Practical Deep Neural Networks

GPU computing perspective

Convolutional Neural Networks

Yuhuang Hu Chu Kiong Loo

Advanced Robotic Lab
Department of Artificial Intelligence
Faculty of Computer Science & IT
University of Malaya

Outline

Outline

Assumed prerequisites

- ☆ Basic signal processing
- ☆ MLP Network (DL book chapter 6)

Suggest Readings

- CS231n: Convolutional Neural Networks, Visualize ConvNet
- UFLDL Tutorial: Feature Extraction Using Convolution, Pooling
- Deep Learning Book chapter 9
- DL Tutorial: Convolutional Neural Networks (LeNet)

Outline

Convolution operation

$$s(t) = \int x(a)w(t-a)\,da$$

the operation is called *convolution*. The convolution operation is typically denoted with $*$:

$$s(t) = (x * w)(t)$$

In discrete form:

$$s[t] = (x * w)(t) = \sum_{a=-\infty}^{\infty} x[a]w[t-a]$$

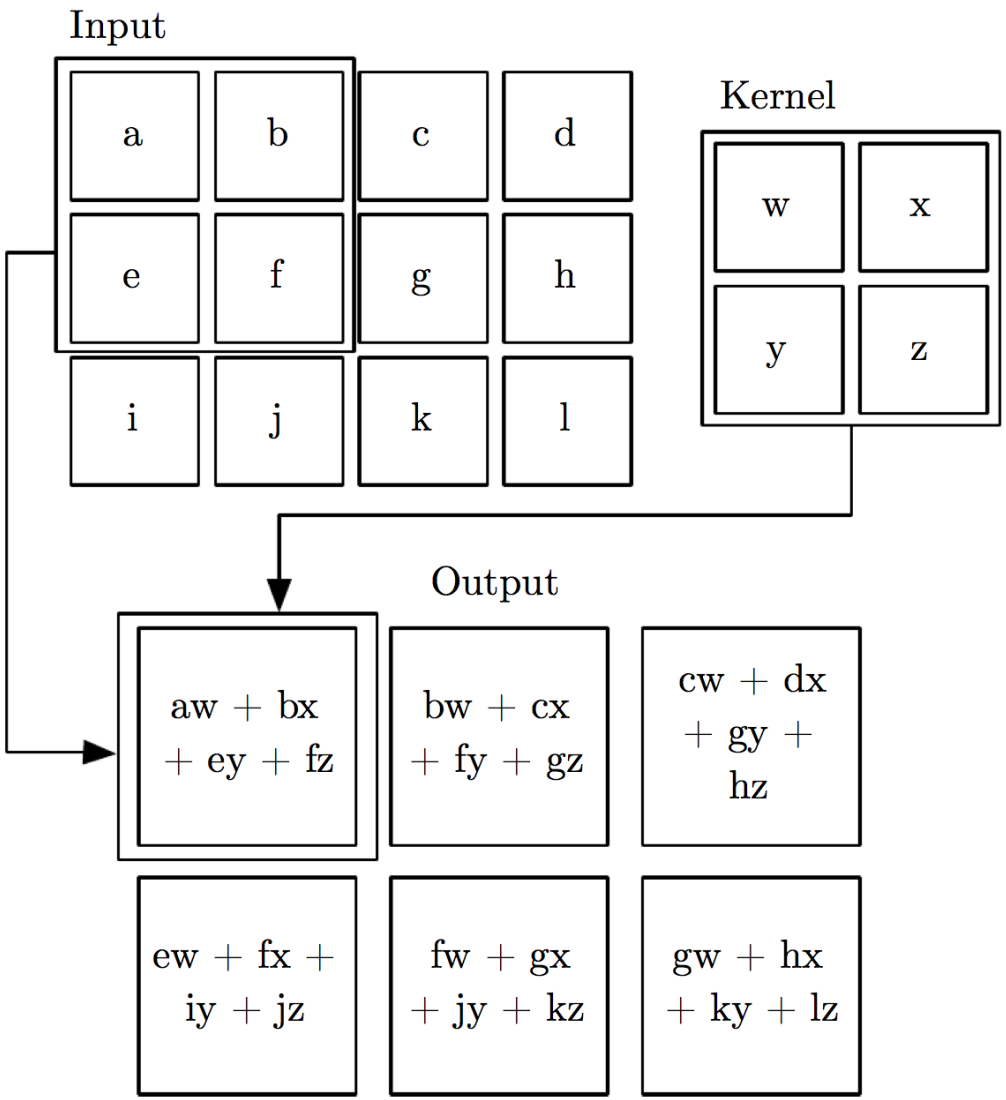
2D convolution operation

$$s[i,j] = (I * K)[i,j] = \sum_m \sum_n I[m,n]K[i-m, j-n]$$

or equivalently:

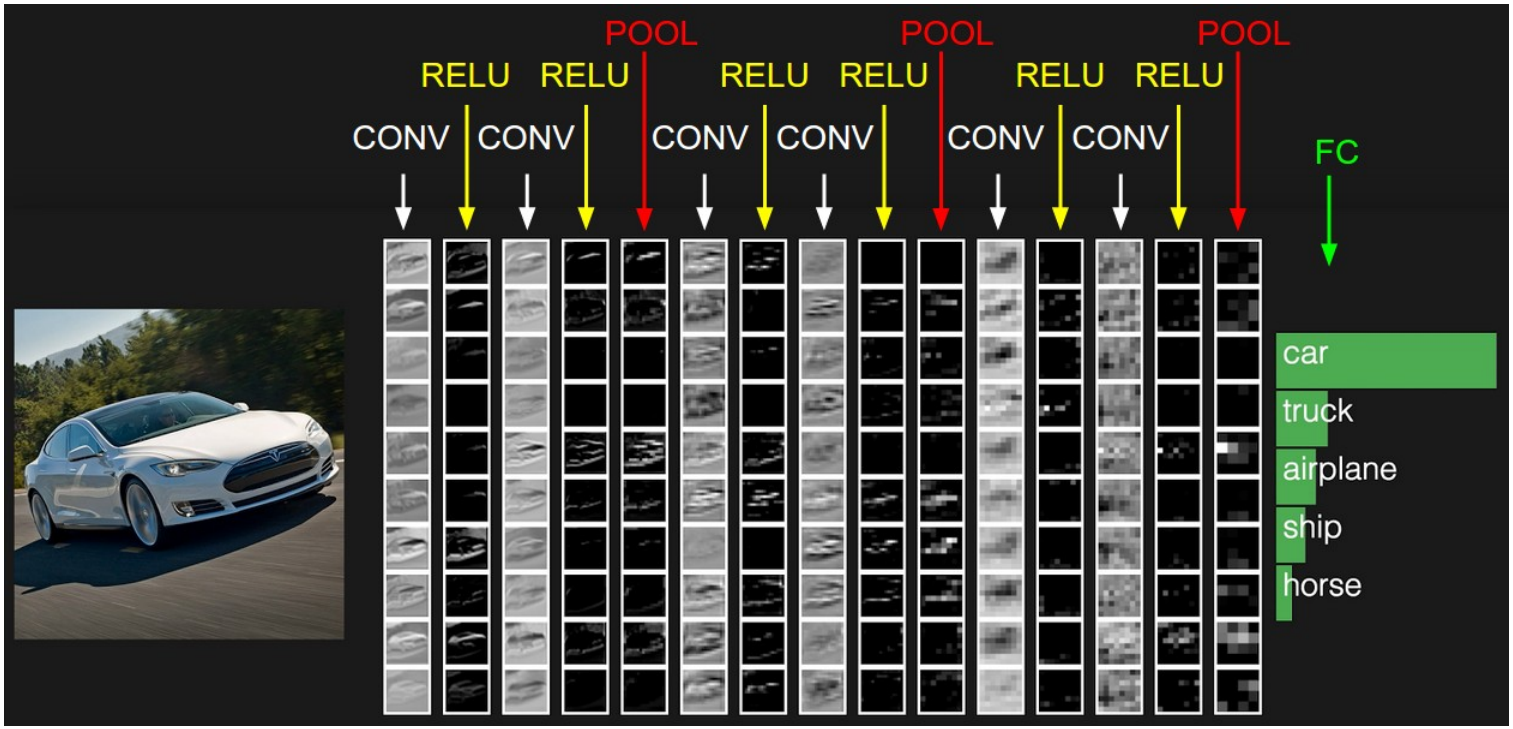
$$s[i,j] = (I * K)[i,j] = \sum_m \sum_n I[i-m, j-n]K[m,n]$$

2D convolution operation

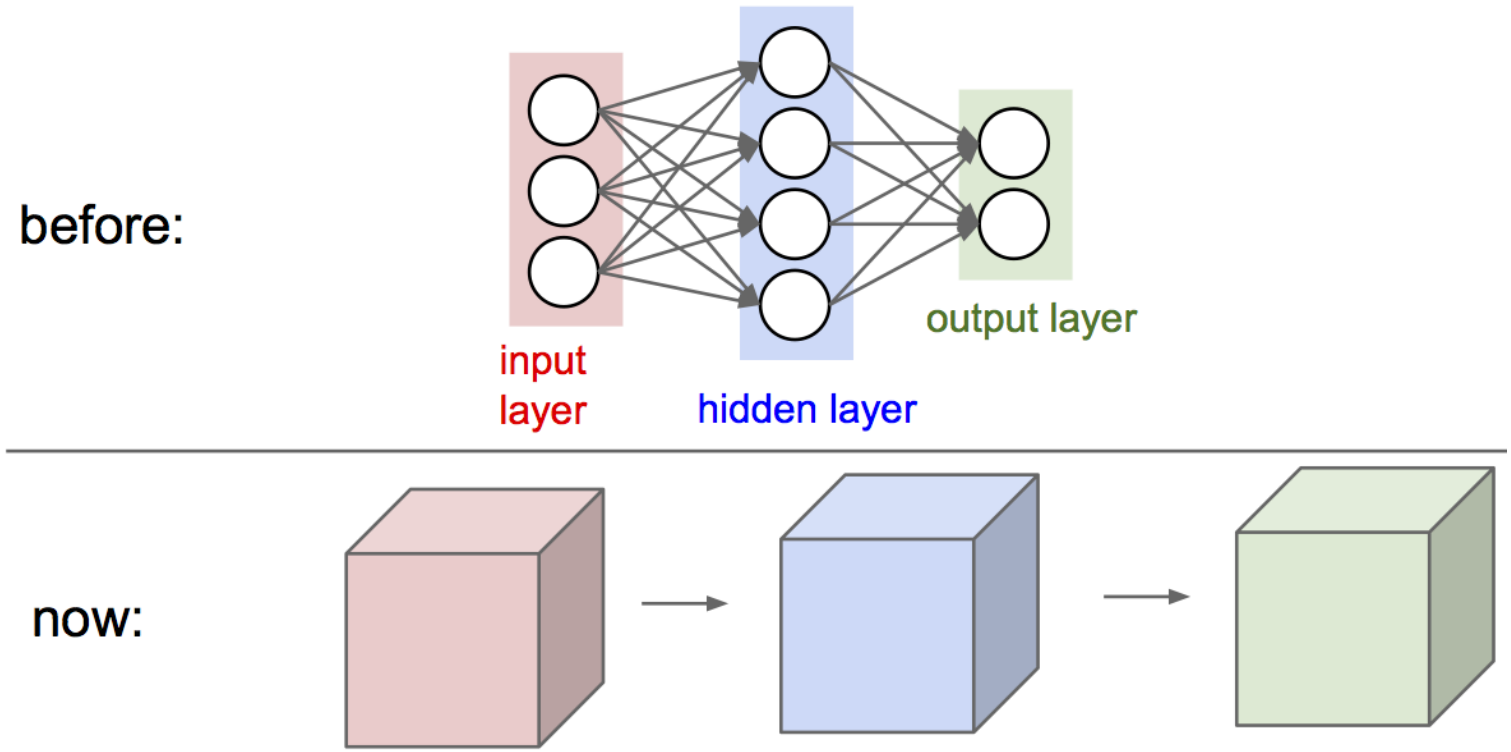


Outline

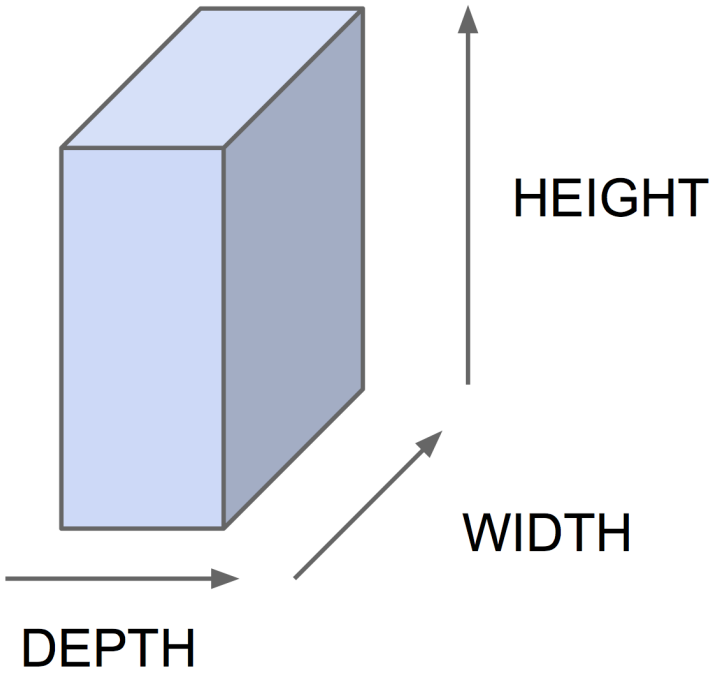
LeNet-5


POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

MLP→ConvNet



Feature maps: activations of ConvNets



- Network activations in ConvNets are **feature maps**.
- All ConvNets feature maps arranged in **3 dimensions**.
- Each feature maps has size of (HEIGHT, WIDTH)
- Input image can be a special kind of feature map (*e.g.* color image is feature maps of some size with depth 3, one for each RGB channel).

Convolution Layer: simple cell

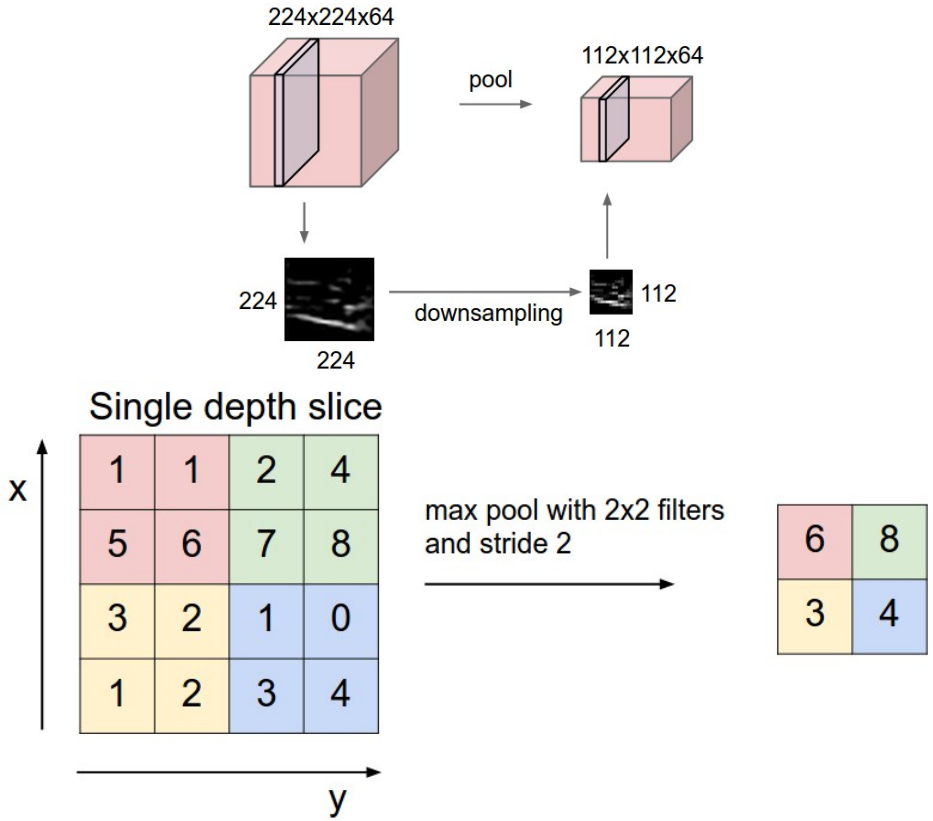
$$\mathbf{h}^{(k)} = f(\mathbf{x} * \mathbf{W}^{(k)} + b_k)$$

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Number of filters K with shape $F \times F \times D_1$, stride S, amount of zero-padding P
- Produce a volume of size $W_2 \times H_2 \times D_2$ where

$$W_2 = (W_1 - F + 2P)/S + 1$$
$$H_2 = (H_1 - F + 2P)/S + 1$$
$$D_2 = K$$

Live Demo of convolution

Pooling Layer: complex cell



Live Demo

Running ConvNets on your browser!

Demo[1]

[1] taken from Andrej Karpathy's ConvNetJS

Q&A