Convolutional Neural Networks

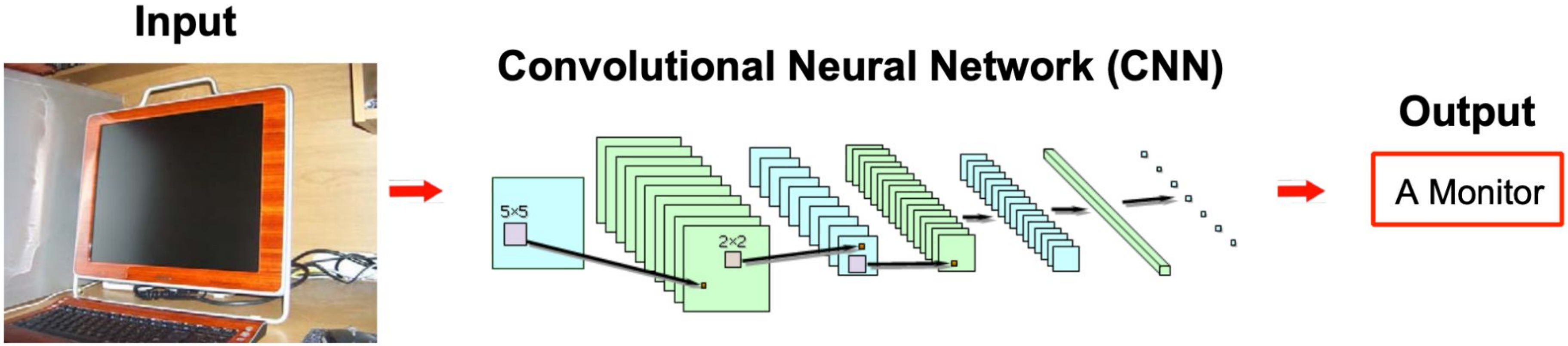
Wen Jiang

Aug 26, 2021

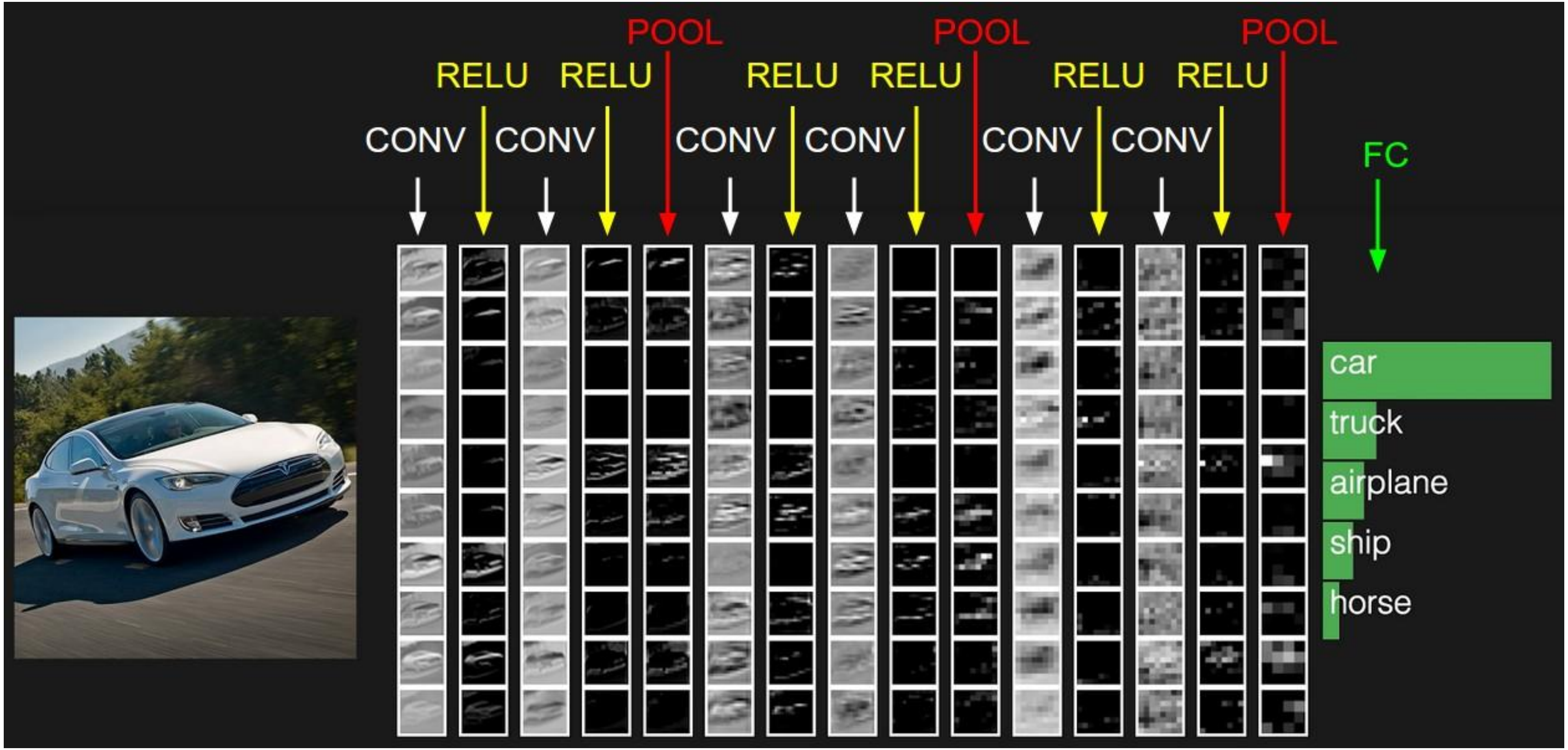
Goal: Given an image, we want to identify what class that image belongs to.

Input: **Output:**

Pipeline:


Input
Convolutional Neural Network (CNN)
5×5
2×2
Output
A Monitor

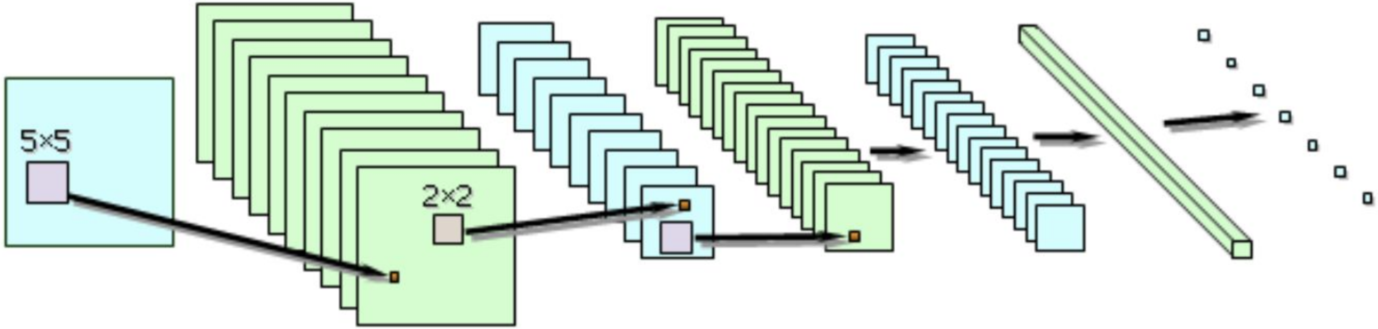
Pipeline:



Feifei Li et al, CS231n Stanford University

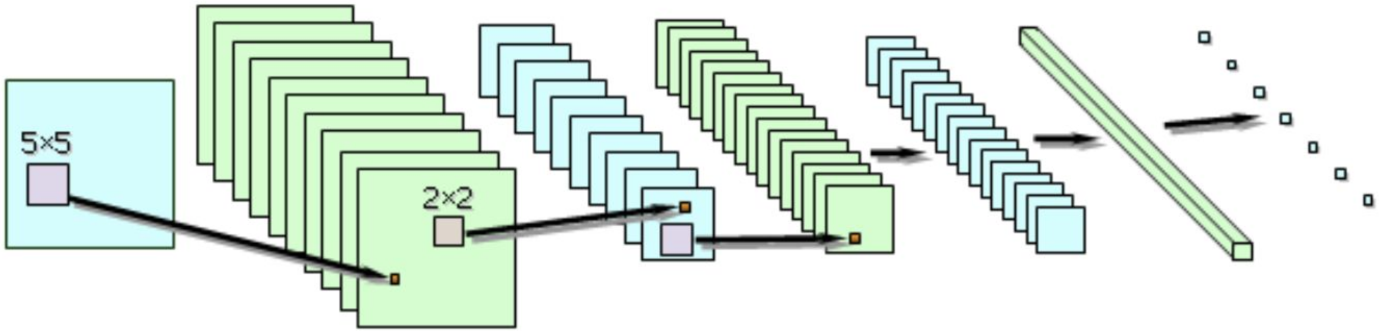
Convolutional Neural Nets (CNNs) in a nutshell:

- A typical CNN takes a raw RGB image as an input.
- It then applies a series of non-linear operations on top of each other.
- These include convolution, sigmoid, matrix multiplication, and pooling (subsampling) operations.
- The output of a CNN is a highly non-linear function of the raw RGB image pixels.



How the key operations are encoded in standard CNNs:

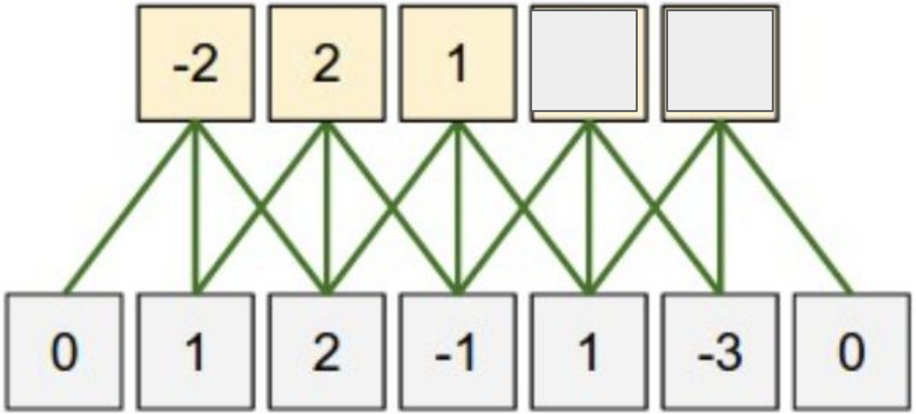
- Convolutional Layers: 2D Convolution
- Fully Connected Layers: Matrix Multiplication
- Sigmoid Layers: Sigmoid function
- Pooling Layers: Subsampling

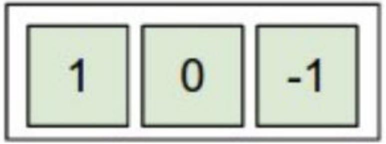


1D Convolution:

M = 2; Two examples with stride=1 and stride=2

$$h_i = \sum_{m=0}^{M} f(i-m)g(m)$$

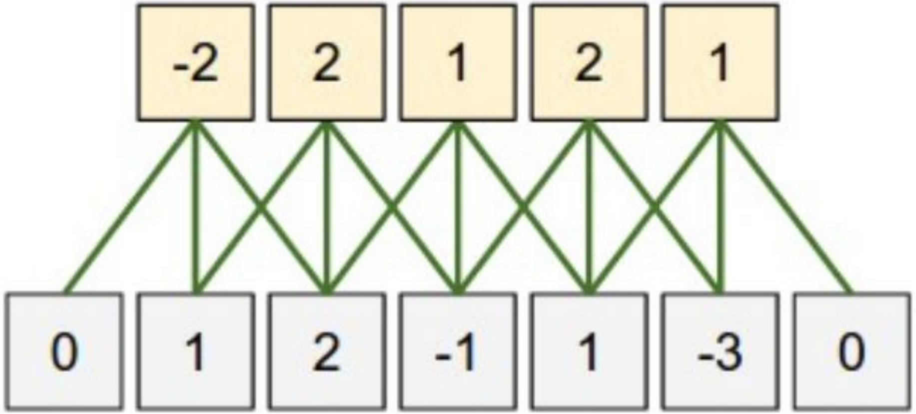


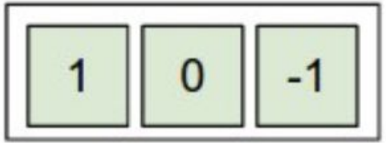


Feifei Li et al, CS231n Stanford University

1D Convolution:

$$h_i = \sum_{m=0}^{M} f(i-m)g(m)$$





Feifei Li et al, CS231n Stanford University

2D convolution:

$$h = f \otimes g$$

f - the values in a 2D grid that we want to convolve

g - convolutional weights of size MxN

$$h_{ij} = \sum_{m=0}^{M} \sum_{n=0}^{N} f(i-m, j-n)\, g(m,n)$$

A sliding window operation across the entire grid f .

$f =$ 

$$g_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad g_2 = \begin{bmatrix} 0.107 & 0.113 & 0.107 \\ 0.113 & 0.119 & 0.113 \\ 0.107 & 0.113 & 0.107 \end{bmatrix} \quad g_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

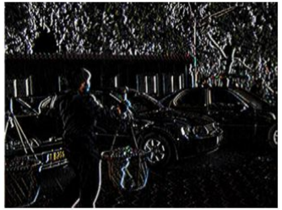
$f \otimes g_1$

Unchanged Image

$f \otimes g_2$

Blurred Image

$f \otimes g_3$



Vertical Edges

$f =$ 

$$g_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad g_2 = \begin{bmatrix} 0.107 & 0.113 & 0.107 \\ 0.113 & 0.119 & 0.113 \\ 0.107 & 0.113 & 0.107 \end{bmatrix} \quad g_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

CNNs aim to learn convolutional weights directly from the data