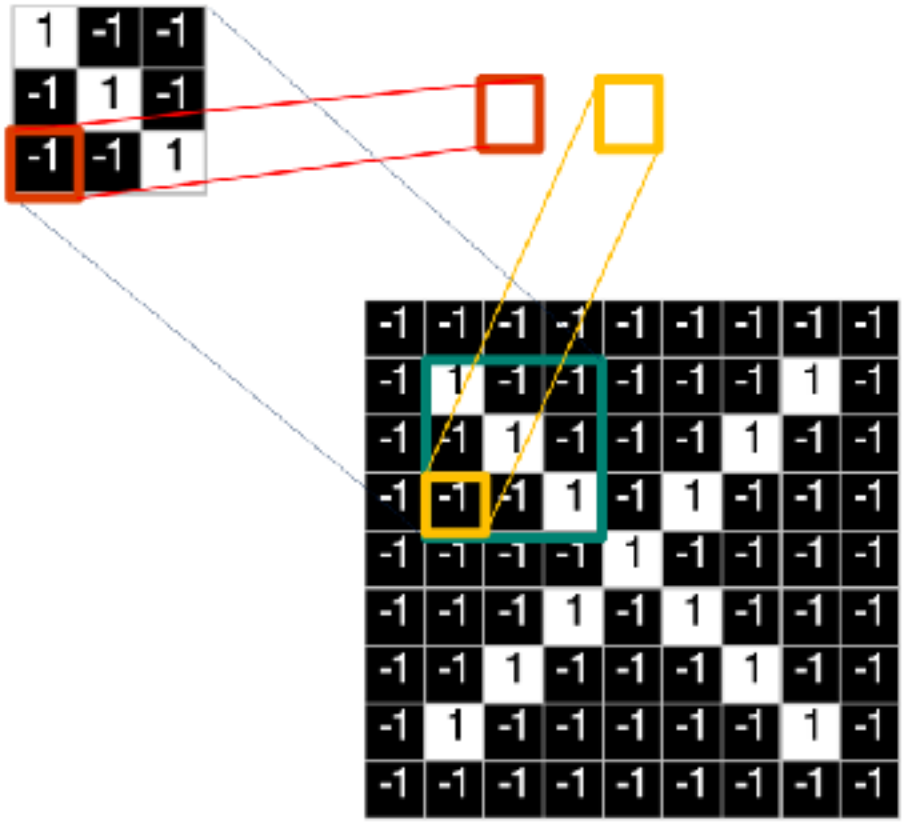
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

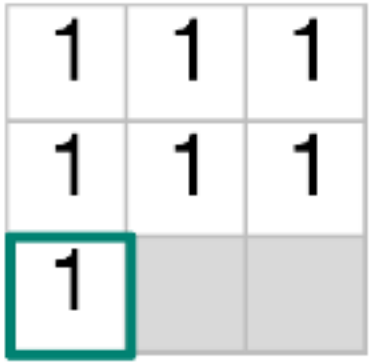
Filtering: The math behind the match



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

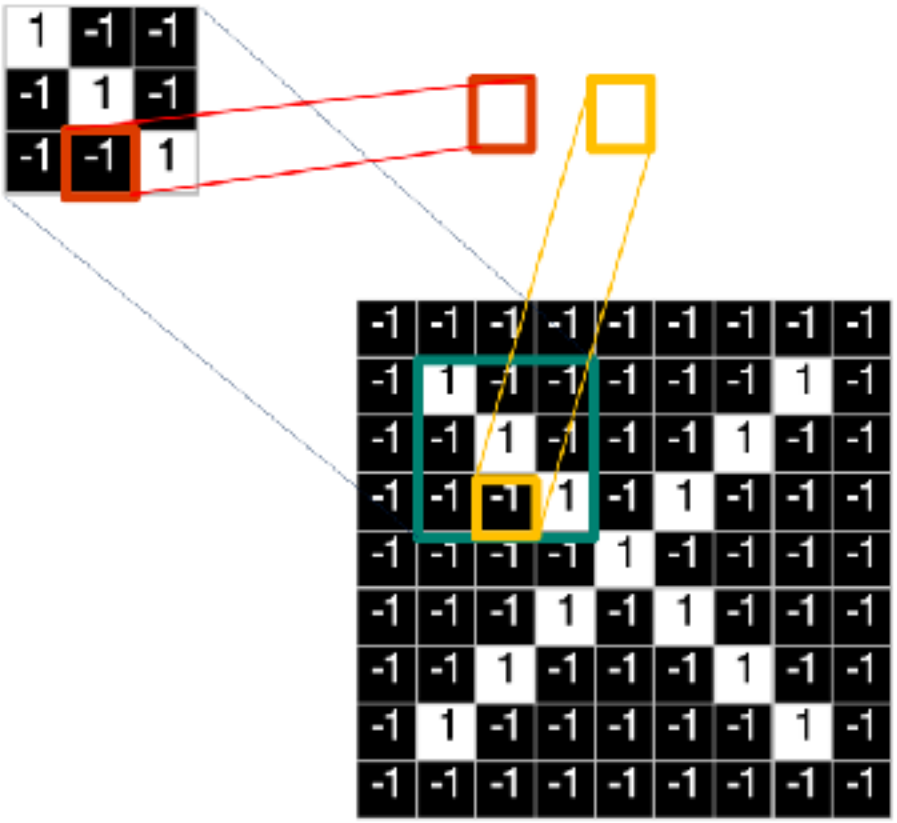
Filtering: The math behind the match





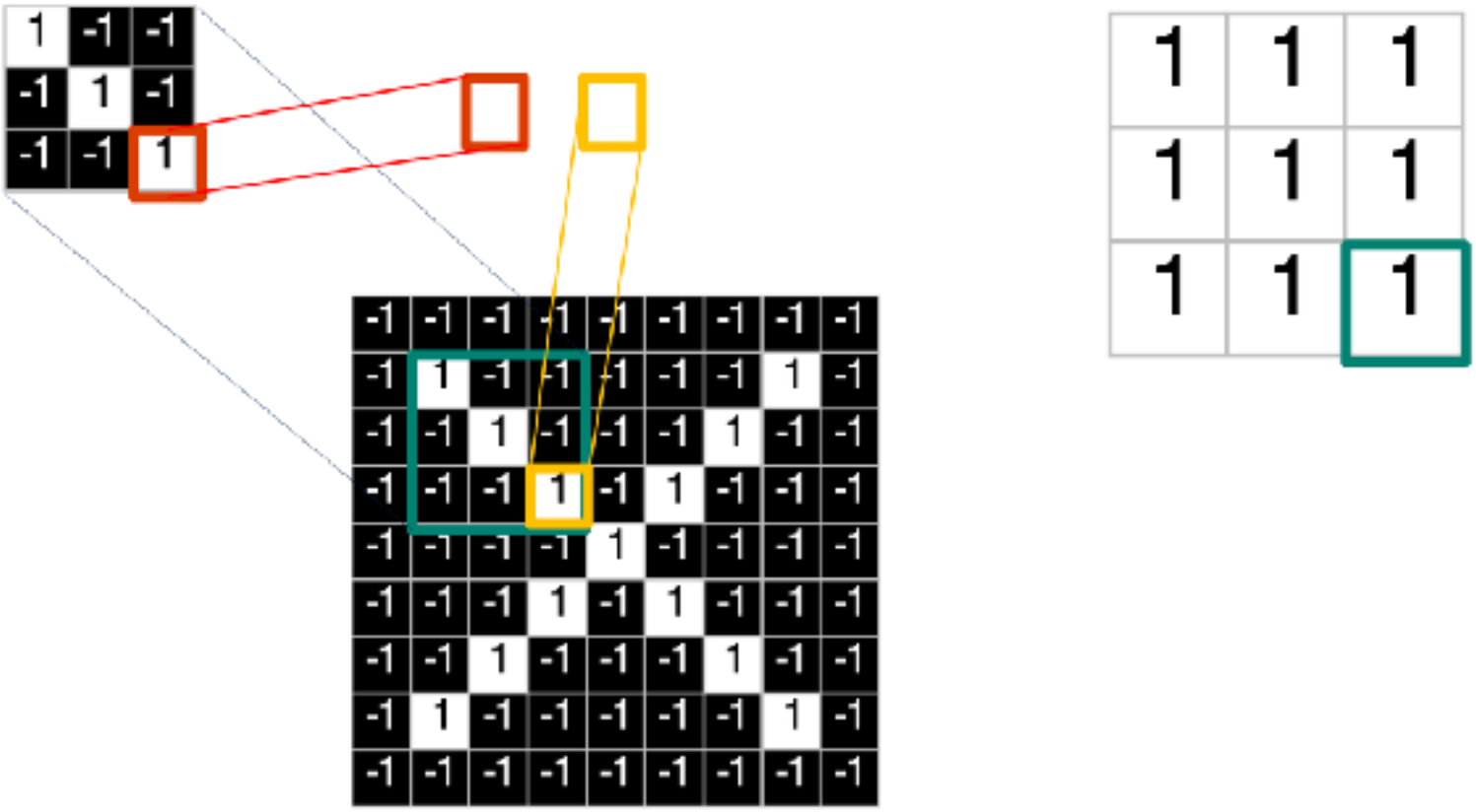
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Filtering: The math behind the match



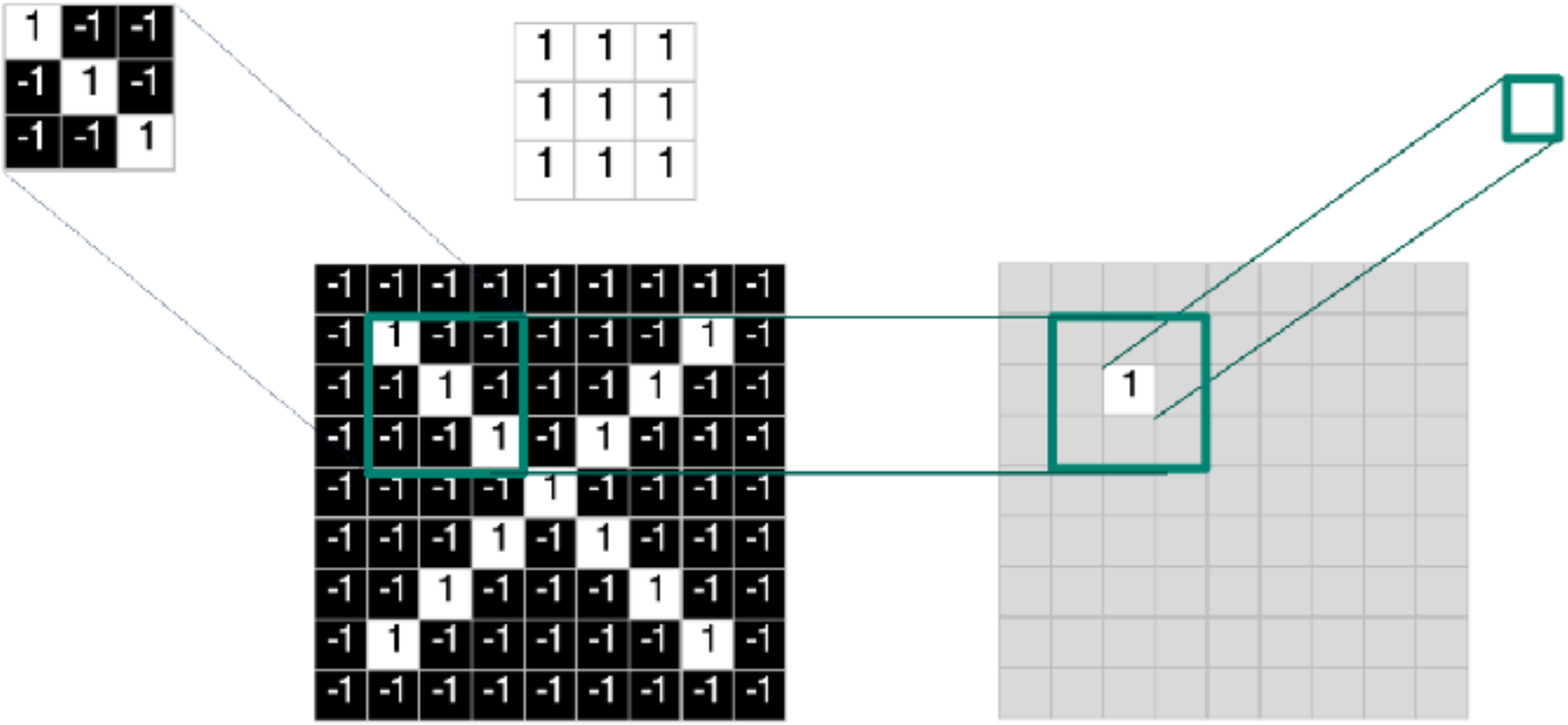
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Filtering: The math behind the match



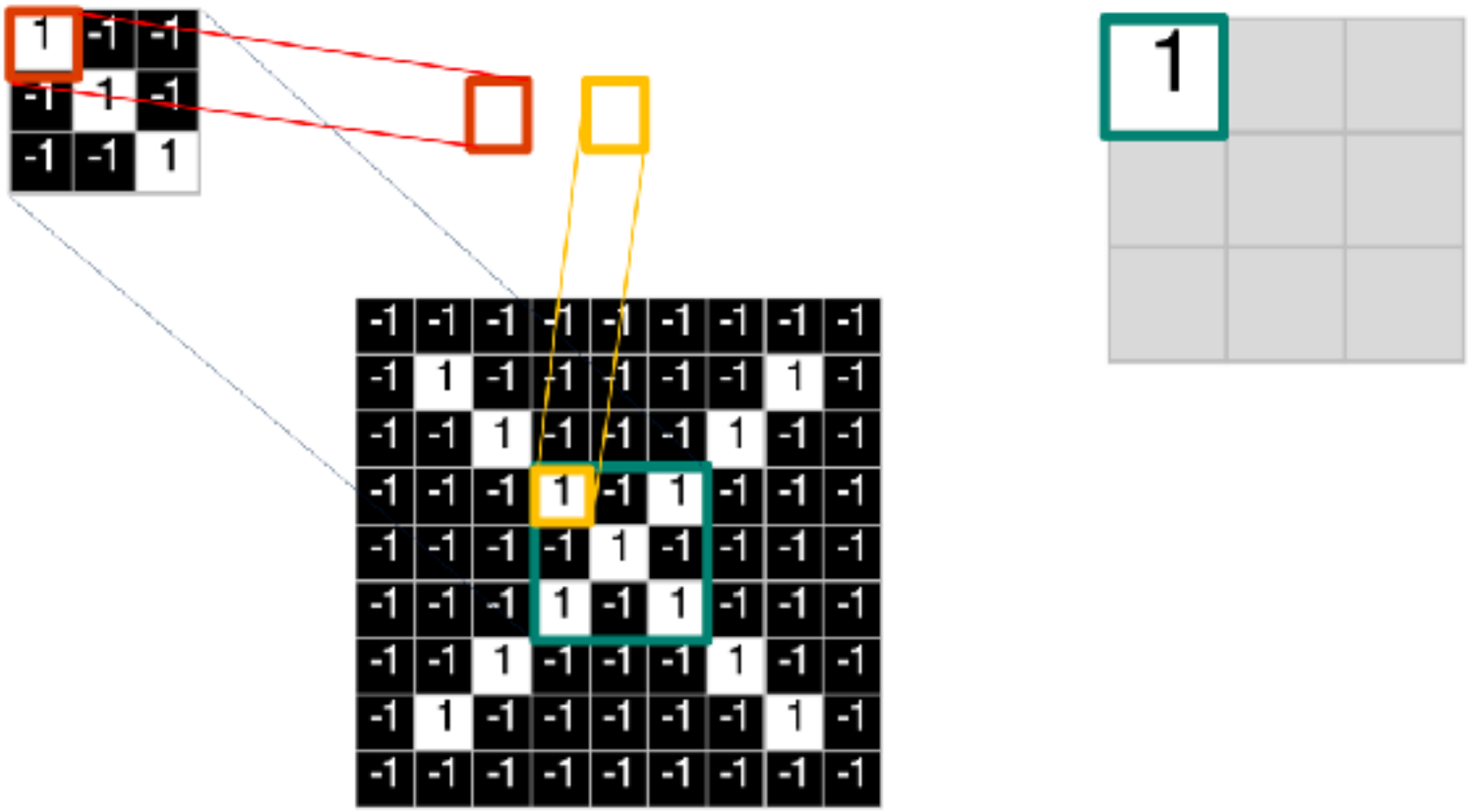
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Filtering: The math behind the match



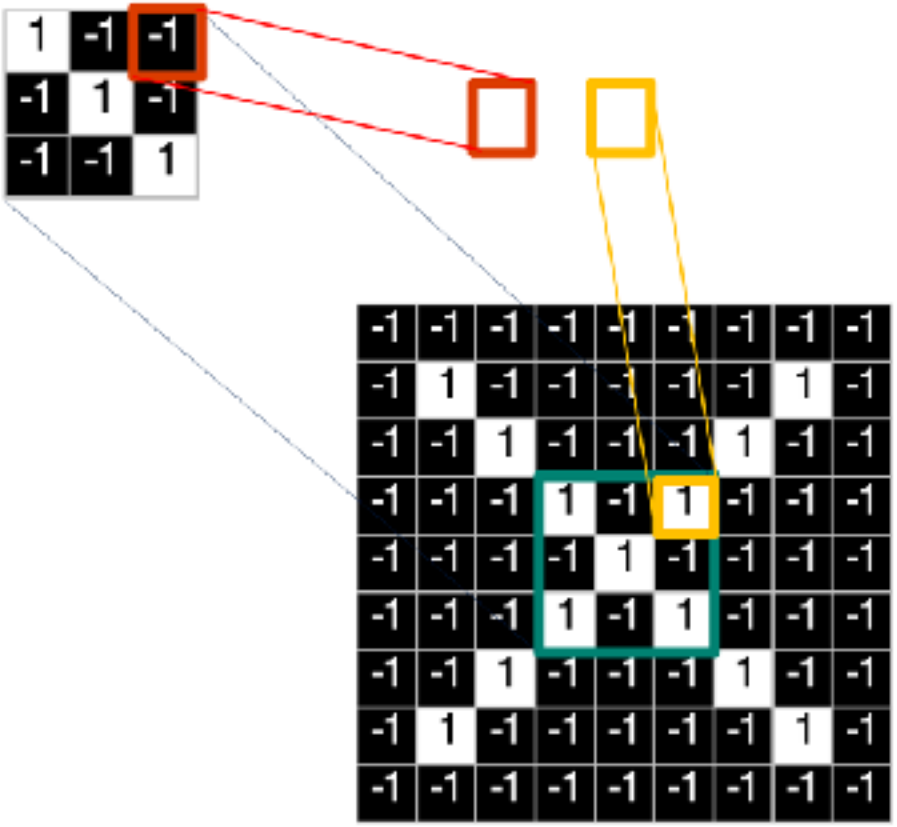
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Filtering: The math behind the match



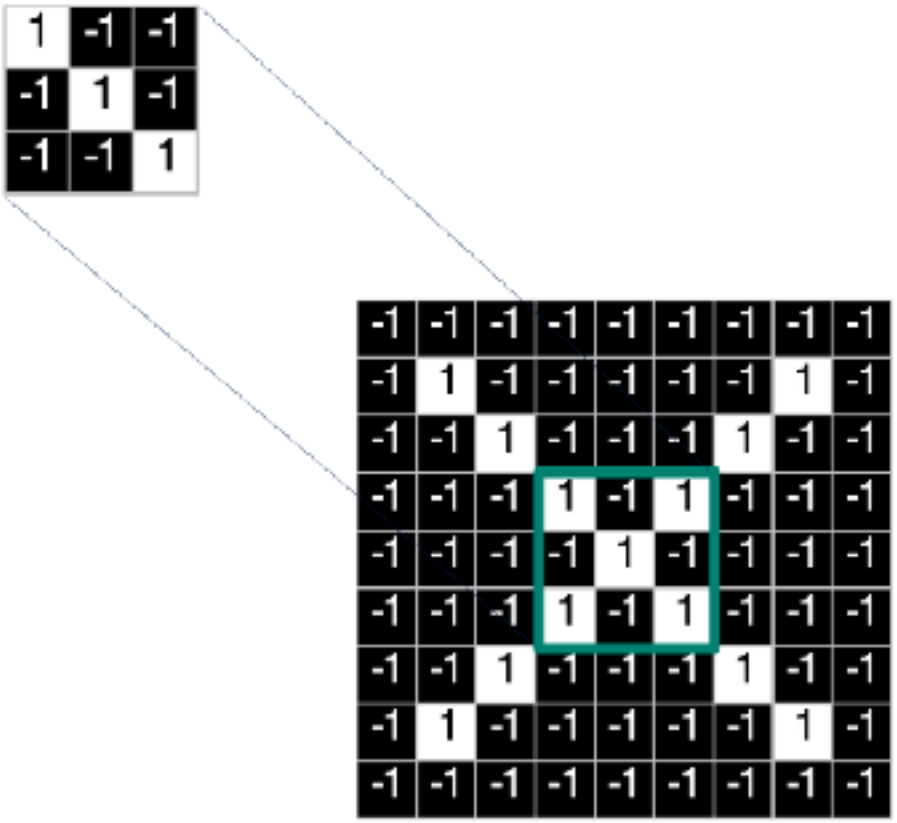
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

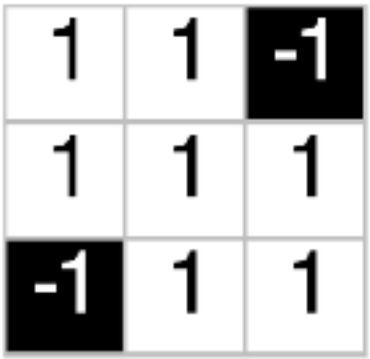
Filtering: The math behind the match



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

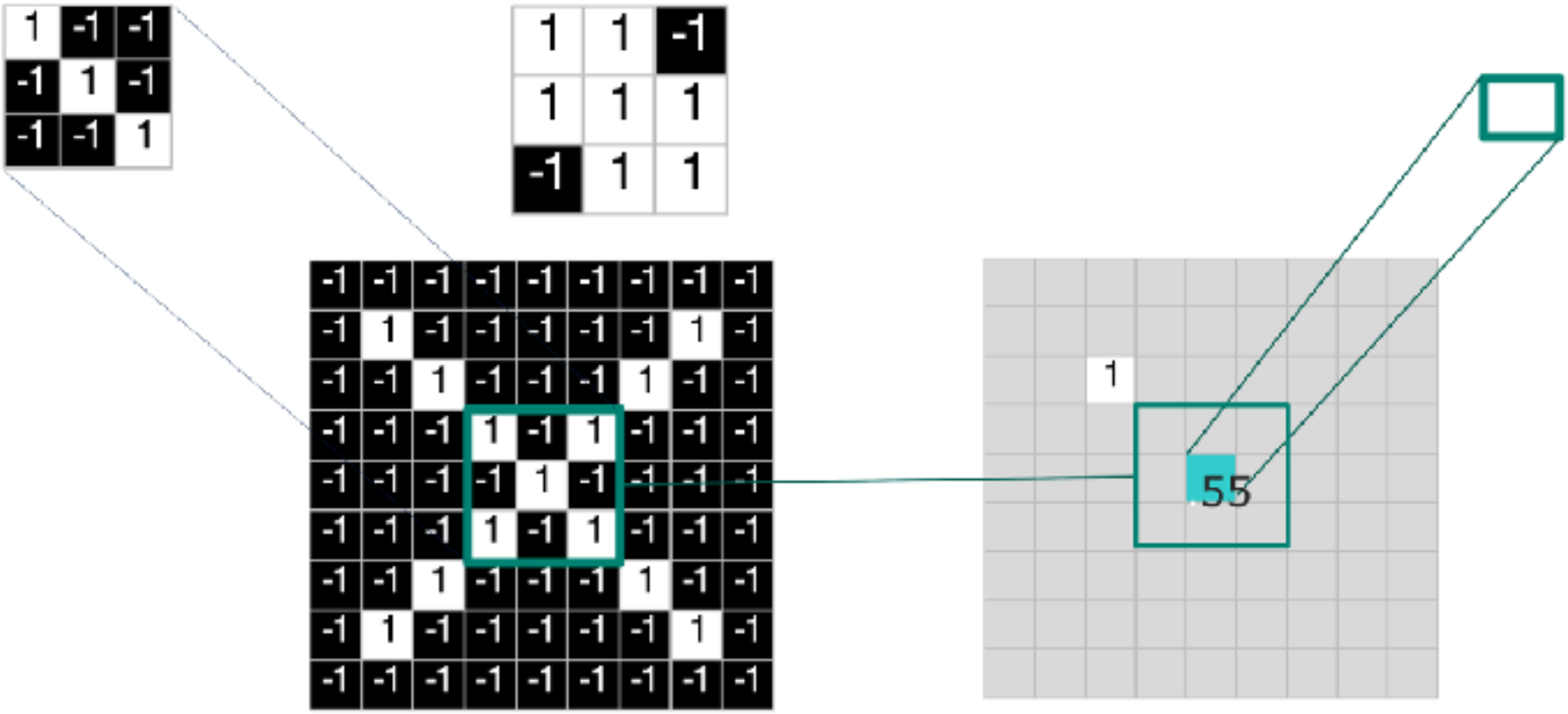
Filtering: The math behind the match





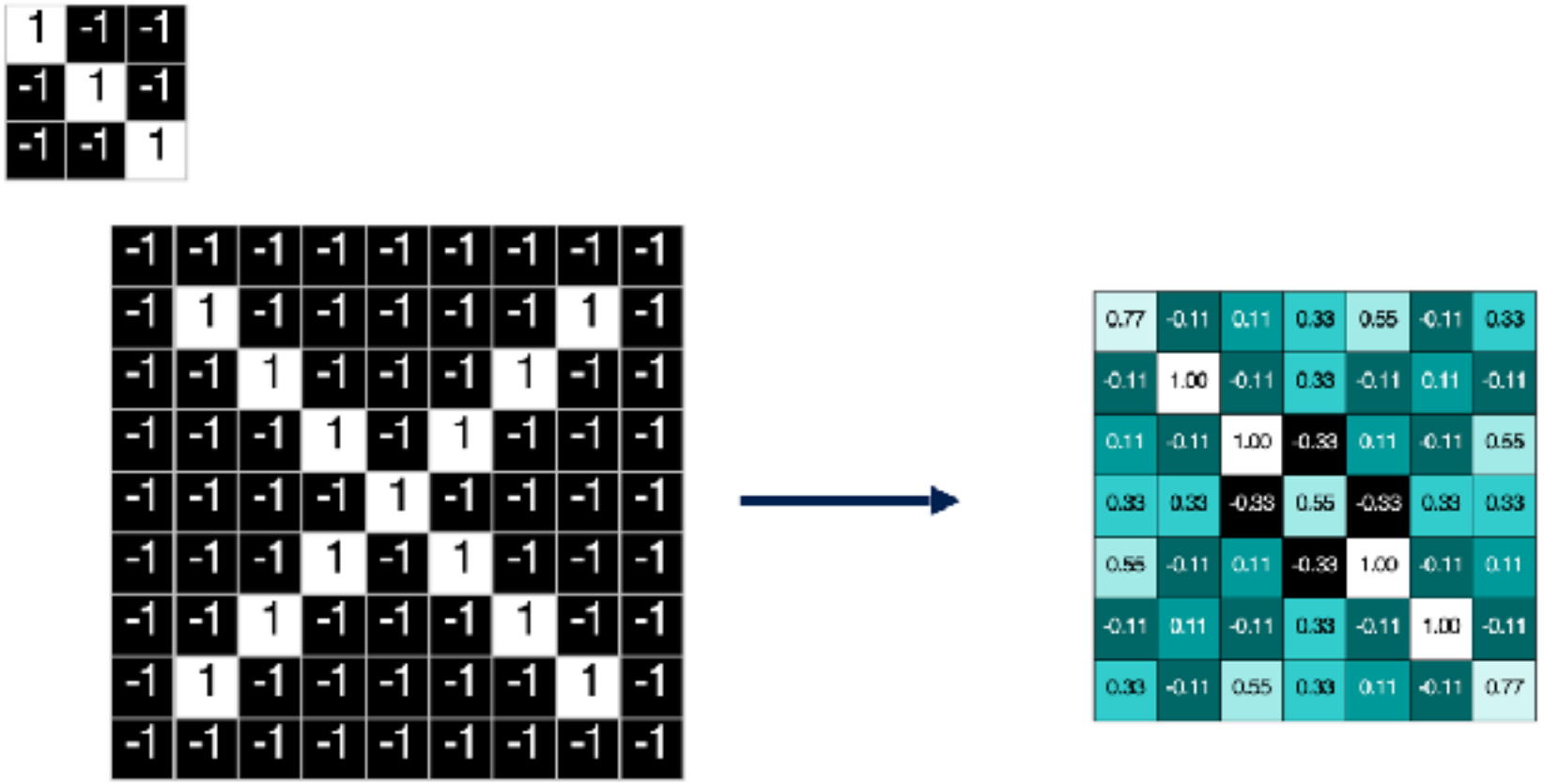
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Filtering: The math behind the match



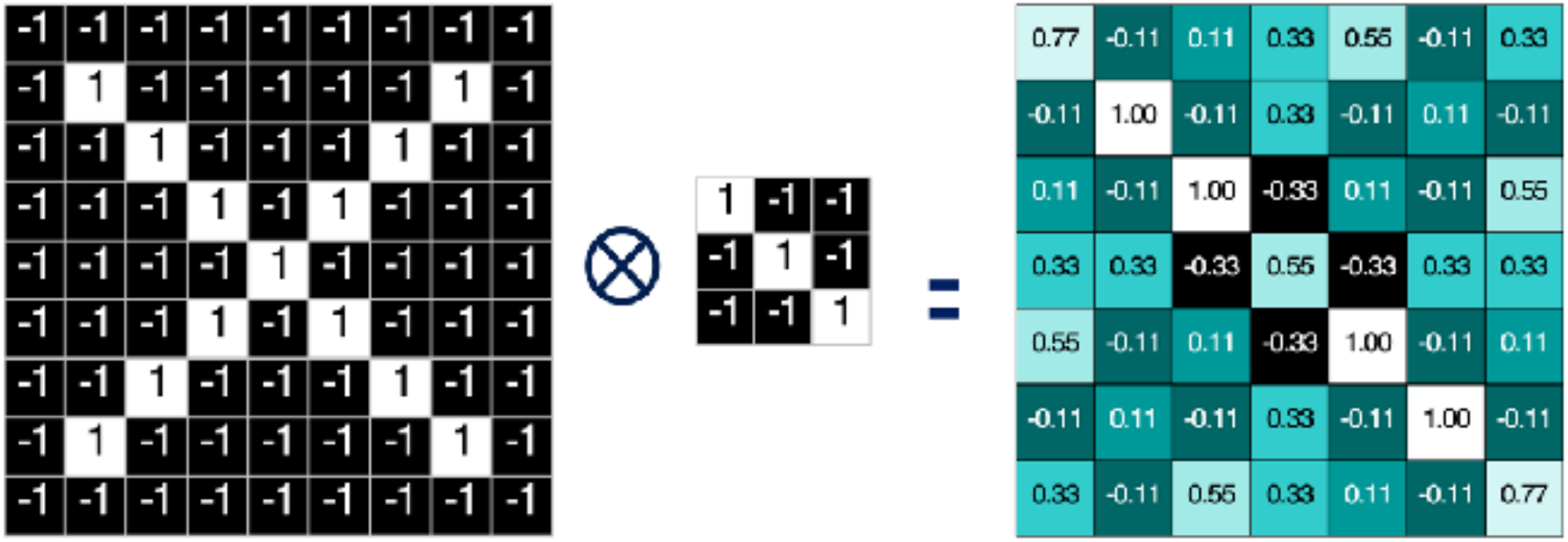
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Convolution: Trying every possible match



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Convolution: Trying every possible match

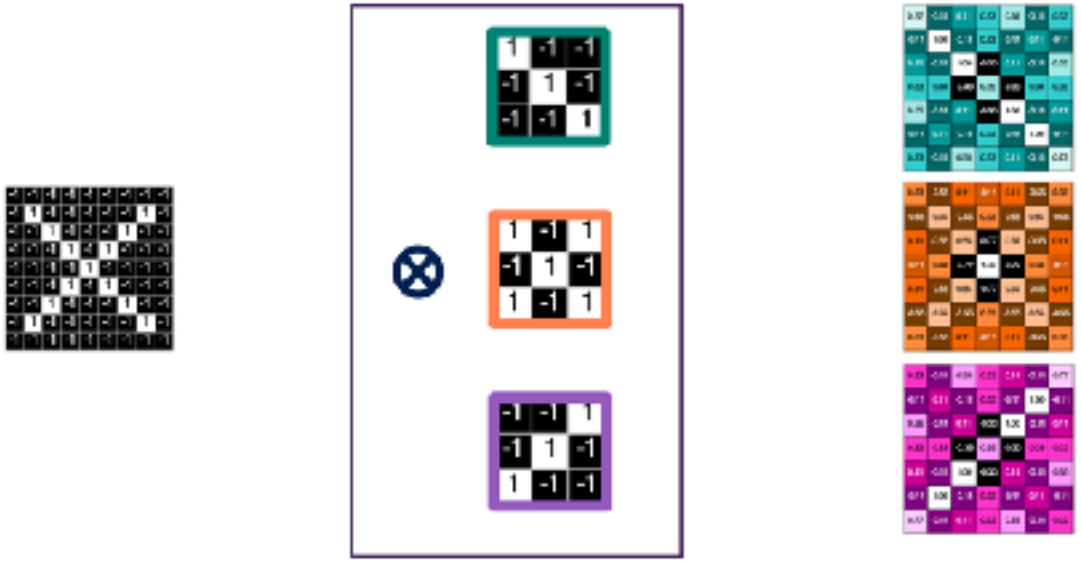


Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Convolution layer

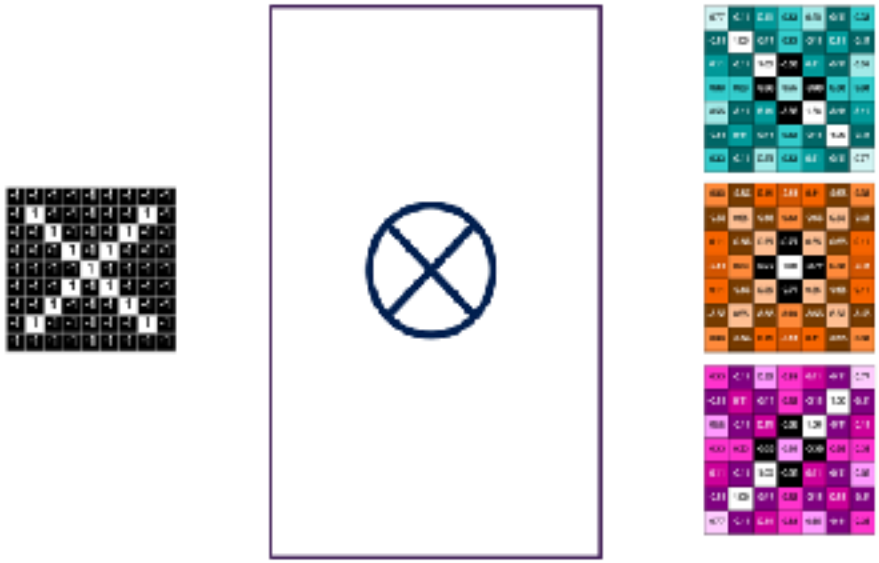
One image becomes a stack of filtered images



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Convolution layer

One image becomes a stack of filtered images



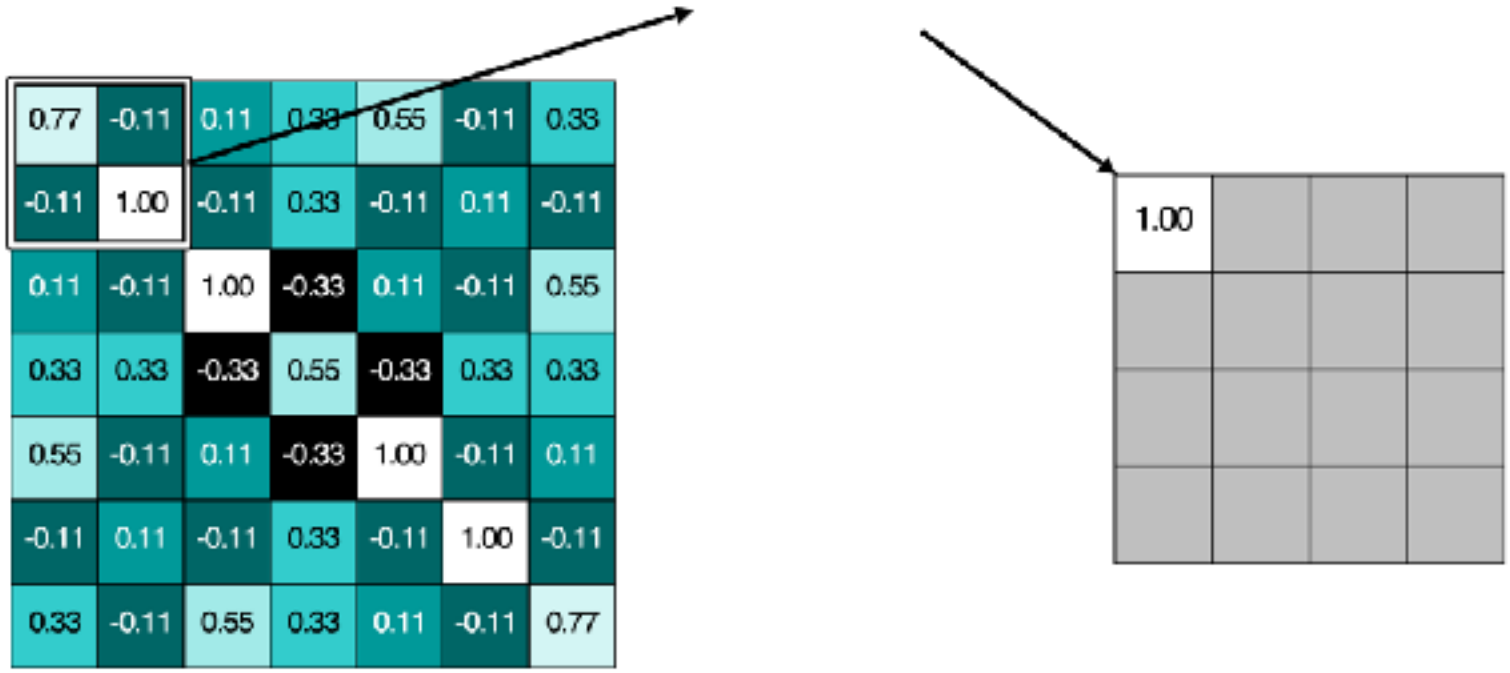
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Pooling: Shrinking the image stack

1. Pick a window size (usually 2 or 3).
2. Pick a stride (usually 2).
3. Walk your window across your filtered images.
4. From each window, take the maximum value.

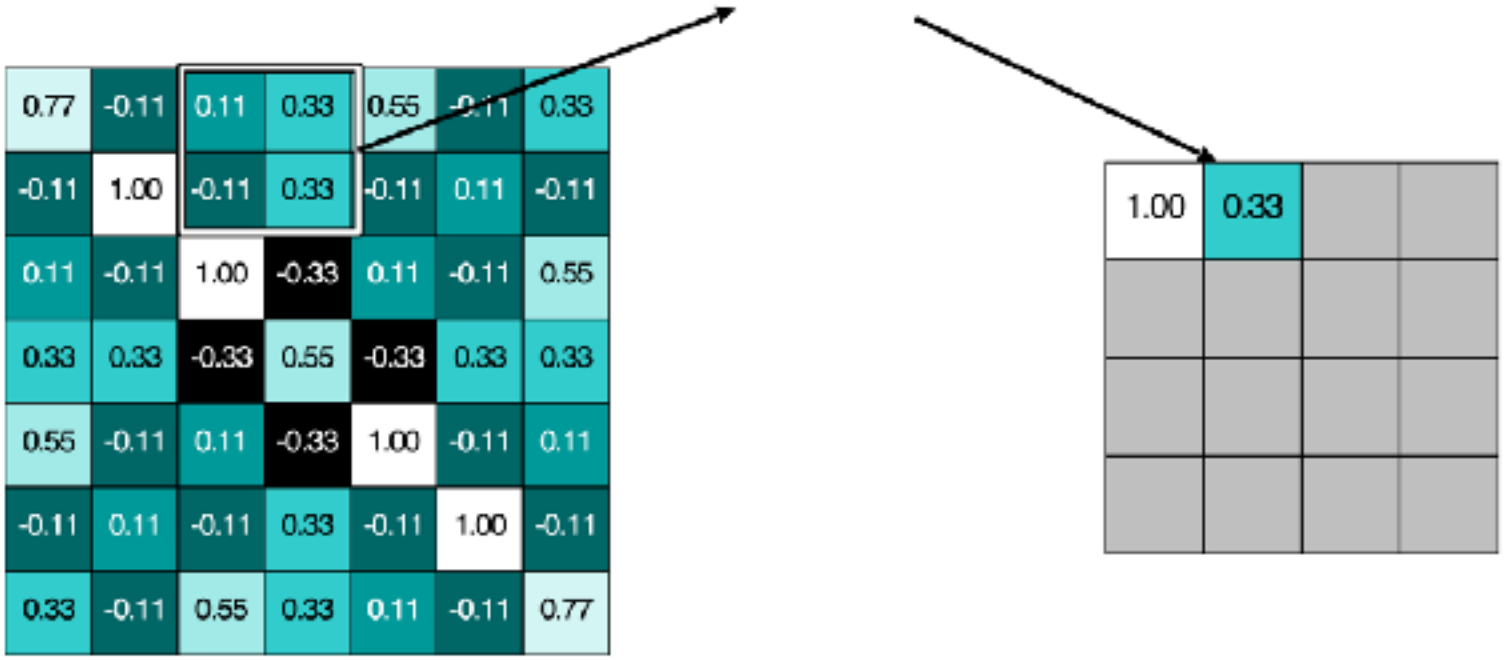
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Pooling



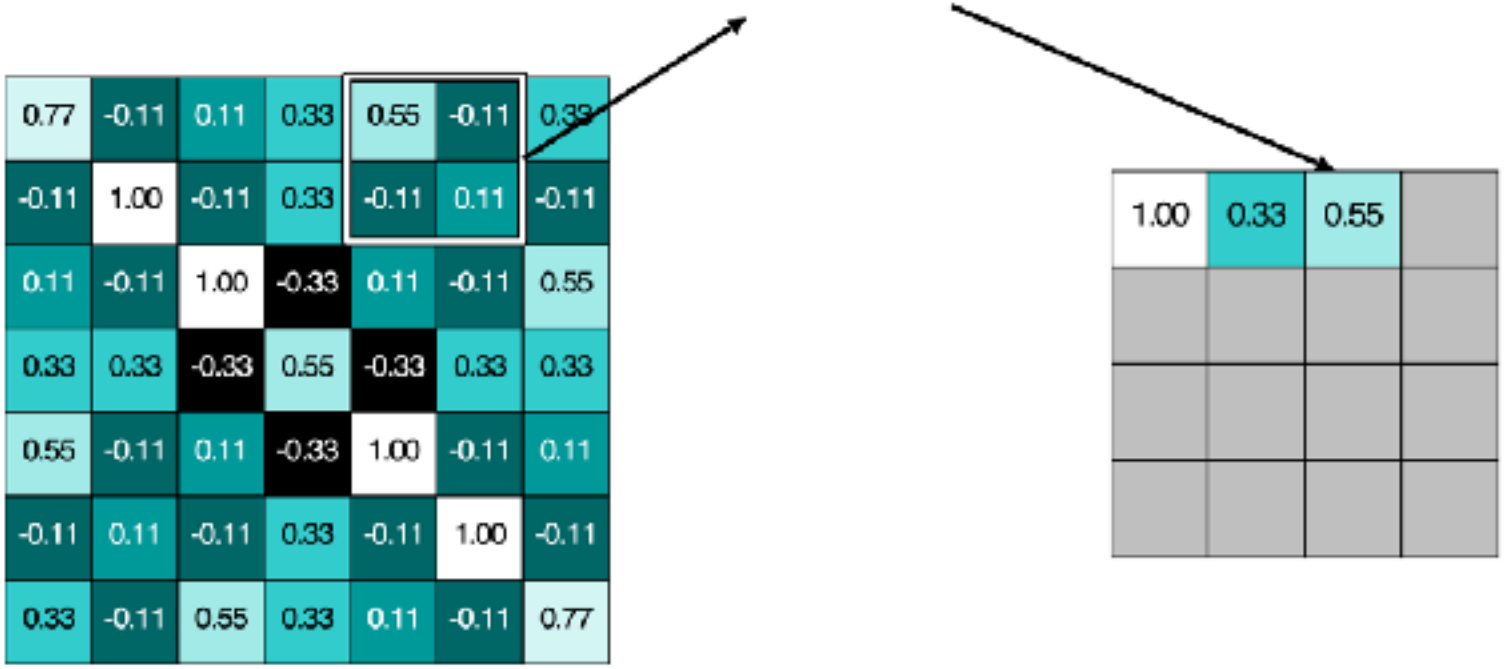
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Pooling



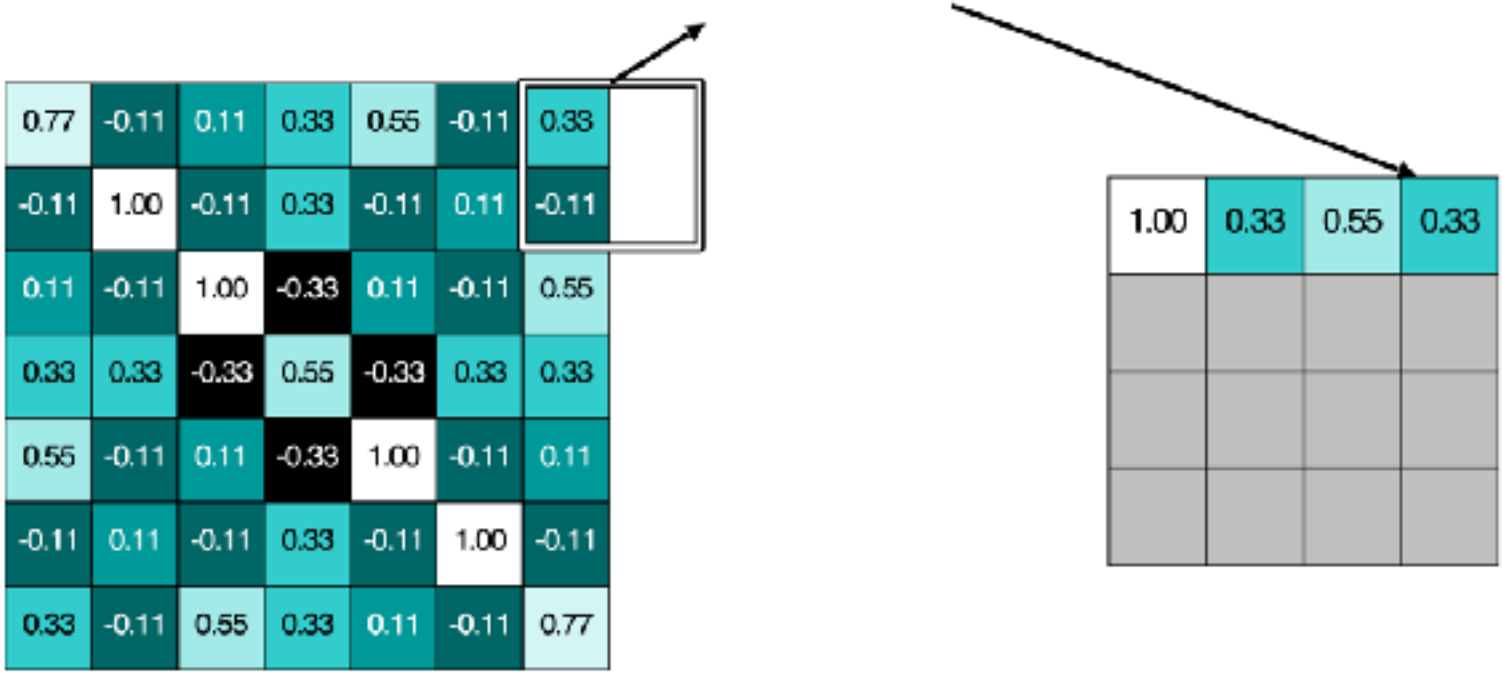
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Pooling



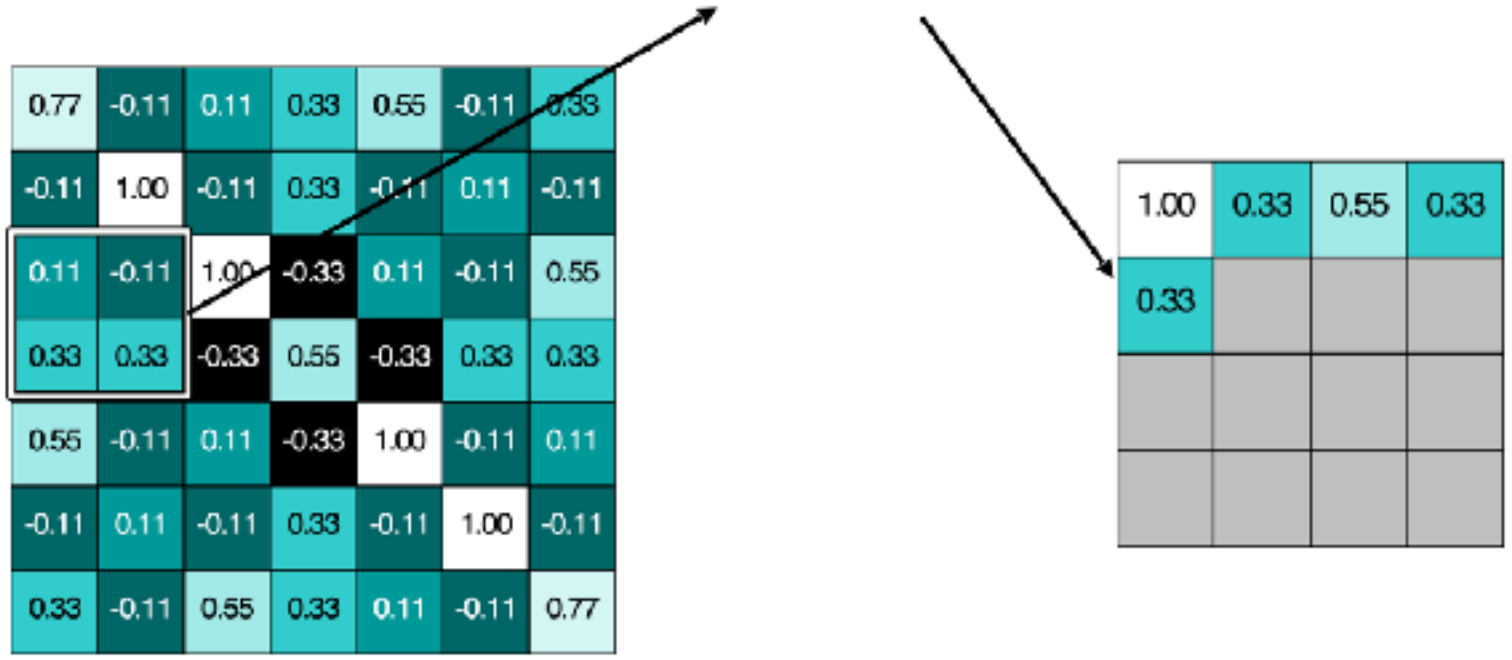
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Pooling



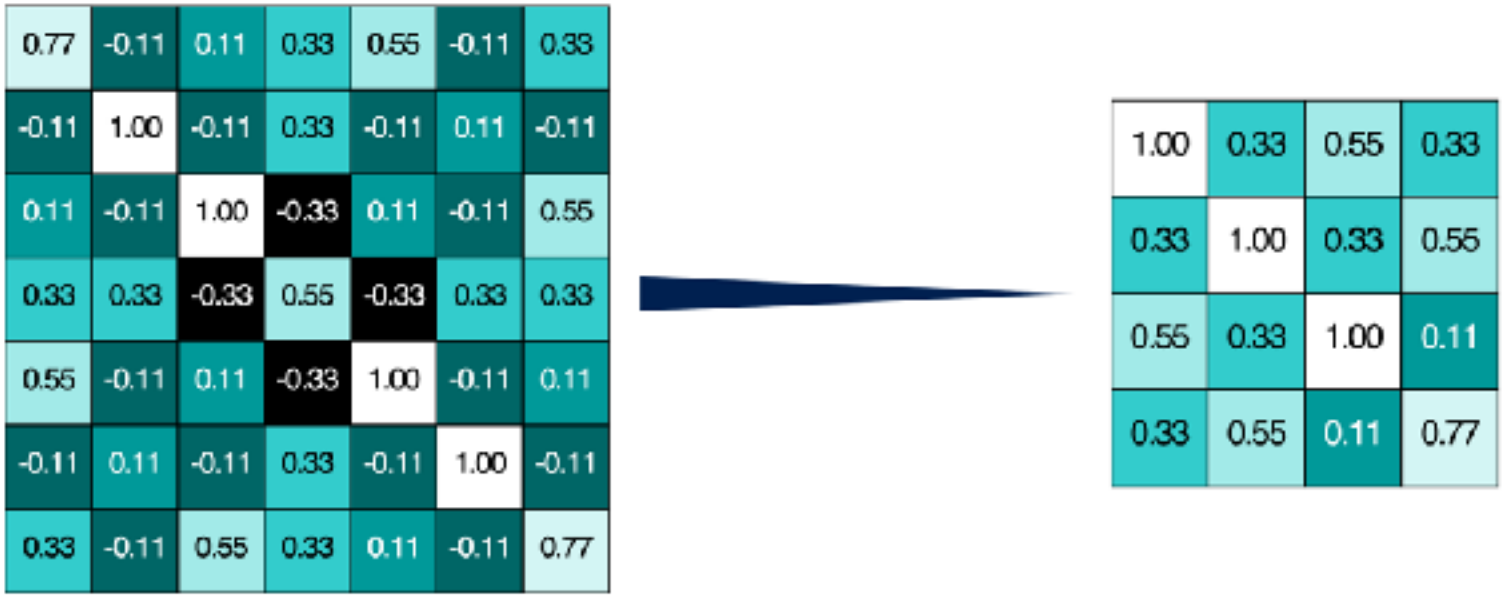
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Pooling

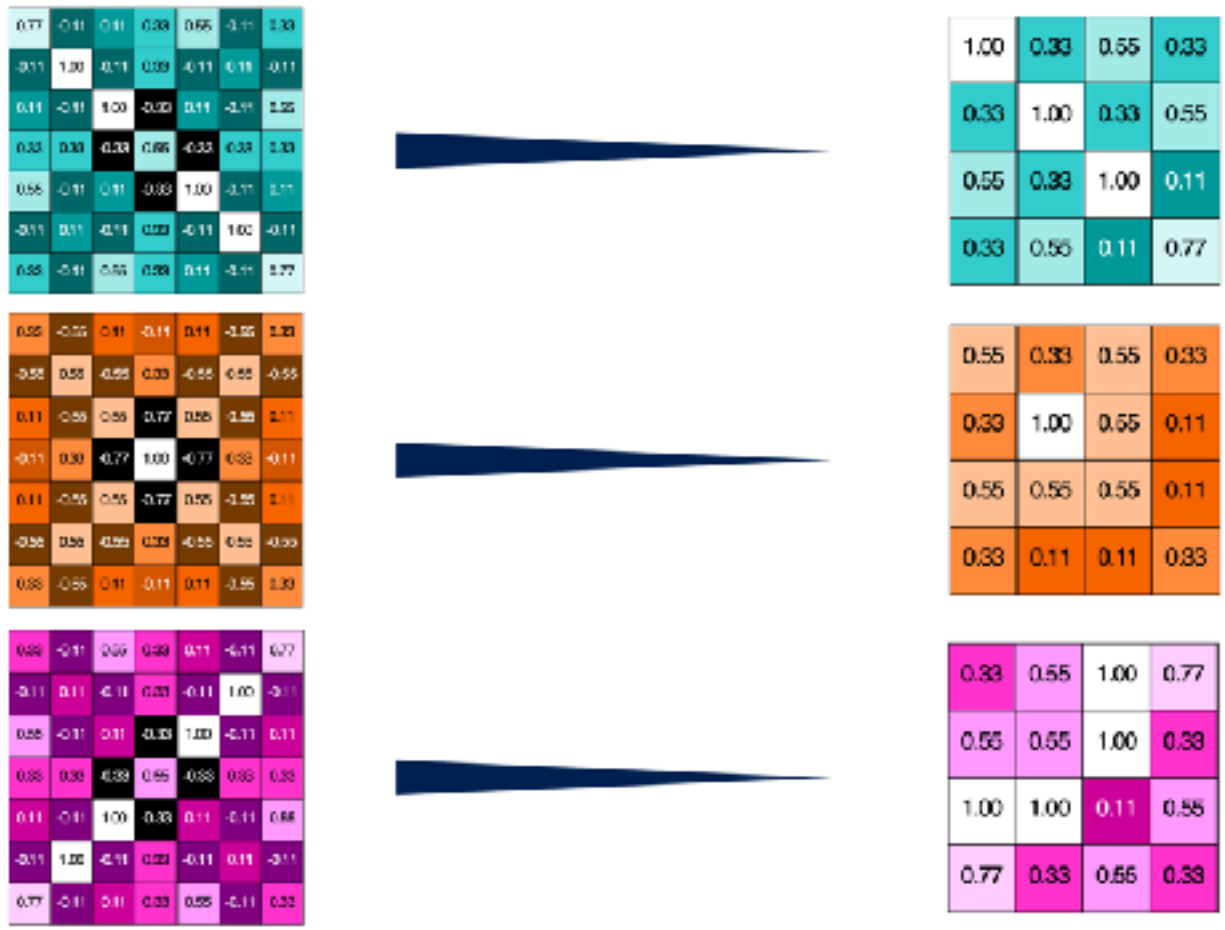


Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Pooling



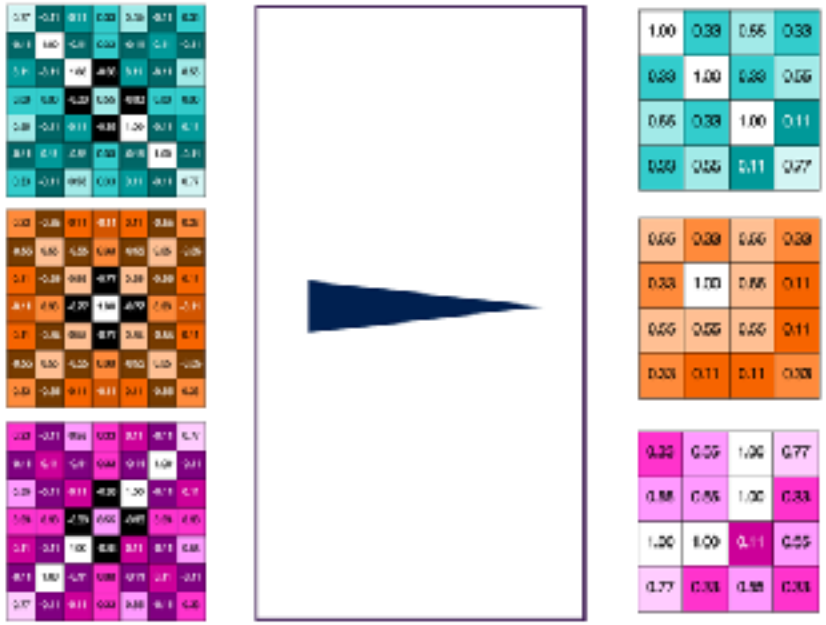
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Pooling layer

A stack of images becomes a stack of smaller images.



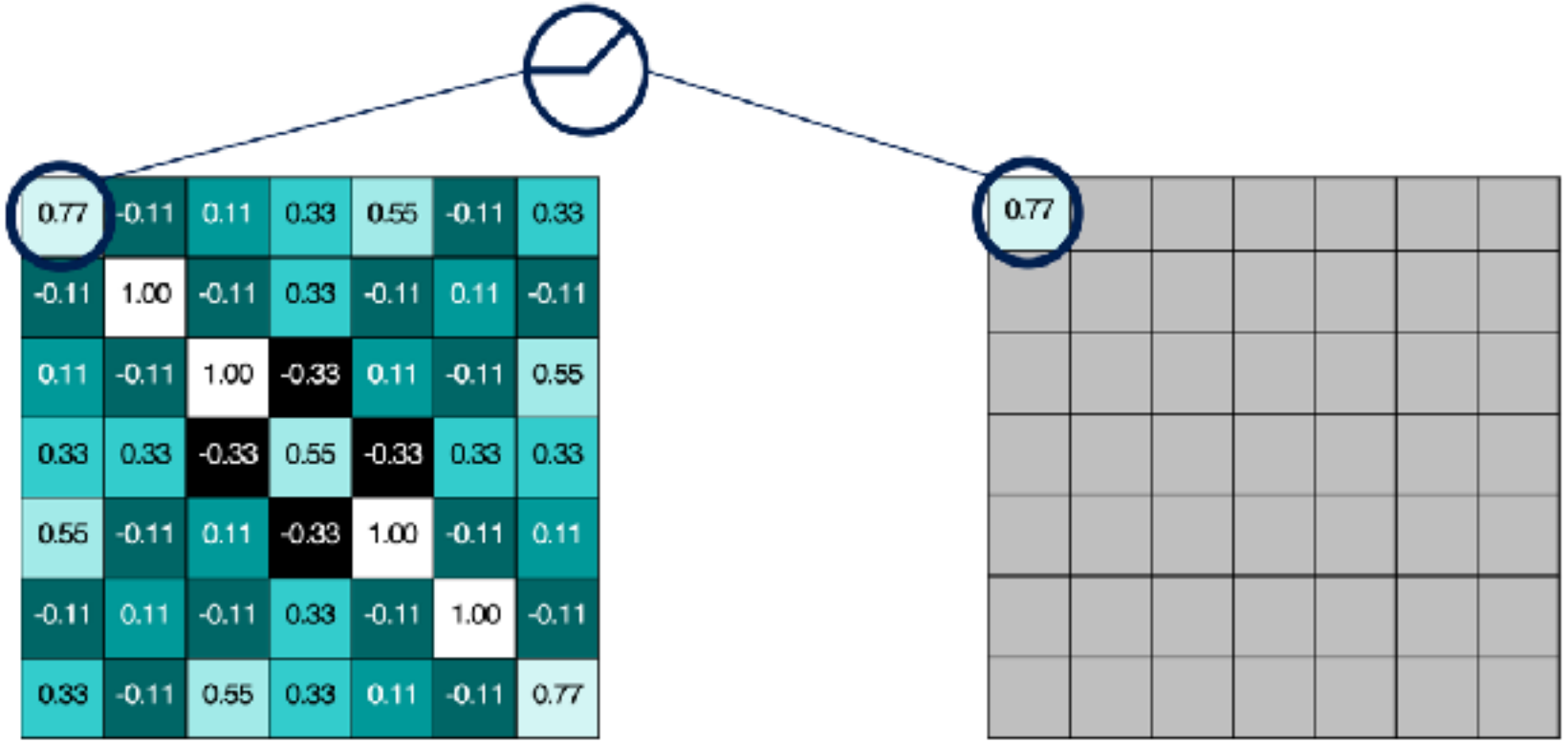
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Normalization

Keep the math from breaking by tweaking each of the values just a bit.

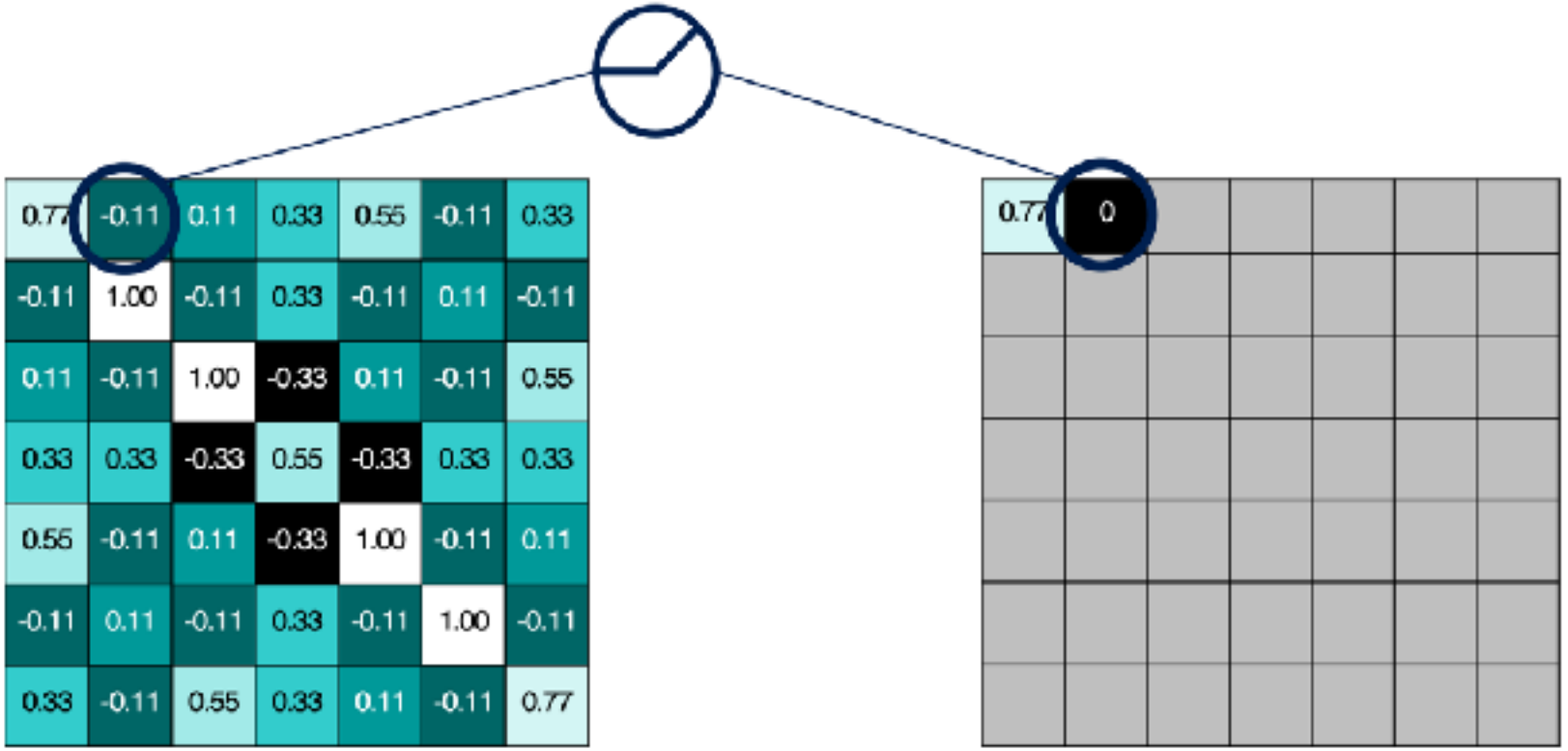
Change everything negative to zero.

Rectified Linear Units (ReLUs)



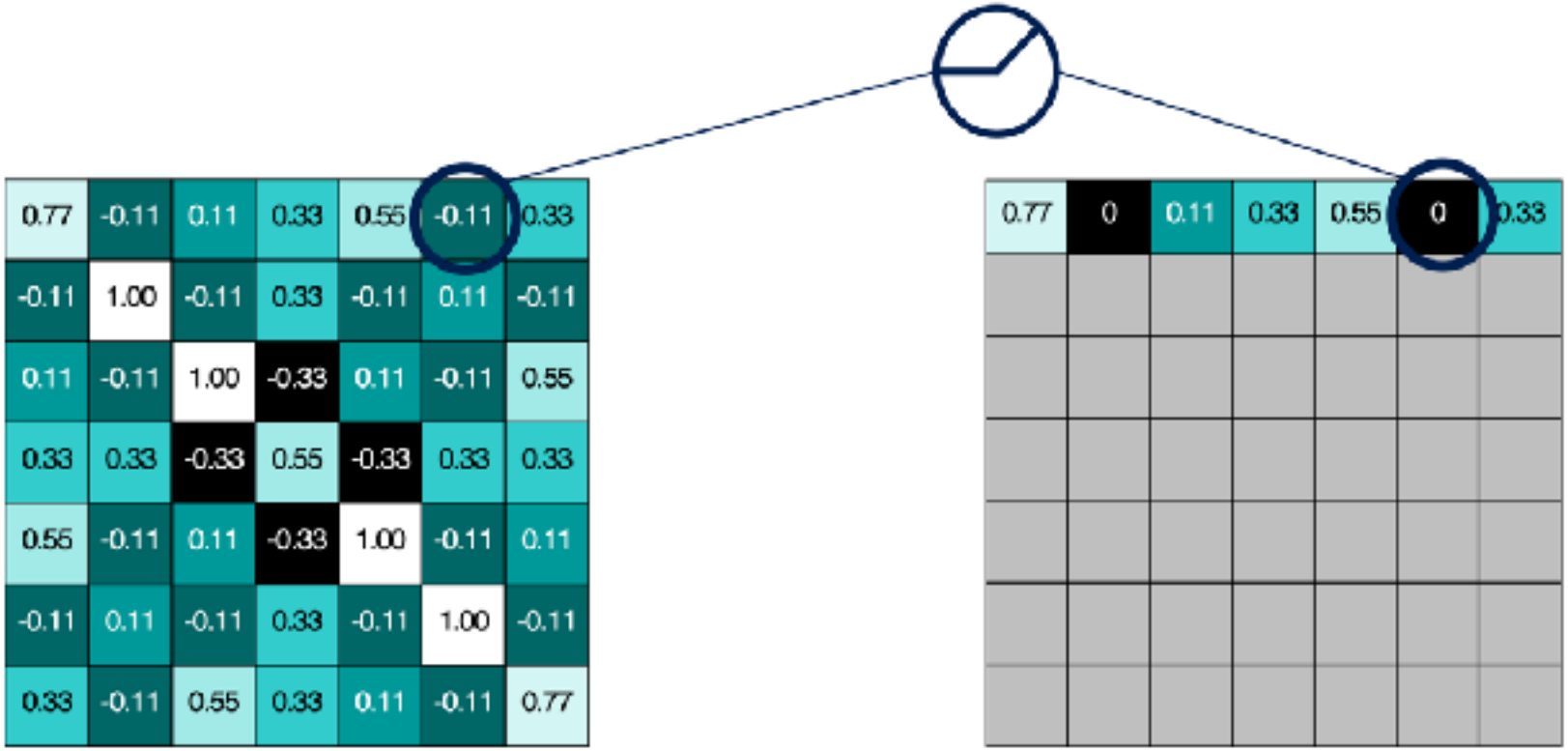
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Rectified Linear Units (ReLUs)



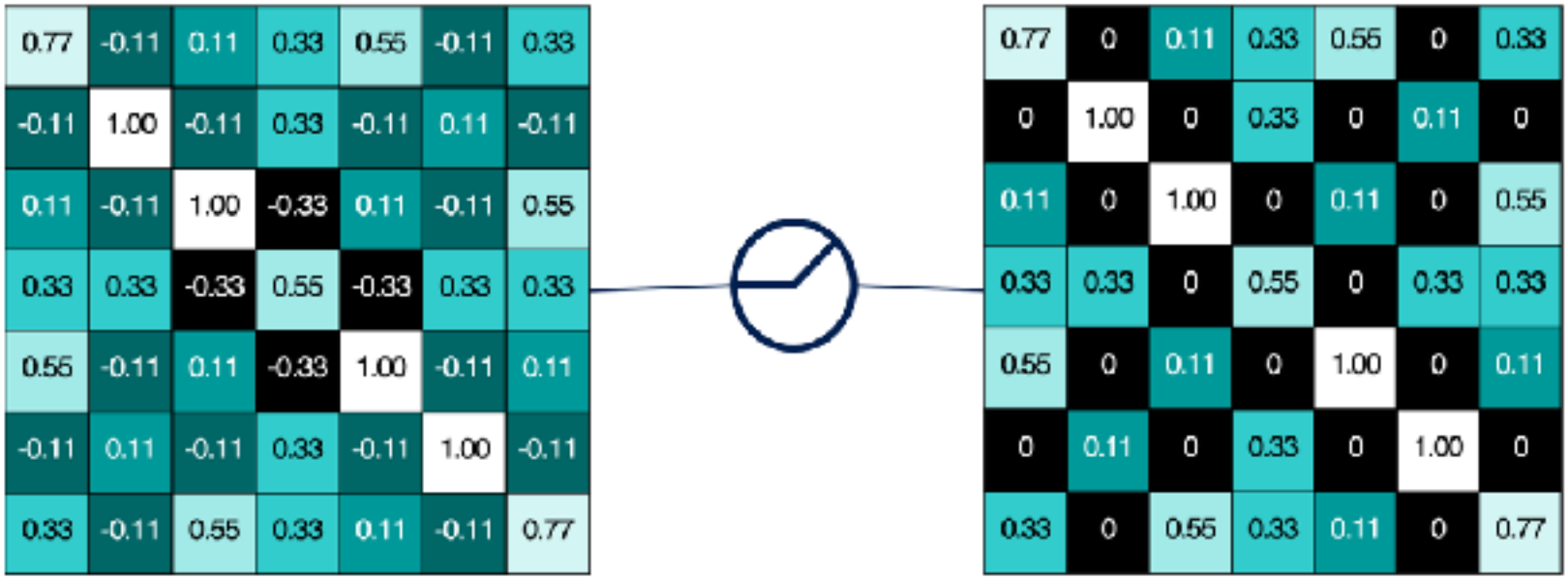
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Rectified Linear Units (ReLUs)



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Rectified Linear Units (ReLUs)



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

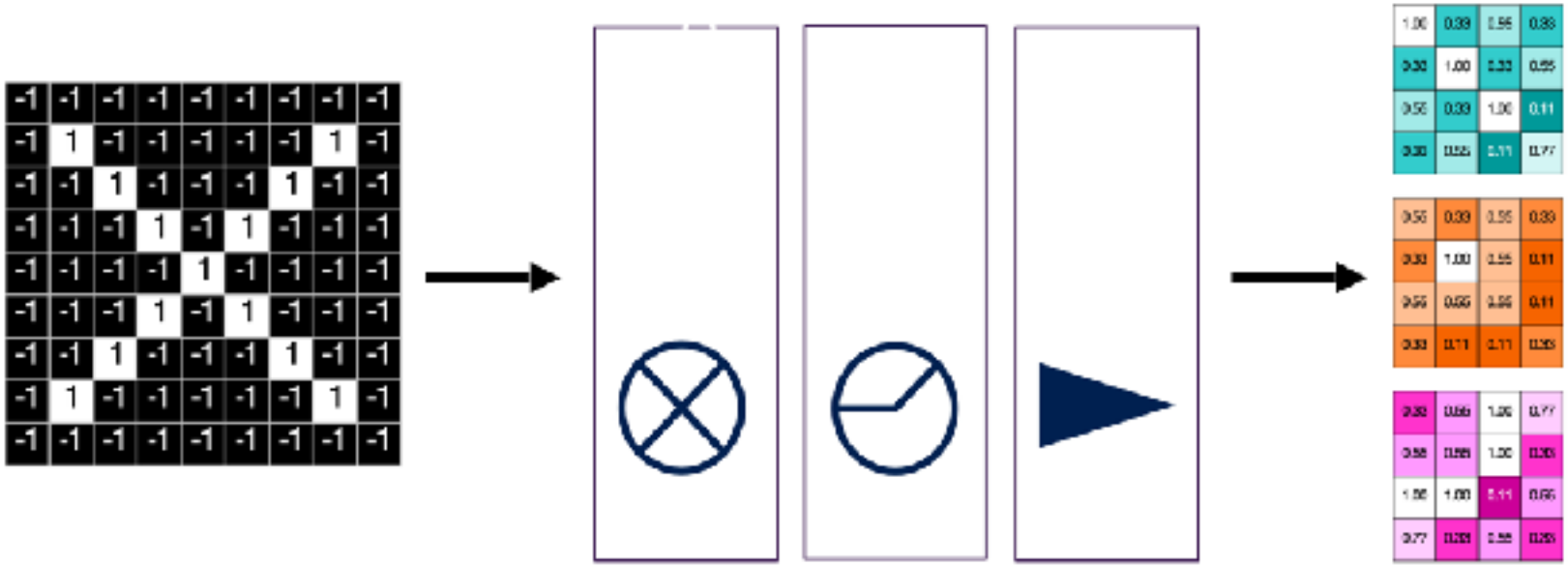
ReLU layer

A stack of images becomes a stack of images with no negative values.

Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Layers get stacked

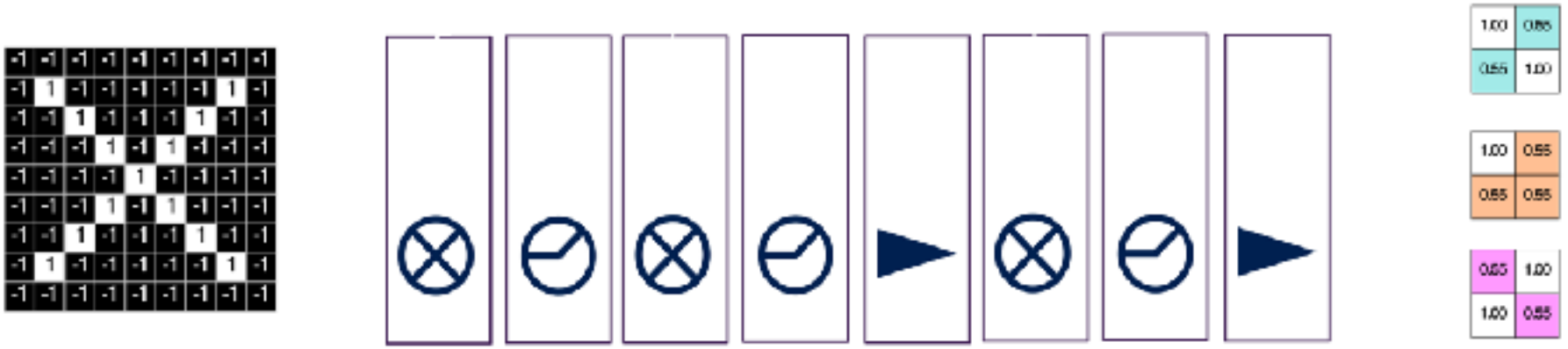
The output of one becomes the input of the next.



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Deep stacking

Layers can be repeated several (or many) times.



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

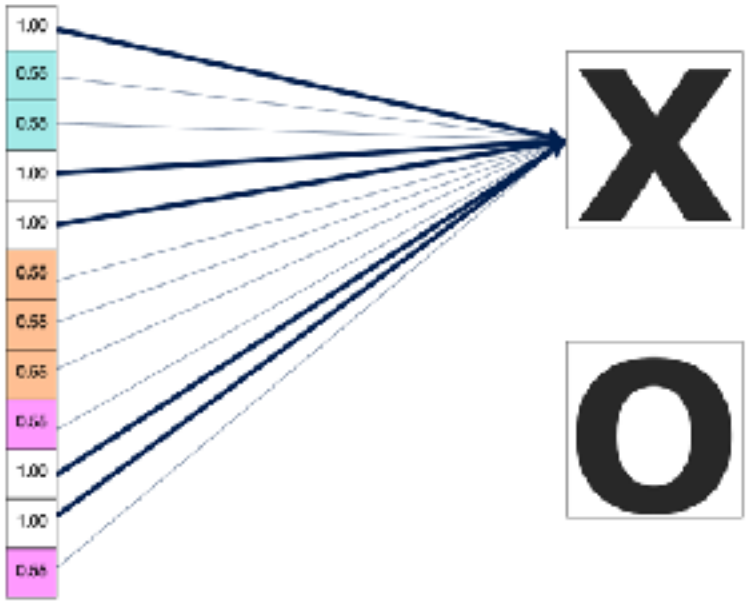
Fully connected layer

Every value gets a vote

Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Fully connected layer

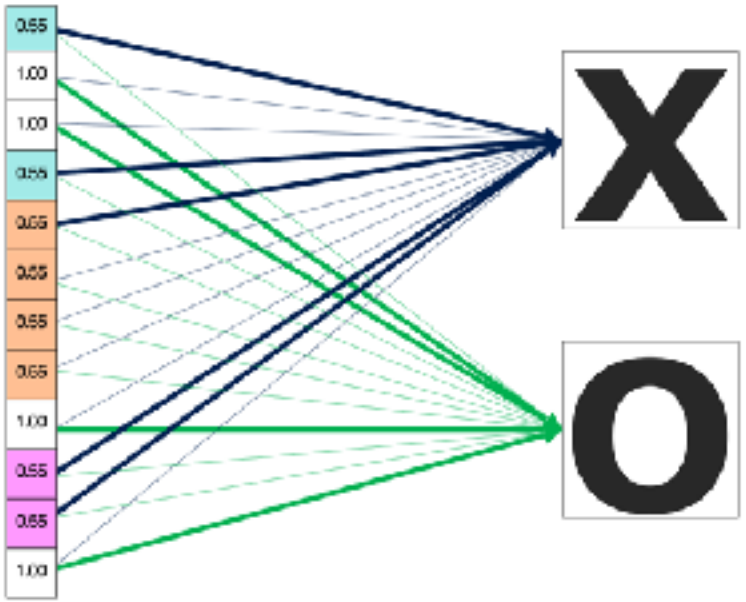
Vote depends on how strongly a value predicts X or O



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Fully connected layer

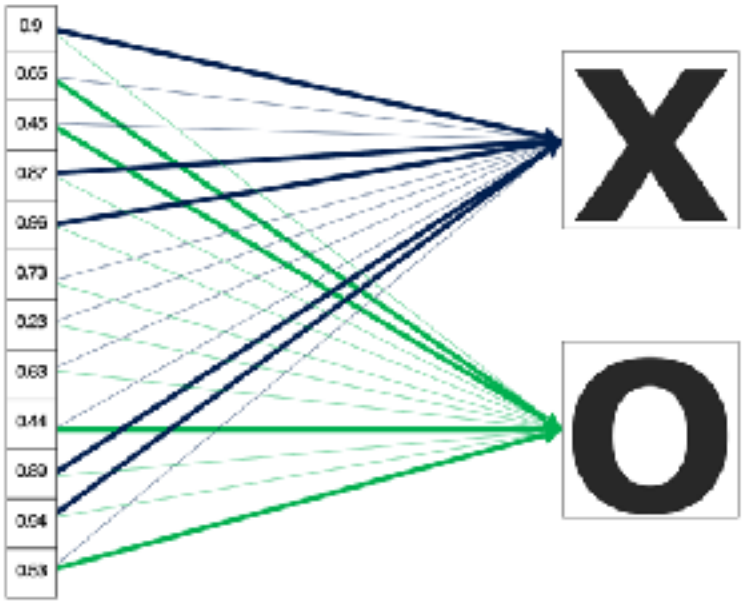
Vote depends on how strongly a value predicts X or O



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Fully connected layer

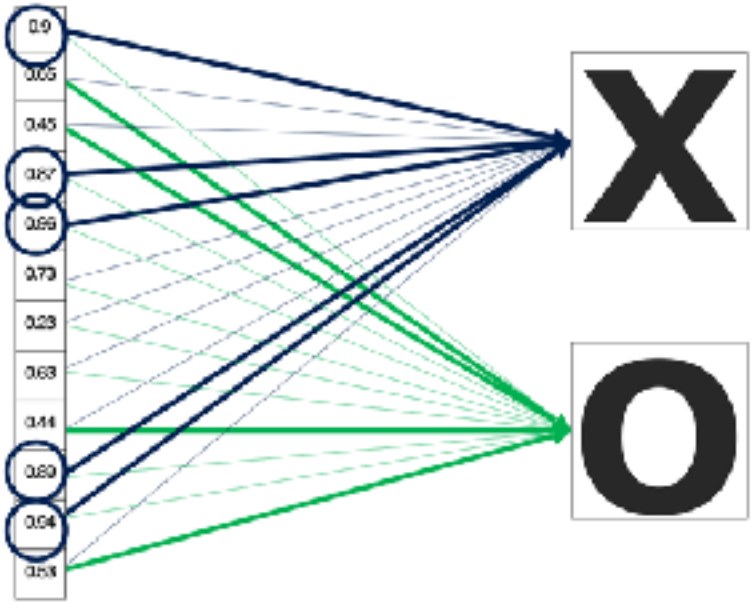
Future values vote on X or O



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Fully connected layer

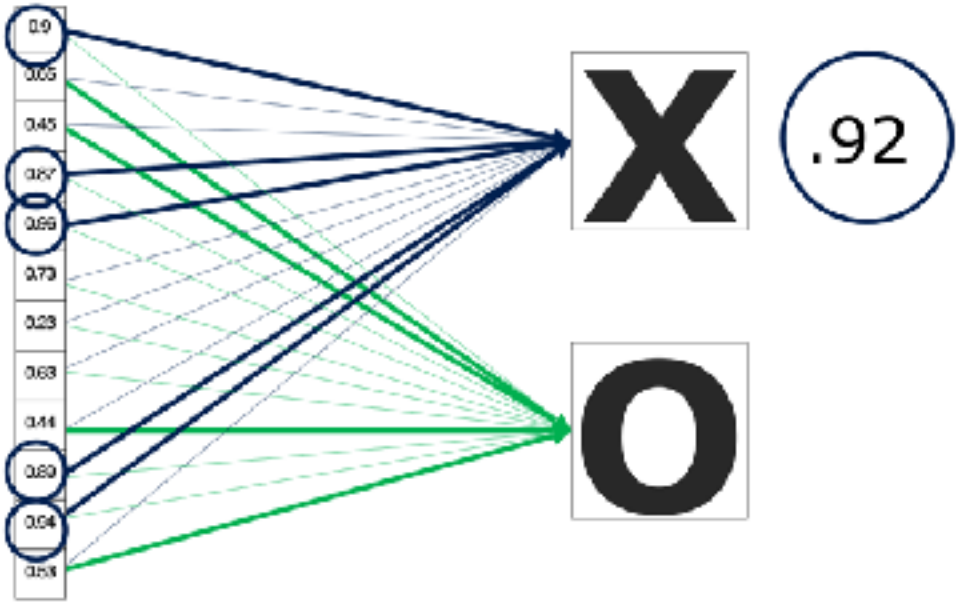
Future values vote on X or O



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Fully connected layer

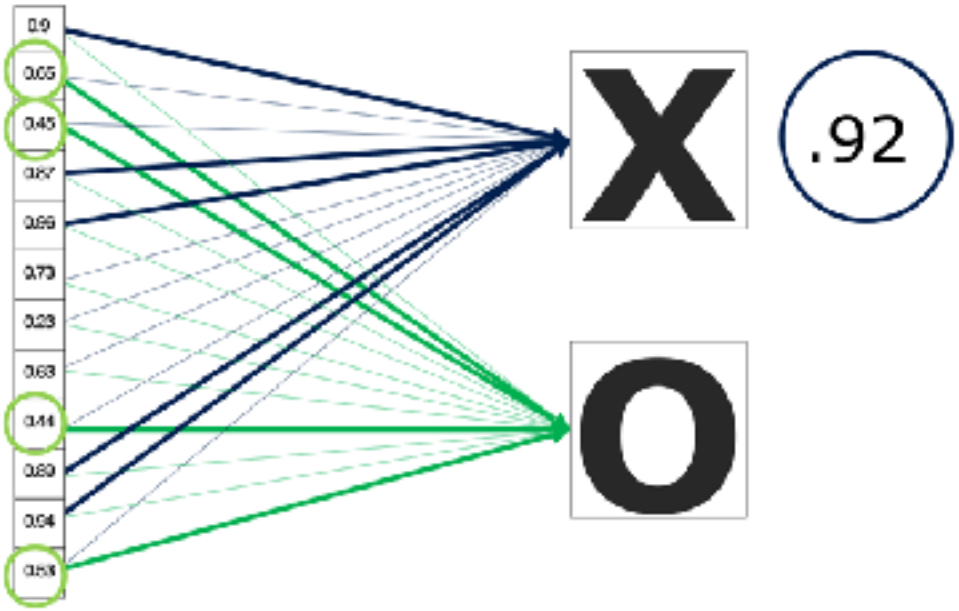
Future values vote on X or O



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Fully connected layer

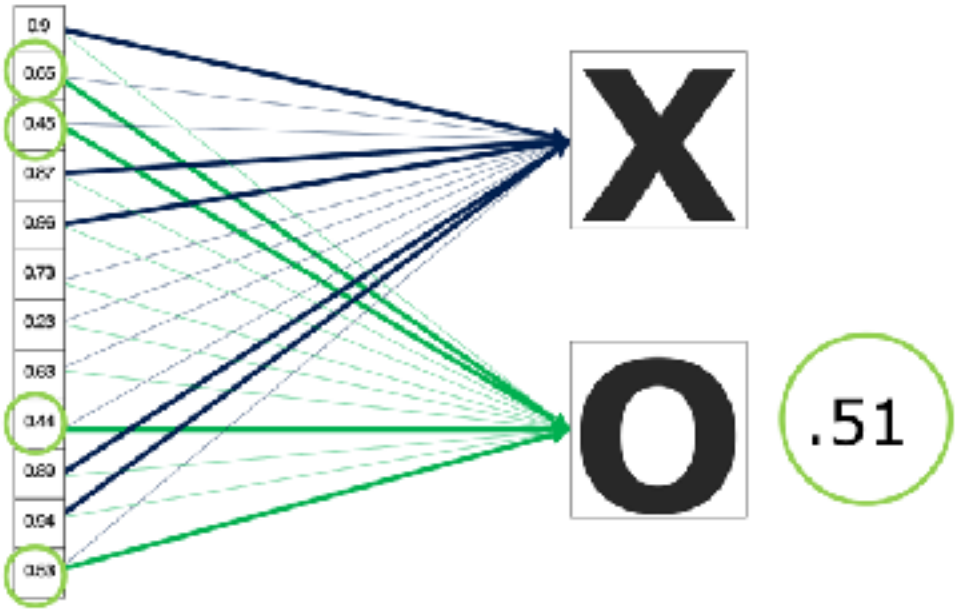
Future values vote on X or O



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Fully connected layer

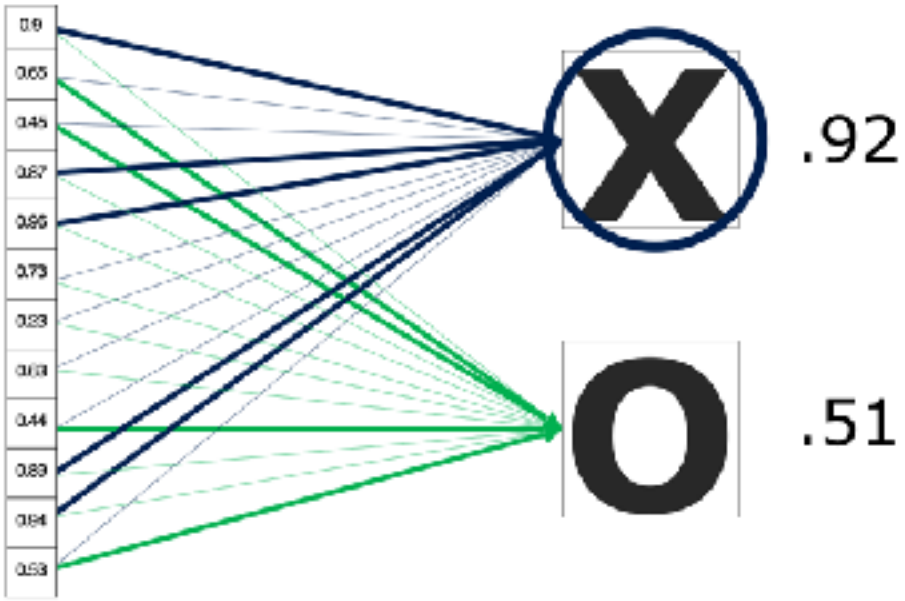
Future values vote on X or O



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Fully connected layer

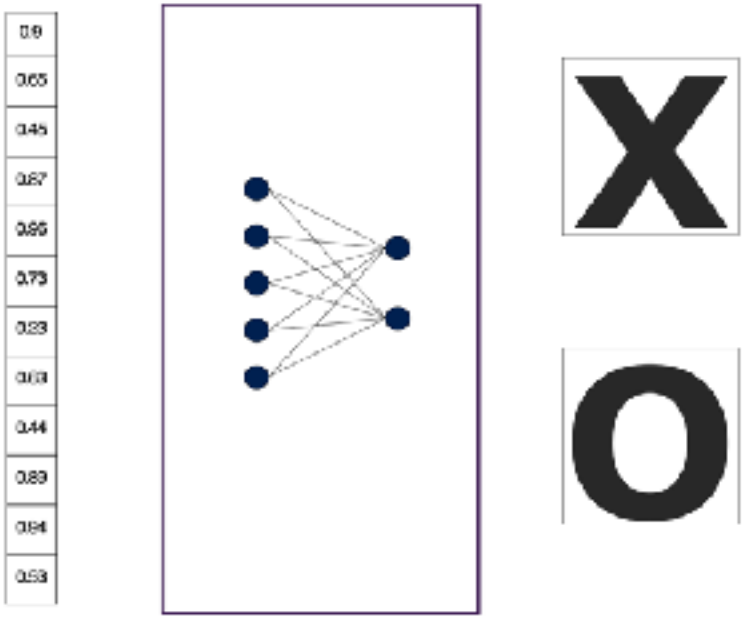
Future values vote on X or O



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Fully connected layer

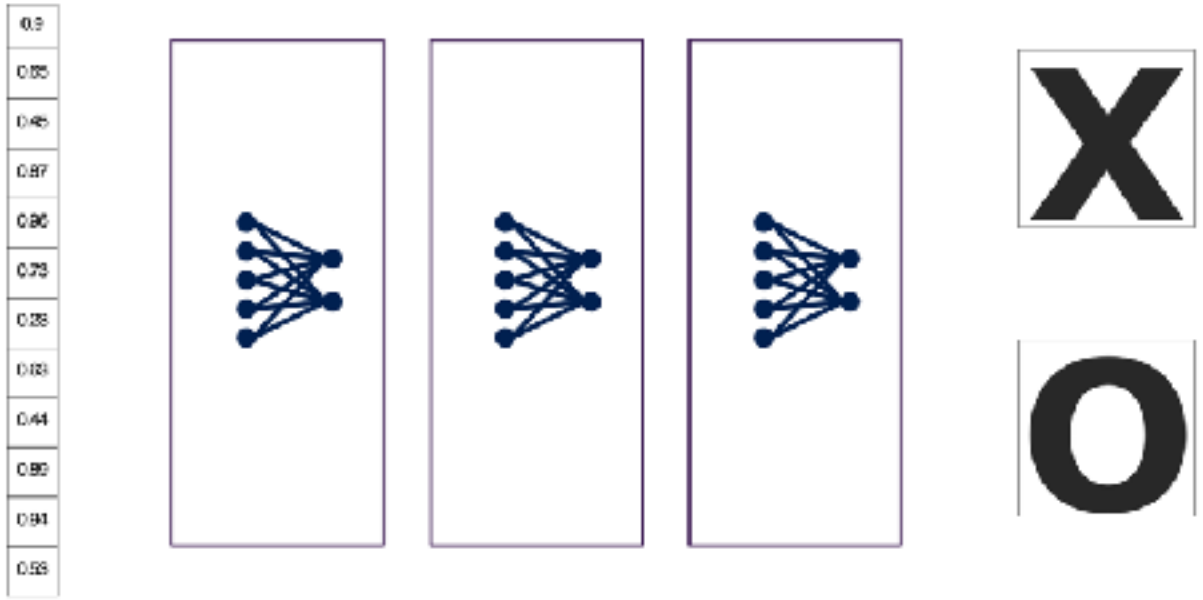
A list of feature values becomes a list of votes.



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Fully connected layer

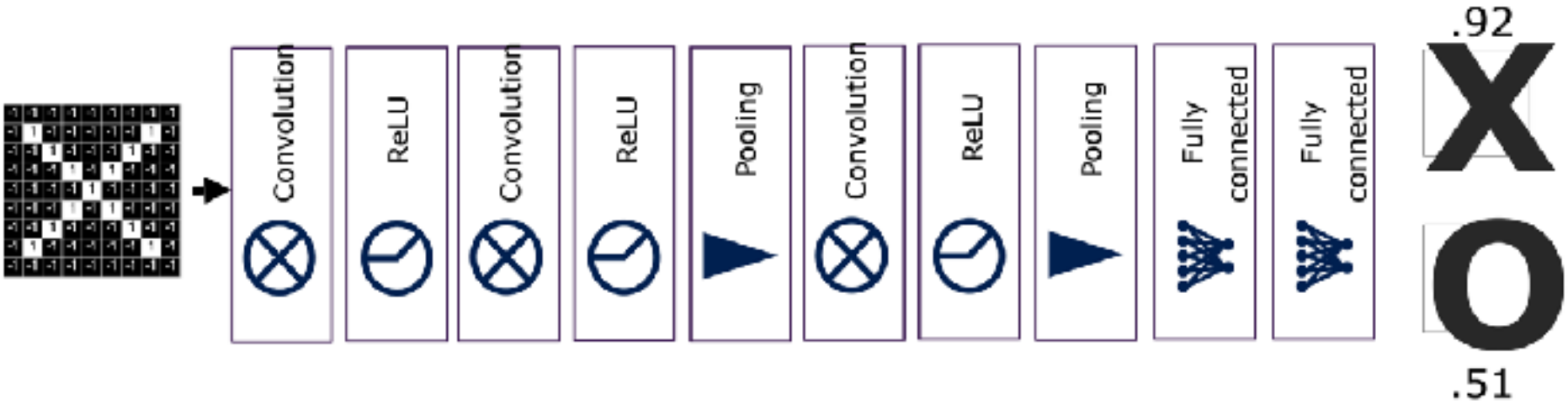
These can also be stacked.



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Putting it all together

A set of pixels becomes a set of votes.



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Learning

Q: Where do all the magic numbers come from?

Features in convolutional layers

Voting weights in fully connected layers

A: Backpropagation

Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Hyperparameters (knobs)

- Convolution
  - Number of features
  - Size of features
- Pooling
  - Window size
  - Window stride
- Fully Connected
  - Number of neurons

Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Architecture

How many of each type of layer?

In what order?

Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

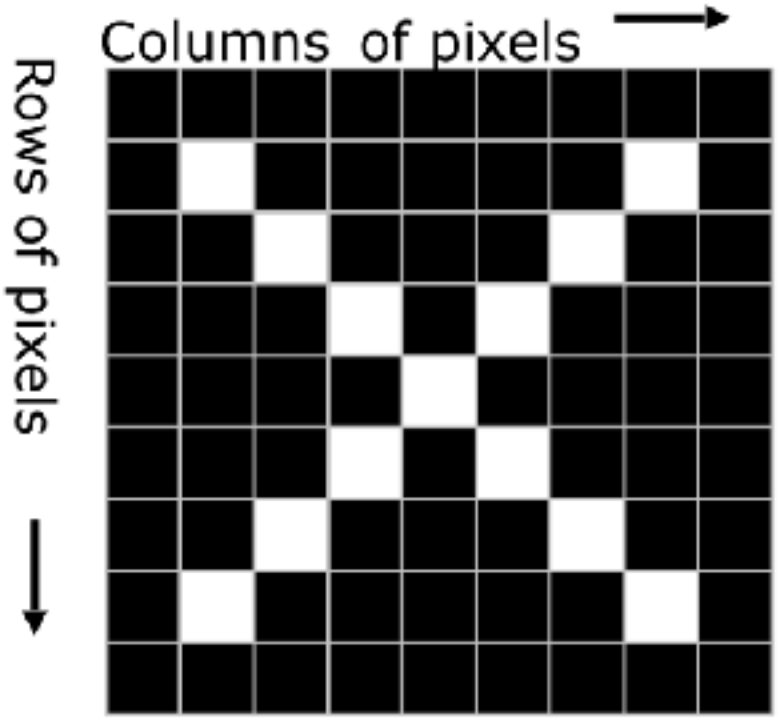
Not just images

Any 2D (or 3D) data.

Things closer together are more closely related than things far away.

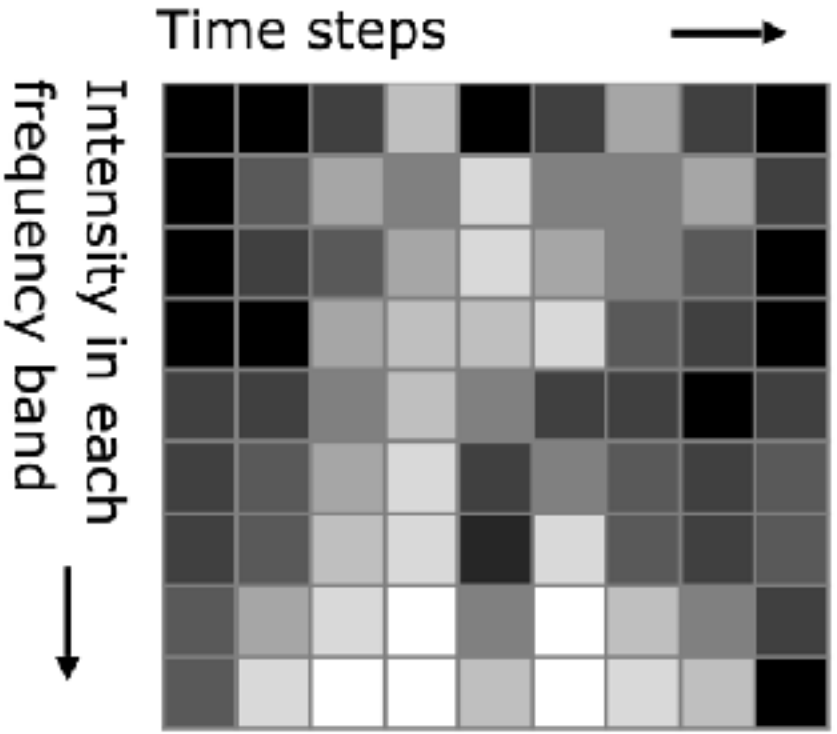
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Images



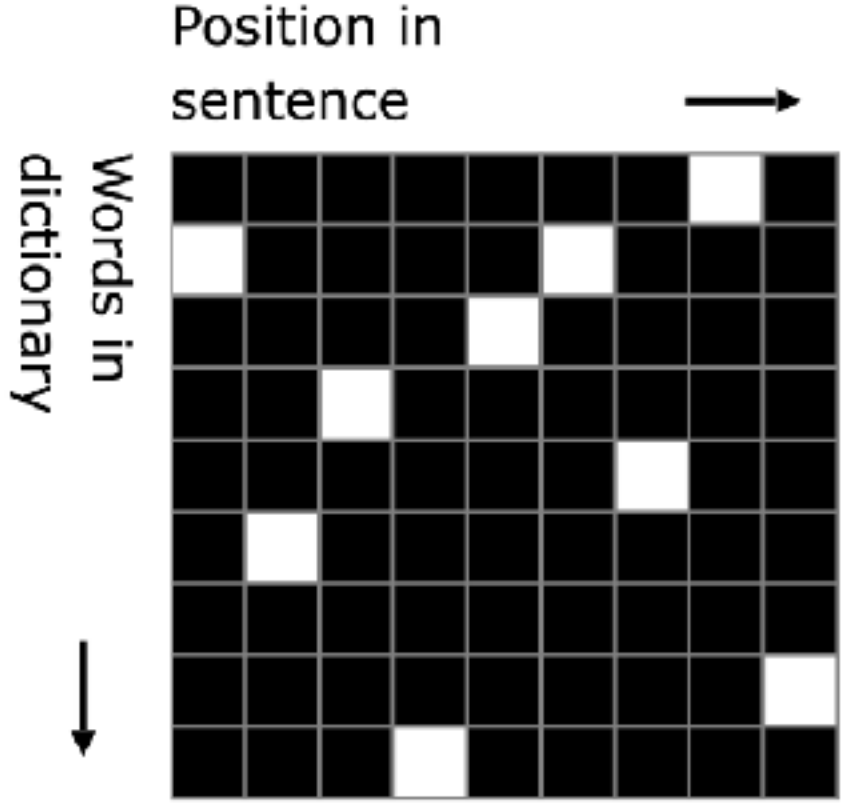
Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Sound



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Text



Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Limitations

ConvNets only capture local “spatial” patterns in data.

If the data can’t be made to look like an image, ConvNets are less useful.

Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Customer data

Name, age,
address, email,
purchases,
browsing activity,…

Customers

A	22	1A	a@a	1	aa	a1.a	123	aa1
B	33	2B	b@b	2	bb	b2.b	234	bb2
C	44	3C	c@c	3	cc	c3.c	345	cc3
D	55	4D	d@d	4	dd	d4.d	456	dd4
E	66	5E	e@e	5	ee	e5.e	567	ee5
F	77	6F	f@f	6	ff	f6.f	678	ff6
G	88	7G	g@g	7	gg	g7.g	789	gg7
H	99	8H	h@h	8	hh	h8.h	890	hh8
I	111	9I	i@i	9	ii	i9.i	901	ii9

Brandon Rohrer https://brohrer.github.io/how_convolutional_neural_networks_work.html

Readings

- DL Book: Convolutional Networks
  - https://www.deeplearningbook.org/contents/convnets.html