Advanced Machine Learning, 27 Aug 2019

Convolutional Neural Networks

Features in small areas of the input

Fully connected network computes a global function

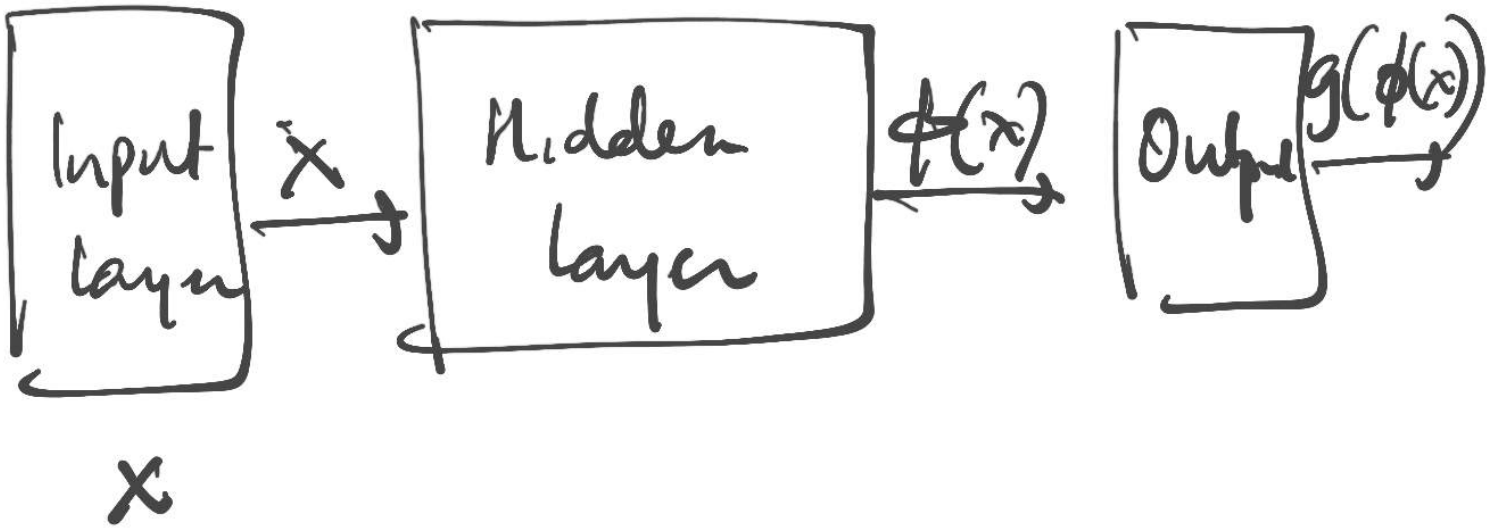
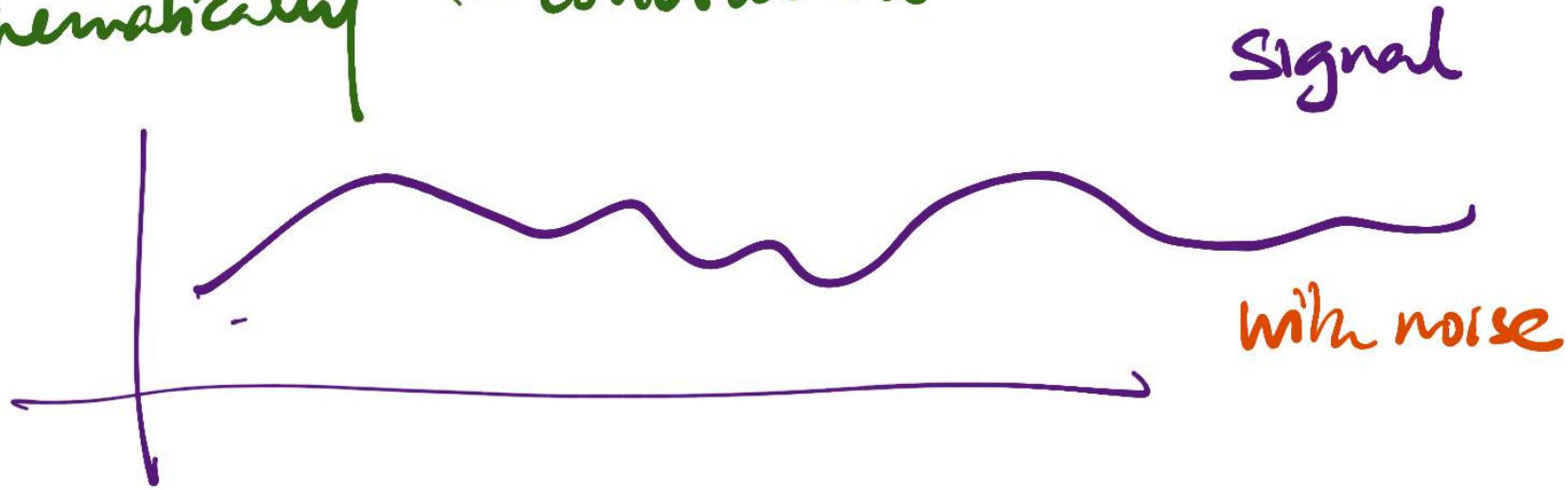
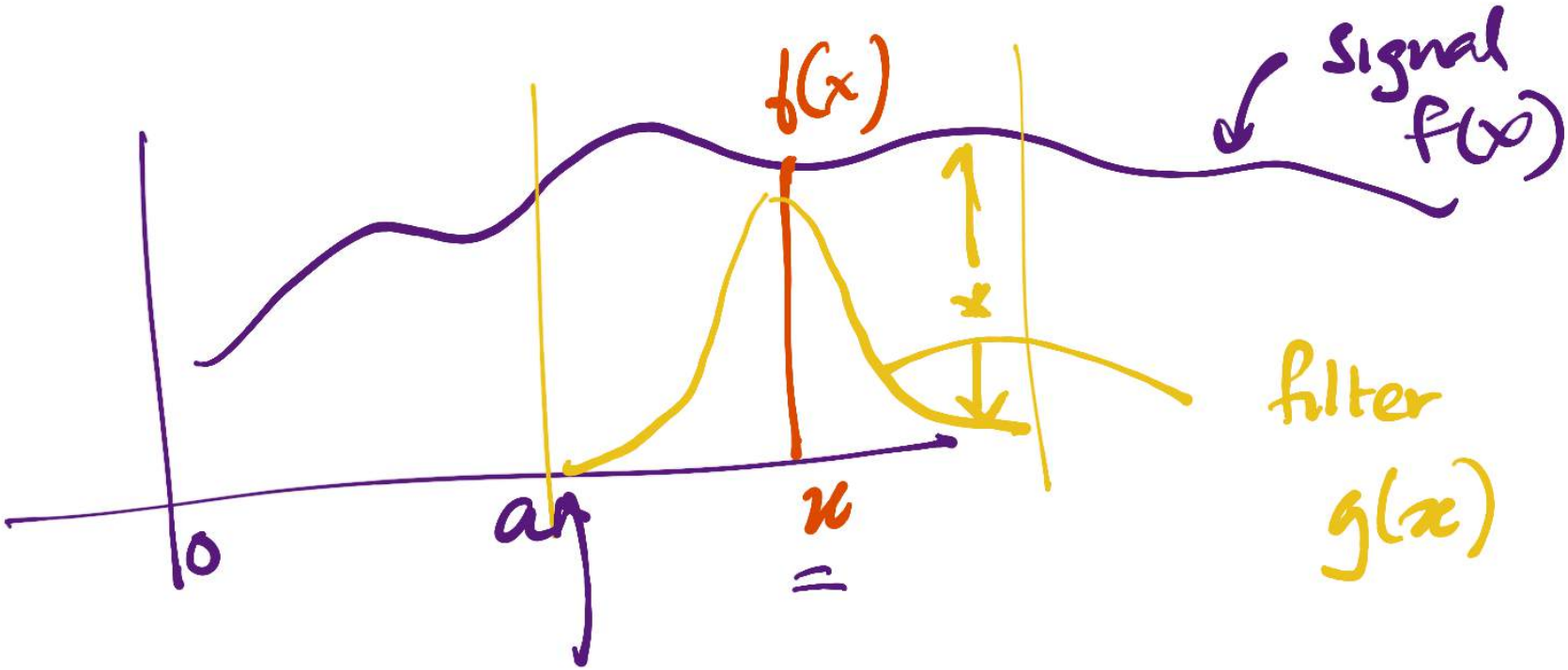


Image detection – look for a "small" pattern

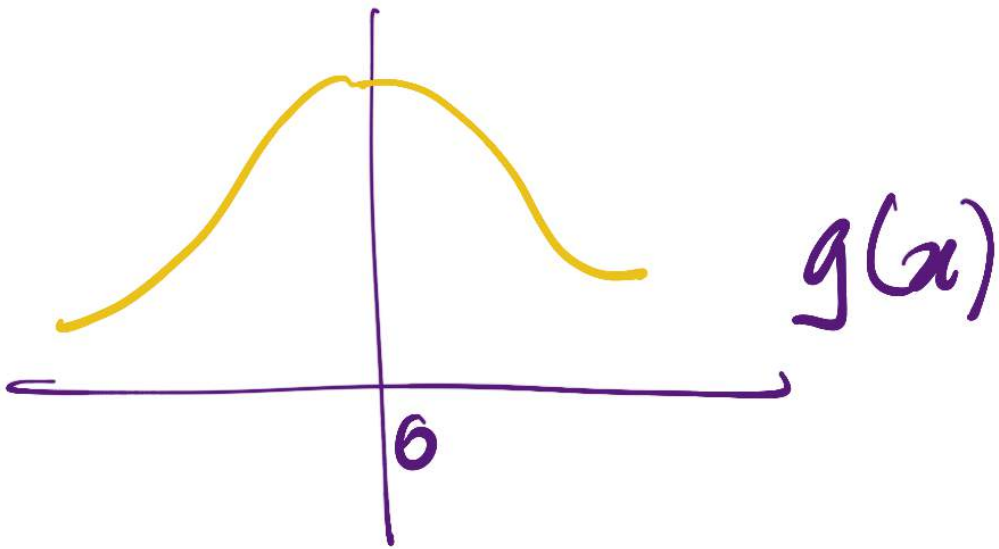
Set up a small "window" to search for the pattern

Mathematically – convolution





$f(a)\, g(x-a)$



At each point new value

$$h(x) = \int_{a=-\infty}^{\infty} f(a)\, g(x-a)\, da$$

$f(a)$: signal; $g(x-a)$: filter

"kernel"

$$(f * g)(x)$$

Computational – we need to do infinite computation at each point!

Assume that window = kernel is nonzero only on a finite interval

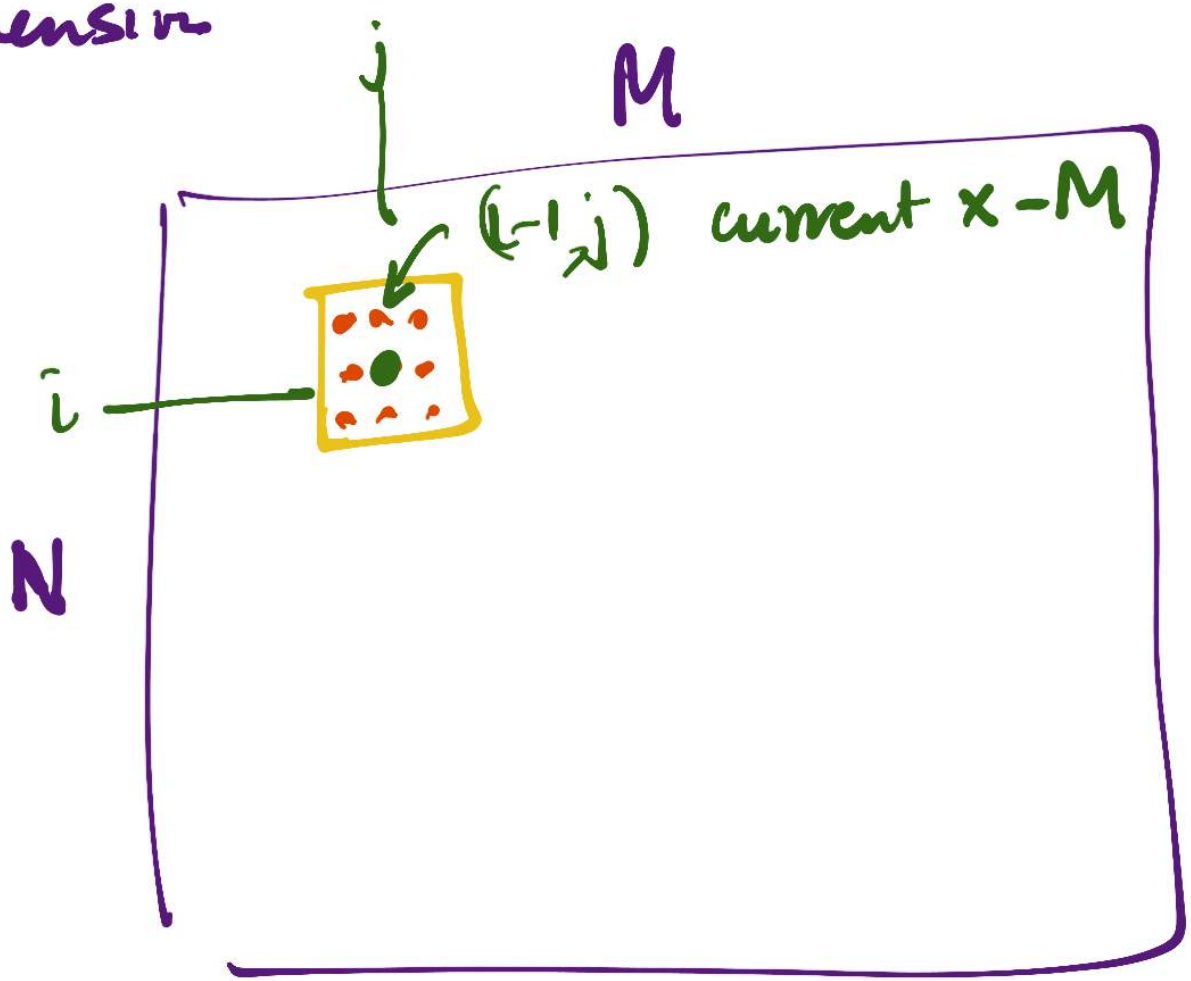
Discrete values + finite kernel

$$K(m) \quad \cdots \quad K(1) \quad K(0) \quad K(-1) \quad \cdots \quad K(-m)$$

$$\times \qquad \times \qquad \times \qquad \times \qquad \cdots \qquad \times$$

$$X(t-m) \quad \cdots \quad X(t-1) \quad X(t) \quad X(t+1) \quad \cdots \quad X(t+m)$$

2 Dimension



Edge detection in 0-1 image

[-1 1]

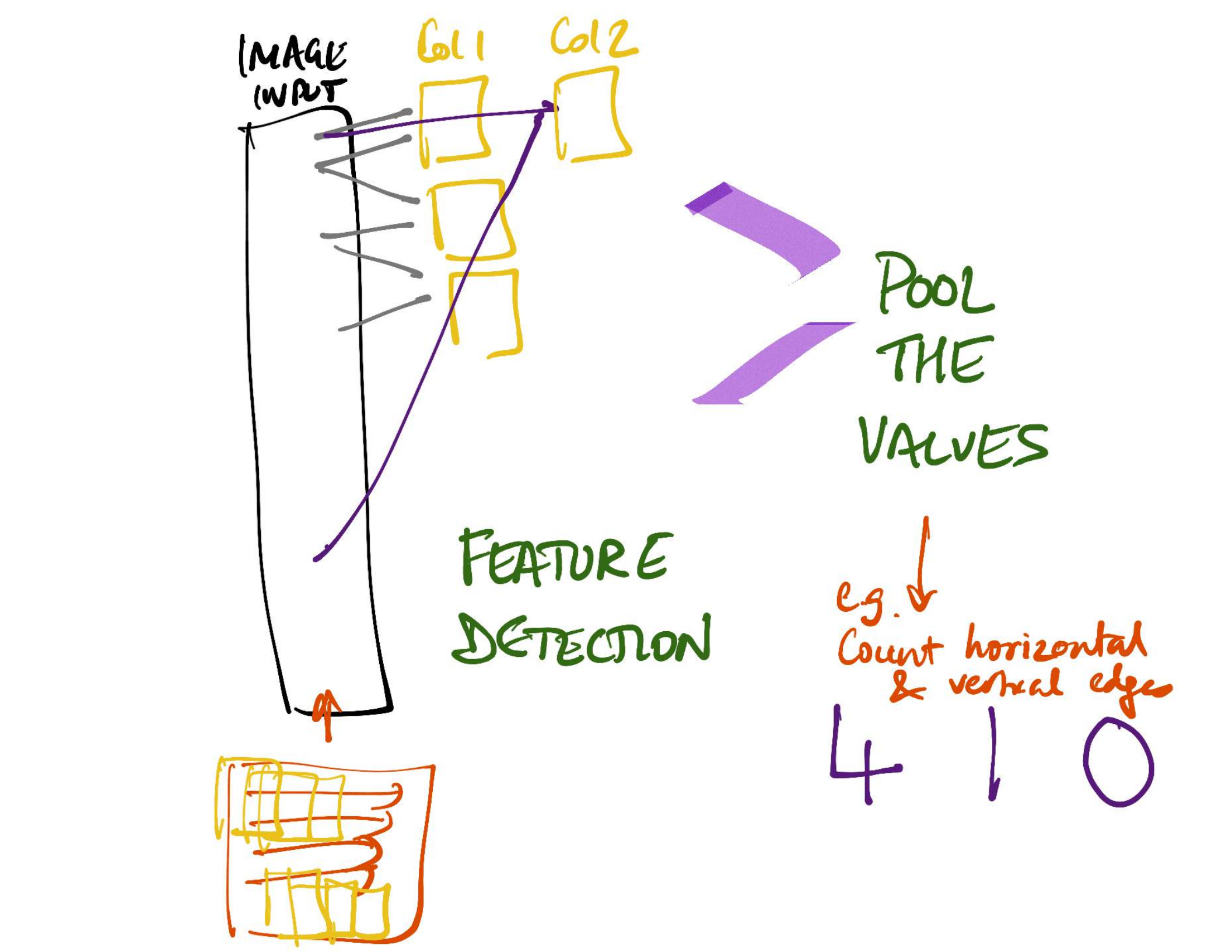
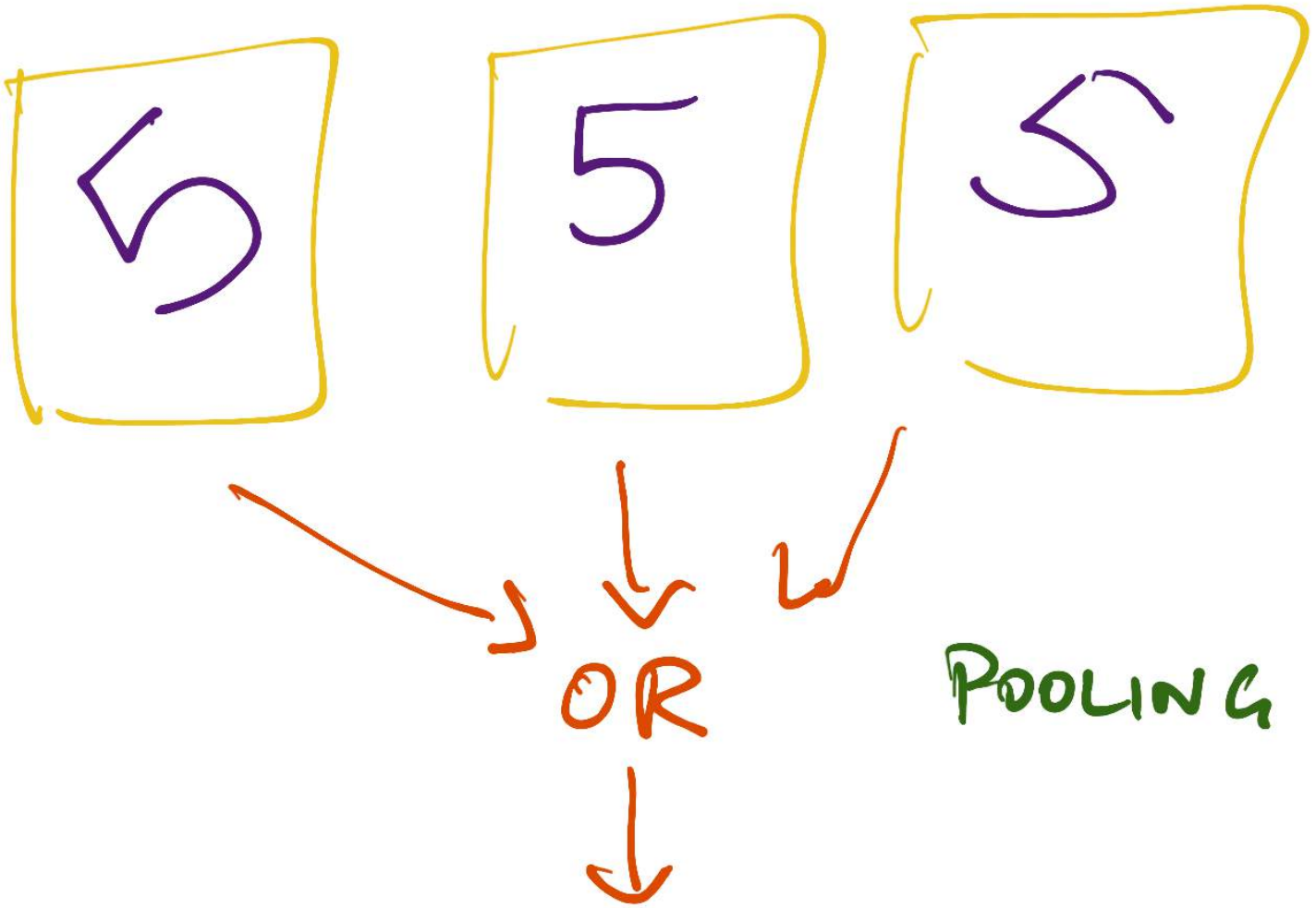


IMAGE INPUT
Col 1
Col 2
POOL THE VALVES
FEATURE DETECTION
e.g.
Count horizontal & vertical edges
4 1 0



Historically, used since early days of neural networks ~ 1990's

Completely connected "deep" networks ~ 2012

Main aspect that makes CNN's attractive computationally is that each hidden node is identical

Learn a single set of weights

- Network is sparse
- Units are identical

Fewer parameters to learn

Less space to "store" the network

Gradient Descent

Backpropagation

1. Forward pass to compute outputs
2. Reverse pass to compute gradients

No saving forward pass

Reverse - one set of parameters per filter

Book says

Maybe a mental block?

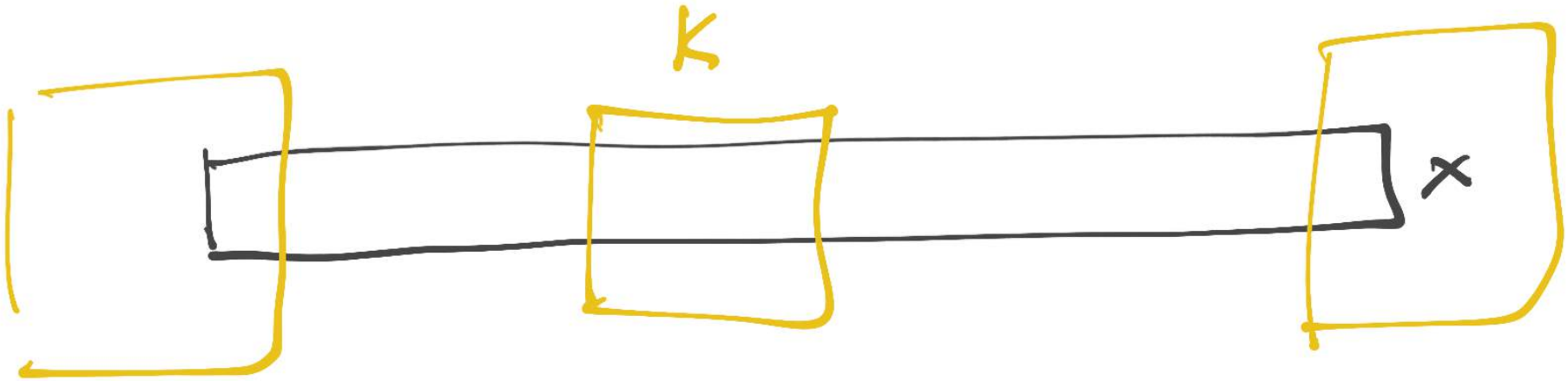
People believed complete networks were too complex to apply backpropagation

Technically

Complete network subsumes CNN

CNN is more "explainable"

Boundaries ?



Value is defined only if K completely overlaps input

- Output is smaller than input
- Information loss in later layers

Padding

Add 0's to supply values of input at boundaries

Valid convolution - no padding
- Shrinks output wrt input

Same convolution - exactly enough padding

Full convolution - extra padding to visit each point multiple times

Shifting the window ("Stride")

Consider (R, G, B) colour image

Stride may vary according to which component we are analyzing

Choosing kernels

By hand

Automated approaches via unsupervised learning

CNN's - Hidden nodes correspond to manually identified features

Deep learning with complete networks

Each hidden layer is an "automatically discovered" feature

- How many layers
- What do these features mean?