Workshop 7

COMP90051 Statistical Machine Learning

Semester 1, 2023

Learning Outcomes

By the end of this workshop you should be able to:

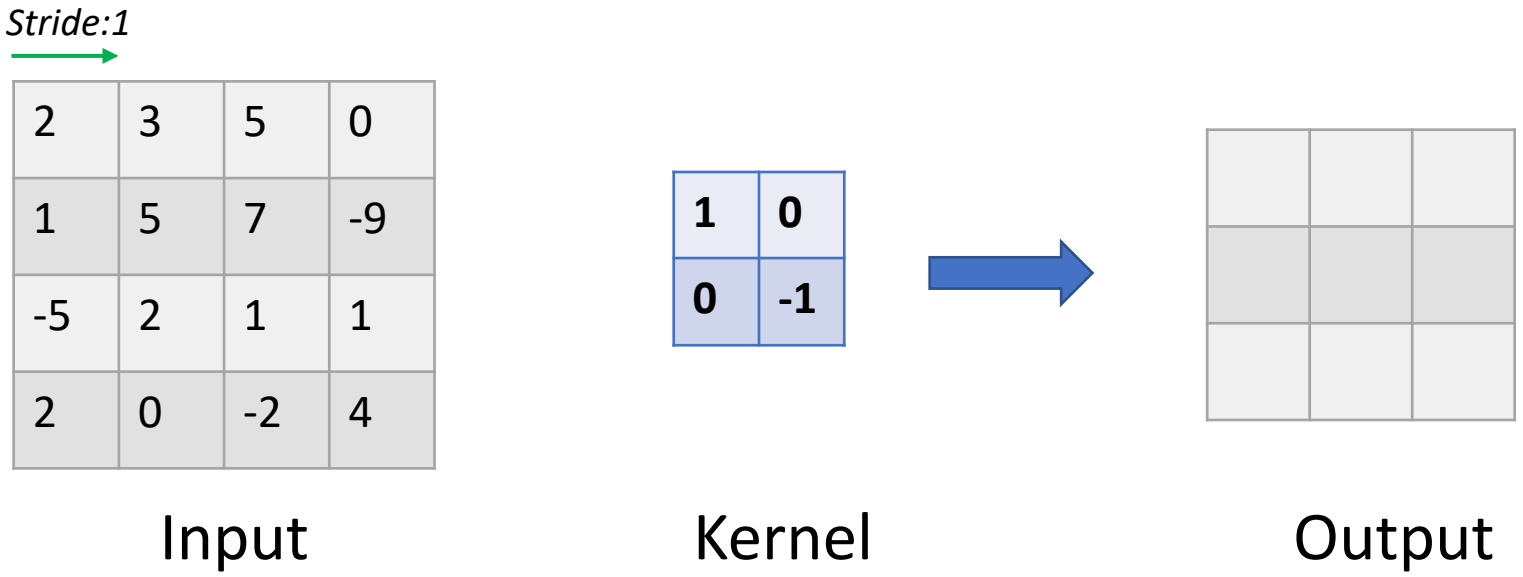
1. Be able to define and fit models in PyTorch
2. Be able to explain the architecture of a basic convolutional neural network
3. Be able to implement autoencoder

Convolutional neural nets

- Convolutional layers
  - Complex input representations based on convolution operation
  - Filter weights are learned from training data
- Downsampling, usually via Max Pooling
  - Re-scales to smaller resolution, limits parameter explosion
- Fully connected parts and output layer
  - Merges representations together

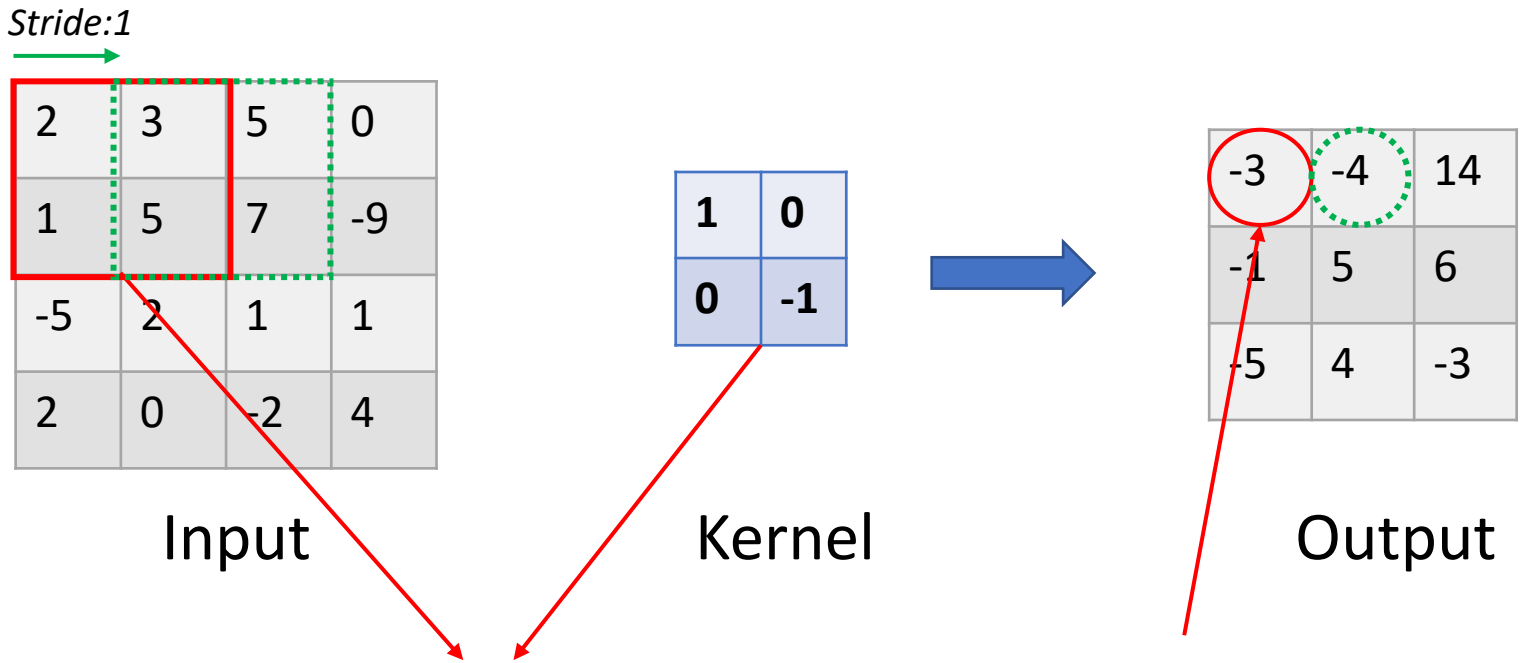
Convolutional in 2D

- Use kernel to perform element-wise multiplication and sum for every local patch



Convolutional in 2D

- Use kernel to perform element-wise multiplication and sum for every local patch



$$2\times1 + 3\times0 + 1\times0 + 5\times(-1) = -3$$

Max Pooling in 2D

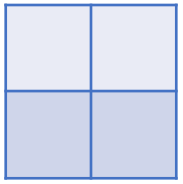
- Use kernel to perform element-wise multiplication and sum for every local patch

Stride:1

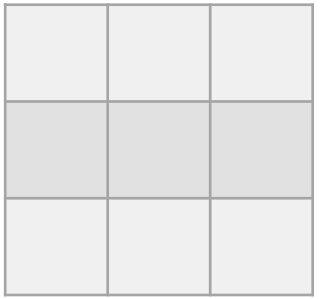
2	3	5	0
1	5	7	-9
-5	2	1	1
2	0	-2	4

Input

no trainable parameters



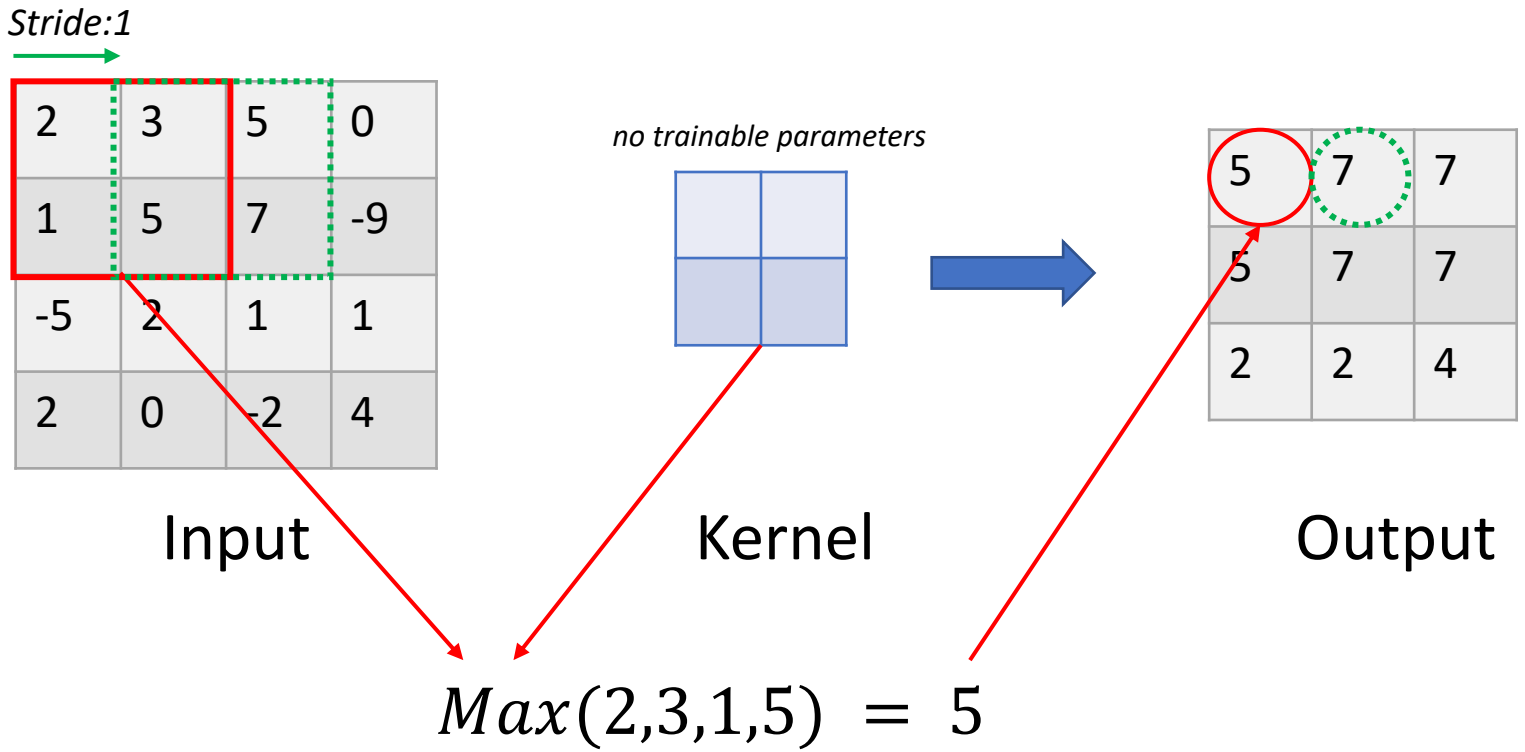
Kernel



Output

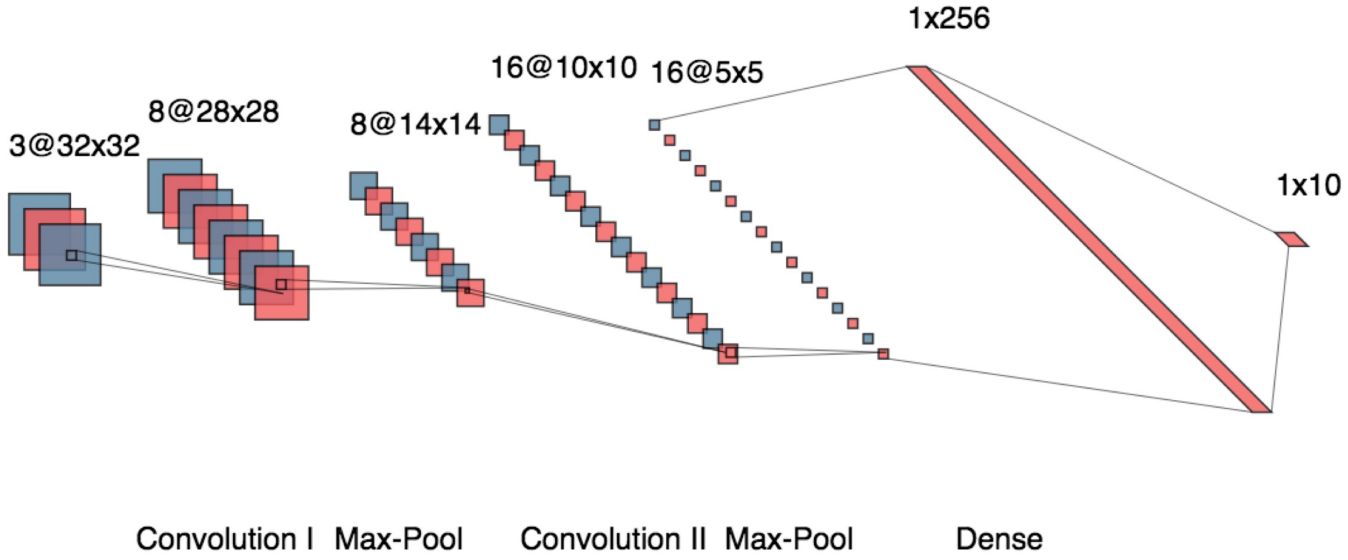
Max Pooling in 2D

- Use kernel to perform element-wise multiplication and sum for every local patch



Convolutional neural nets

- Local connectivity pattern between adjacent layers
- Shared weights—filters are replicated across the spatial dimensions of the input
- Pooling reduces the spatial extent deeper into the network



We'll implement this architecture for CIFAR-10

Worksheet 7