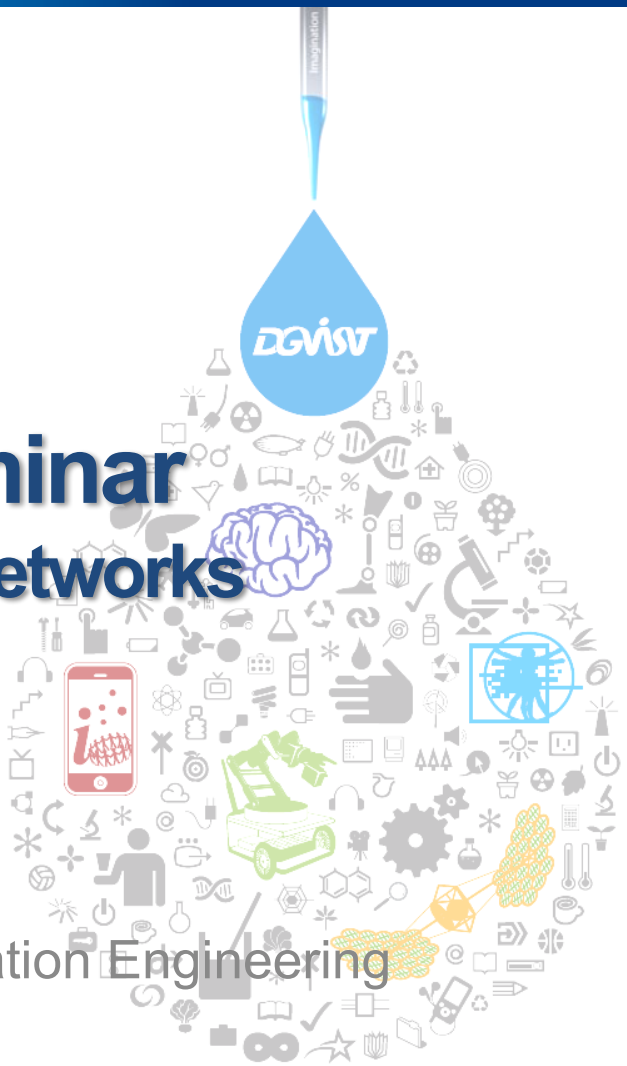


Deep Learning Seminar
Chapter 9. Convolutional Networks

Keonwoo Noh

Department of Information and Communication Engineering

DGIST

2017.09.27

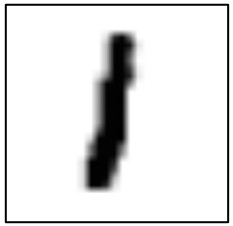
Chapter 9. Convolutional Networks

- **Part 1**
  - 9.0 Introduction
  - 9.1 The Convolution Operation
  - 9.2 Motivation
  - 9.3 Pooling
  - 9.4 Convolution and Pooling as an Infinitely Strong Prior
  - 9.5 Variants of the Basic Convolution Function
  - **Part 2**
  - 9.6 Structured Outputs
  - 9.7 Data Types
  - 9.8 Efficient Convolution Algorithms
  - 9.9 Random or Unsupervised Features
  - 9.10 The Neuroscientific Basis for Convolutional Networks
  - 9.11 Convolutional Networks and the History of Deep Learning

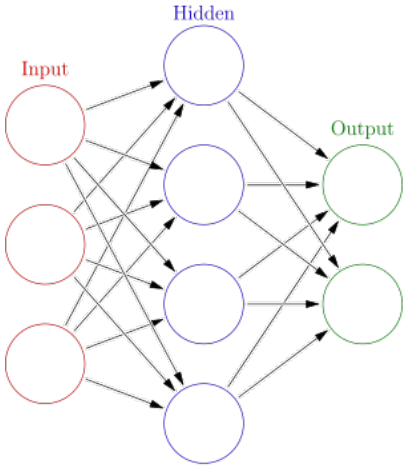
Introduction

Multi-Layer Perceptron (MLP) for image recognition task

- **Use intensity of each pixel as 1D input vector**
- **There several problems such as:**
  - **Required extremely large number of parameter**
  - **No invariance to shifting, scaling**
  - **Unable to deal with variable size input data**



28x28 image



784 input neurons and large number of parameters

Image source : https://en.wikipedia.org/wiki/Artificial_neural_network

Convolutional neural network (CNN)

- **CNN consists of:**
  - Convolutional layer
  - Pooling layer
  - Fully connected layer
- **CNN learns convolutional layer's kernel**
  - To make computation easier , CNN uses cross-correlation instead of convolution

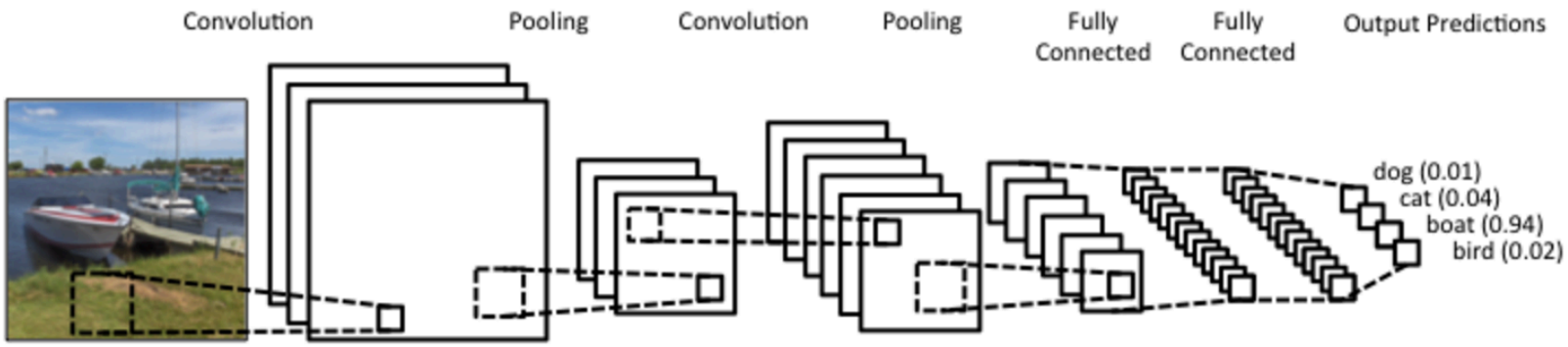


Image source : https://www.kdnuggets.com/2015/11/understanding-convolutional-neural-networks-nlp.htm

ILSVRC

- **ImageNet Large Scale Visual Recognition (ILSVRC)**
  - One of the biggest computer vision contest
  - 1M images
  - 1000 labels

Convolutional Neural Network

- AlexNet (Krizhevsky et al.) won the ILSVRC with significant difference about accuracy with 2nd highest accuracy one
- It also showed CNN can be accelerated through GPU

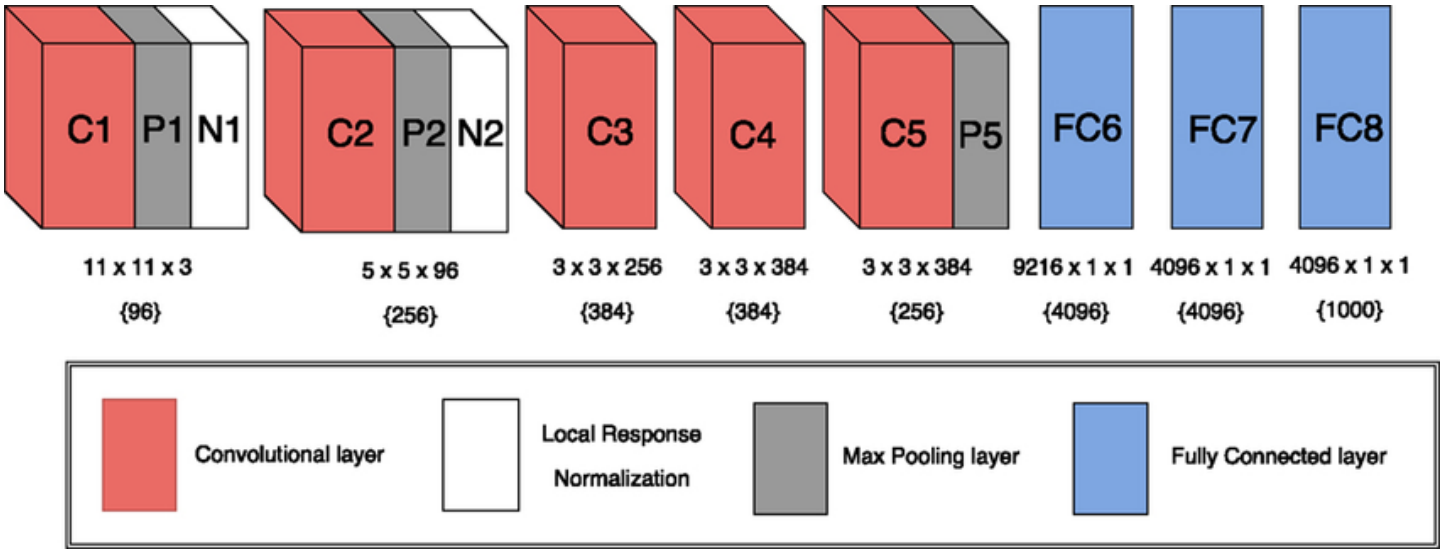


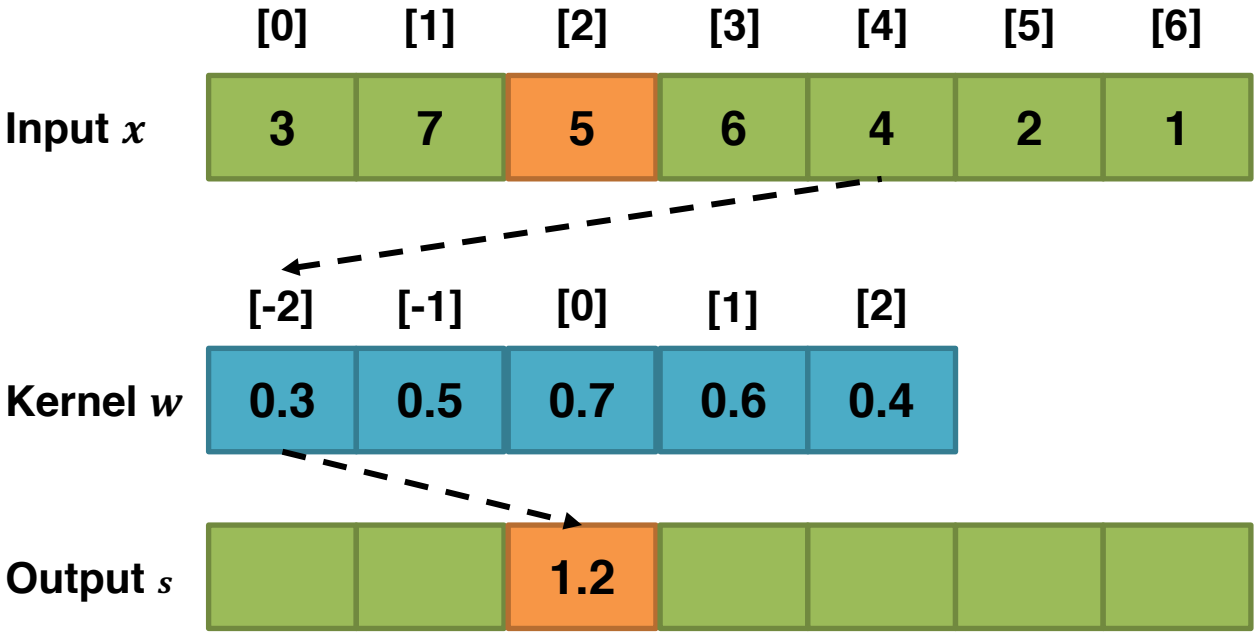
Image source : https://www.researchgate.net/figure/289928157_fig7_Figure-1-An-illustration-of-the-weights-in-the-AlexNet-model-Note-that-after-every

The Convolution Operation

Convolution in the context of discrete

- **Discrete 1D convolution in computer**

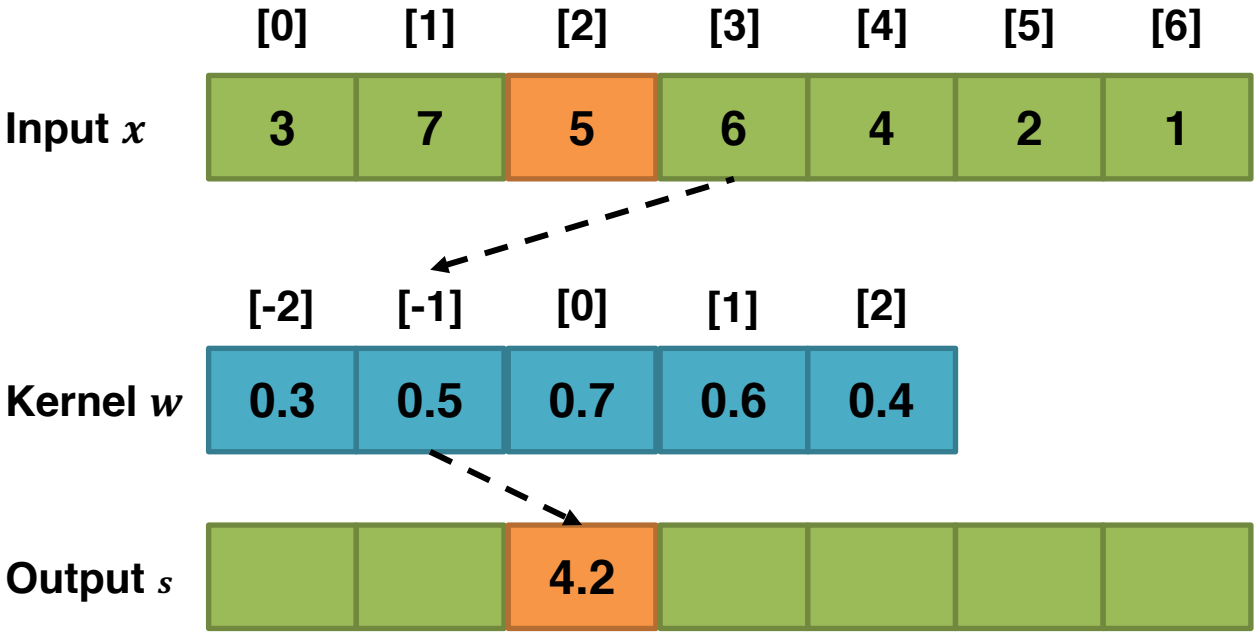
$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$



Convolution in the context of discrete

- **Discrete 1D convolution in computer**

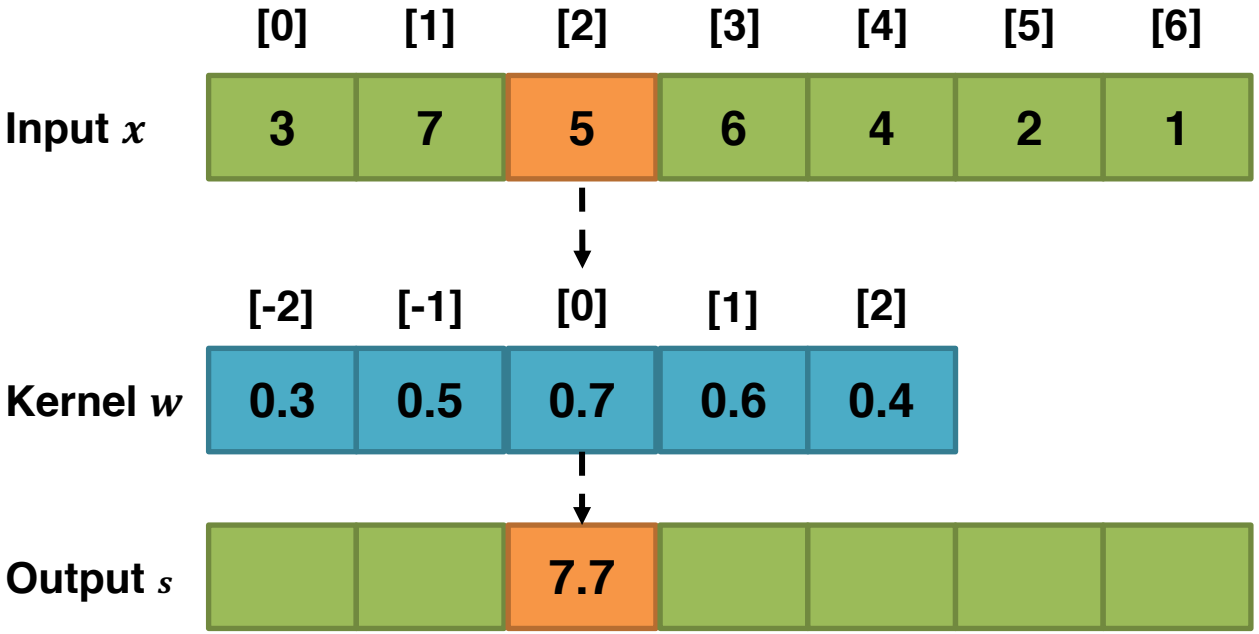
$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$



Convolution in the context of discrete

- **Discrete 1D convolution in computer**

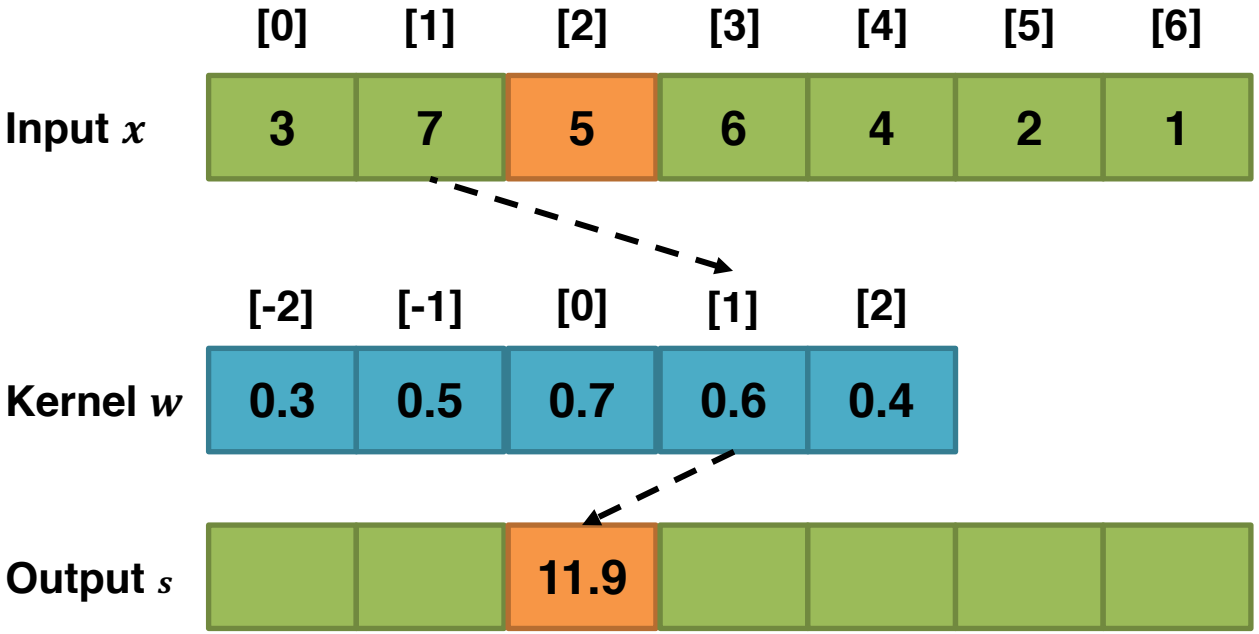
$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$



Convolution in the context of discrete

- **Discrete 1D convolution in computer**

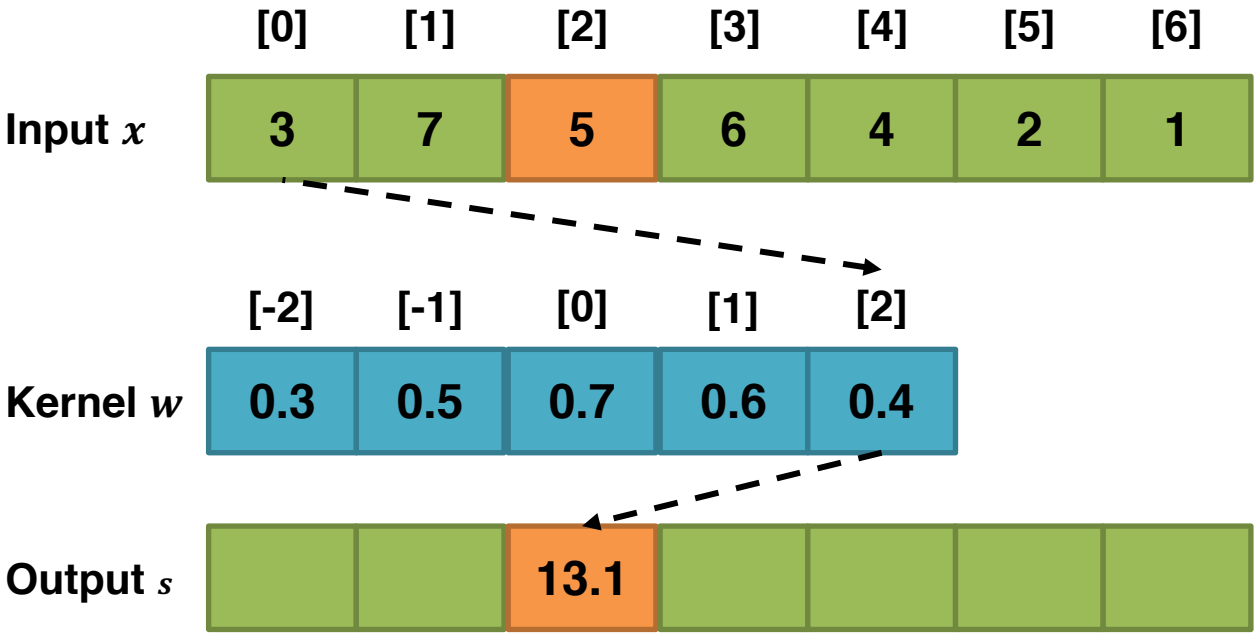
$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$



Convolution in the context of discrete

- **Discrete 1D convolution in computer**

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



Convolution vs. Cross-correlation

- **Convolution**

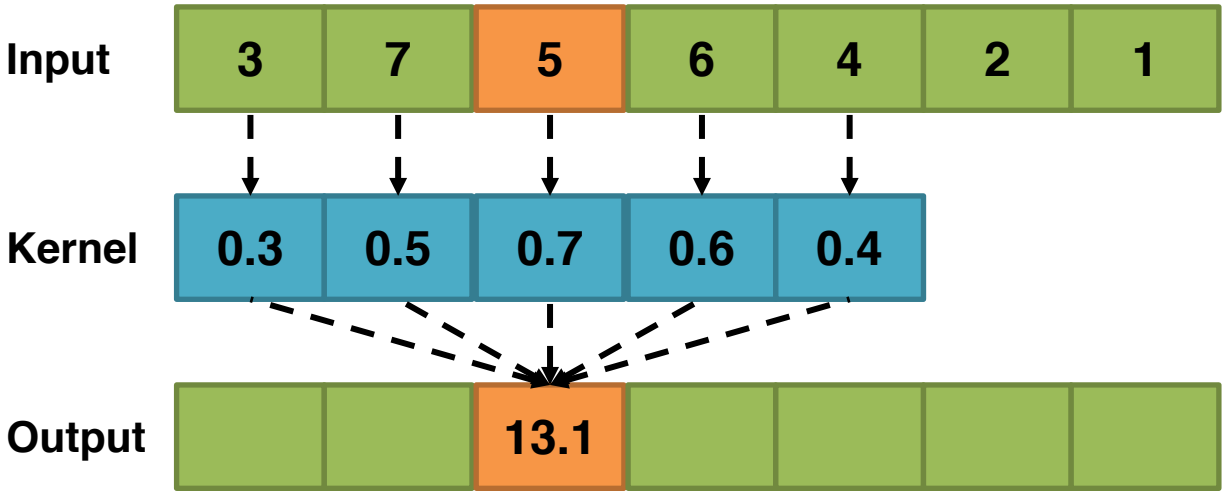
$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n) K(m,n).$$

- **Cross-correlation**

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n).$$

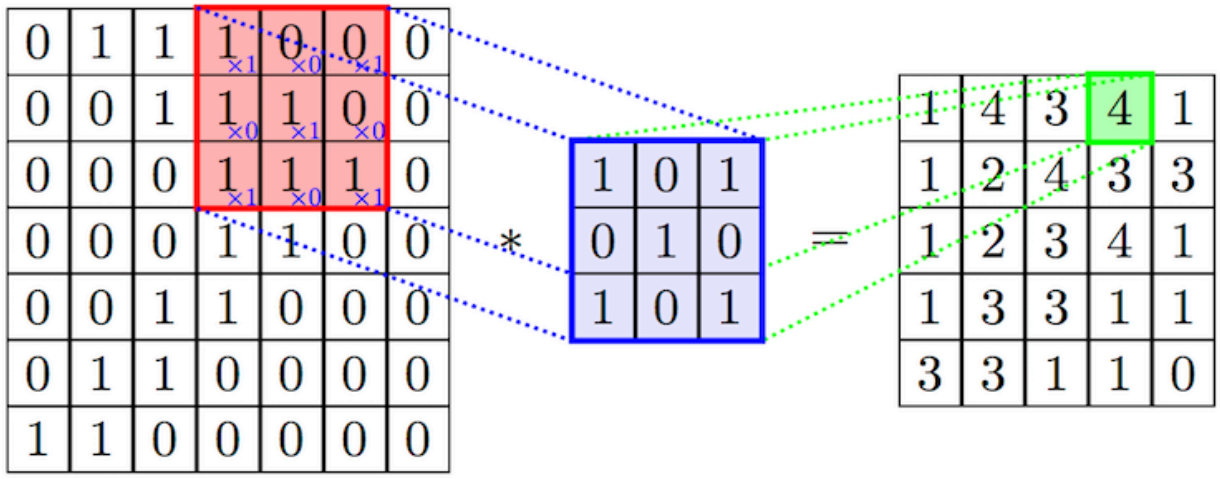
Cross-correlation

- Discrete 1D cross-correlation



2D convolution

- Convolutional Neural Network (CNN) learns appropriate value of kernel
- CNN learns flipped kernel if use convolution instead of cross-correlation



source : https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html

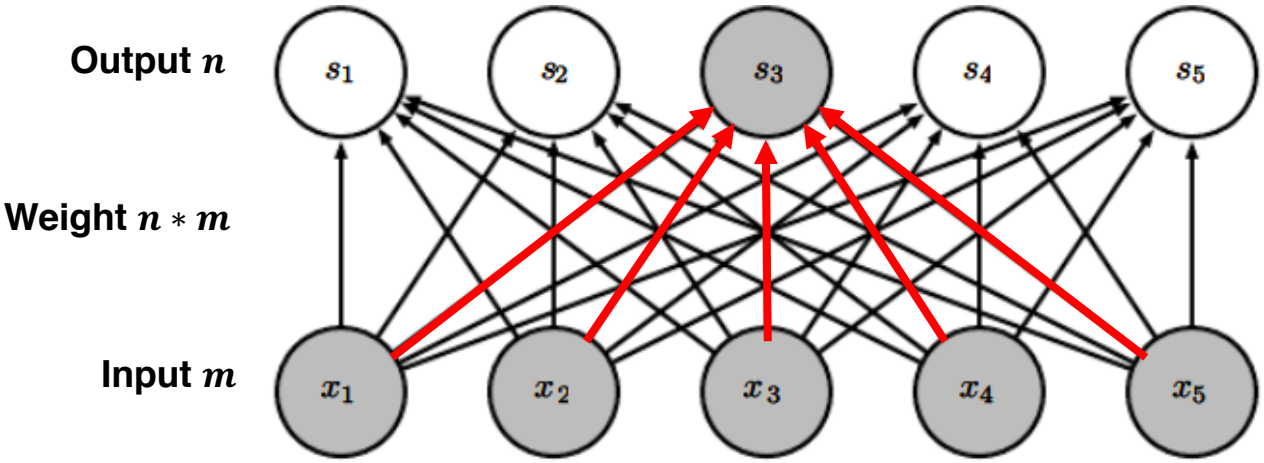
Motivation

Benefits when includes convolution layer in deep learning

- **Convolution leverages three important ideas :**
  - Sparse interaction
  - Parameter sharing
  - Equivariant representation

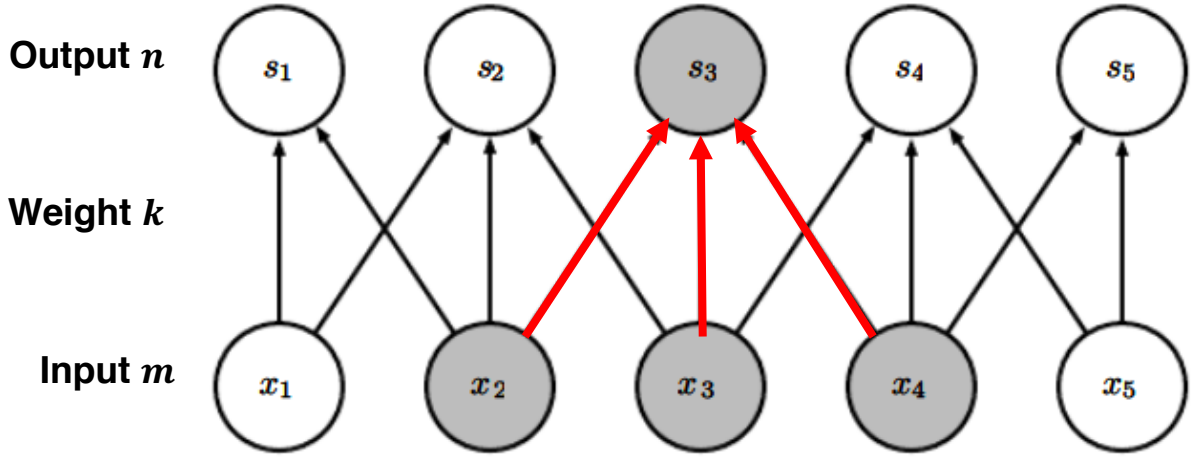
Sparse interaction (traditional neural net)

- Neurons of input layer and hidden layer are “fully connected”
- Complexity is $O(n * m)$



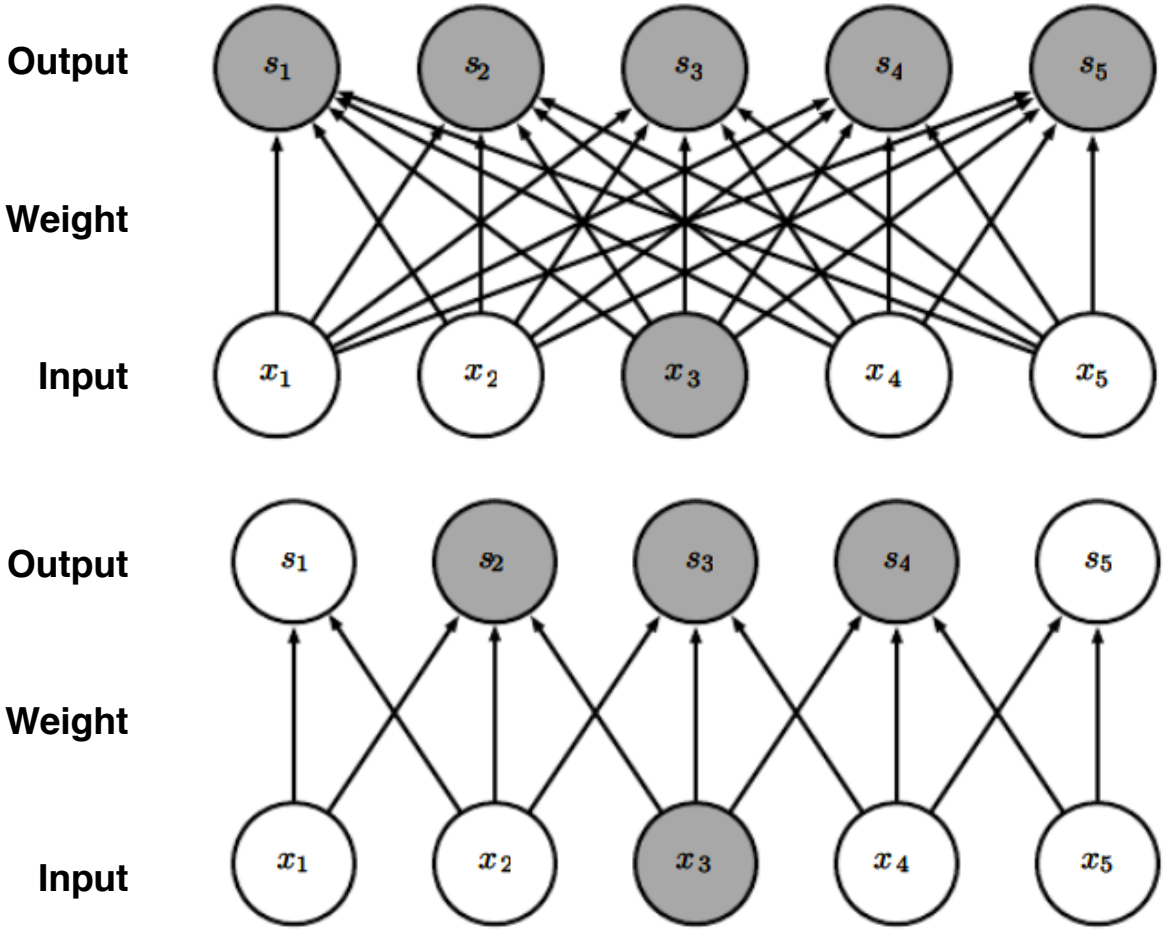
Sparse interaction (conv)

- To make the kernel smaller than the input causes sparsity
- Complexity is $O(n * k)$
- Fewer parameters are required to be stored and computed



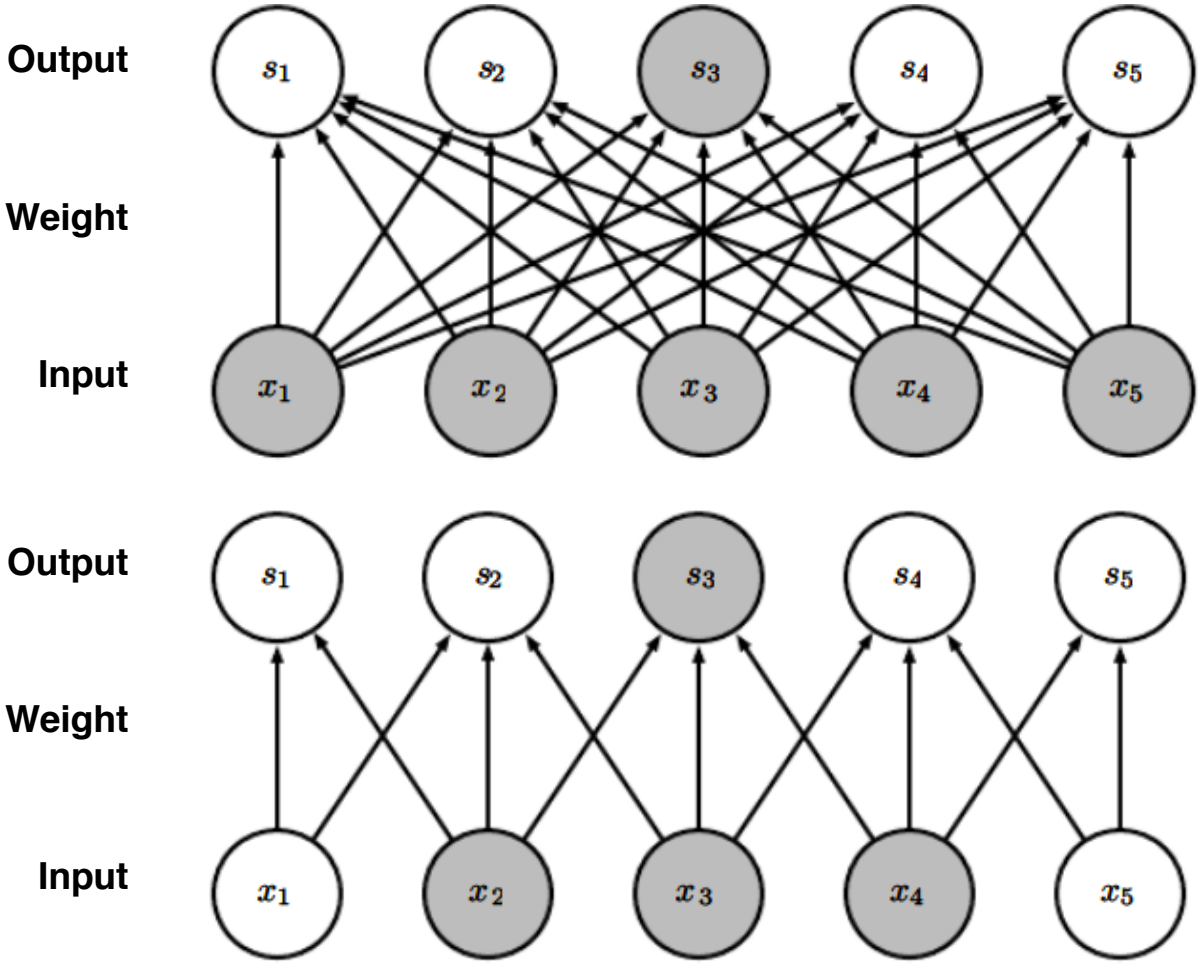
Sparse interaction (comparison)

- Output neurons influenced by single input neuron



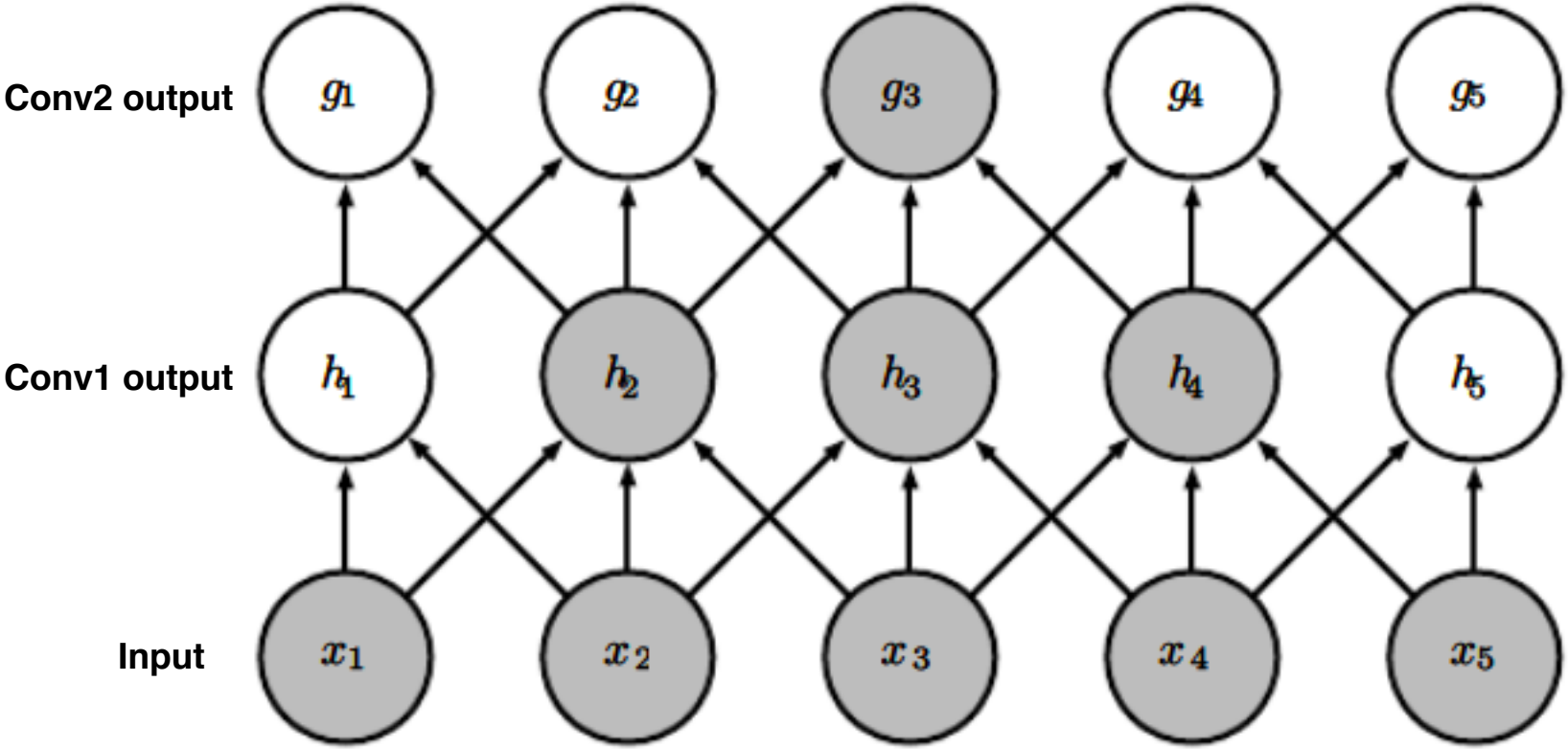
Sparse interaction (comparison)

- Input neurons influence single output neuron



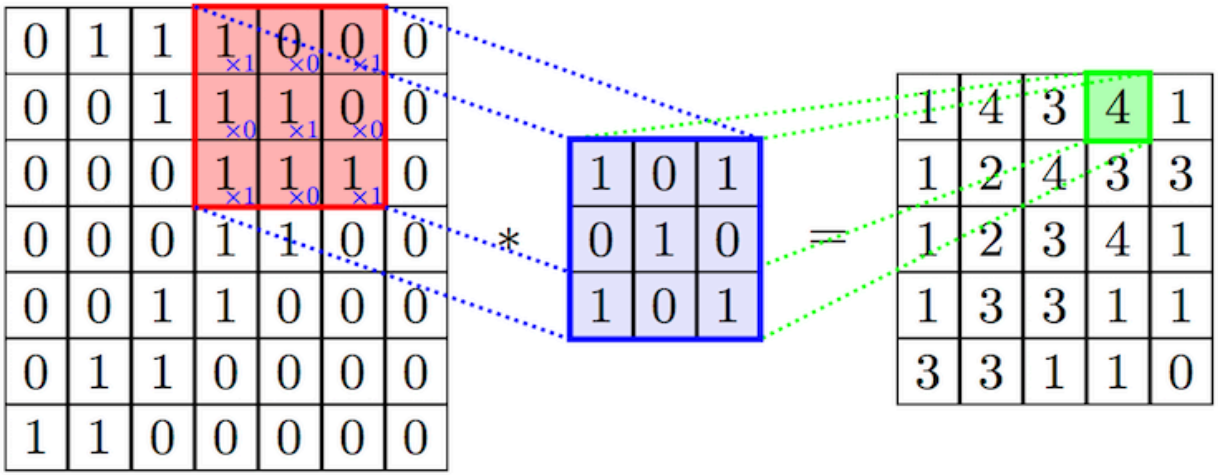
Sparse interaction (indirectly fully influenced)

- Even though direct connection is sparse, indirect connection is influenced almost all of the input neurons if the neural net is deep enough

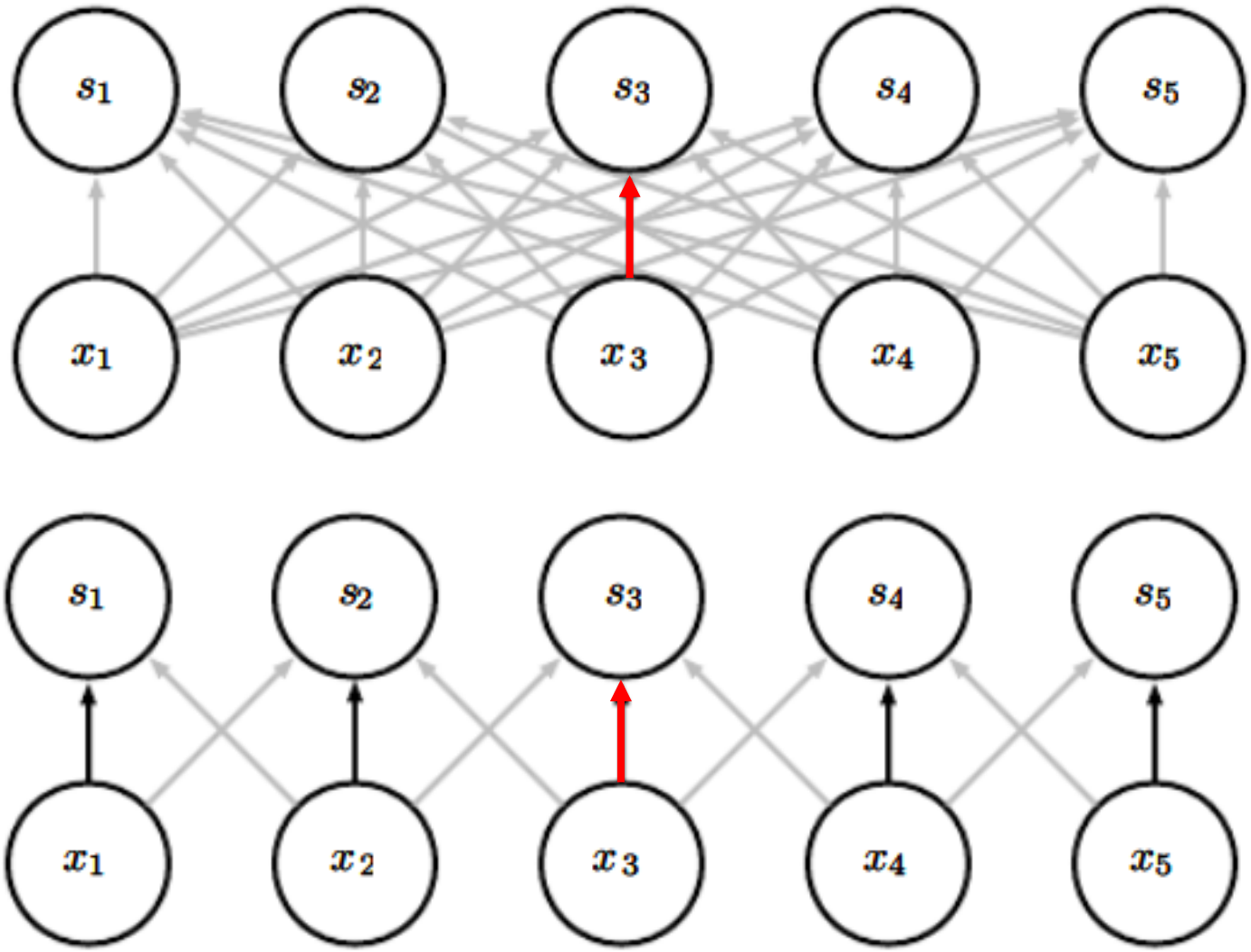


Parameter sharing

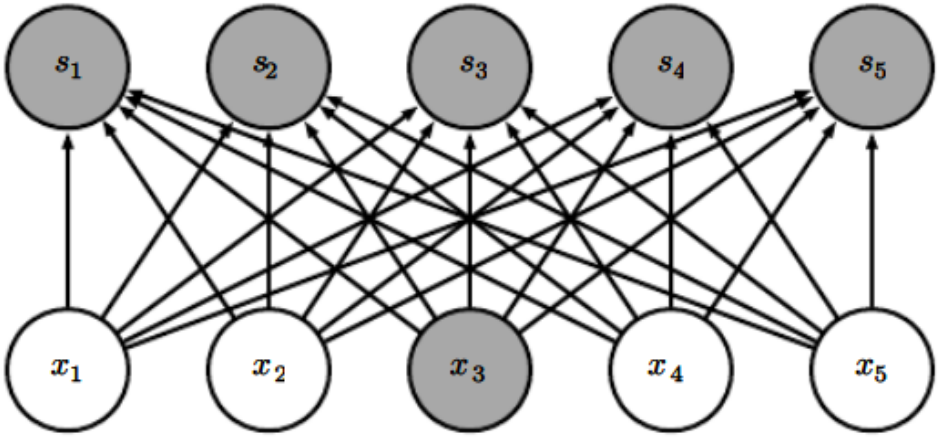
- Kernel is shared to input located every position
- Parameter sharing has some effects :
  - Extracts same features for every location
  - Reduce memory space

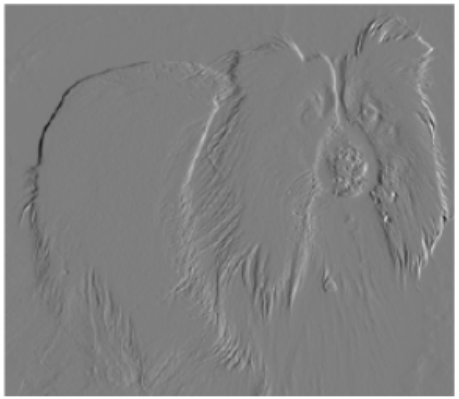


Parameter sharing (extract same feature)



Parameter sharing (memory efficiency)



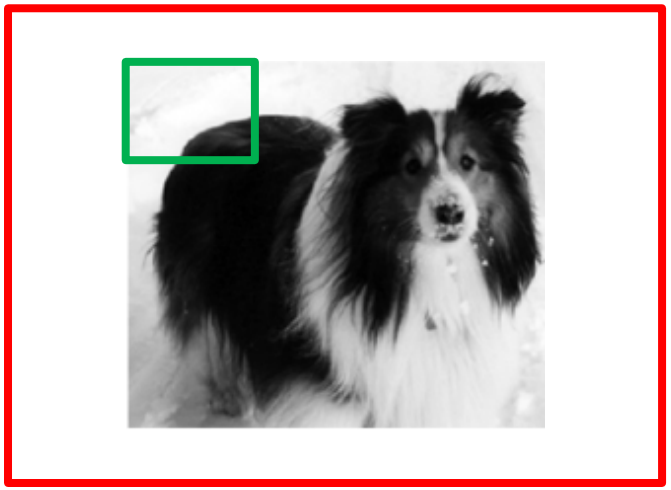


Equivariant representation

- **Convolution is equivariant operation about translate because it exploits shared kernel and stride from the scratch to the end**

Kernel

Image



Dog in the center of the image

Dog in the right side of the image

Pooling

Max pooling

- Summarizes the neighborhood of input

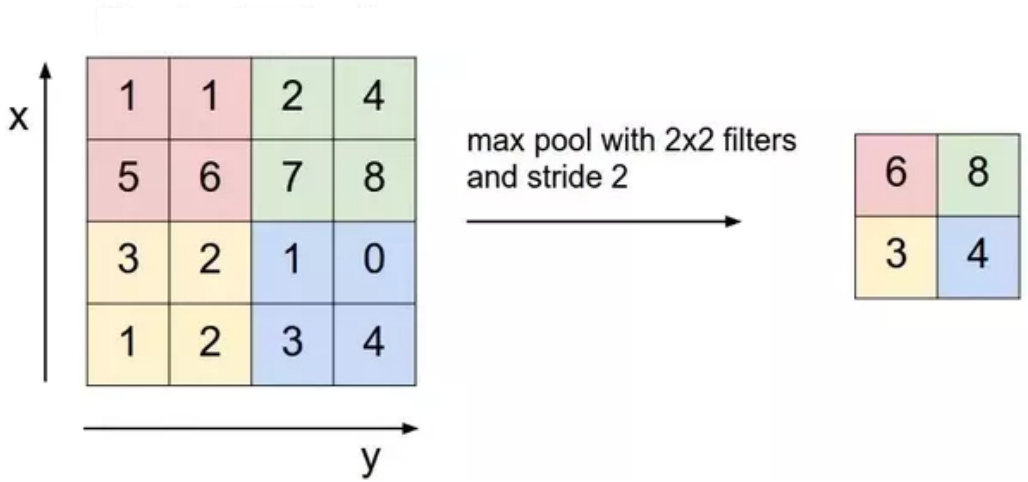
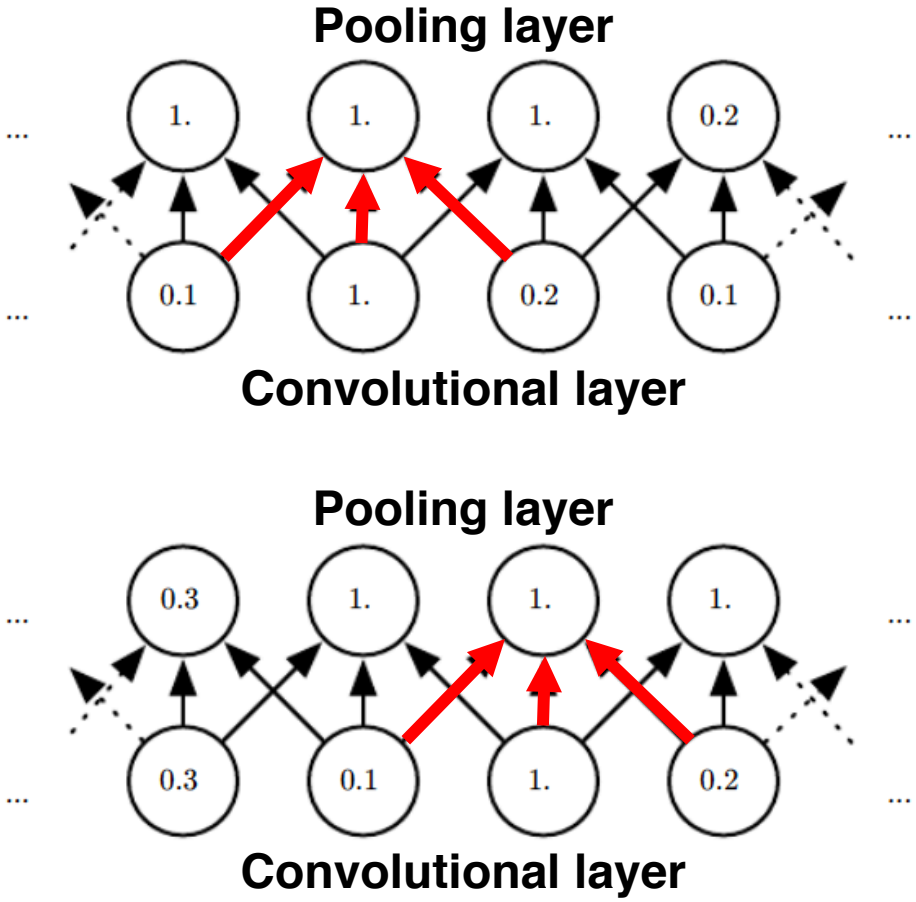


Image source : https://www.pexels.com/photo

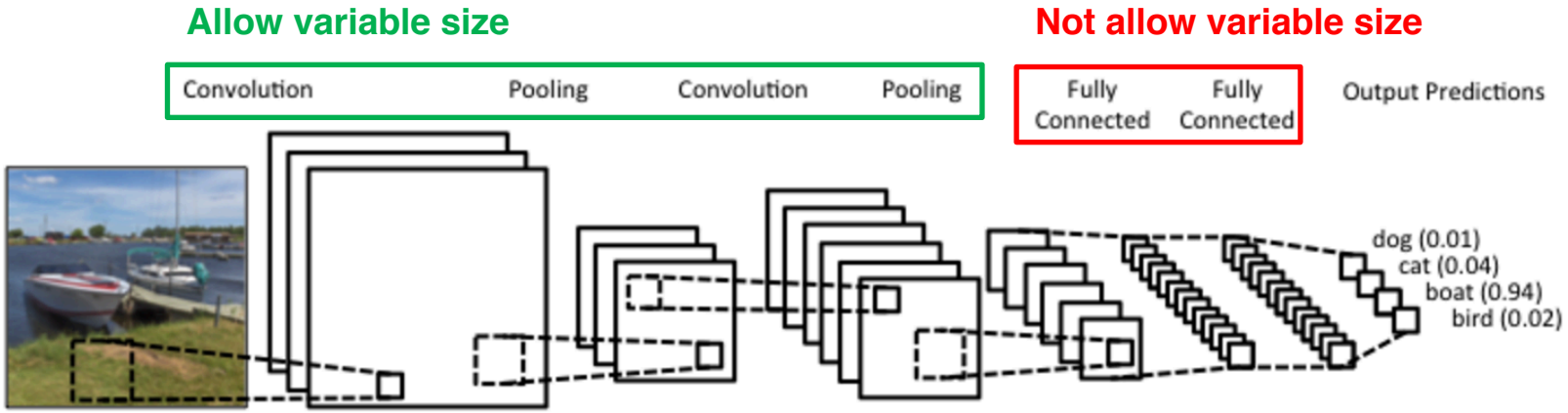
Invariance to local translation

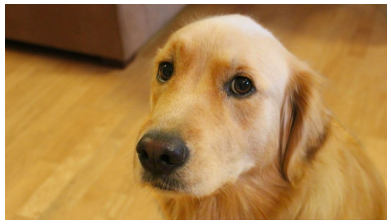
- This property is useful (statistically efficient) if we care more about whether some feature is present than exactly where it is



Input having variable size

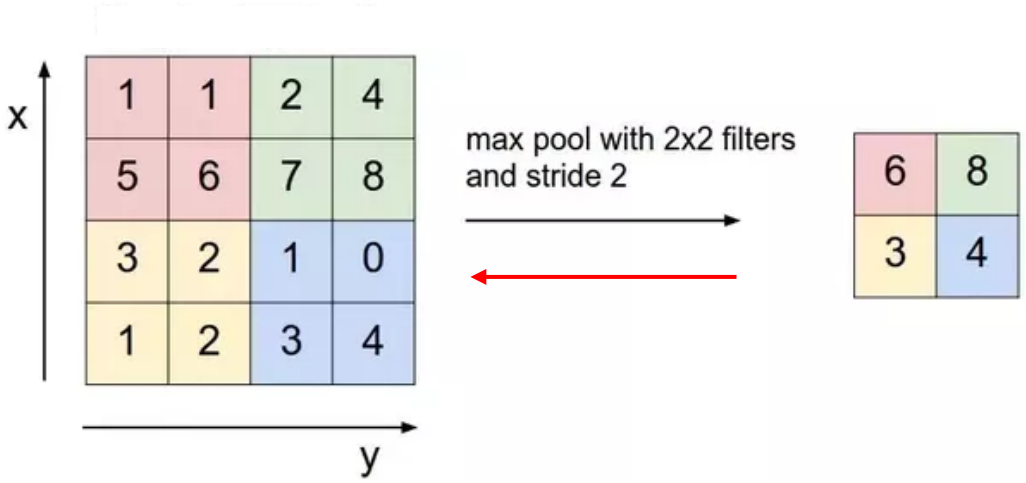
- Classification layers require fixed size of their input
- Pooling make their output fixed size by changing their pooling size, stride and so on





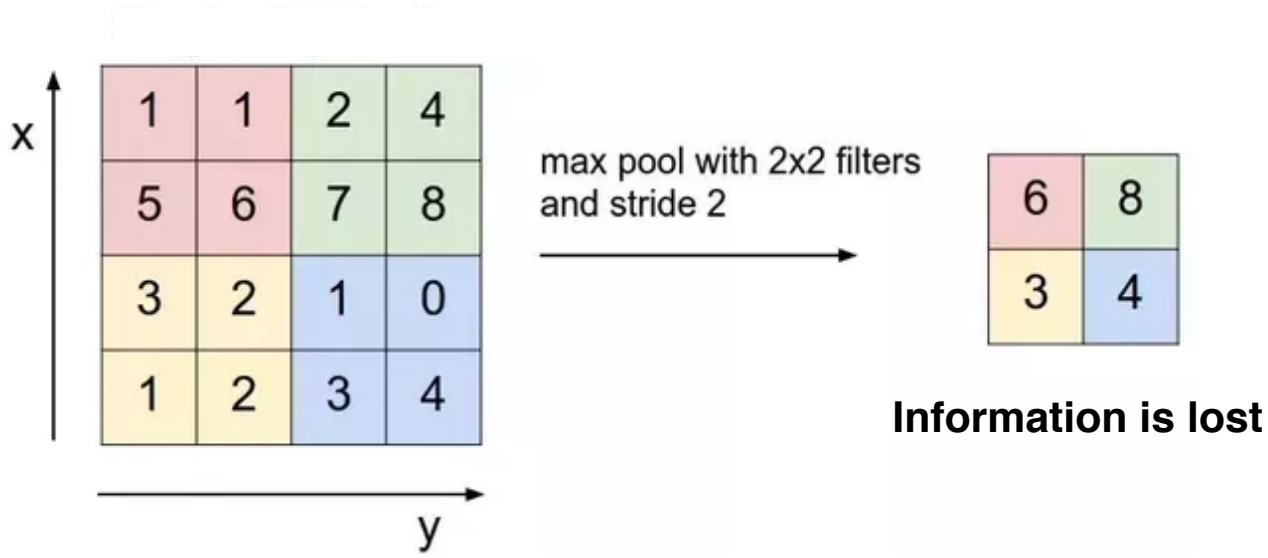
Additional temper of pooling

- Pooling hasn’t inverse operation
- Pooling makes network to invariant to rotation
- In some case, Pooling causes underfitting



Pooling and underfitting

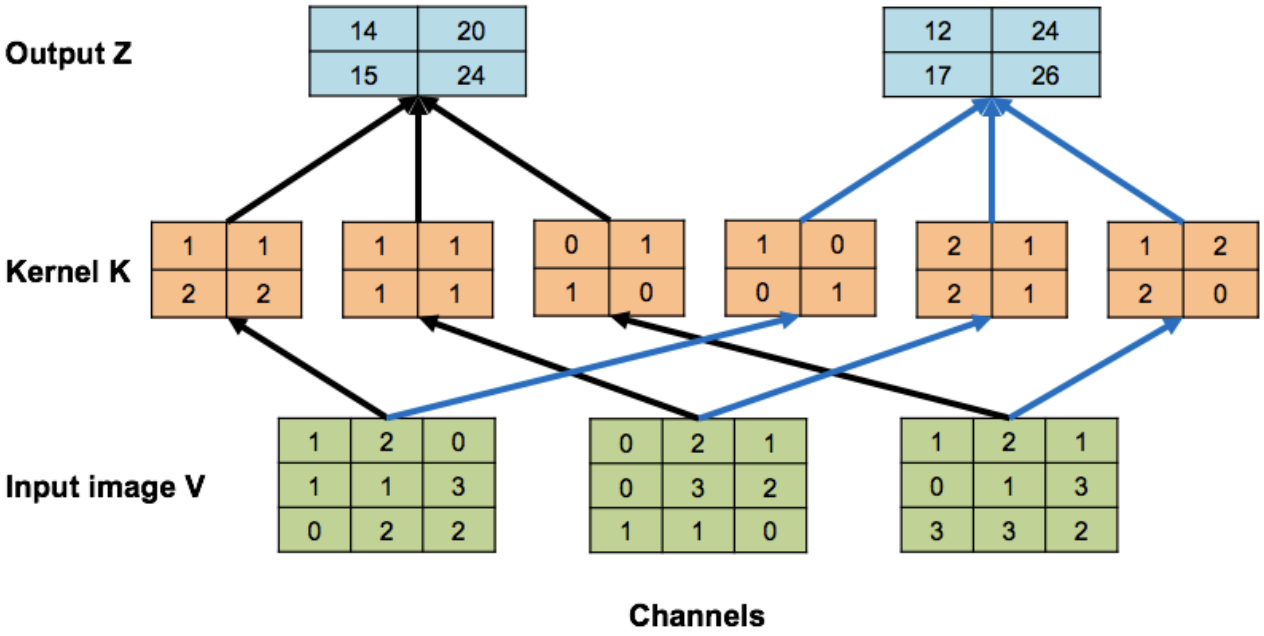
- If a task relies on preserving precise spatial information, then using pooling on all features can increase the training error



Variants of the Basic Convolution Function

3 channels 2D convolution

- Image is usually consists of 3 channels (Red, Green Black)

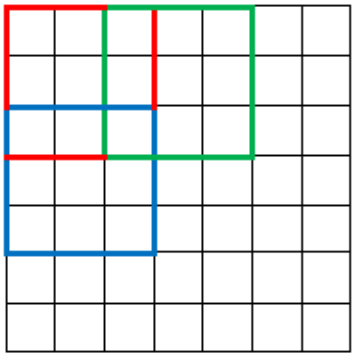


$$Z_{i,j,k} = \sum_{l,m,n} V_{l,j+m,k+n} K_{i,l,m,n}$$

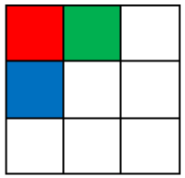
Stride

stride : 2

7 x 7 Input Volume

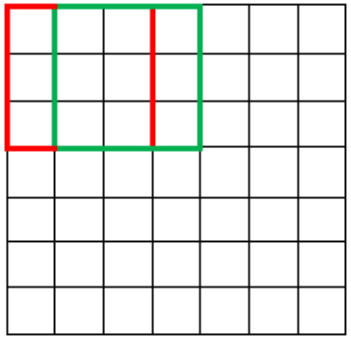


3 x 3 Output Volume

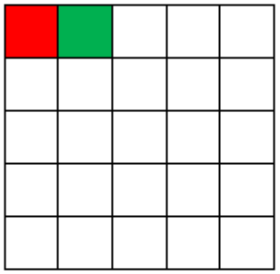


stride : 1

7 x 7 Input Volume

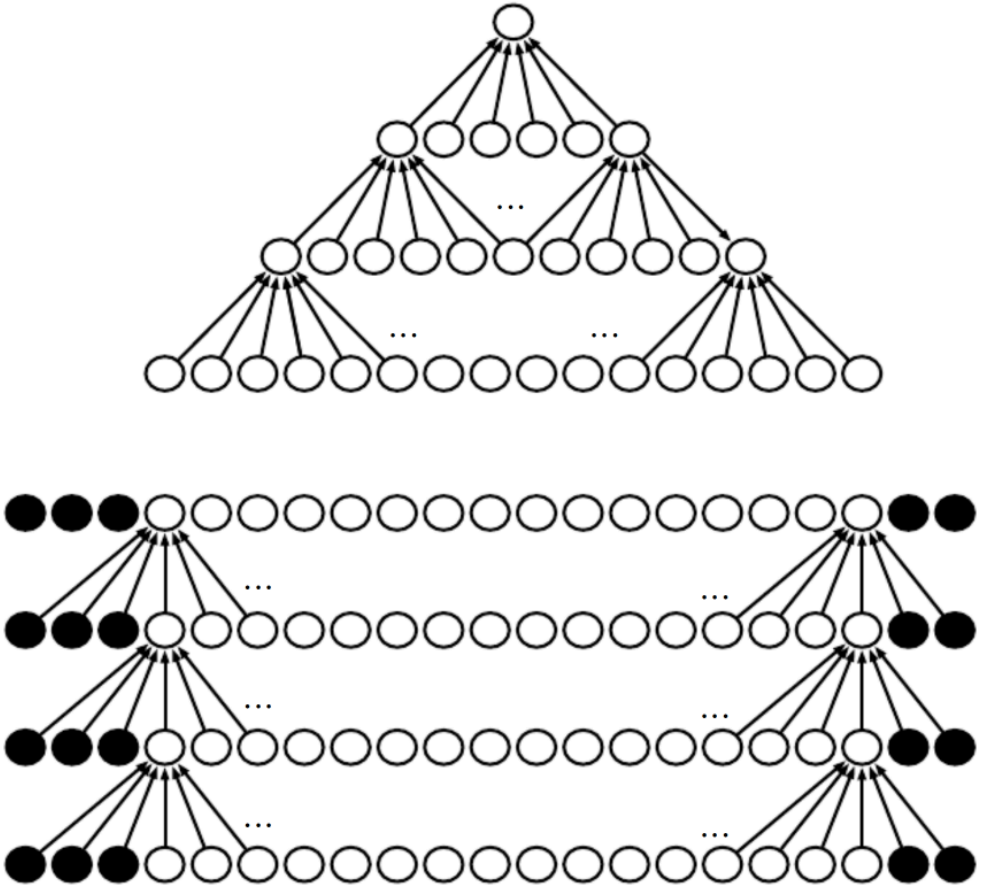


5 x 5 Output Volume



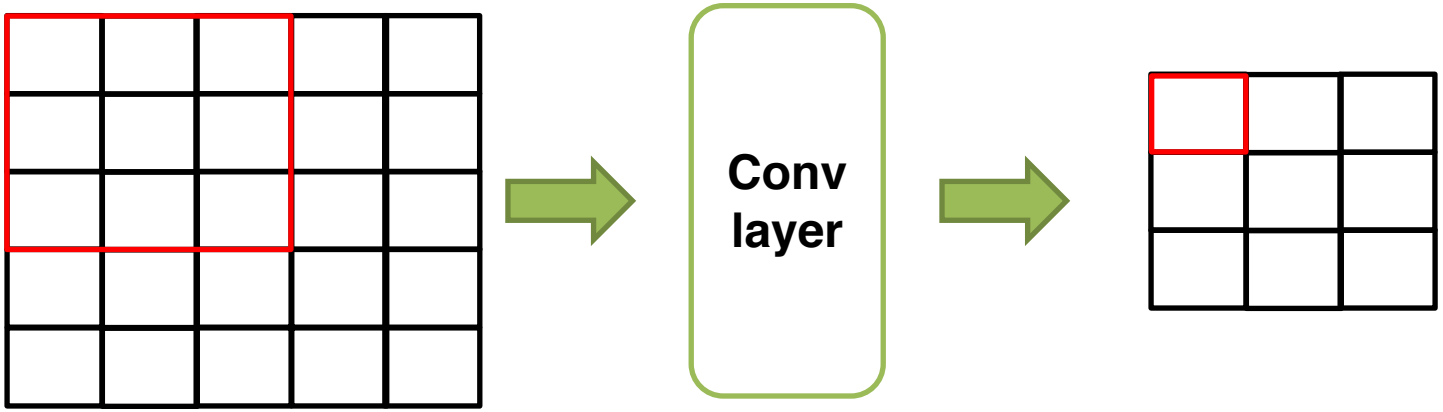
Padding

- Stacked convolutional layer causes shrunk data



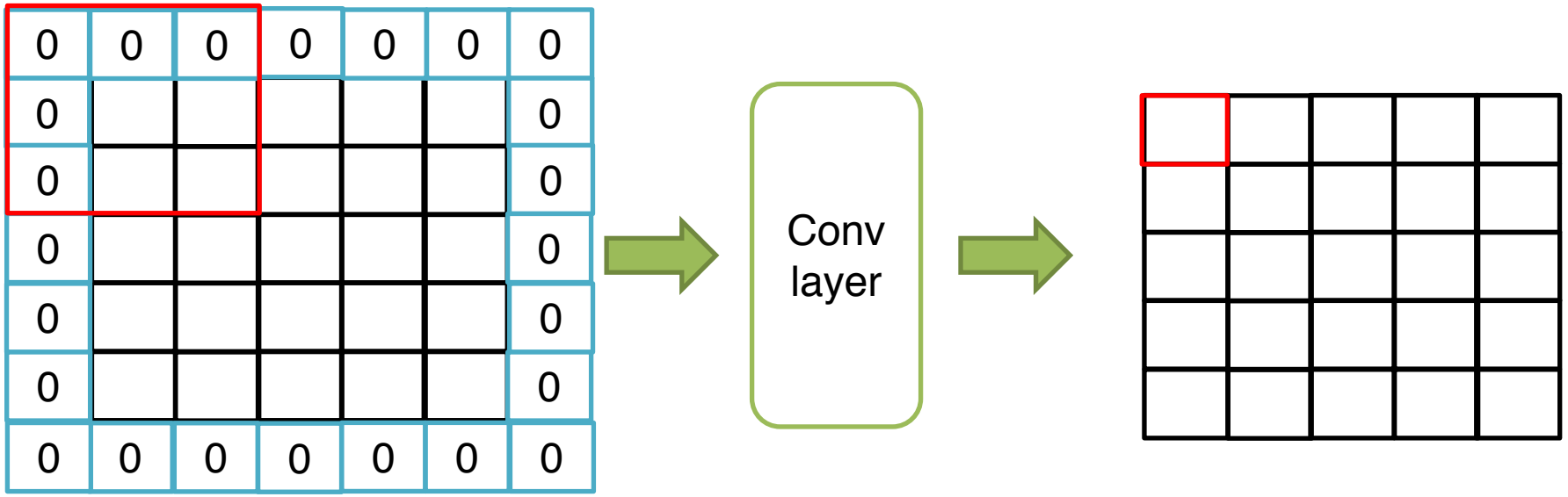
Special case of zero padding

- Valid convolution

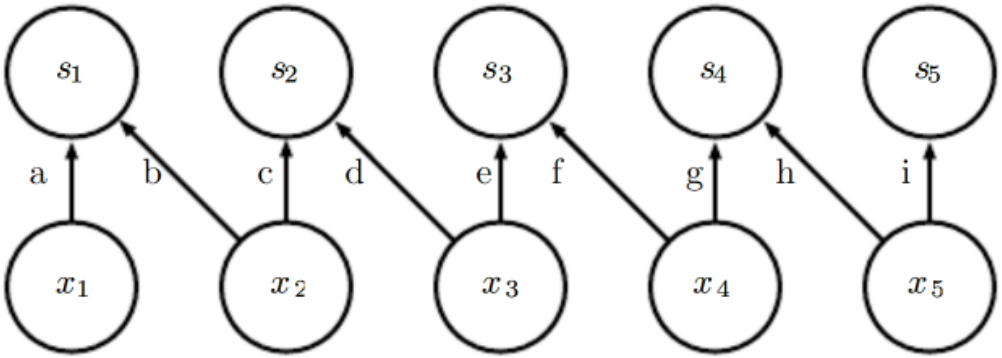


Special case of zero padding

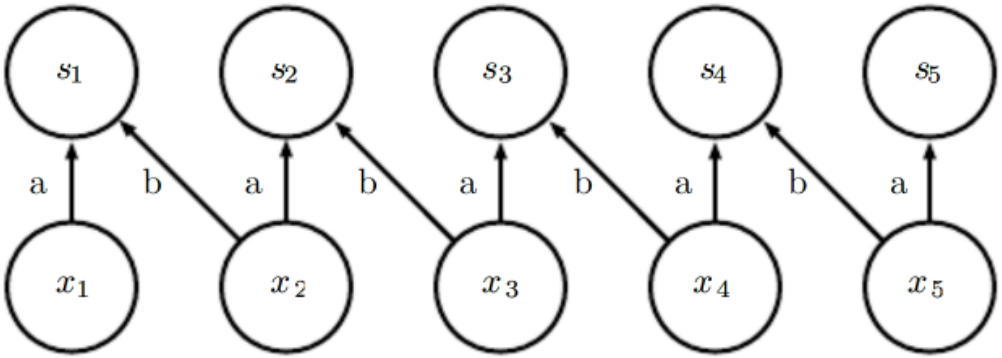
- **Same convolution**



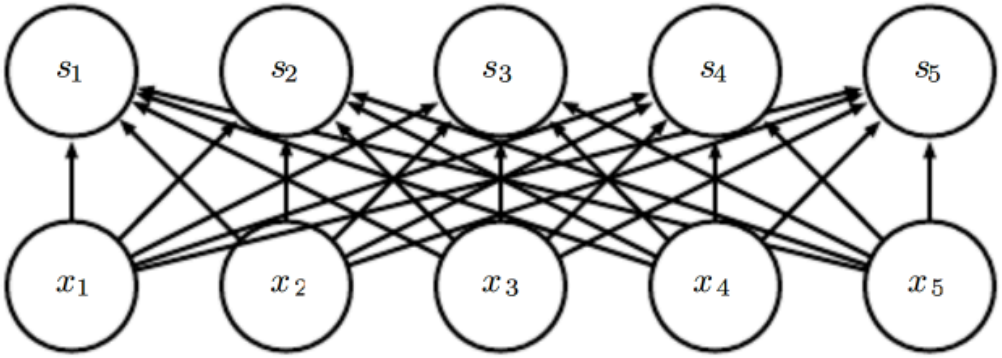
Locally connected layer



Locally connected layer



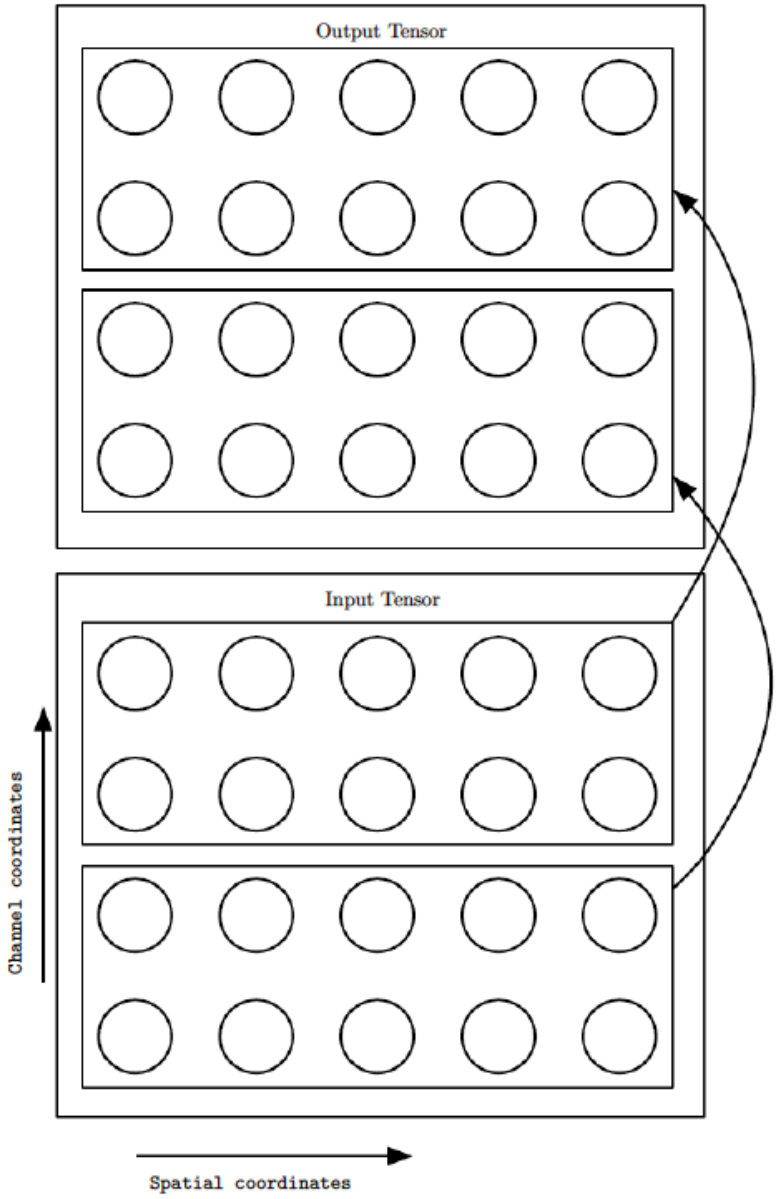
Convolutional layer



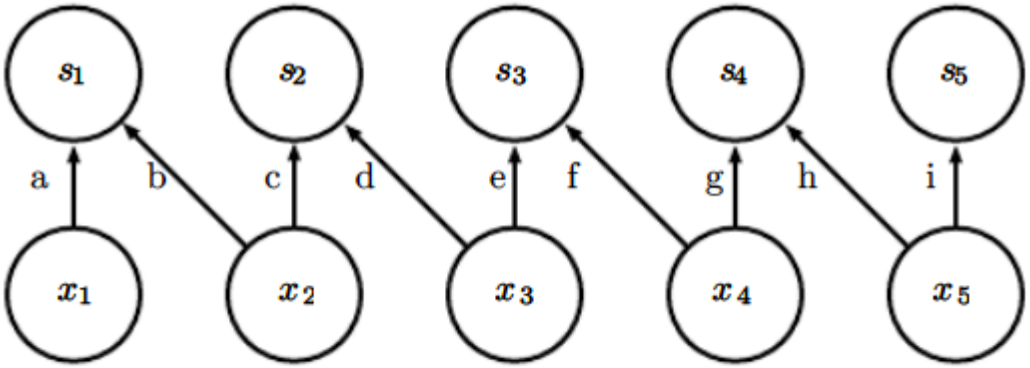
Fully connected layer

Restricted connectivity in convolutional layer

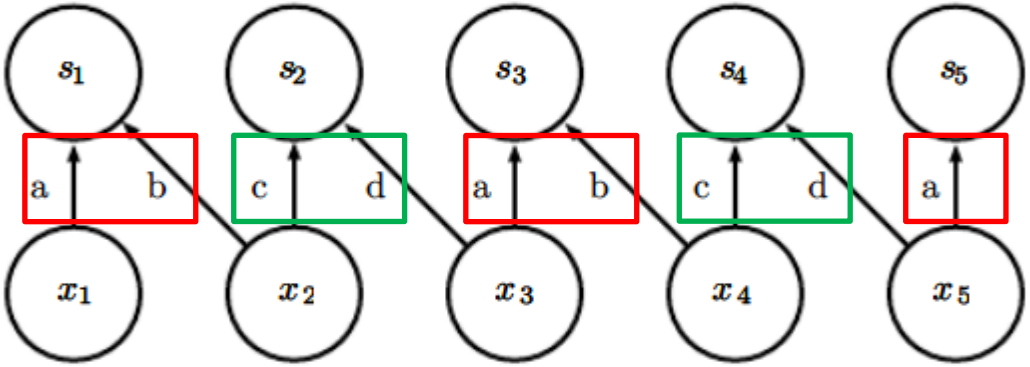
- **In order to reduce :**
  - Required memory
  - Computation



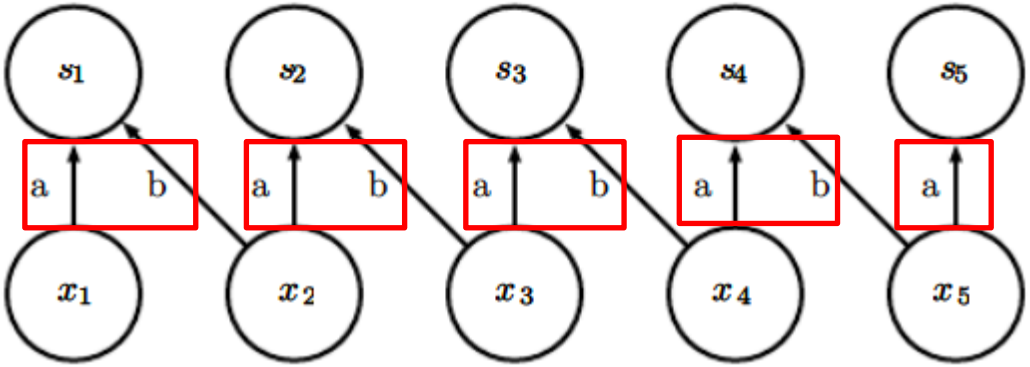
Tiled convolutional layer



Locally connected layer



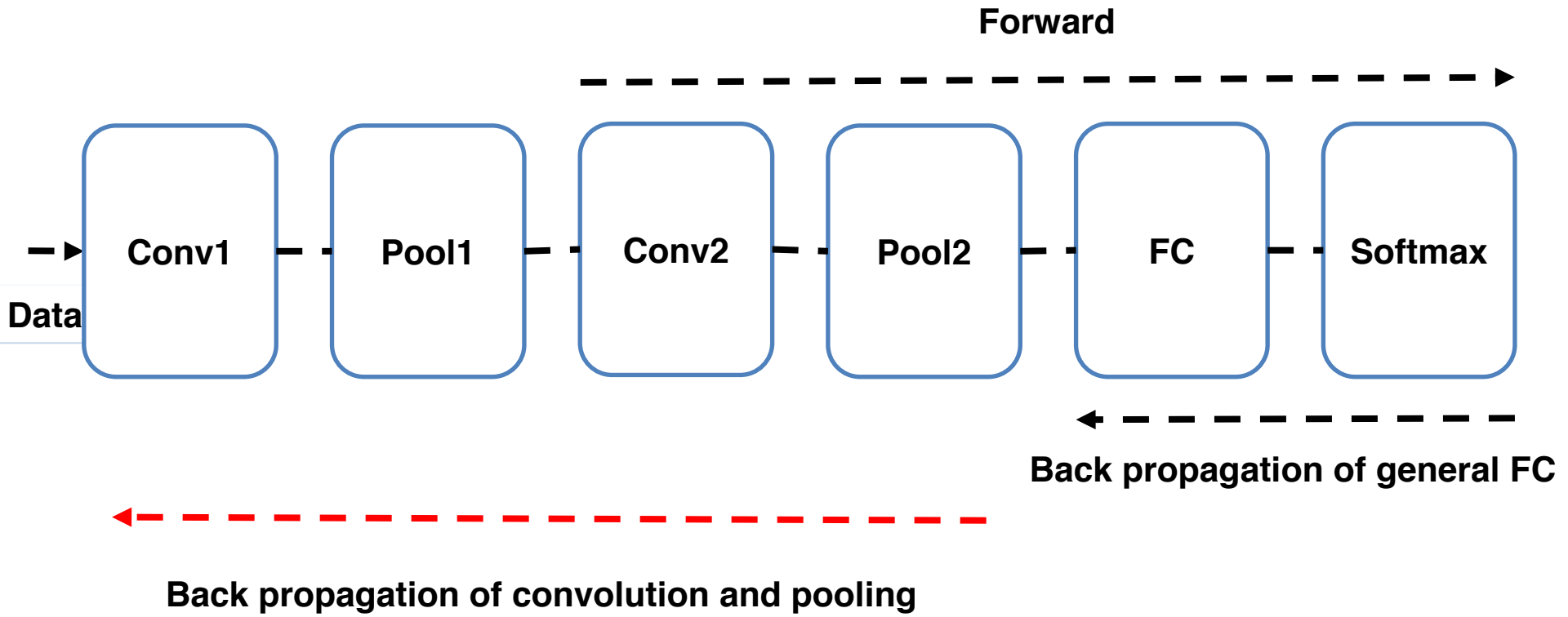
Tiled convolutional layer



Convolutional layer

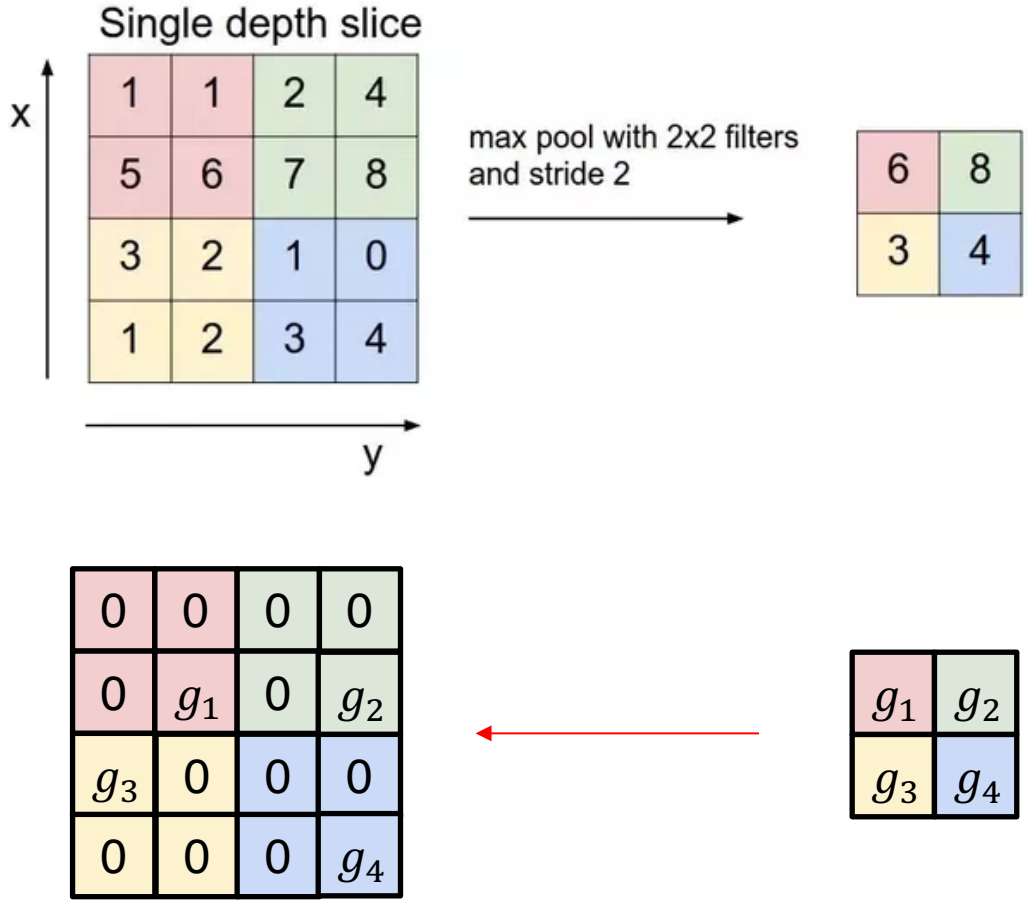
Backpropagation (Outline)

- Neural net has to propagate gradient



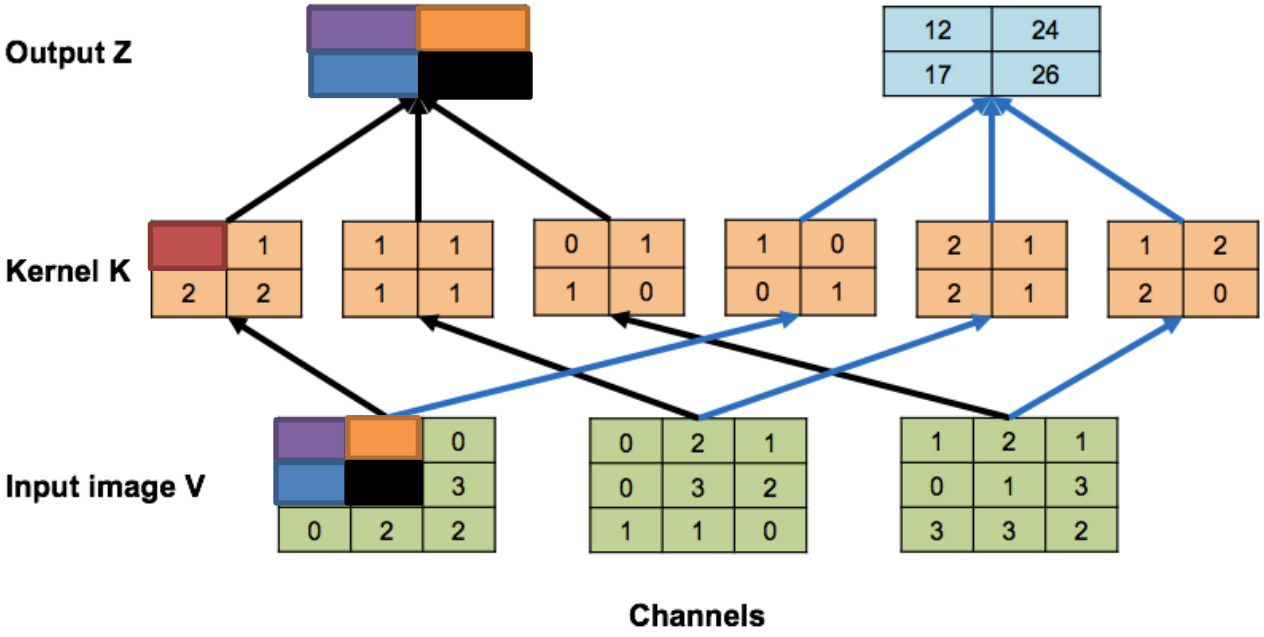
Backpropagation (Pooling)

- Neural net has to propagate gradient



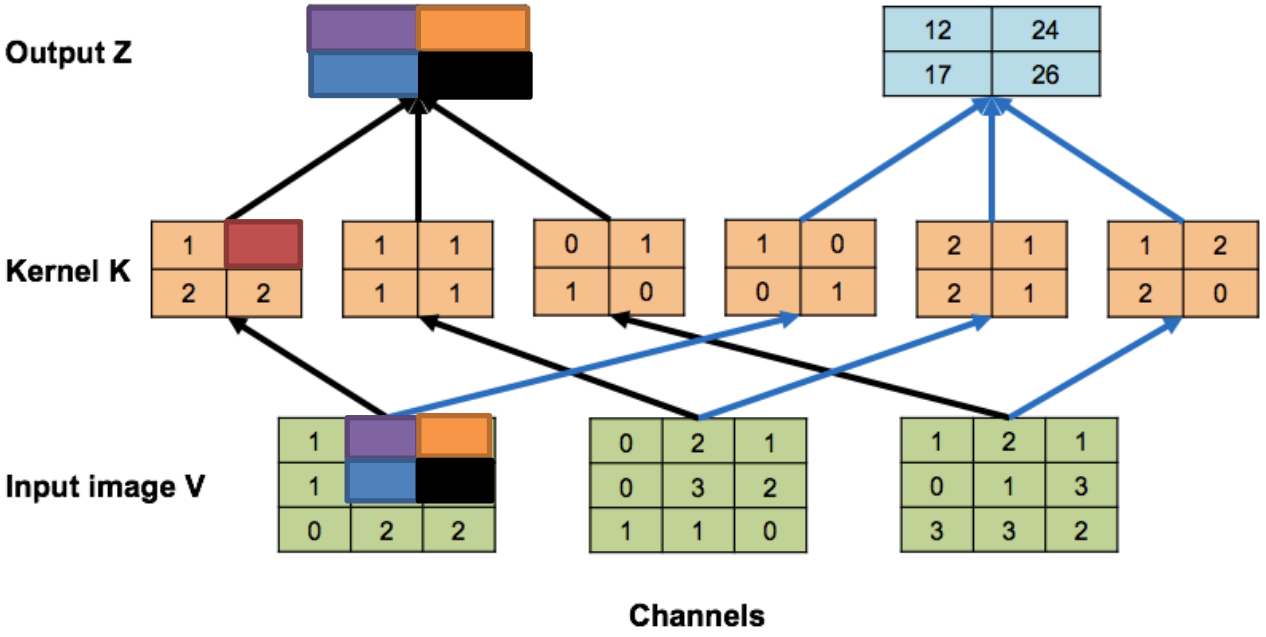
Backpropagation (Convolution - kernel)

- Gradient of kernel



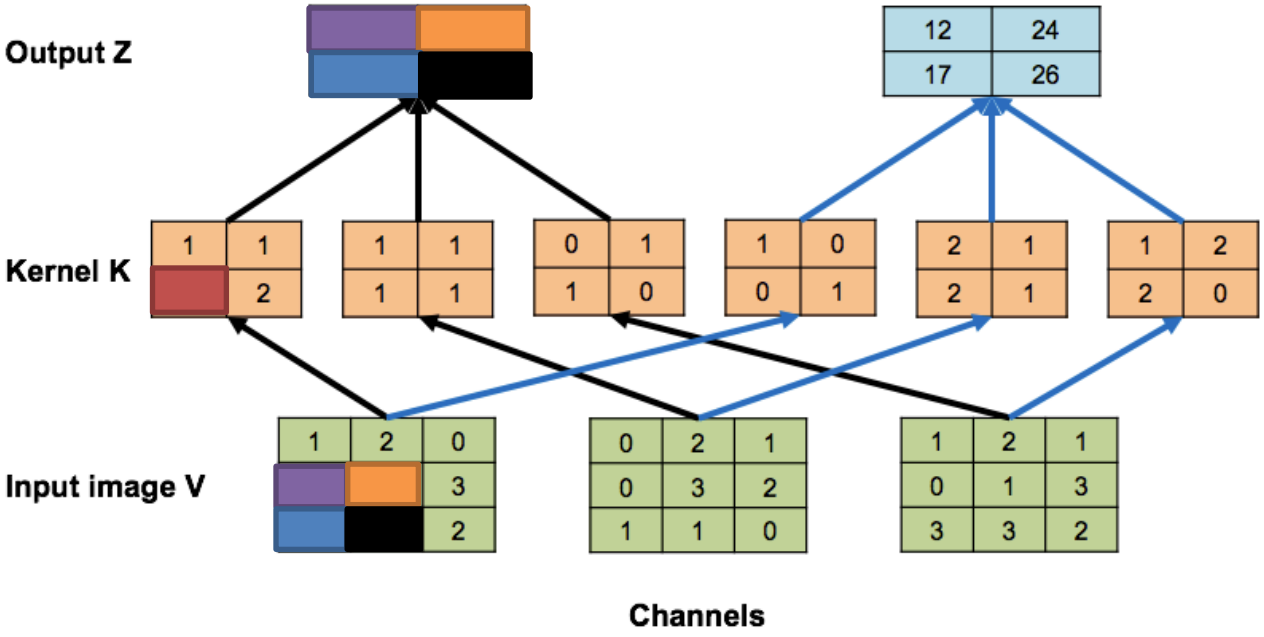
Backpropagation (Convolution - kernel)

- Gradient of kernel



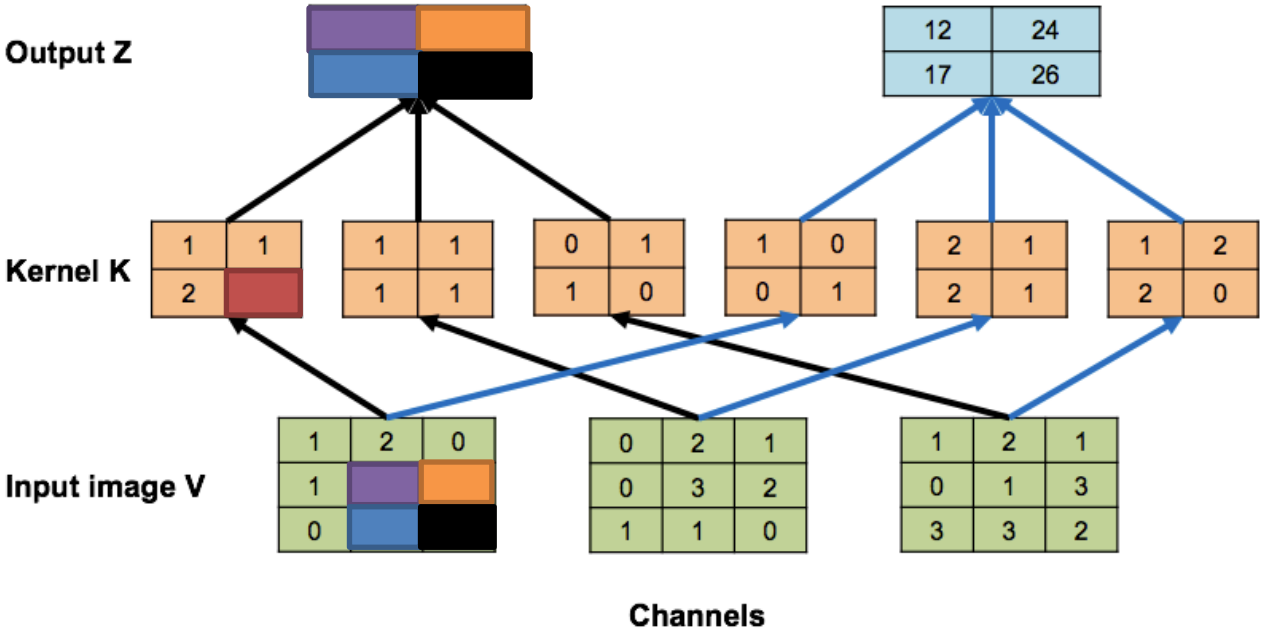
Backpropagation (Convolution - kernel)

- Gradient of kernel



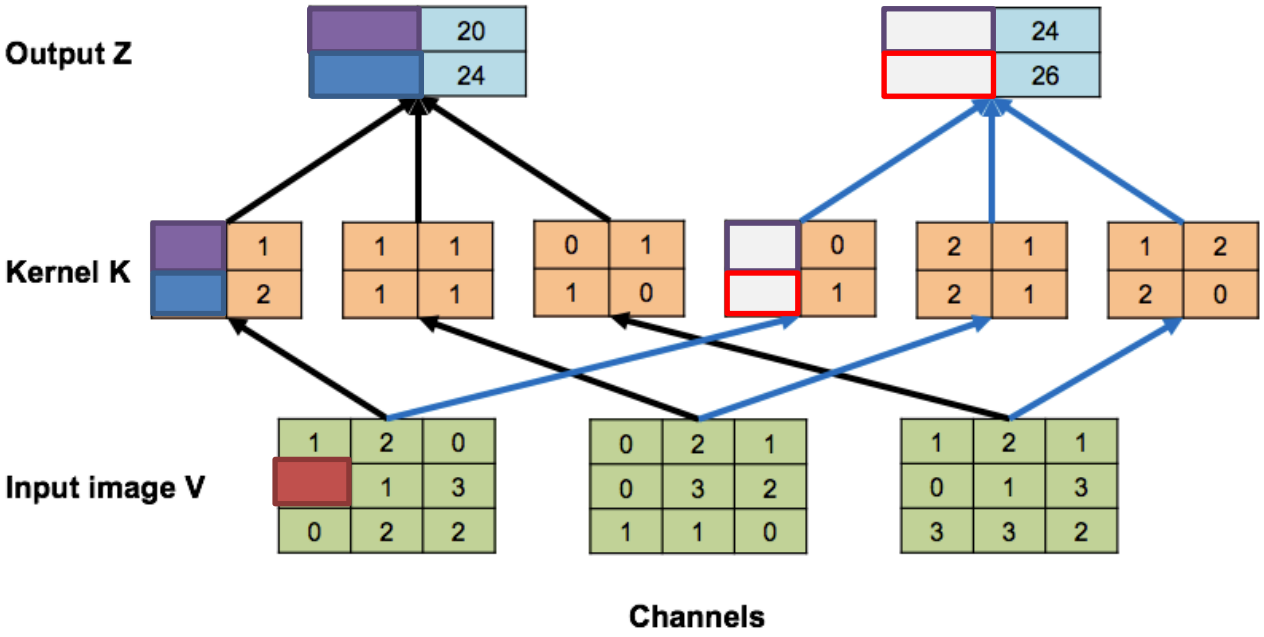
Backpropagation (Convolution - kernel)

- Gradient of kernel



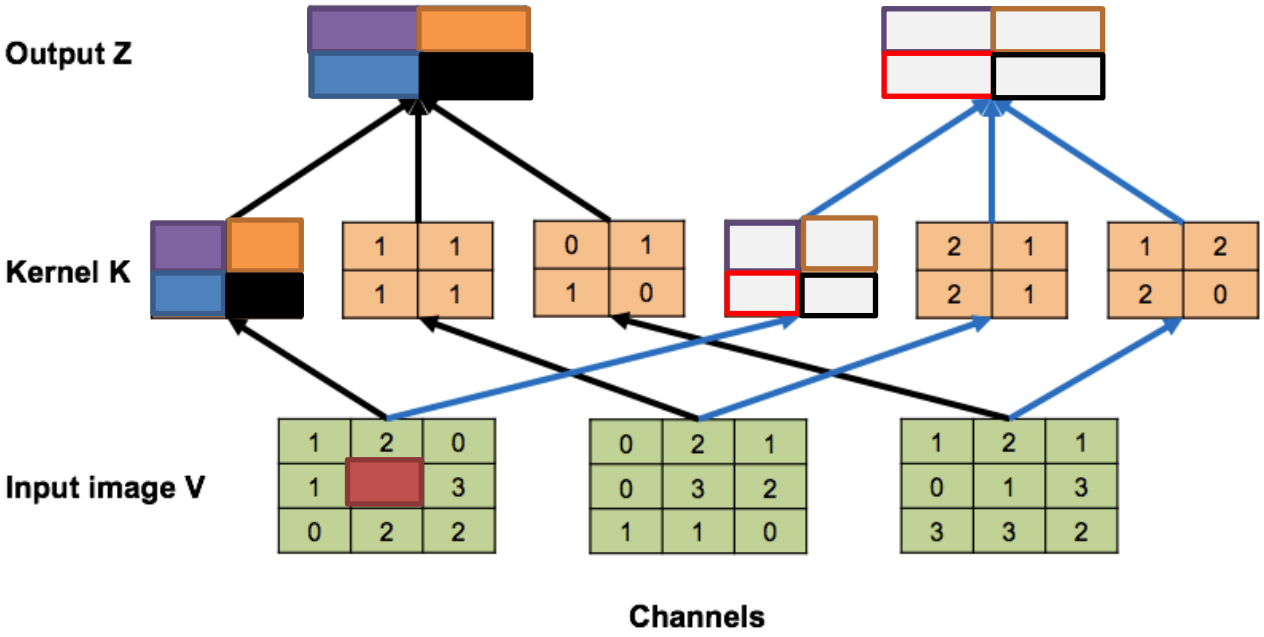
Backpropagation (Convolution - input)

- Gradient of input



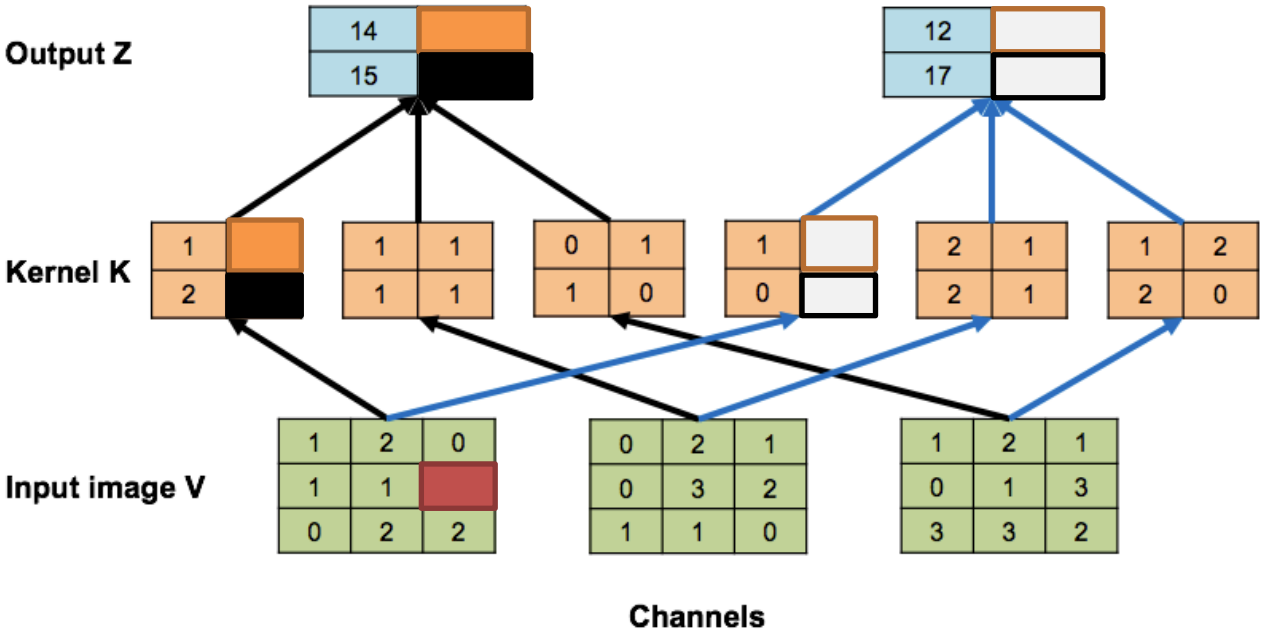
Backpropagation (Convolution - input)

- Gradient of input



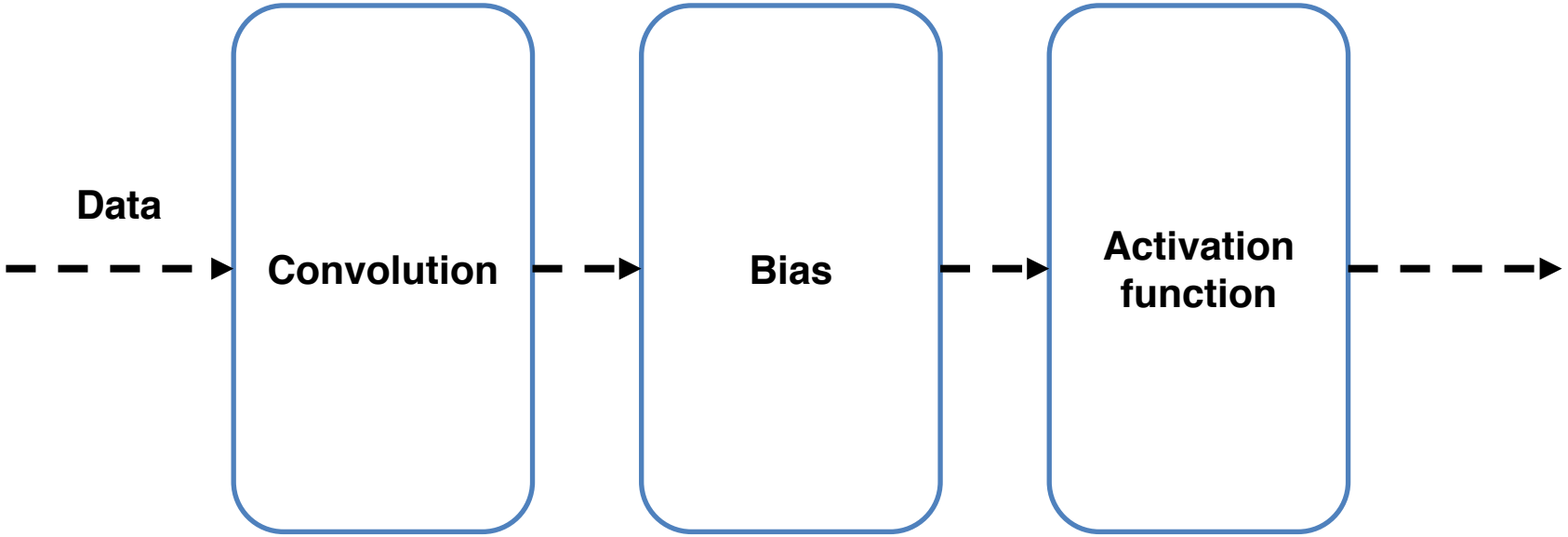
Backpropagation (Convolution - input)

- Gradient of input

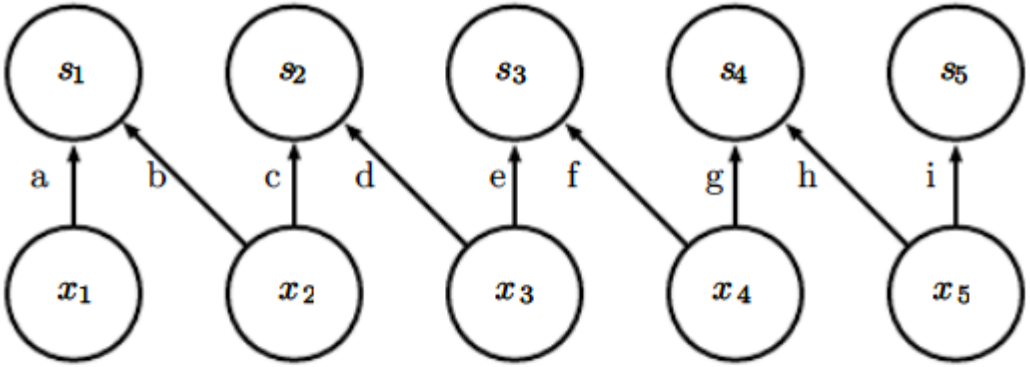


Bias and activation function

- Add bias term before applying activation function
  - Sigmoid
  - Tanh
  - ReLU

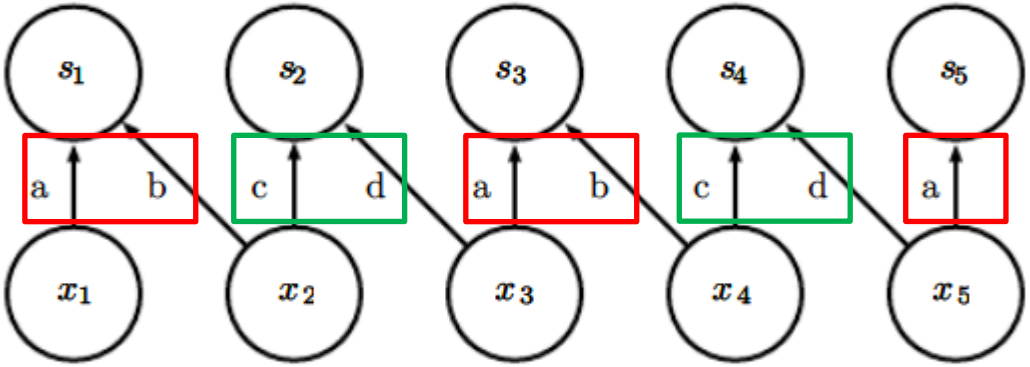


Various convolutional layer and its bais



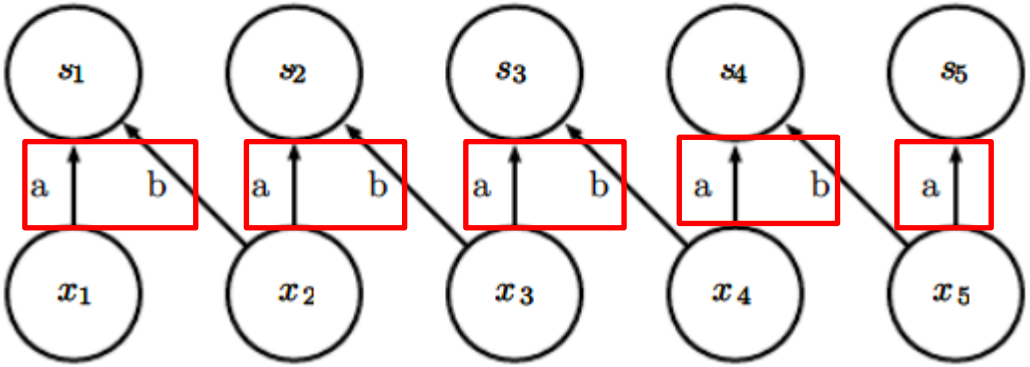
Locally connected layer

(Different bias for every filter)



Tiled convolutional layer

(Same bias for same tile pattern)



Convolutional layer

(Same bias for same channel)