University of Rome "La Sapienza"

Master in Artificial Intelligence and Robotics

Machine Learning

A.Y. 2018/2019

Prof. Luca Iocchi

Sapienza University of Rome, Italy
Master in Artificial Intelligence and Robotics
Machine Learning (2018/19)

15. Convolutional Neural Networks

Luca Iocchi

Summary

- Convolution
- Convolutional layers
- Pooling
- Example: LeNet
- "Famous" CNNs
- Resources

Motivation

Up to now we treated inputs as general feature vectors

In some cases inputs have special structure:

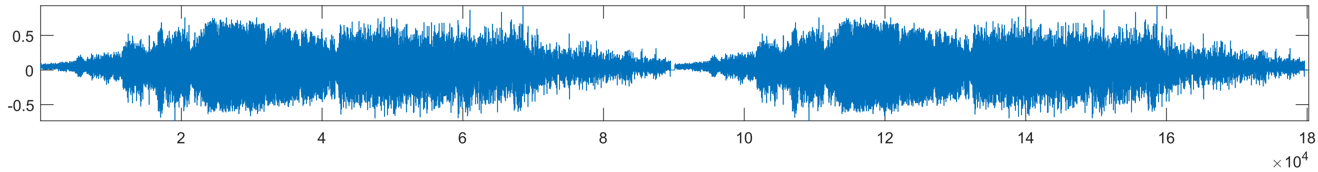
- Audio
- Images
- Videos

Signals: Numerical representations of physical quantities

Deep learning can be directly applied on signals by using suitable operators

Motivation - Examples

Audio



...	0.0468	0.0468	0.0468	0.0390	0.0390	0.0390	0.0546	0.0625	0.0625	0.0390	0.0312	0.0468	0.0625	...

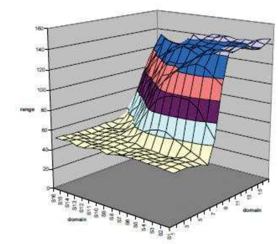
Note: variable length 1D vectors (1D tensor)

Sapienza University of Rome, Italy - Machine Learning (2018/2019)

Motivation - Examples

Images

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120



Note: multi-channel 2D matrices (3D tensor)

Video

A sequence of color images sampled through time
Note: sequence of multi-channel 2D matrices (4D tensor)

Convolution

Continuous functions

$$(x * w)(t) \equiv \int_{a=-\infty}^{\infty} x(a)\, w(t-a)\, da$$

Discrete functions

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

Discrete limited 2D functions:

$$(I * K)(i,j) \equiv \sum_m \sum_n I(m,n) K(i-m, j-n)$$

I: input, K: kernel

Convolution

Commutative

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

Cross-correlation

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

implemented in machine learning libraries (called convolution).

Image Convolutions

Original

Emboss

Interactive example: `http://setosa.io/ev/image-kernels/`

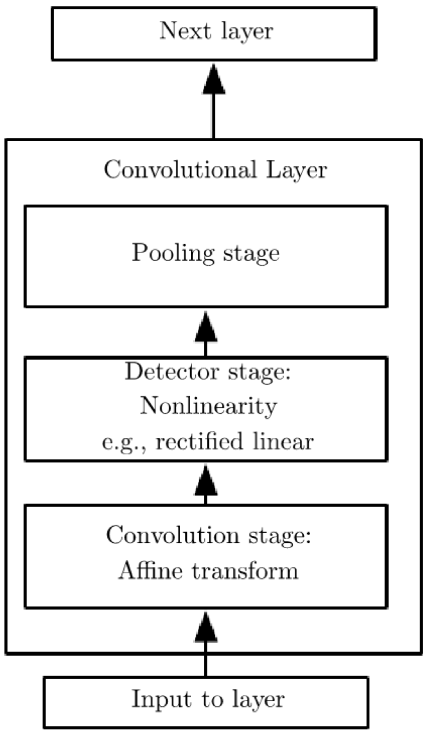
Convolutional Neural Networks

FNN with one or more convolutional layers.

Convolutional layer

Three stages:

- convolutions between input and kernel
- non-linear activation function (detector)
- pooling



Convolutional layer: Convolution

1. Convolution stage

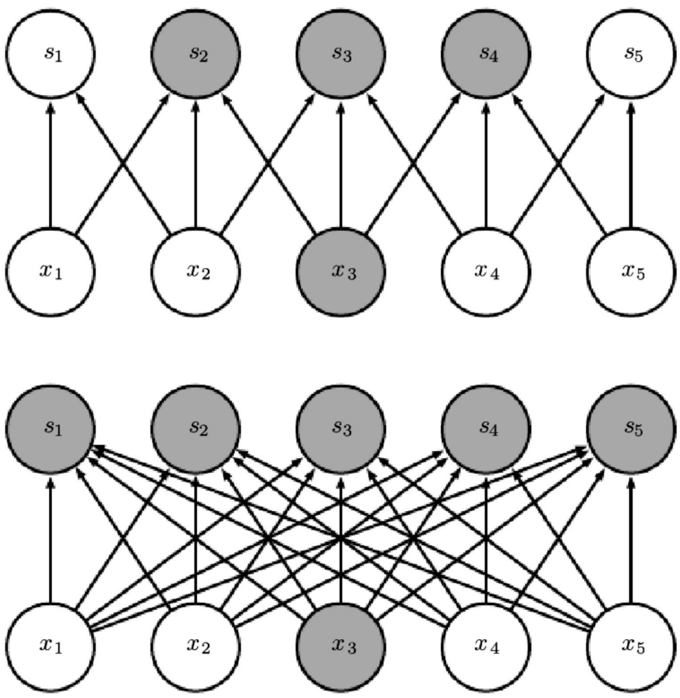
$$(I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n)$$

Very efficient in terms of memory requirements and generalization accuracy.

- Sparse connectivity
- Parameter sharing

Sparse connectivity

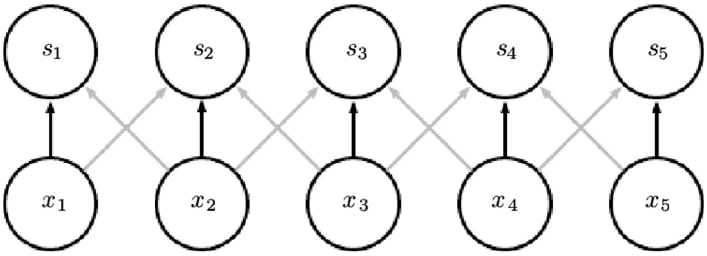
sparse interactions/ sparse connectivity: outputs depend only on a few inputs (as kernel is usually much smaller than input)

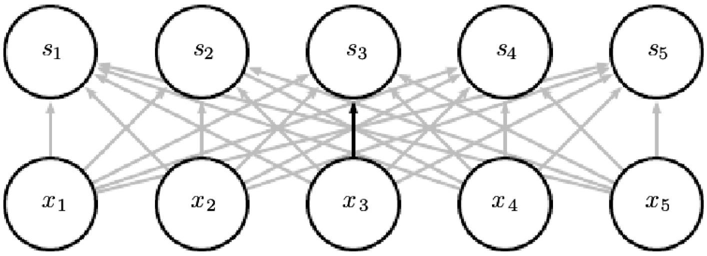


Parameter sharing

Learn only one set of parameters (for the kernel) shared to all the units.

k parameters instead of $m \times n$ (note: $k \ll m$)



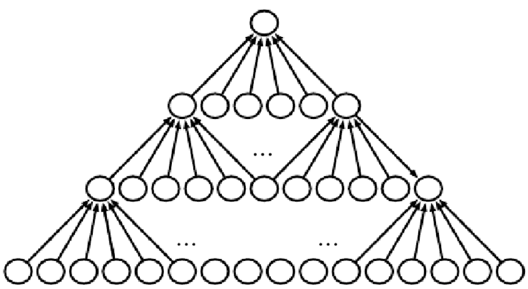


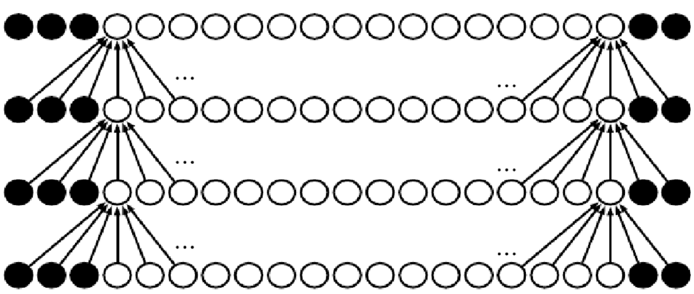
Sapienza University of Rome, Italy - Machine Learning (2018/2019)

Padding

valid padding: output contains only valid values (depends on kernel size)

same padding: input layer is padded with zeros, output size independent of kernel size.





Convolutional layer: Detector

2. Detector stage

Use non-linear activation functions.

- ReLU
- tanh
- ...

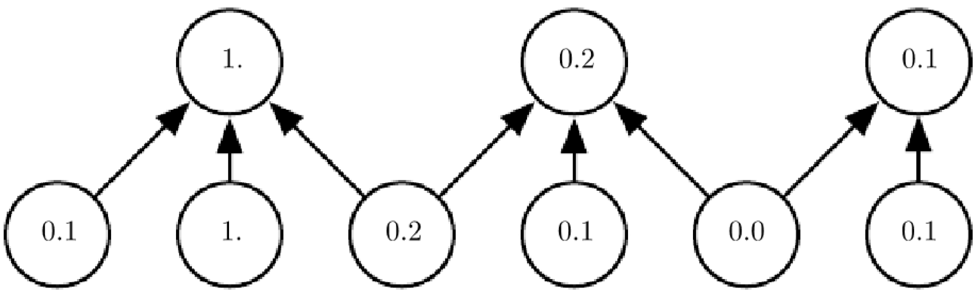
Convolutional layer: Pooling

3. Pooling stage
Implements *invariance to local translations*.

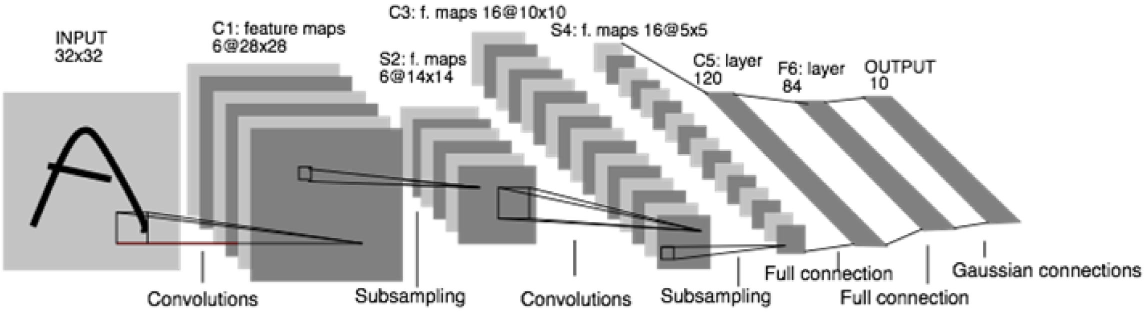
max pooling returns the maximum value in a rectangular region.
average pooling returns the average value in a rectangular region.

When applied with *stride*, it reduces the size of the output layer.

Example: pool width 3 and stride 2



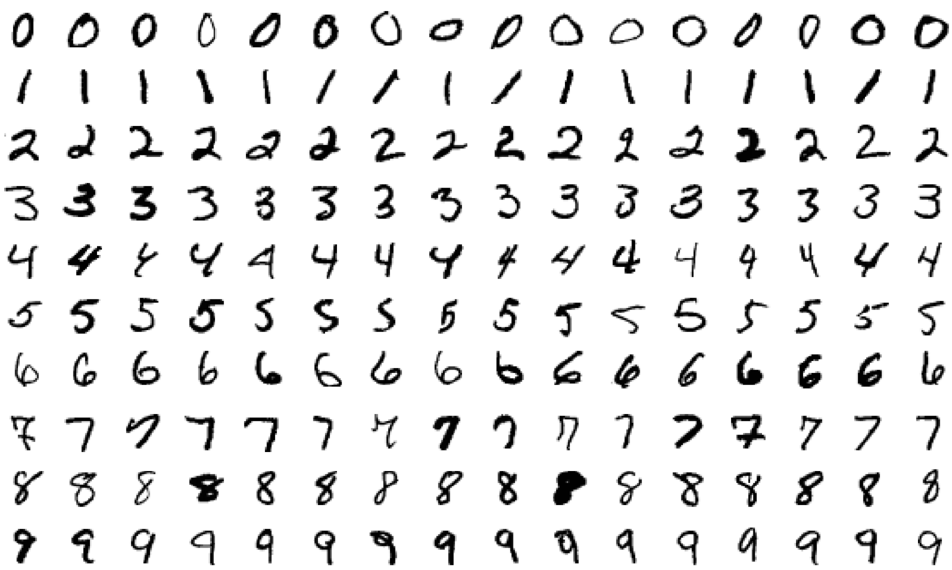
LeNet



Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11):2278-2324, November 1998

Handwritten recognition with LeNet

MNIST database



Accuracy up to 98 %.

ImageNet and ILSVRC

ImageNet Huge dataset of images
over 14 M labelled high resolution images
about 22 K categories

ILSVRC Competitions of image classification at large scale (since 2010)
1.2 M images in 1 K categories
5 guesses about image label

`http://image-net.org`

"Famous" CNNs

- AlexNet
- GoogLeNet
- VGG
- ResNet

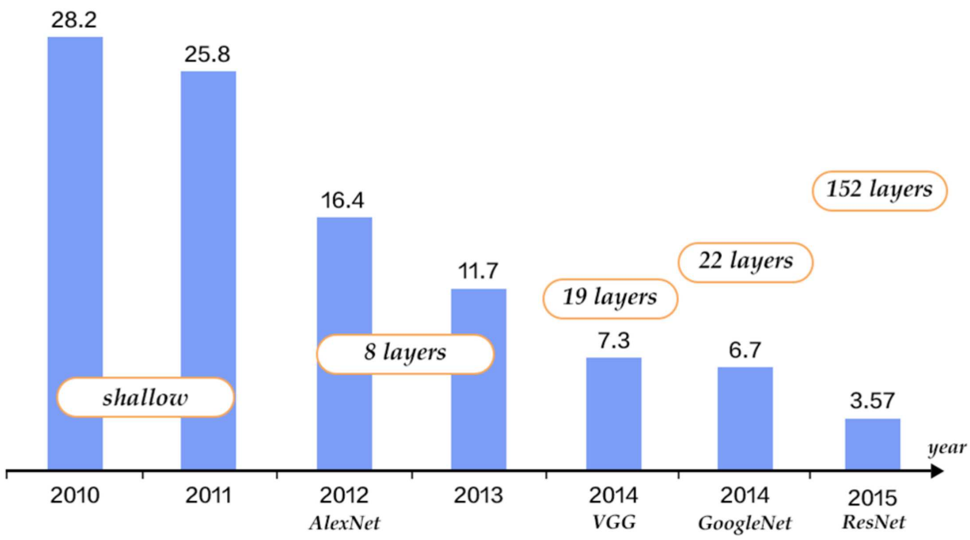


Image classification error rate (ILSVRC)

Common uses of famous CNNs

- Train the model on a new dataset
- Use pre-trained models (e.g., trained on ImageNet) to
  - predict ImageNet categories for new images
  - extract features to train another model (e.g., SVM)
- Refine pre-trained models on a new set of classes

Examples: `https://keras.io/applications/`

Resources

Frameworks

- Caffe/Caffe 2 (UC Berkeley) - C/C++, Python, Matlab
- TensorFlow (Google) - C/C++, Python, Java, Go
- Theano (U Montreal) - Python
- CNTK (Microsoft) - Python, C++ , C#/.Net, Java
- Torch/PyTorch (Facebook) - Lua/Python
- MxNet (DMLC) - Python, C++, R, Perl,
- Darknet (Redmon J.) - C

High-level application libraries and models

- Keras
- TFLearn

Conclusions

- CNNs are deep networks using convolution
- Very efficient (memory and generalization accuracy)
- Many successful examples
- Requires very large amount of data and very high computational resources.