Convolutional Neural Networks

Léo Bois

May 2021

Outline

Introduction

Convolution

CNN Architectures

Table of Contents

Machine Learning

Data

- Variables
- Images
- Speech
- Time Series

Task

- Regression
- Classification
- Generation
- Clustering
- Dimension Reduction

Model

- Linear Model
- Decision Tree
- Random Forest
- Support-Vector Machine
- Neural Network

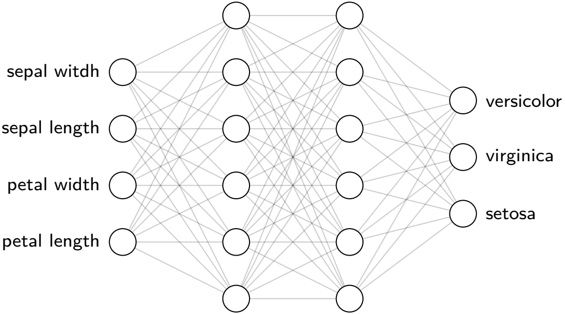
Learning

- Supervised
- Unsupervised
- Reinforcement

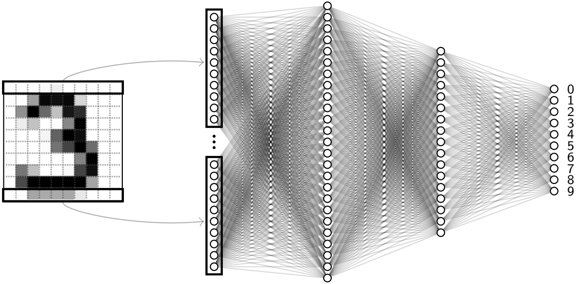
Basics of Neural Networks

Neural Network "Vaguely inspired by the biological neural networks that constitute animal brains." (Wikipedia)
Function relying on many parameters, that can be optimized by gradient descent with the back-propagation algorithm.

Iris Versicolor (Wikipedia)



Dense Networks and Image Recognition



Quoting [LeCun et al., 1995]: "there are problems"

- "Typical images [...] are large, often with several hundred variables. [...] **Overfitting problems** may occur if training data is scarce."
- "But, the main deficiency of unstructured nets for image [...] applications is that they have **no built-in invariance with respect to translations, or local distortions of the inputs**. [...] In principle, a fully-connected network of sufficient size could learn to produce outputs that are invariant with respect to such variations. However, learning such a task would probably result in multiple units with identical weight patterns positioned at various locations in the input."
- "Secondly, a deficiency of fully-connected architecture is that the **topology of the input is entirely ignored**. [...] On the contrary, images [...] have a strong 2D local structure [...]: variables (or pixels) that are spacially [...] nearby are highly correlated."

Main Ideas of CNNs

Quoting again [LeCun et al., 1995]:
"Convolutional networks combine three architectural ideas to ensure some degree of shift and distortion invariance:

- local receptive fields,
- shared weights (or weight replication), and, sometimes,
- spatial or temporal subsampling."

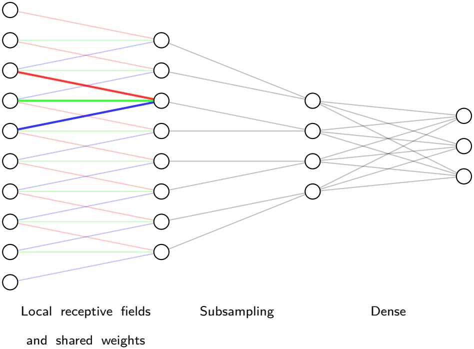
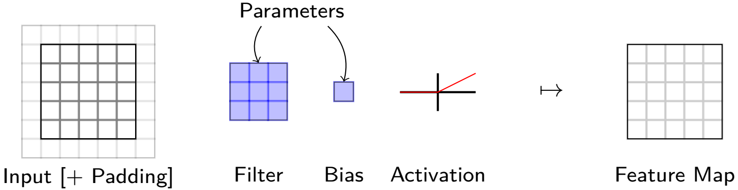


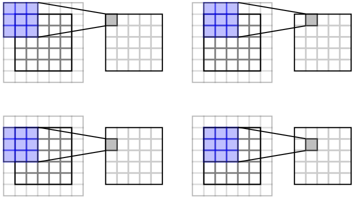
Table of Contents

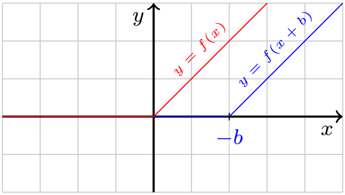
Base Recipe

- Ingredients:



- Directions:

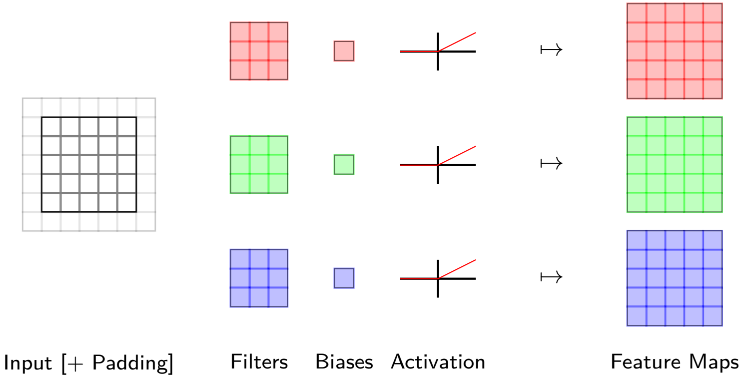




1. Scalar Products (Input·Filter)
2. Add bias and apply activation, term by term

More Flavors

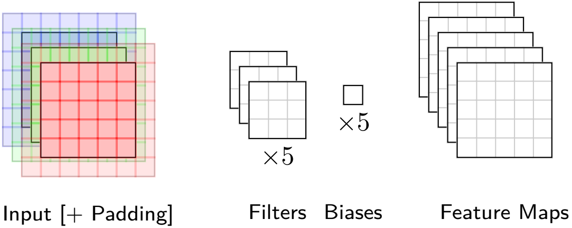
In practice, multiple filters are used in each convolution, resulting in multiple feature maps.



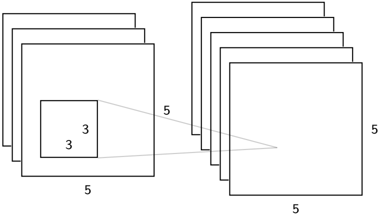
This raises the question of convolutions on input with multiple channels.

Input with Multiple Channels

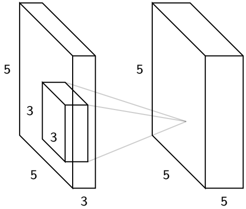
The current standard way is to use filters with the same depth as the input:



Representation of such a convolution in the litterature:



or



Summary

Hyper-parameters of a convolution:

- Number of filters (determines the depth of the output)
- Size of the filters (width and height; their depth is the same as the input)
- Padding (no padding, zero padding, mirror padding, ...)
- Activation function (ReLU, sigmoid, tanh, parameterized function, ...)

Properties of a convolution:

- The number of parameters is given by

  $$[(\text{filters' size}) \times (\text{input depth}) + 1] \times (\text{number of filters}).$$

  It does not depend on the width and height of the input.
- The number of connexions (ie multiplications) is about

  $$(\text{input's size}) \times (\text{filters' size}) \times (\text{input depth}) \times (\text{number of filters}).$$

Here the size denotes the quantity width$\times$height.

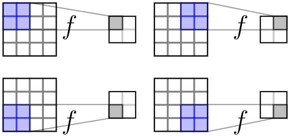
Subsampling

Goals:

- Removing useless information about exact location.
- Reducing the size of the feature maps, therefore decreasing the number of subsequent connexions and parameters needed.

Current implementations:

- Max-Pooling: f is the $\max$ function.
  Close to a logical OR operation.
- Convolution with *stride* 2: f is a 2×2 trainable filter, followed by a trainable bias and an activation function.



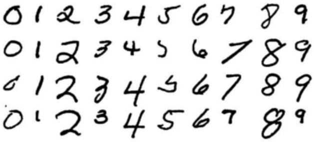
Remarks:

- Historically, the subsampling consisted in adding the four inputs, multiplying the result by a trainable parameter, adding a trainable bias and applying the sigmoid function, thus using 2 parameters per feature map [Lecun et al., 1998].
- In the experiments of [Hutchison et al., 2010], the max-pooling has proven more effective than this subsampling. In this paper, other alternatives have also been tested.
- The use of convolutions instead of max-pooling layers was explored in [Springenberg et al., 2015]. It can yield better results but introduces more parameters.

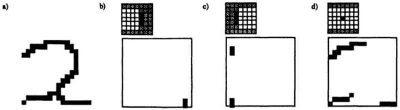
Table of Contents

Early Digit Recognition

[Denker et al., 1988]



The authors use convolutions as part of the preprocessing.
They use 49 hand-engineered filters to extract features.



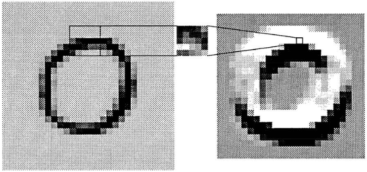
The feature maps are then "Coarse Coded" (ie subsampled).

The authors then compare several classifiers: with enough data, the (fully-connected) neural networks give the best results.

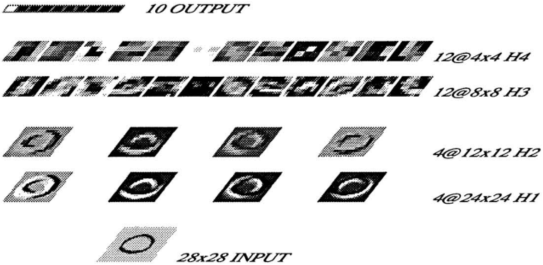
Early Digit Recognition

First CNN (1989), [LeCun et al., 1990]

The authors integrate the feature extraction to the model: the filters are learned by gradient descent.



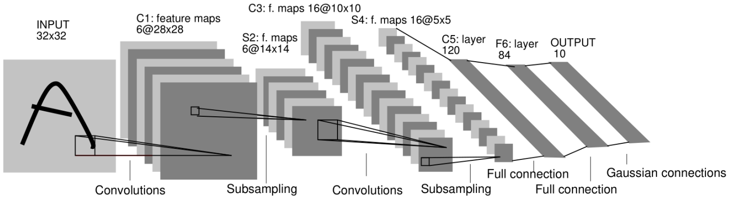
The resulting neural network has 98442 connections, and 2578 parameters:



Early Digit Recognition

LeNet-5 (1995), [Lecun et al., 1998]

LeNet-5 is an improved version of the previous model. It has 340 908 connections and 60 000 parameters.



In the second convolutional layer, each of the 16 filters combine a specific set of the preceding features maps:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

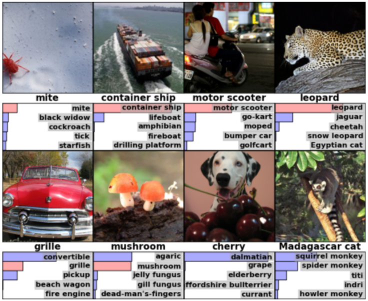
TABLE I

EACH COLUMN INDICATES WHICH FEATURE MAP IN S2 ARE COMBINED BY THE UNITS IN A PARTICULAR FEATURE MAP OF C3.

ImageNet ILSVRC

ImageNet Large-Scale Visual Recognition Challenge

- Each year since 2010
- Data consists of hand labeled photographs
- Labels are from 1000 object categories (e.g. centipede, millipede)
- Several tasks: classification, classification with localization, detection, ...
- Classification dataset: 1.2M training images, 100k validation images, 50k test images
- Errors for classification: Top-1 Error Rate & Top-5 Error Rate

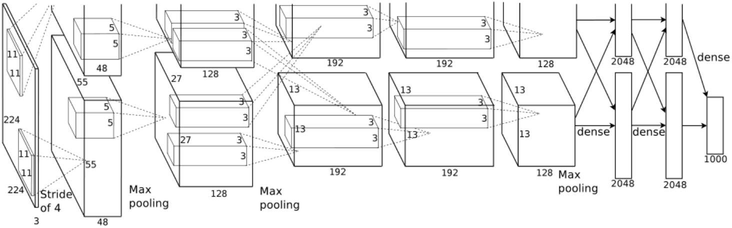


Classification examples ([Krizhevsky et al., 2017])

ILSVRC Winners

AlexNet (2012), [Krizhevsky et al., 2017]

Architecture:



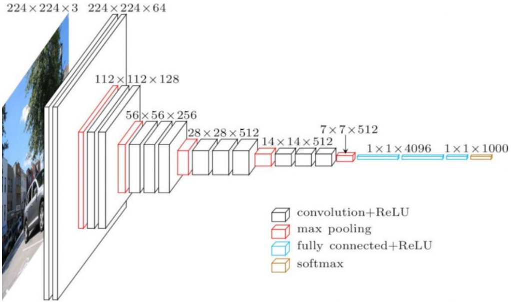
Comments:

- Top-5 Error Rate of 15.3% (second place 26.2%).
- 60M parameters, 5-6 days of training with 2 GPUs.
- Architecture distributed on the 2 GPUs.
- Uses Rectified Linear Units (ReLUs) (cite).
- Uses max-pooling with 3×3 windows overlapping by 1 pixel.
- Reduces overfitting with data augmentation and dropout.

ILSVRC Winners (or not)

VGG16 (2014), [Simonyan and Zisserman, 2015]

Architecture:



Found here, original source unknown

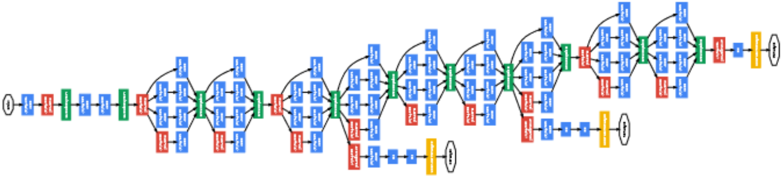
Comments:

- Top-5 Error Rate of 7.0% (with single net; better than GoogLeNet with 7.9%).
- ∼140M parameters, 2-3 weeks of training with 4 GPUs.
- Only uses filters of size 3×3 with padding.
- Reduces overfitting with data augmentation, dropout, weight decay.

ILSVRC Winners

GoogLeNet (2014), [Szegedy et al., 2015]

Architecture:



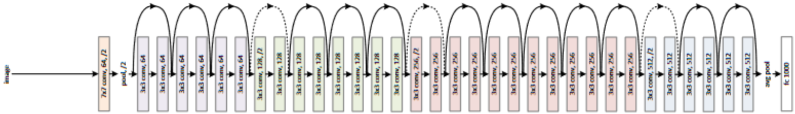
Comments:

- Top-5 Error Rate of 6.67% (with 7 nets; better than VGG with 7.3%).
- $\sim$7.4M parameters, about 1 week of training with "a few high-end GPUs".
- Uses "inception modules" with 1×1, 2×2 and 3×3 convolutions.
- Adds auxiliary classifiers connected to intermediate layers for back-propagation.

ILSVRC Winners

ResNet (2015), [He et al., 2015]

Architecture:



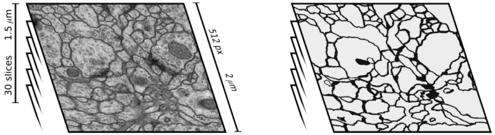
Comments:

- Top-5 Error Rate of 3.57% (with ensembles; 4.49% with single net).
- Architectures with up to 152 layers and ∼60M parameters.
- The authors explore a model with 1202 layers (19.4M parameters), with higher error rate (overfitting).
- Uses "identity shortcuts" for better training.

Image Segmentation

[Ciresan et al., 2012]

Task: Segmentation of images of slices of neurons into membrane and not membrane pixels.



Approach: Pixel classifier with a deep CNN.

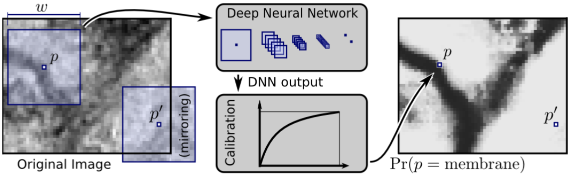
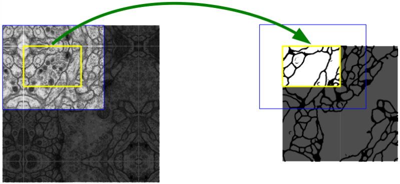


Image Segmentation

[Ronneberger et al., 2015]

Approach: Segmentation by "tile".



Architecture:

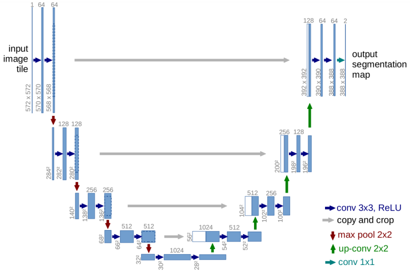
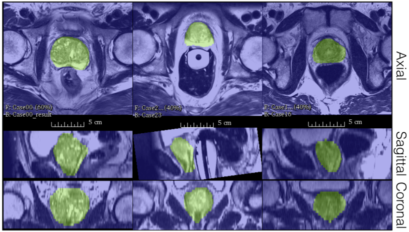


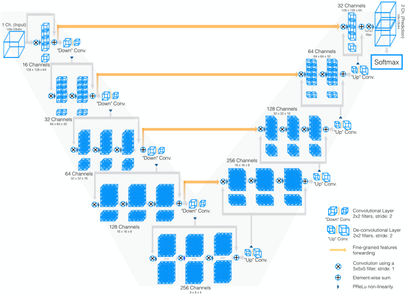
Image Segmentation

[Milletari et al., 2016]

Approach: Segmentation of the whole 3D image (prostate).



Architecture:



Bibliography I

Ciresan, D., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2012).
Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images.
Advances in neural information processing systems, page 9.

Denker, J. S., Gardner, W. R., Graf, H. P., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D., Baird, H. S., and Guyon, I. (1988).
Neural Network Recognizer for Hand-Written Zip Code Digits.
page 9.

He, K., Zhang, X., Ren, S., and Sun, J. (2015).
Deep Residual Learning for Image Recognition.
arXiv:1512.03385 [cs].
arXiv: 1512.03385.

Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Scherer, D., Müller, A., and Behnke, S. (2010).
Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition.
In Diamantaras, K., Duch, W., and Iliadis, L. S., editors, *Artificial Neural Networks – ICANN 2010*, volume 6354, pages 92–101. Springer Berlin Heidelberg, Berlin, Heidelberg.
Series Title: Lecture Notes in Computer Science.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017).
ImageNet classification with deep convolutional neural networks.
Communications of the ACM, 60(6):84–90.

Bibliography II

LeCun, Y., Bengio, Y., and Laboratories, T. B. (1995).
Convolutional Networks for Images, Speech, and Time-Series.
page 15.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1990).
Handwritten Digit Recognition with a Back-Propagation Network.
page 9.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998).
Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11):2278–2324.
Conference Name: Proceedings of the IEEE.

Milletari, F., Navab, N., and Ahmadi, S.-A. (2016).
V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation.
arXiv:1606.04797 [cs].
arXiv: 1606.04797.

Ronneberger, O., Fischer, P., and Brox, T. (2015).
U-Net: Convolutional Networks for Biomedical Image Segmentation.
arXiv:1505.04597 [cs].
arXiv: 1505.04597.

Bibliography III

Simonyan, K. and Zisserman, A. (2015).
Very Deep Convolutional Networks for Large-Scale Image Recognition.
arXiv:1409.1556 [cs].
arXiv: 1409.1556.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2015).
Striving for Simplicity: The All Convolutional Net.
arXiv:1412.6806 [cs].
arXiv: 1412.6806.

Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015).
Going deeper with convolutions.
In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, Boston, MA, USA. IEEE.