DATASCI 347 Machine Learning

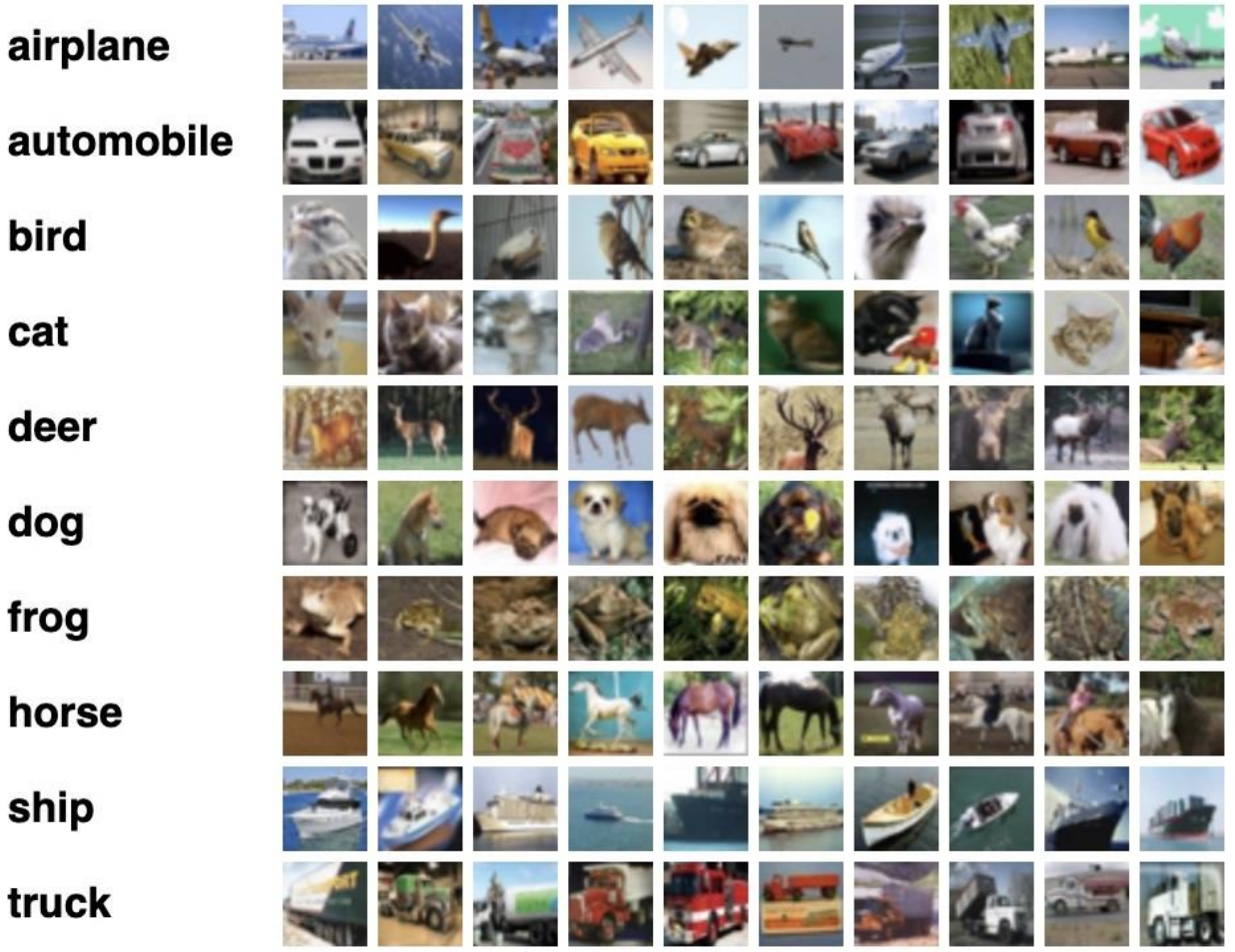
Lecture 20: Convolutional Neural Networks

Ruoxuan Xiong

Suggested reading: ISL Chapter 10

Object recognition

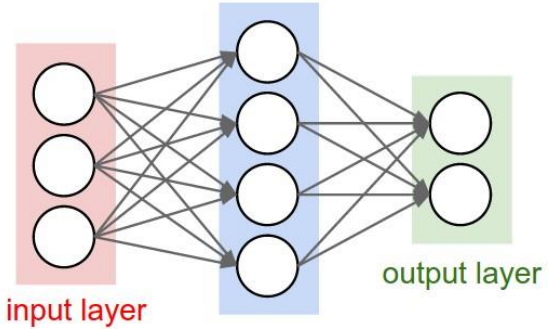
- CIFAR-10



Object recognition

- ImageNet (1000 classes)

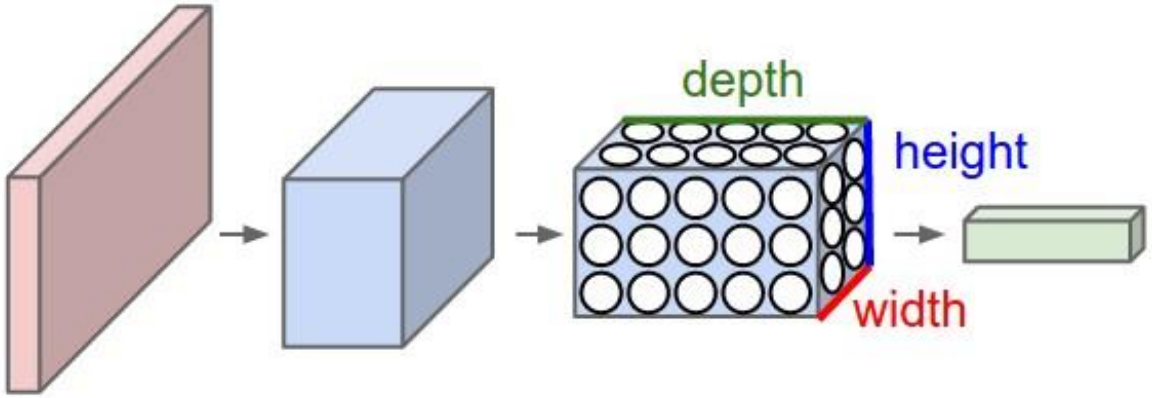
Issues of using feedforward neural networks for large images



- Feedforward neural networks use fully-connected layers to transform the input
- Fully-connected layers do not scale to large images
  - A black-and-white digit in MNIST has size 28 by 28. A colored image in CIFAR-10 has size 32 by 32 by 3
  - For MNIST, a fully-connected neuron needs $28 \times 28 = 784$ weights
  - For CIFAR-10, a fully-connected neuron needs $32 \times 32 \times 3 = 3{,}072$ weights
  - Processing larger images requires more parameters

CNN only uses local connections

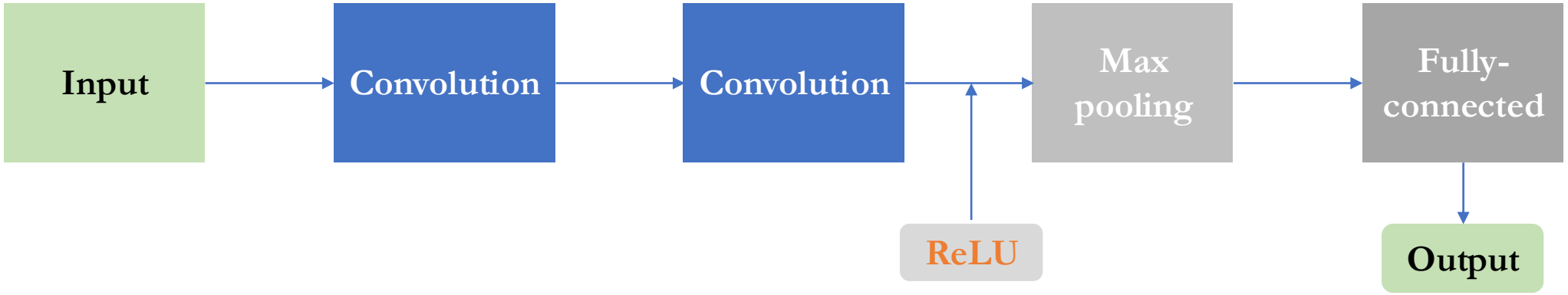
- In convolutional neural networks (CNN), a neuron only connects to a small local region of the image
  - Example: A colored (2D) image is specified by width, height, and depth

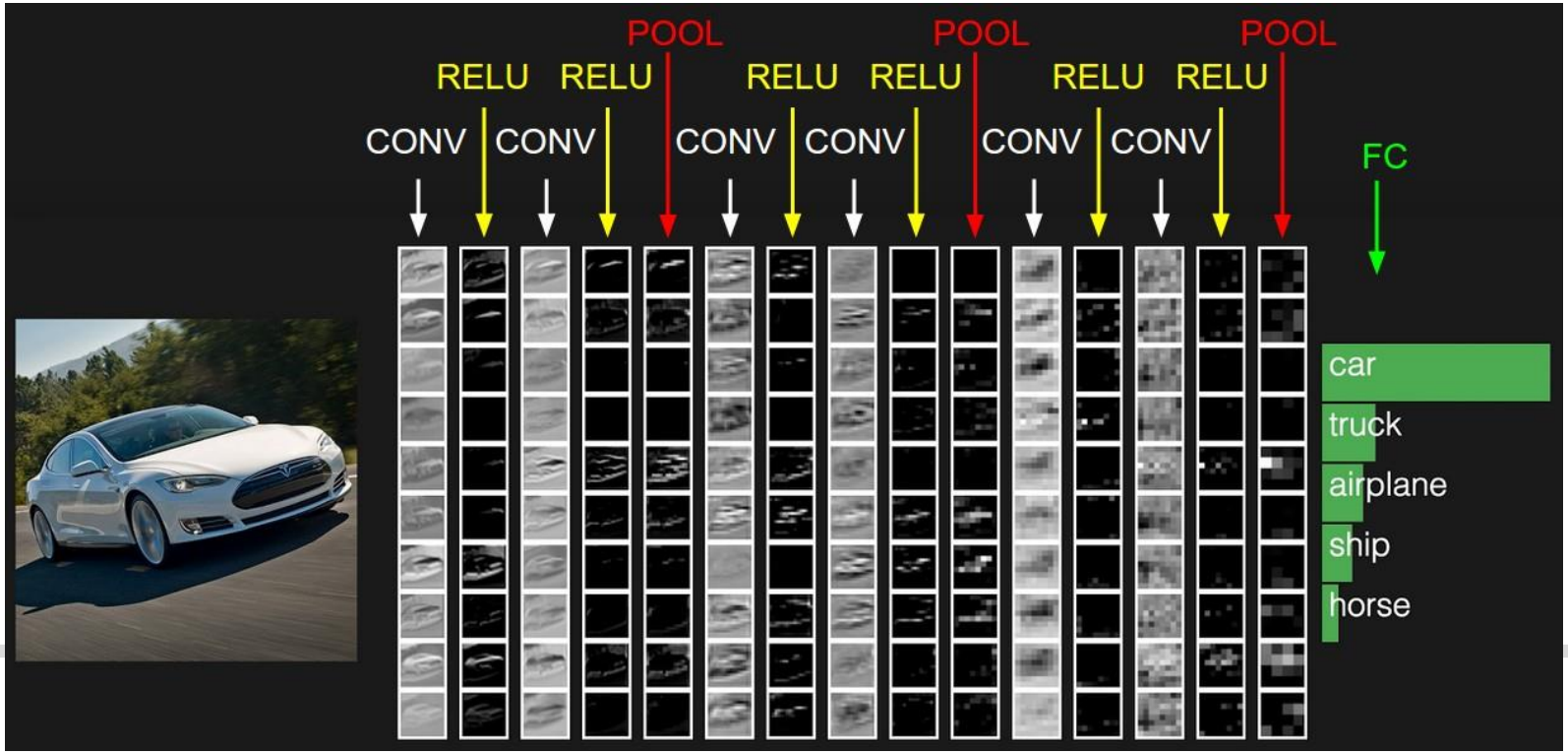


Types of layers

- A CNN involves a combination of the following types of layers
  - **Input layer:** Raw pixel values of the image
  - **Convolution layer:** Combine pixel values in a local region
  - **Pooling layer:** Down sample pixels
  - **Fully-connected layers:** Classification/prediction

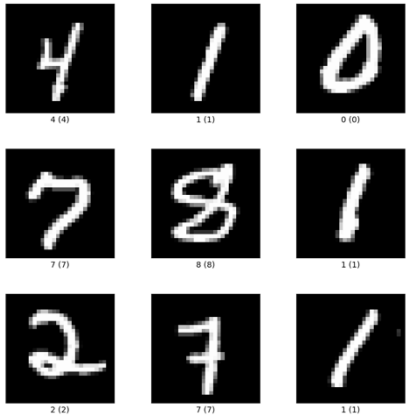
Illustration of CNN architectures





Convolution layer

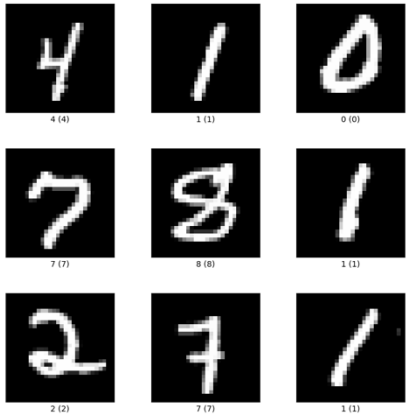
- **Example** (MNIST)
  - **Input** size: 28 by 28
  - **Convolutional layer:**
    - Filter size: (3, 3)
      - Looks at 3x3 pixels at a time. This local region is called receptive field
      - Contains learnable weights
    - Stride: (1, 1)
      - Moves 1 pixel at a time horizontally and moves 1 pixels at a time vertically
    - Zero padding size: 0
      - Padding controls whether you add extra pixels around the border (usually 0)
      - Padding is added to left, right, top and bottom

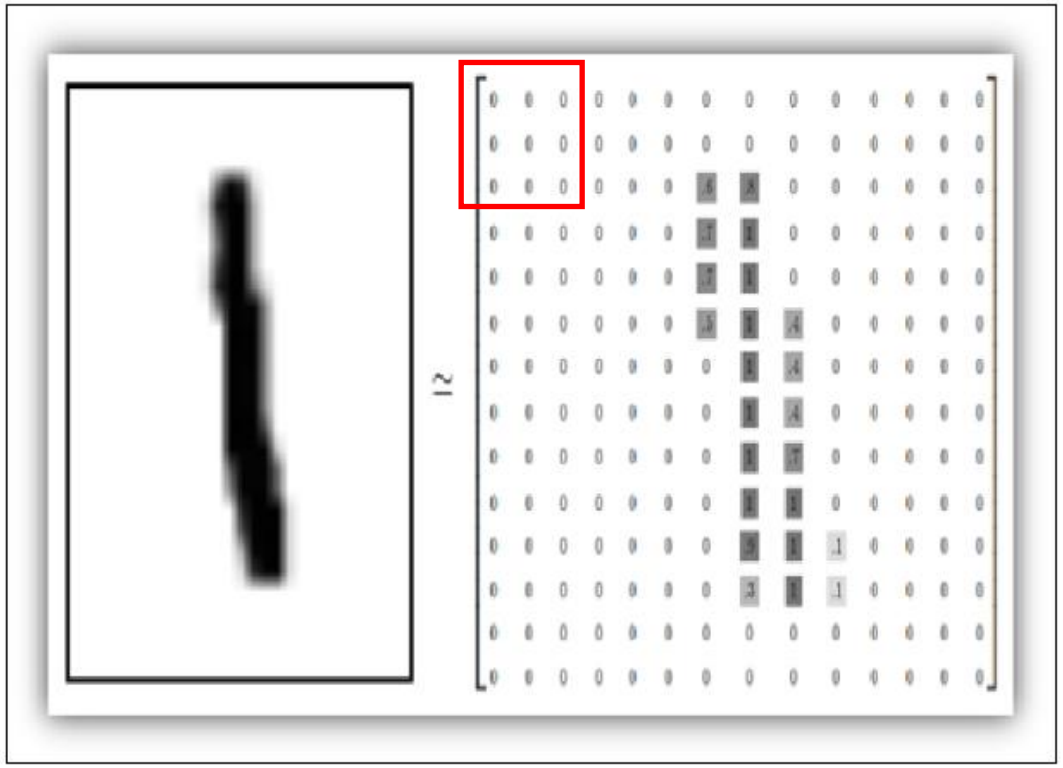


$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}$$

Convolution layer

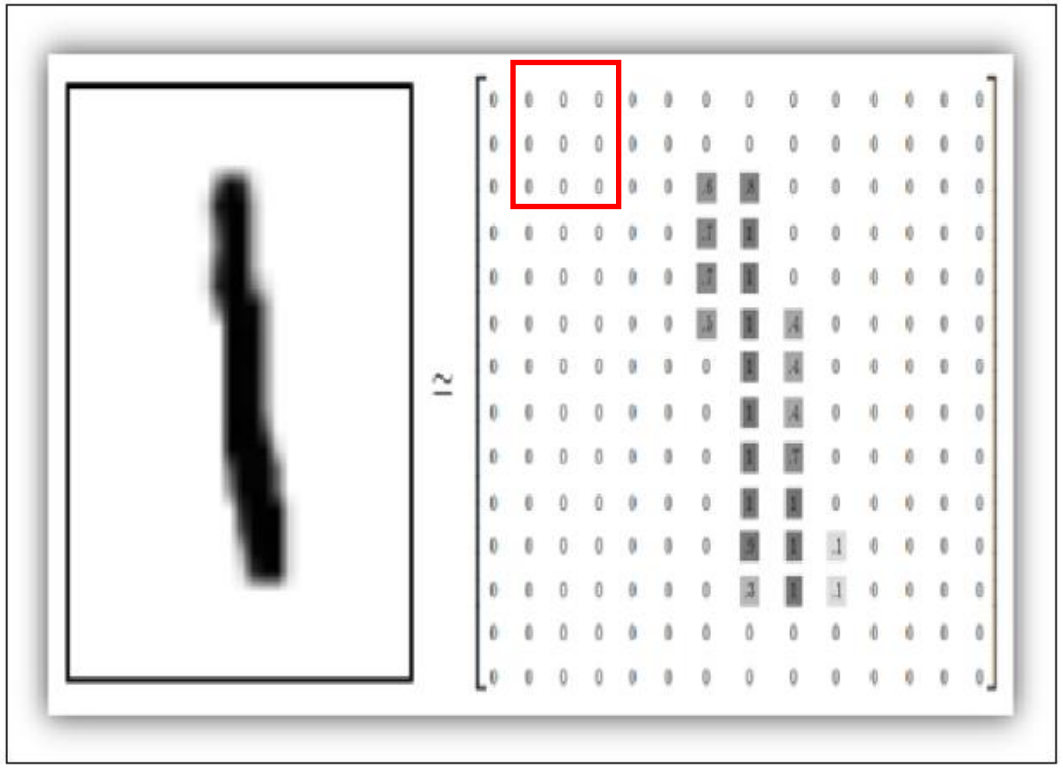
- **Example** (MNIST)
  - **Input** size: 28 by 28
  - **Convolutional layer:**
    - Filter size: (3, 3)
    - Stride: (1, 1)
    - Zero padding size: 0
  - **First row, first patch**





Convolution layer

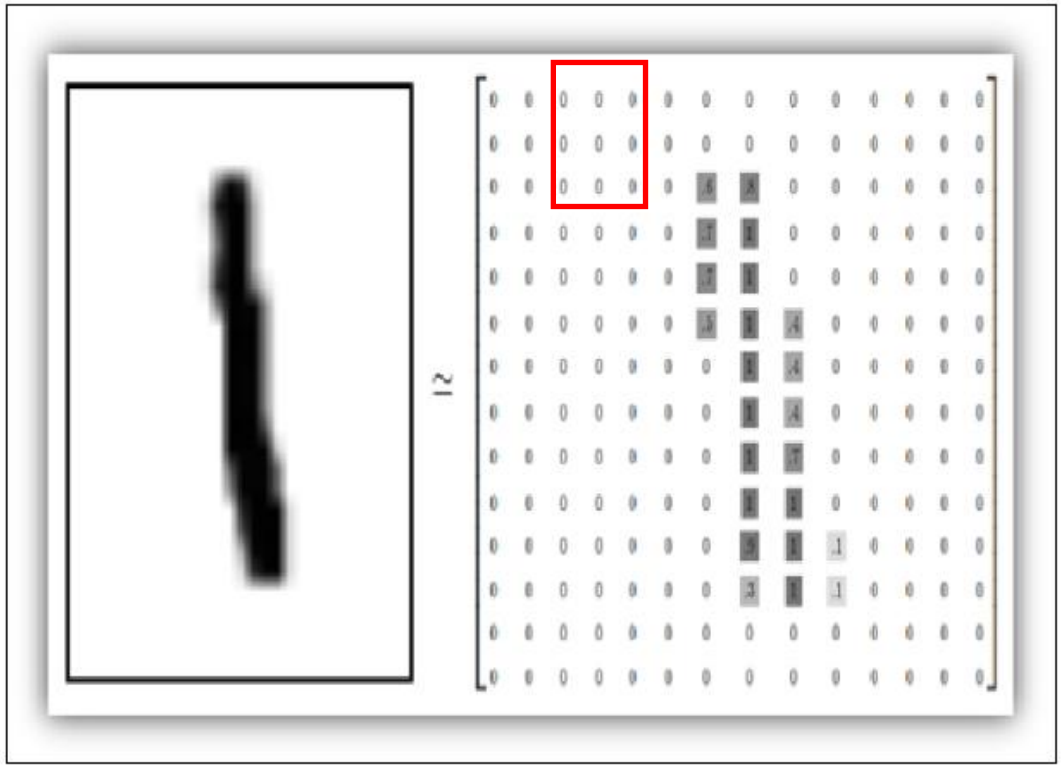
- **Example** (MNIST)
  - **Input** size: 28 by 28
  - **Convolutional layer:**
    - Filter size: (3, 3)
    - Stride: (1, 1)
    - Zero padding size: 0
  - **First row, second patch**



$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}$$

Convolution layer

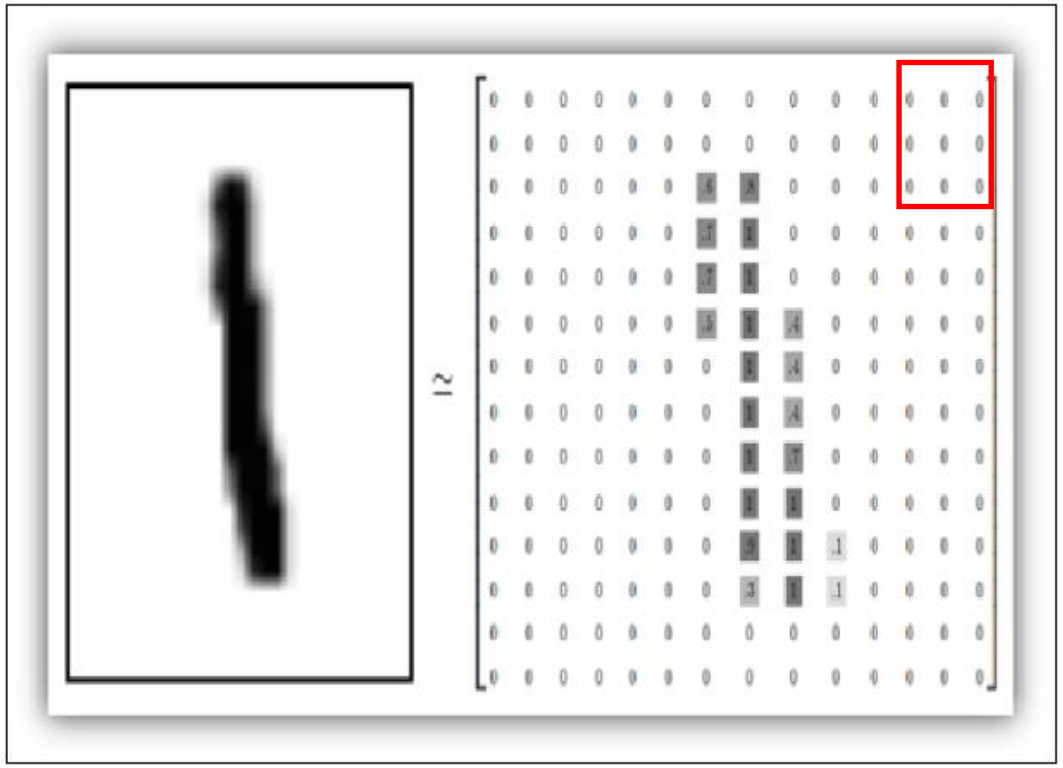
- **Example** (MNIST)
  - **Input** size: 28 by 28
  - **Convolutional layer:**
    - Filter size: (3, 3)
    - Stride: (1, 1)
    - Zero padding size: 0
  - **First row, third patch**



$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}$$

Convolution layer

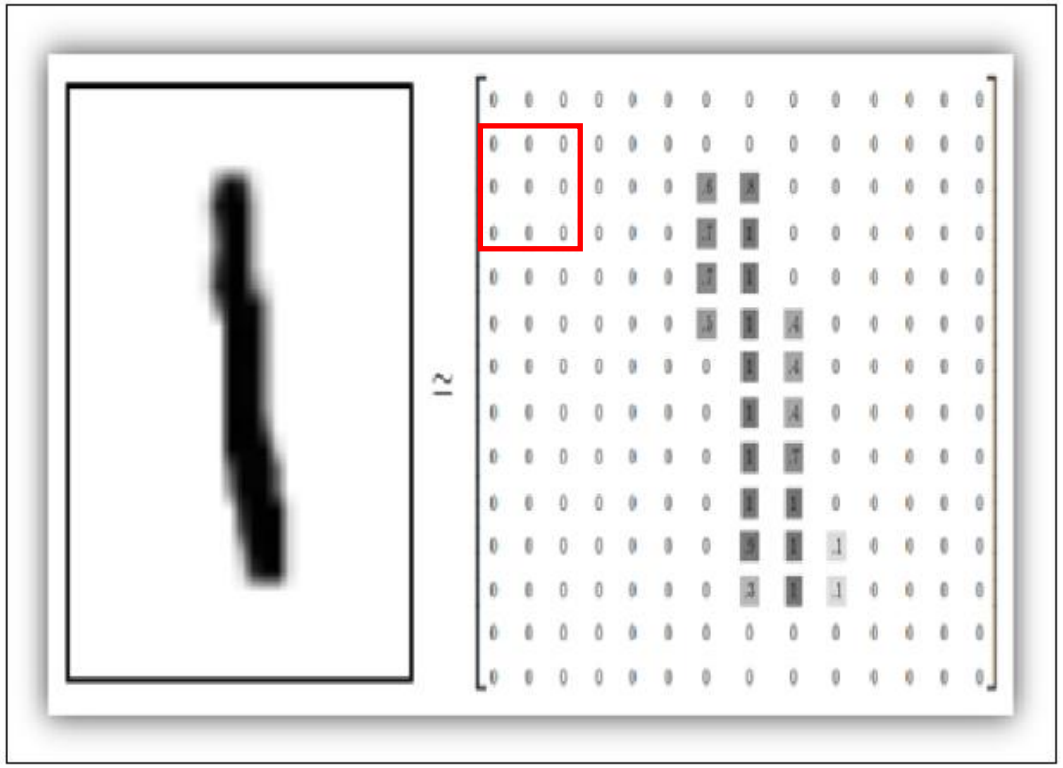
- **Example** (MNIST)
  - **Input** size: 28 by 28
  - **Convolutional layer:**
    - Filter size: (3, 3)
    - Stride: (1, 1)
    - Zero padding size: 0
  - **First row, last patch**



$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}$$

Convolution layer

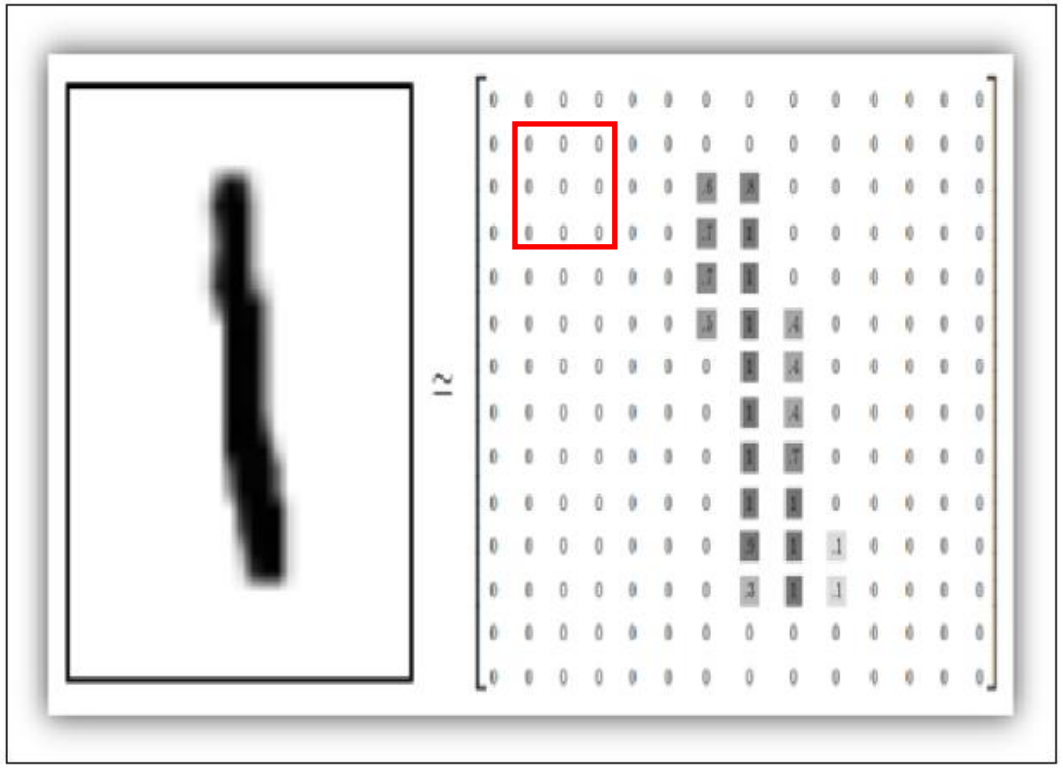
- **Example** (MNIST)
  - **Input** size: 28 by 28
  - **Convolutional layer:**
    - Filter size: (3, 3)
    - Stride: (1, 1)
    - Zero padding size: 0
  - **Second row, first patch**



$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}$$

Convolution layer

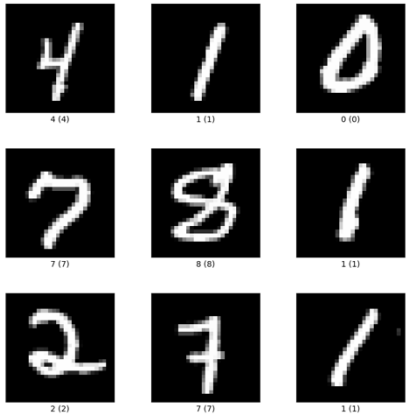
- **Example** (MNIST)
  - **Input** size: 28 by 28
  - **Convolutional layer:**
    - Filter size: (3, 3)
    - Stride: (1, 1)
    - Zero padding size: 0
  - **Second row, second patch**

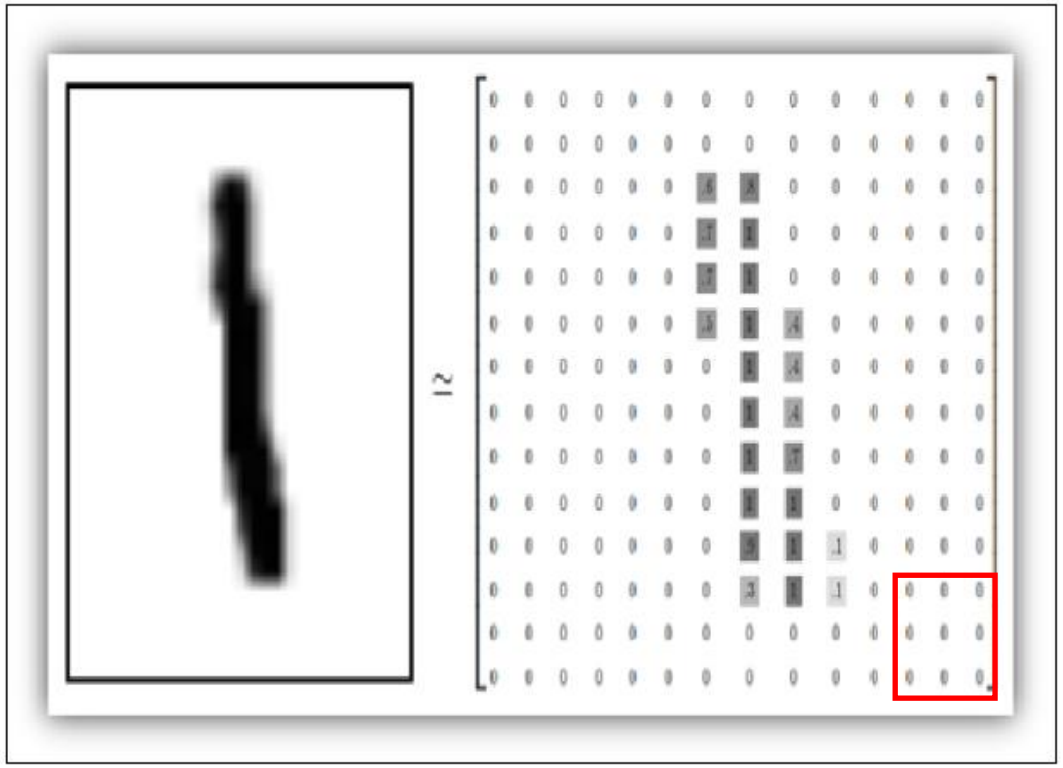


$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}$$

Convolution layer

- **Example** (MNIST)
  - **Input** size: 28 by 28
  - **Convolutional layer:**
    - Filter size: (3, 3)
    - Stride: (1, 1)
    - Zero padding size: 0
  - **Last row, last patch**





- **Question:** What is the final output size?

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}$$

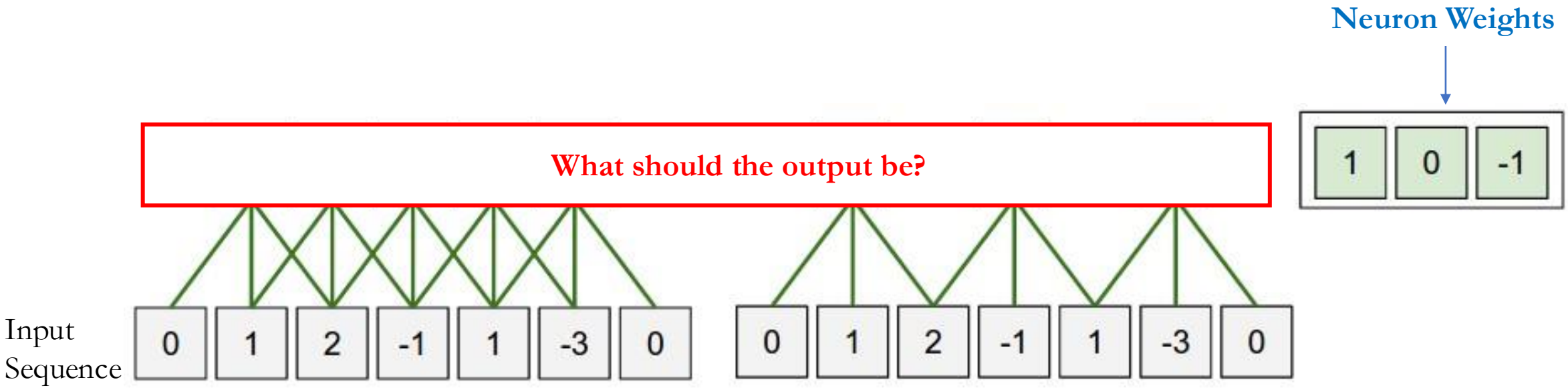
Convolution layer

- **Filter (depth times width)**
  - Larger filter captures broader patterns in one step. Need fewer layers to get large receptive fields
  - Smaller filters capture fine-grained details. Need more layers to get large receptive fields
- **Stride (depth times width)**
  - Small stride (e.g., 1): high spatial resolution, more detailed feature map
  - Large stride (e.g., 2): downsample quickly, reduce computation
- **Padding size**
  - No padding: output shrinks after each layer
  - With padding: preserves spatial size

Illustration

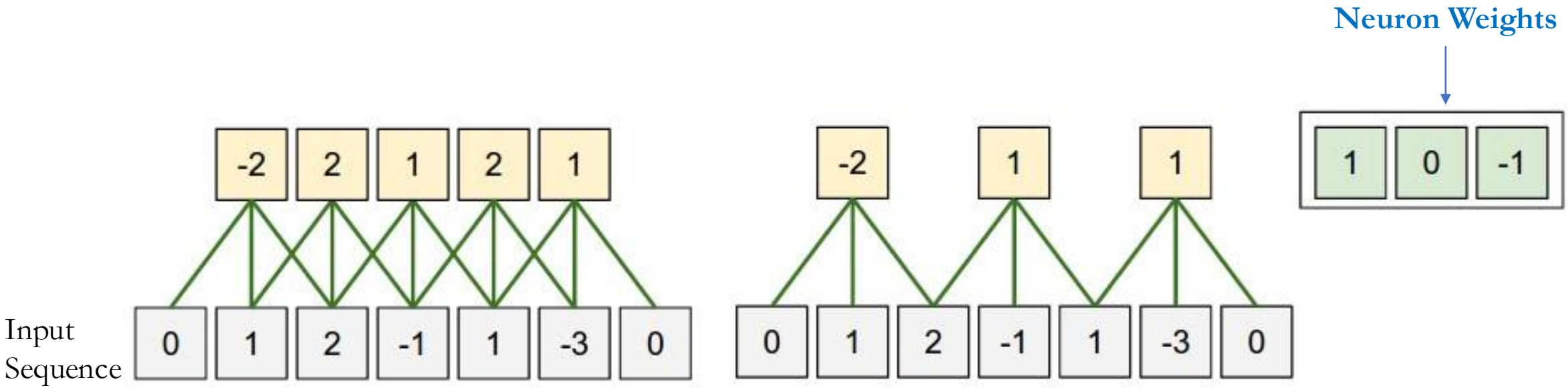
- Input dimension is one, filter size is (3), stride is (1)

- Multiply the input with the neuron weights pixel-by-pixel



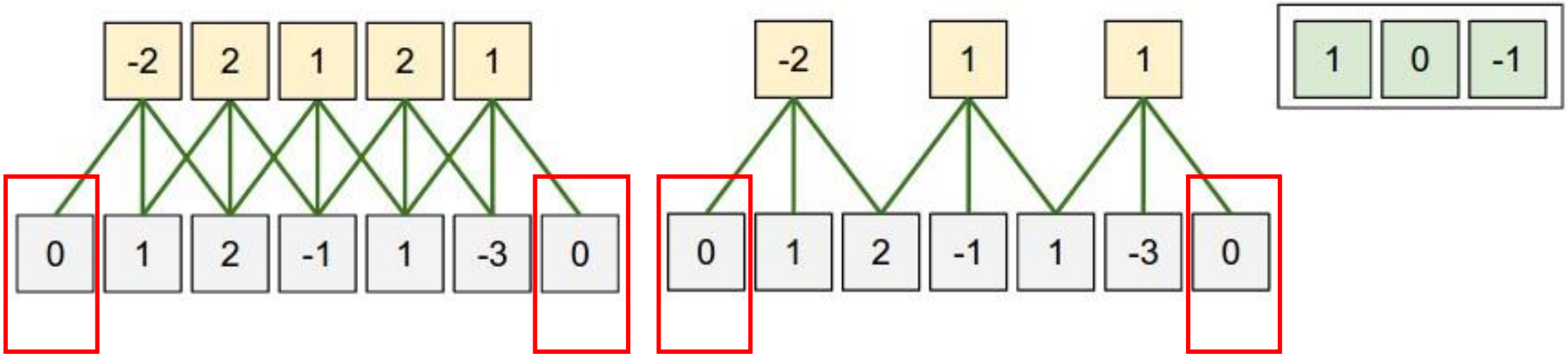
Illustration

- Illustration of spatial arrangement with a simplified example
  - Filter size is (3)
  - Stride is (1)



Explaining zero padding size

- This example uses **a single zero padding** on **both left and right**



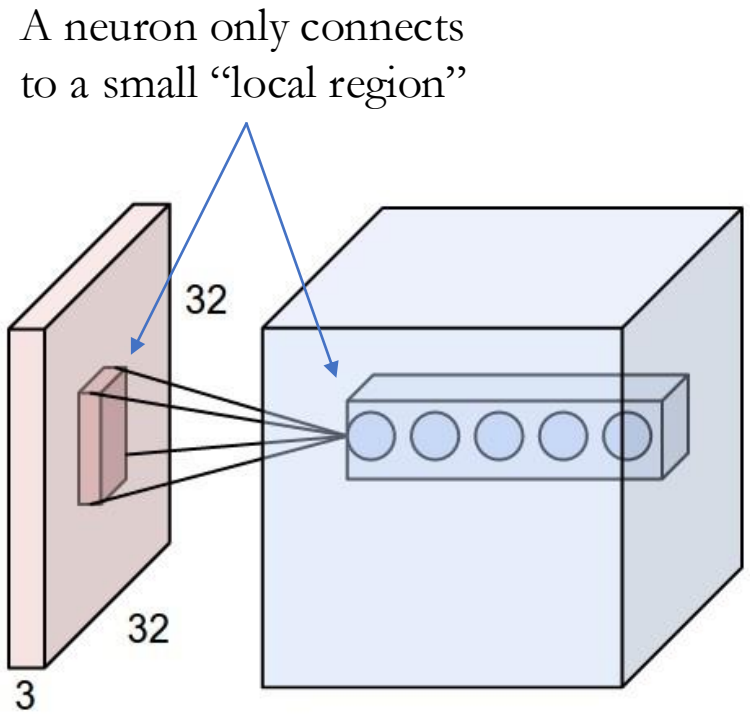
- We can use zero padding to adjust the output dimension, e.g., in sentence classification, use zero padding for fixed (max) length sentences

Stride size

- **Constraints**
  - Filter size and stride size must satisfy that: (image width – filter size) should be divisible by (stride size)
  - **Otherwise, often add zero padding**
  - **Question:** What goes wrong if this constraint is not satisfied?

Example (CIFAR-10)

- Illustrating the convolution operation for an image of size $(32, 32, 3)$



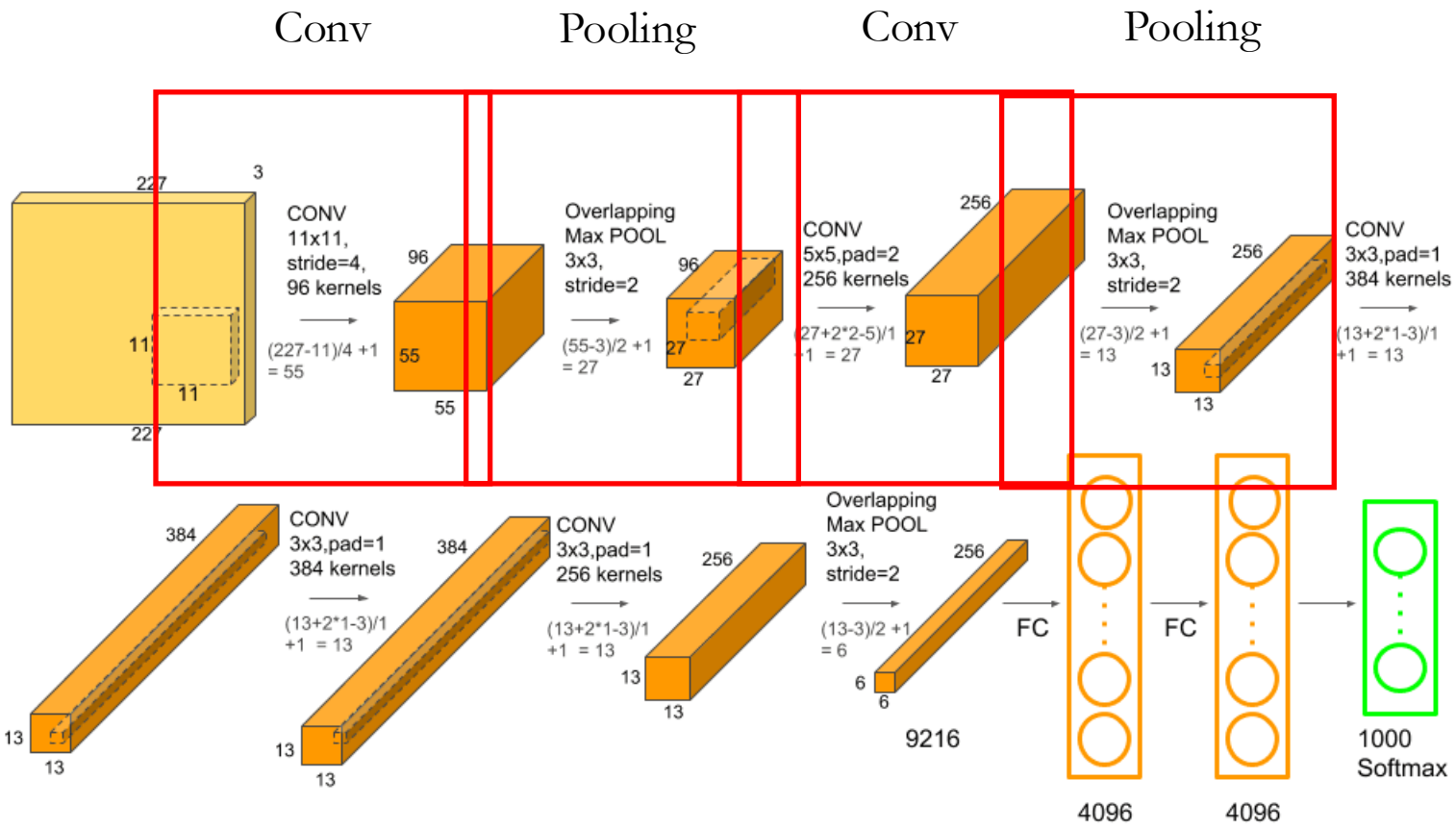
  - Within each neuron, perform convolution with possible nonlinear activation
- **Question:** can you specify a convolution layer configuration for CIFAR-10?

Example (Image

- ImageNet: Each image has size (227, 227, 3)
- AlexNet (2012), led by Goeff Hinton at Google
  - First convolution layer uses
    - **Filter size**: 11 by 11 by 3
    - **Stride**: 4 by 4
    - **Zero-padding**: 0
    - (227 – 11) is divisible by 4
  - Number of different filters is 96
  - **Question: Final output size?**
    - (227 – 11) / 4 + 1 = 55: 55 by 55 by 96

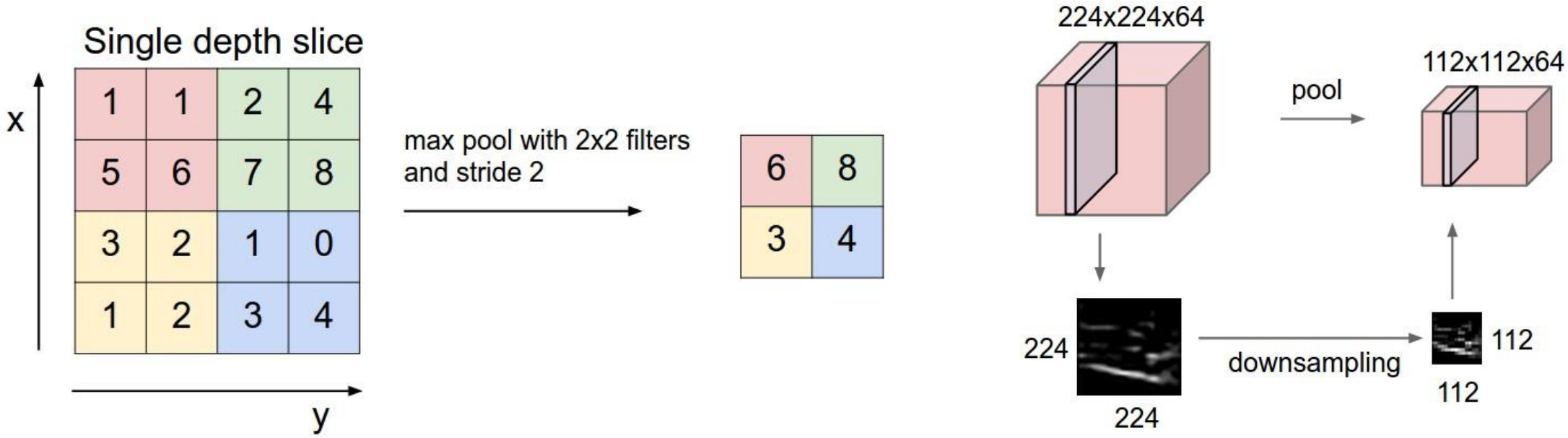
Nobel prize in physics 2024!!

Example (ImageNet)



Pooling layer

- **Pooling** reduces the spatial size of the input: Insert a pooling layer between convolution layers



Comparison of number of parameters

- In ImageNet, each image has size (227, 227, 3)
  - If we use a fully-connected layer: Suppose there are 100 filters, the total number of parameters is 227*227*3*100; this is very large
  - If we use a convolution layer: 11*11*3*100=36,300
- **Key idea:** parameter sharing, i.e., we use the same parameters in every filter
  - Leverages the geometry already present in visual images