CS167: Machine Learning

Convolutional Neural Network (CNN)
Coding CNN using PyTorch

Monday, April 27nd, 2026

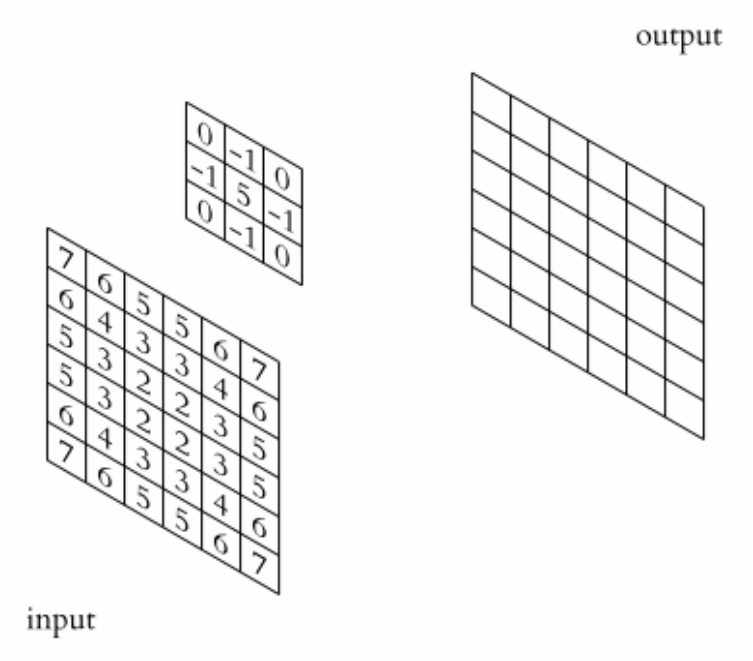
Drake
UNIVERSITY

Today's Agenda

- Convolutional Neural Network (CNN): another type of neural network
  - Convolution operation
  - Nonlinearity
  - Pooling operation
  - CNN: convolutional layer + nonlinearity + pooling layer
  - Coding CNN in PyTorch

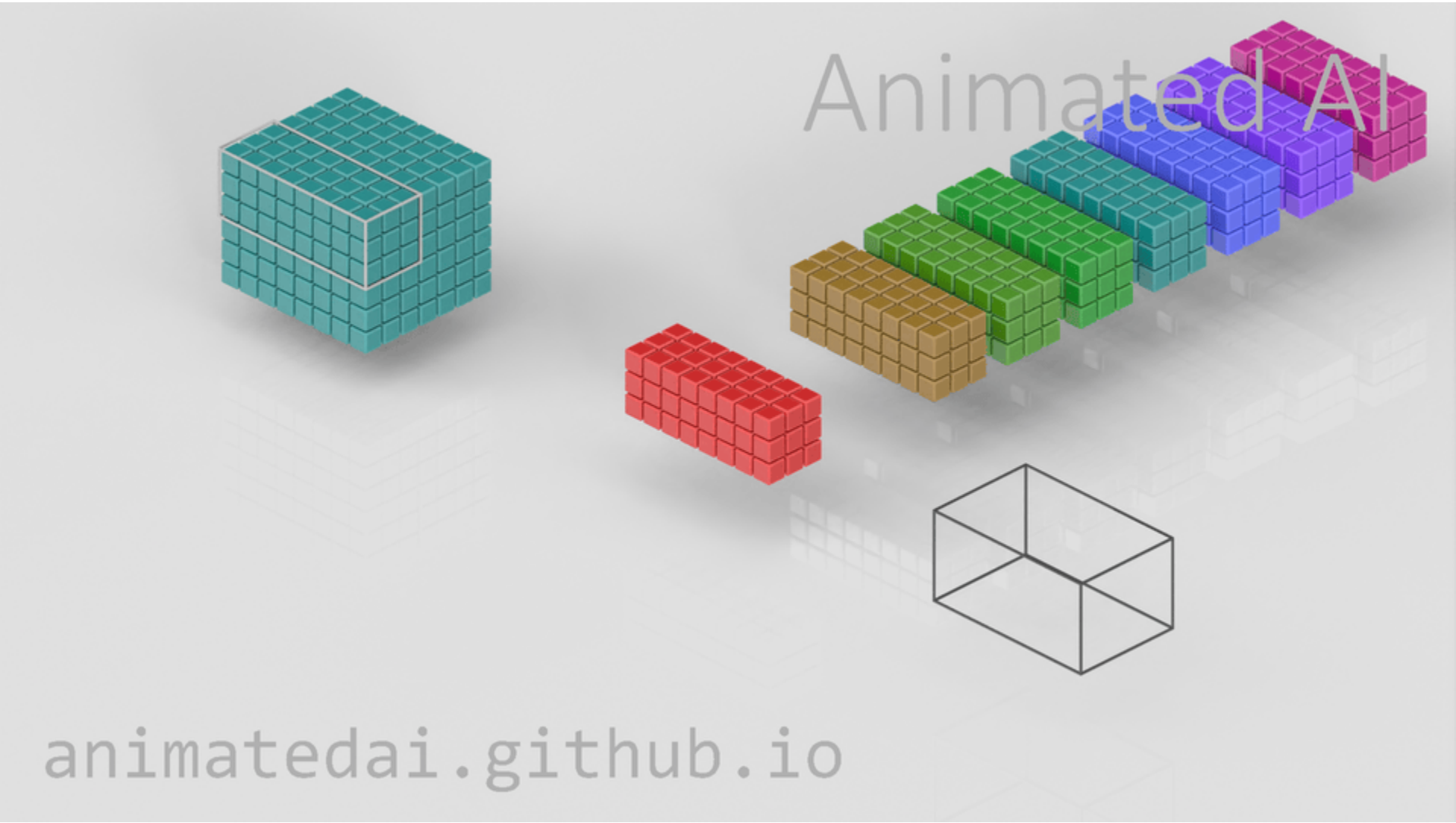
Review: Convolution Operation Animation!

- What does a **convolution operation** do?
- **convolution operation can be achieved with a series of dot products between portions of input feature map and a convolution filter (kernel) weights**

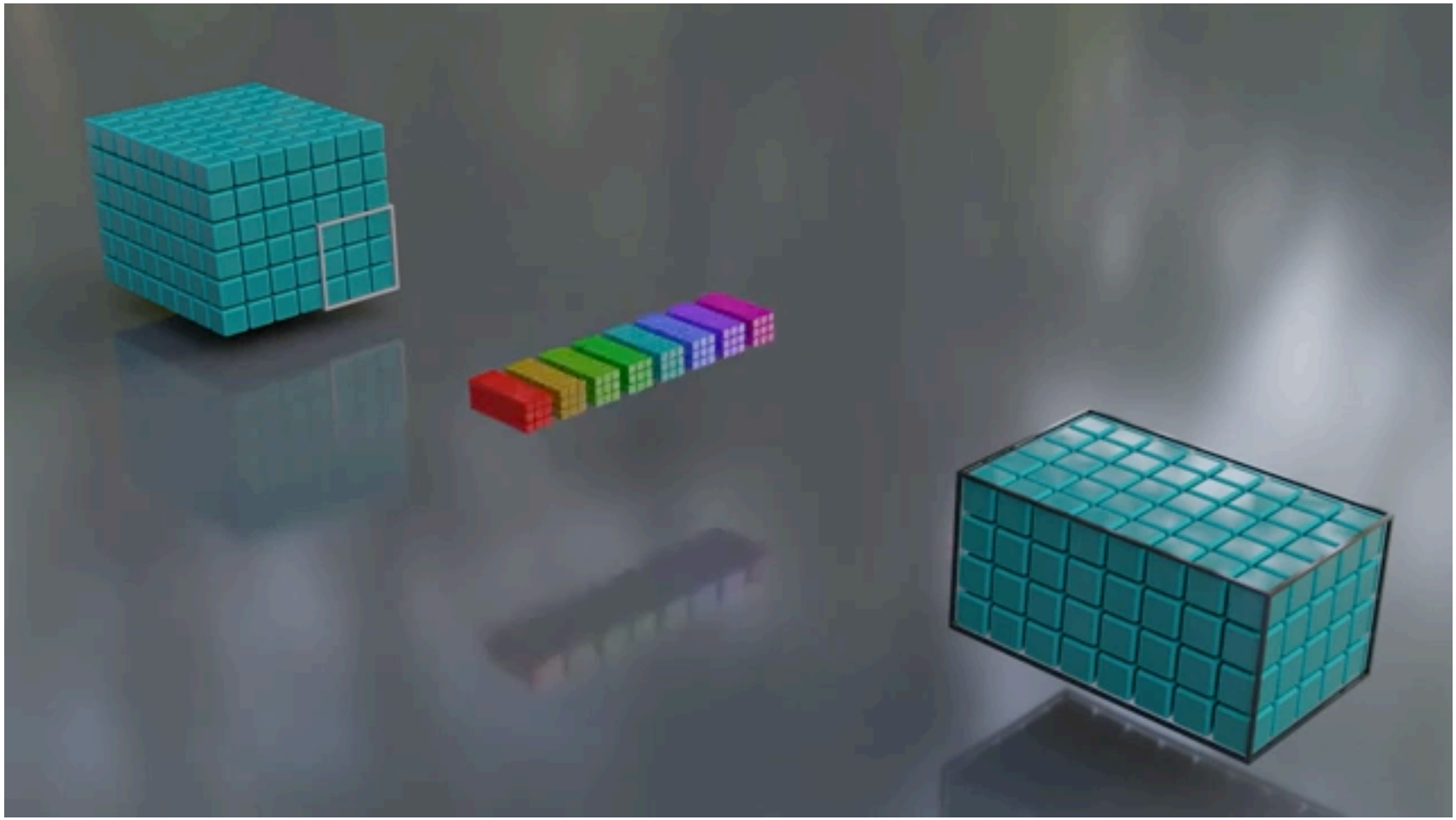


Another visualization shows a convolution filter applied to an image, resulting in the convolved feature

Review: How to calculate the output volume size?



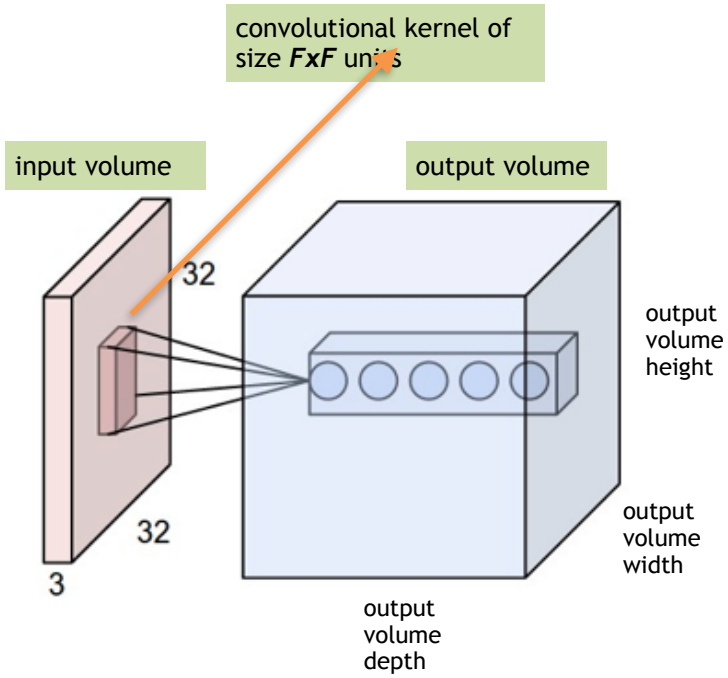
Review: How to calculate the output volume size?



https://www.youtube.com/watch?v=w4kNHKcBGzA&t=210s

Review: How to calculate the output volume size?

- An input volume has size ***(WxWx3)***, eg, (227, 227, 3)
- Filter size/receptive field is ***(FxF)***, eg, (11x11)
- Spatial Stride ***S***, eg, ***S***=4
- Padding size ***P***, eg, ***P***=0
- Number of filters ***K***, eg, ***K***=96

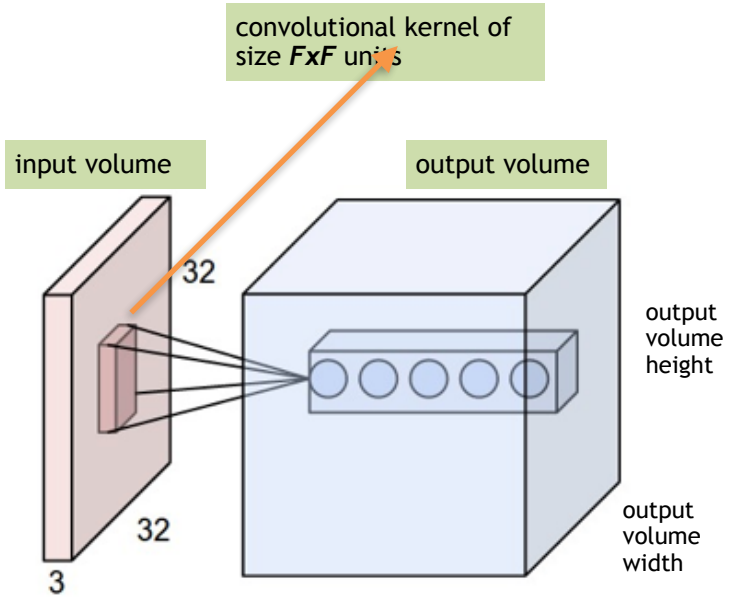


- Size of the output volume width and output volume height as a function of ***W***, ***F***, ***S***,and ***P*** as follows:

$$\text{output volume width/height} = \frac{(W - F + 2P)}{S} + 1 = \frac{(227 - 11 + 2*0)}{4} + 1 = 54+1 = 55$$

Review: How to calculate the output volume size?

- An input volume has size **(W_1 x H_1 x D_1)**
  - Filter size/receptive field is ***(FxF)***
  - Spatial stride size ***S***
  - Padding size ***P***
  - Number of filters ***K***



- Spatial sizes of the output volume ***(W_2 x H_2 x D_2)***

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

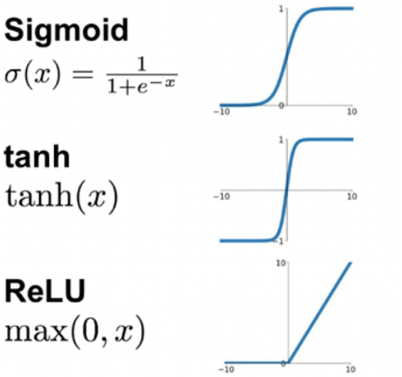
$$D_2 = K$$

- Number of filter weight parameters ***= (F x F x D_1) x K***
- Number of bias parameters = ***K***

Today's Agenda

- Convolutional Neural Network (CNN): another type of neural network
  - Convolution operation
  - Nonlinearity
  - Pooling operation
  - CNN: convolutional layer + nonlinearity + pooling layer

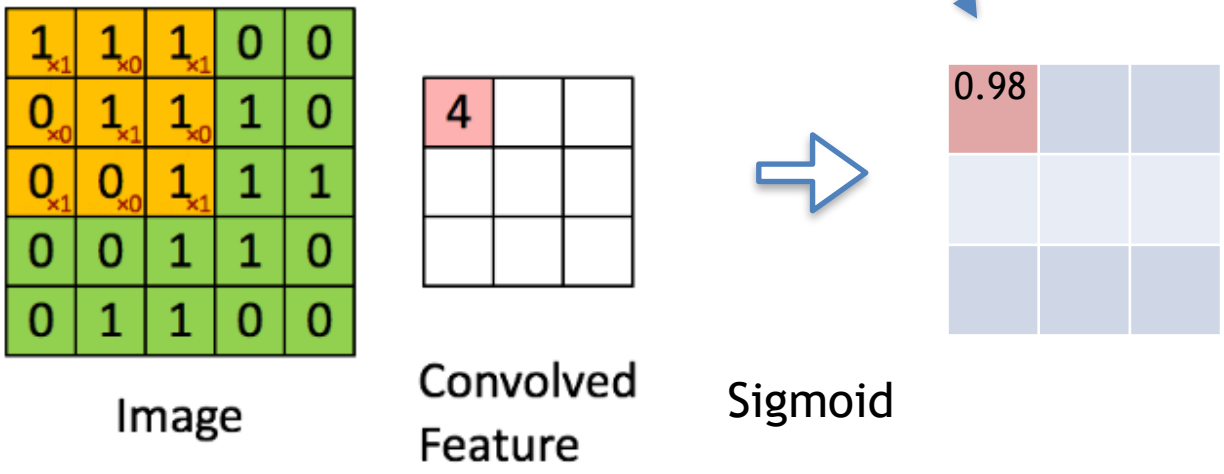
Nonlinear Function



- Just like an MLP, each convolutional output goes through a non-linear function such as Sigmoid, Tanh, or Rectified Linear Unit (ReLU)

$$convolution = 1*1+1*0+1*1+0*0+1*1+1*0+0*1+0*0+1*1 = 4$$

$$Sigmoid(4) = \frac{1}{1+exp(-4)} = 0.98$$



Today's Agenda

- Deep Learning

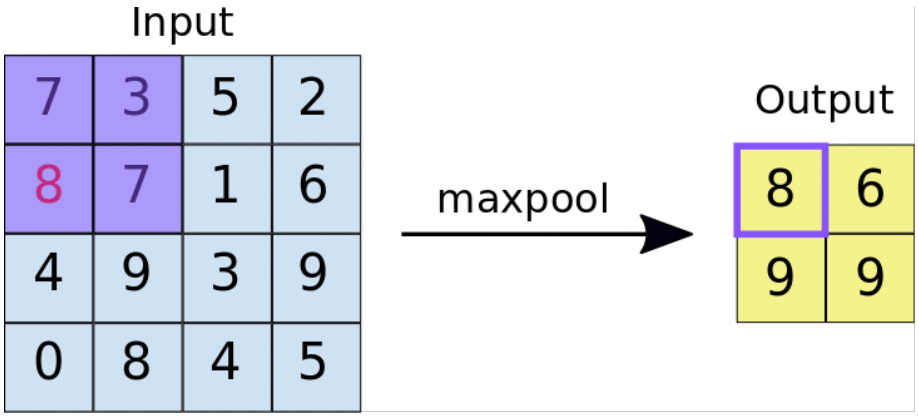
- Convolutional Neural Network (CNN)
  - Convolution operation
  - Nonlinearity
  - Pooling operation
  - CNN: convolutional layer + nonlinearity + pooling layer

Pooling Operation

- Image data can get computationally inefficient, really quickly. To avoid this, we often toss in a layer that helps us to **summarize** and **downsample** the data

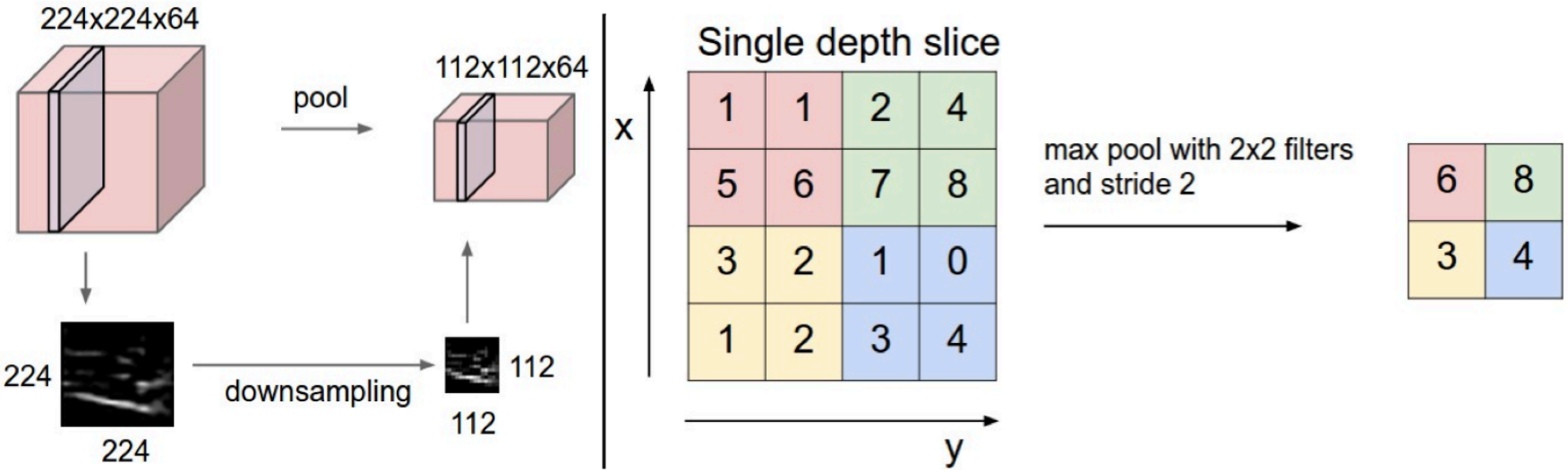
- In classical CNN, we find another useful operation called **pooling operation**

- A common pooling operation is **max pooling**, and its goal is to locally summarize the convolution. It performs something like a convolution, but rather than taking the dot product, it takes the maximum element in the filter area



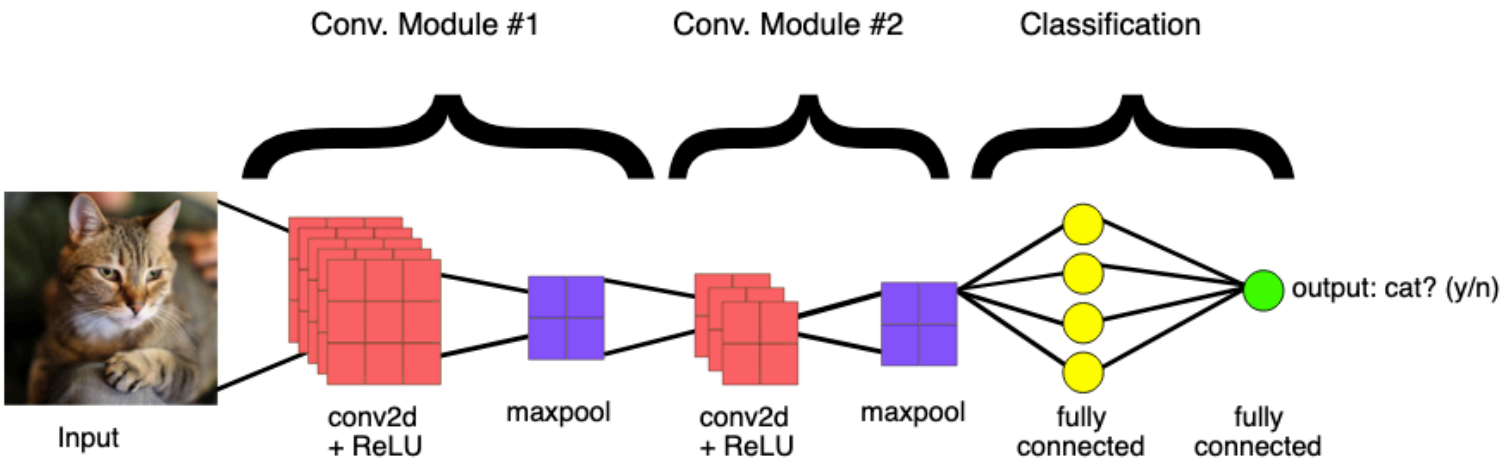
Pooling Operation

- Pooling operation downsamples the volume spatially, i**ndependently in each depth slice of the input volume**
- Besides max pooling, other pooling operations include: **sum pooling**, average pooling



CNN: A Composition of Convolutional Layers

- We've talked about **image data**, **convolutions**, **nonlinearity**, **max pooling**, and how they are related to some computer vision tasks. Let's connect the dots
  - input is an image (in this case a color image, so 3 channels—red, green, and blue)
  - there are several filters, not just one.
  - `Conv2D` layers with `ReLU` are often followed by `maxpool`
  - towards the end of the model, we switch to fully connected (`Dense`) layer
  - We have as many output nodes as we have classes to predict



Reference

Building CNN using PyTorch

- Let's build CNN using PyTorch

  https://github.com/alimoorreza/CS167-sp26-notes/blob/main/Day25_simple_CNN.ipynb

Notebook#5

- It has two parts. Start working on part#2 which is CNN

https://github.com/alimoorreza/CS167-SP26-Notebook-5

Notebook #5

Deep learning with MLP and CNN for an image recognition task.

The Data: 📊

For this notebook, you will train two deep neural networks (DNN): **Part 1) a multilayer perceptron (MLP)***, and Part 2) a convolutional neural network (CNN) for an image recognition task, such as the cat vs. dog example. In this case, the algorithm will utilize a dataset of images to learn useful features for classifying an image into one of 10 classes. You should be using the CIFAR-10 dataset.