Convolutional neural networks

E. Decencière

MINES ParisTech
PSL Research University
Center for Mathematical Morphology

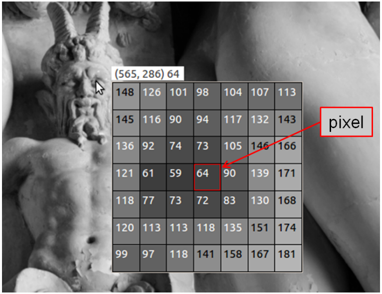
Contents

Contents

A picture is worth a thousand words

Definition

- Classically, an image is a matrix of values belonging to $[0, \ldots, 255]$ (grey level images) or to $[0, \ldots, 255]^3$ (color images).
- More generally, an image is a q-dimensional array of values belonging to R^d.



Grey level values around the left eye of the faun

Extracting semantic information from an image

- Where is the phone? (localization task)
- How many mugs are there? (quantification task)
- Is there a window in the room?
- At what time of the day was the photograph taken?

Extracting semantic information from an image

- Where is the phone? (localization task)
- How many mugs are there? (quantification task)
- Is there a window in the room?
- At what time of the day was the photograph taken?

Designing computer vision systems that are able to extract semantic information from an image is a difficult task.

The role of annotated image databases

Image databases including *annotations* (typically some kind of high level information) are essential to the development of *supervised* machine learning methods for image analysis.

Annotations

- Image class
- Measure(s) obtained from the image
- Position of objects within the image
- Segmentation

MNIST database [Lecun et al., 1998]

- The Modified National Institute of Standards and Technology (MNIST) database contains $60\,000$ training images of hand-written digits, and $10,000$ test images.
- Image size: 28×28
- It has been used since 1998
- Human performance on a similar database (NIST) is reported to be around 1.5% error [Simard et al., 1993]
- Best methods, based on convolutional neural networks, give around 0.21% test error.

MNIST database

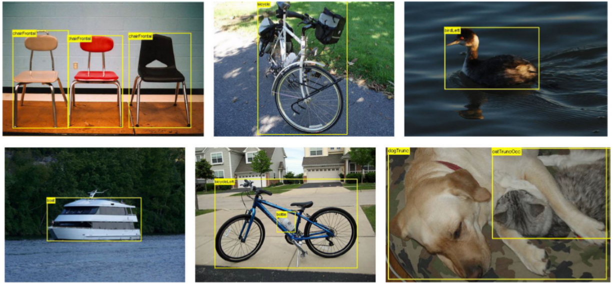
Credits: Images from MNIST assembled by Josef Stepan (licensed under CC BY-SA 4.0)

Pascal VOC project [Everingham et al., 2010, Everingham et al., 2014]

This project organized a challenge from 2005 to 2012, divided into several tasks, including an image classification task.

Pascal VOC image classification task (2012)

Train/val: $11\,540$ images where the presence of 20 categories of objects was annotated. The test dataset is unknown and tests are run online (still available).



Credits: From [Everingham et al., 2014]

ImageNet project

Since 2010, ImageNet organizes and annual challenge: The ImageNet Large Scale Visual Recognition Challenge (ILSVRC), that constitutes a breakthrough in the design of image analysis challenges by its size.

Image classification task (since 2012)

- Training: $1\,281\,167$; validation: $50\,000$; test: $100\,000$.
- $1\,000$ classes (90 dog breeds!).

ImageNet projet

Examples from the *acoustic guitar* class

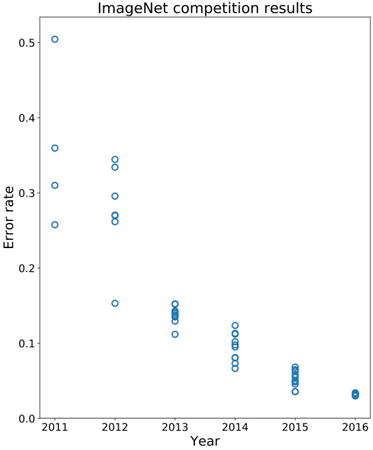


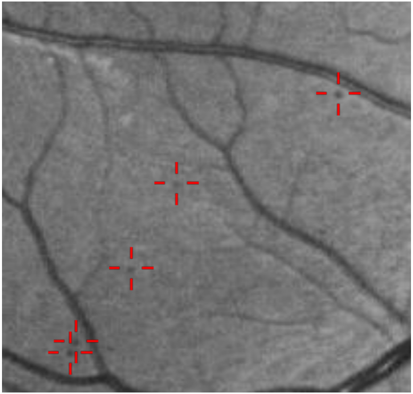
Image processing approach

- Build a geometrical model for the objects of interest
- Implement this model using image processing operators

Credits: [zhang et al., 2011]

Image processing approach

- Build a geometrical model for the objects of interest
- Implement this model using image processing operators

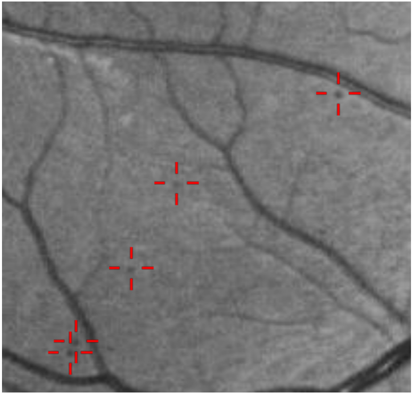


Detail of eye fundus image with microaneurysms to be detected

Credits: [zhang et al., 2011]

Image processing approach

- Build a geometrical model for the objects of interest
- Implement this model using image processing operators


Detail of eye fundus image with microaneurysms to be detected

+ This approach works correctly when the objects are not too complex.
- If objects are difficult to model, machine learning methods can bring a solution.

Credits: [zhang et al., 2011]

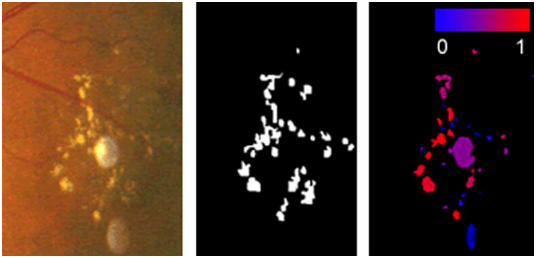
Classical machine learning approach

- Compute features from the image
- Apply machine learning to those features

Credits: [Zhang et al., 2014]

Classical machine learning approach

- Compute features from the image
- Apply machine learning to those features

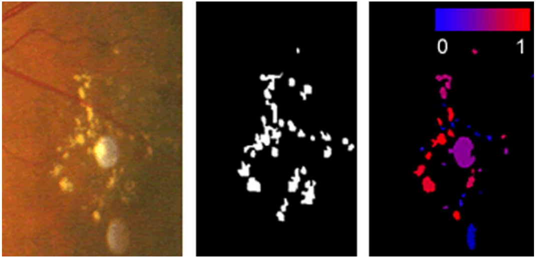


Exudates segmentation: original image, ground-truth and candidates with associated probabilities obtained with machine learning

Credits: [Zhang et al., 2014]

Classical machine learning approach

- Compute features from the image
- Apply machine learning to those features



Exudates segmentation: original image, ground-truth and candidates with associated probabilities obtained with machine learning

\+ Works well with the right features

\- An expert is required to define those features - and this can be a long process

\- Annotated data is required

Credits: [Zhang et al., 2014]

Deep learning approach

- Directly take as input the image pixels
- The network is supposed to build its own features

+ Good (impressive!) results
- A large amount of annotated data is required

Some accomplishments

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

2012: *AlexNet* [Krizhevsky et al., 2012] won this challenge by a large margin

The database contains more that 1 million training images, belonging to 1000 different classes.

Some accomplishments (cont.)

- 2011: first super-human performance, IJCNN 2011 traffic sign recognition contest [Cireșan et al., 2011]
- 2012: visual object detection (Mitosis detection in breast cancer histology [Cireșan et al., 2013])
- 2012: segmentation competition (neuronal membranes in electron microscopy images [Ciresan et al., 2012])
- 2016: AlphaGo beats Lee Sedol, one of the best go players, in a 5-game match

Deep learning image applications

- classification
- object localization
- semantic segmentation
- instance segmentation
- transformation (filtering, in-painting, editing, colorization...)
- quantification
- compression
- image caption generation
- 2D to 3D (stereo matching, 3D reconstruction, ...)
- motion estimation
- Style transfer
- Anomalous image detection
- Image generation

Convolutional neural networks in deep learning

- They are pivotal to many of the successes achieved by neural networks these recent years
- They are interesting for dealing with regular structured data, such as images (or board games!)

Acronyms

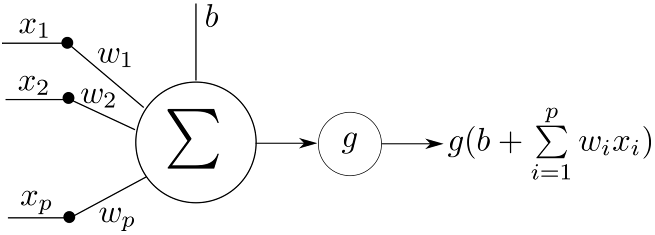
CNN and *ConvNet*

Essential milestones

- 1979: Neocognitron (CNN architecture) [Fukushima, 1979, Fukushima, 1980]
- 1989: Backpropagation applied to CNNs [LeCun et al., 1989]
- 2006, 2010: GPU implementation [Chellapilla et al., 2006, Cireșan et al., 2010]
- 2010: Availability of large databases (ImageNet, ...)

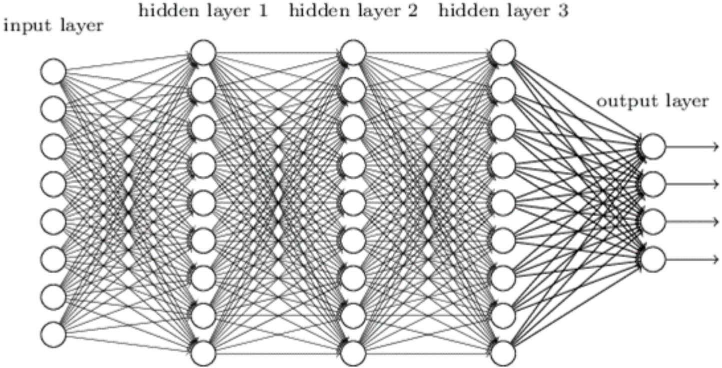
Contents

Reminder: Artificial neuron



- $b, w_1, \ldots, w_n$ are the neuron parameters, to be learnt
- g is the activation or transfer function

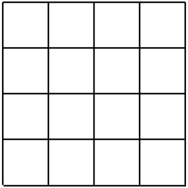
Reminder: Neural network



(from http://www.jtoy.net)

Input image, input neurons

In the scalar case (single-valued images), each input pixel is considered as an input neuron.



Input image, input neurons

In the scalar case (single-valued images), each input pixel is considered as an input neuron.

Image classification problem

Classification problem:

- Input: image $\mathbf{x}$
- Output: class $y \in \{label_1, label_2, \ldots, label_q\}$

Class coding

Often, classes are denoted by integers, but this is only a coding commodity. For instance, it would be meaningless to use a regression approach for this problem.

Class coding

If there are q possible classes, then a class will be coded as a vector $\mathbf{y}$ of length q. If its class is r then for $0 \leq i < q$:

$$\mathbf{y}[i] = \begin{cases} 1, & \text{if } i = r \\ 0, & \text{otherwise} \end{cases}$$

Example with 4 classes

- Label $0 \longmapsto [1, 0, 0, 0]$
- Label $1 \longmapsto [0, 1, 0, 0]$
- Label $2 \longmapsto [0, 0, 1, 0]$
- Label $3 \longmapsto [0, 0, 0, 1]$

Image classification with a neural network

Input

The input image, containing p pixels, is transformed into a vector of length p.

Output

For q classes, the output will be a vector of length q.

Image classification with a neural network

Input

The input image, containing p pixels, is transformed into a vector of length p.

Output

For q classes, the output will be a vector of length q.

Example: image of size 4×2, 4 possible classes

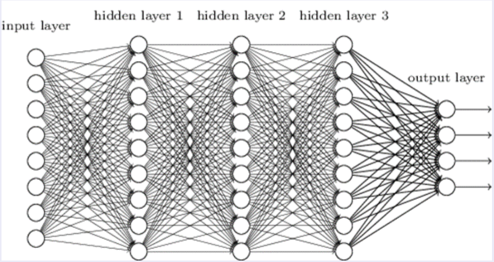
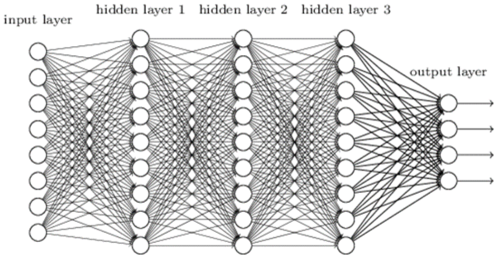
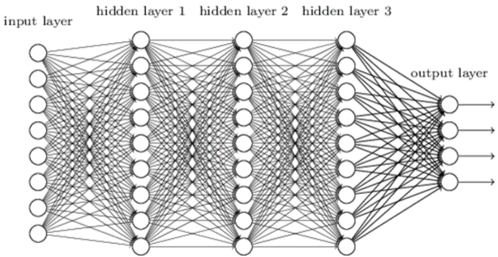


Image classification using fully-connected layers



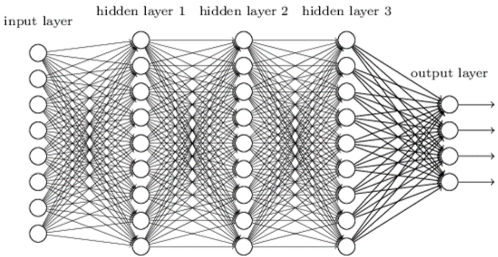
- A small image contains at least $100\,000$ pixels

Image classification using fully-connected layers



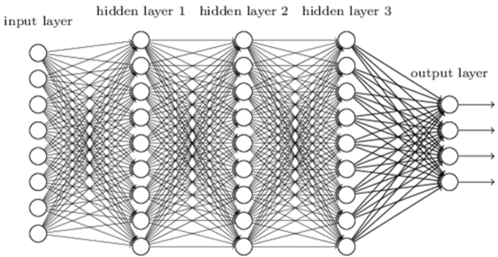
- A small image contains at least $100\,000$ pixels
- The number of parameters between two layers of that size is $10^5 \times (10^5 + 1)$!

Image classification using fully-connected layers



- A small image contains at least $100\,000$ pixels
- The number of parameters between two layers of that size is $10^5 \times (10^5 + 1)$!
- This approach is not feasible...

Image classification using fully-connected layers



- A small image contains at least $100\,000$ pixels
- The number of parameters between two layers of that size is $10^5 \times (10^5 + 1)$!
- This approach is not feasible...
- Moreover, this approach does not take into account the local structure of images.

Activations

Different activations (typically ReLU) can be used in the intermediate layers.

Concerning the last layer: Given that the aim is a vector containing zeros except for a one, two designs are commonly used:

- Use a sigmoid as last activation
- Use any activation, but follow it by a soft-max operator

Softmax operator

Definition

The softmax operator $\sigma : \mathbb{R}^d \to \mathbb{R}^d$ is given by:

$$\forall \mathbf{x} \in \mathbb{R}^d, \forall k \in \{1, \ldots, d\} : \quad \sigma(\mathbf{x})_k = \frac{e^{\mathbf{x}_k}}{\displaystyle\sum_{i=1}^{d} e^{\mathbf{x}_i}}$$

Softmax operator

Definition

The softmax operator $\sigma : \mathbb{R}^d \to \mathbb{R}^d$ is given by:

$$\forall \mathbf{x} \in \mathbb{R}^d, \forall k \in \{1, \ldots, d\} : \quad \sigma(\mathbf{x})_k = \frac{e^{\mathbf{x}_k}}{\sum\limits_{i=1}^{d} e^{\mathbf{x}_i}}$$

Some properties

- $0 < \sigma(\mathbf{x})_k < 1$
- $\sum\limits_{i=1}^{d} \sigma(\mathbf{x})_i = 1$

Softmax operator

Definition

The softmax operator $\sigma : \mathbb{R}^d \to \mathbb{R}^d$ is given by:

$$\forall \mathbf{x} \in \mathbb{R}^d, \forall k \in \{1, \ldots, d\} : \quad \sigma(\mathbf{x})_k = \frac{e^{\mathbf{x}_k}}{\sum_{i=1}^{d} e^{\mathbf{x}_i}}$$

Some properties

- $0 < \sigma(\mathbf{x})_k < 1$
- $\sum_{i=1}^{d} \sigma(\mathbf{x})_i = 1$

Example

$$\mathbf{x} = \begin{pmatrix} 10.1 \\ 0 \\ -4.3 \\ 1.33 \end{pmatrix} \quad \sigma(\mathbf{x}) \approx \begin{pmatrix} 0.9998 \\ 0.000041 \\ 0.00000056 \\ 0.00016 \end{pmatrix}$$

Loss function for classification: cross-entropy

The preferred loss function for classification is cross-entropy:

For $\mathbf{y}$ in $[0, 1]^d$ and $\hat{\mathbf{y}}$ in $]0, 1]^d$:

$$H(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^{d} \mathbf{y}_i \log(\hat{\mathbf{y}}_i)$$

Loss function for classification: cross-entropy

The preferred loss function for classification is cross-entropy:

For $\mathbf{y}$ in $[0, 1]^d$ and $\hat{\mathbf{y}}$ in $]0, 1]^d$:

$$H(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^{d} \mathbf{y}_i \log(\hat{\mathbf{y}}_i)$$

For this to work, each $\hat{\mathbf{y}}_i$ must be strictly positive. Therefore when used in a NN the cross-entropy has to be preceded by a convenient operator, such as a sigmoid or a softmax.

Loss function for classification: cross-entropy

The preferred loss function for classification is cross-entropy:

For $\mathbf{y}$ in $[0, 1]^d$ and $\hat{\mathbf{y}}$ in $]0, 1]^d$:

$$H(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^{d} \mathbf{y}_i \log(\hat{\mathbf{y}}_i)$$

For this to work, each $\hat{\mathbf{y}}_i$ must be strictly positive. Therefore when used in a NN the cross-entropy has to be preceded by a convenient operator, such as a sigmoid or a softmax.

Note that the binary cross-entropy we saw during the first lecture is a particular case of cross-entropy.

Conclusion on fully-connected networks for image classification

Fully connected layers:

- scale badly to large size images
- do not take into account the local structure of images

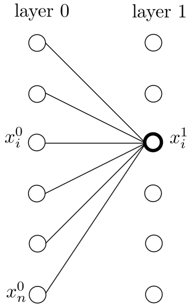
Today:

- Fully-connected networks are almost never used for image analysis.
- Fully-connected layers are only used in the middle (auto-encoders) or at the end (classification) of the pipeline.

Contents

Towards convolutional layers

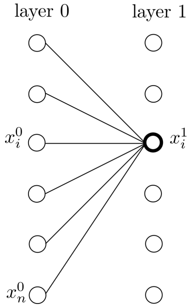
For illustration purposes, in the following slides images and filters will be displayed as rows of neurons - these can be seen as 1D arrays or as sections of 2D arrays.



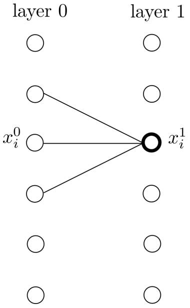
Fully connected layer:
$n(s+1)$ weights

Towards convolutional layers

For illustration purposes, in the following slides images and filters will be displayed as rows of neurons - these can be seen as 1D arrays or as sections of 2D arrays.



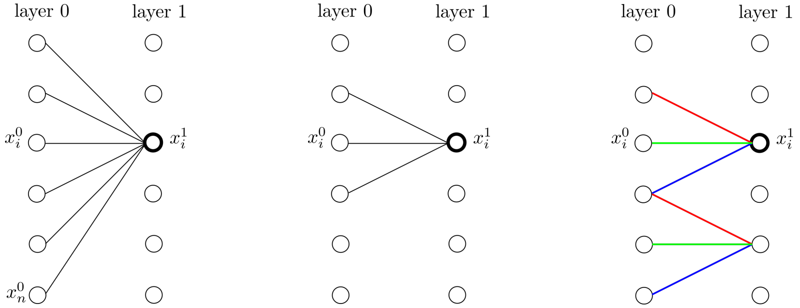
Fully connected layer:
$n(s+1)$ weights



Locally conn. layer:
$n(s+1)$ weights

Towards convolutional layers

For illustration purposes, in the following slides images and filters will be displayed as rows of neurons - these can be seen as 1D arrays or as sections of 2D arrays.



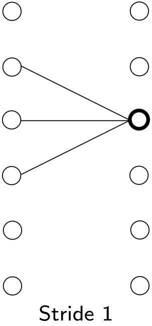
Fully connected layer: $n(s+1)$ weights

Locally conn. layer: $n(s+1)$ weights

Weight replication: $s+1$ weights

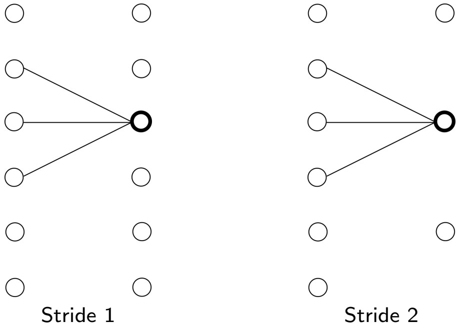
Stride

A convolutional layer can at the same time downsample the image by applying a sampling step, or *stride*.



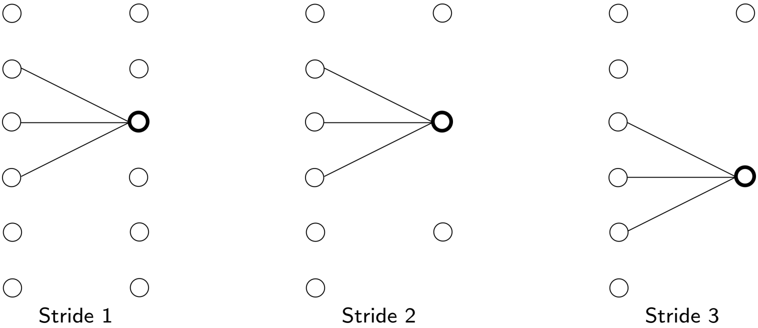
Stride

A convolutional layer can at the same time downsample the image by applying a sampling step, or *stride*.



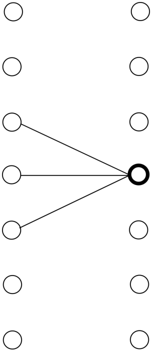
Stride

A convolutional layer can at the same time downsample the image by applying a sampling step, or *stride*.



Dilated convolutions

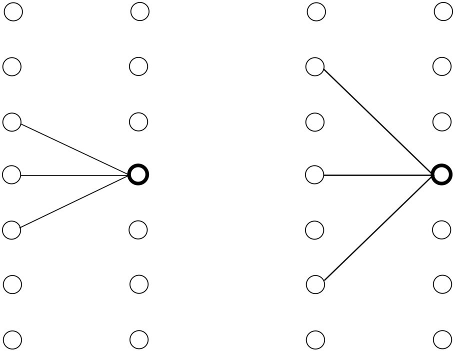
Dilated convolutions are used the increase the size of the receptive field of the network.



Normal convolution

Dilated convolutions

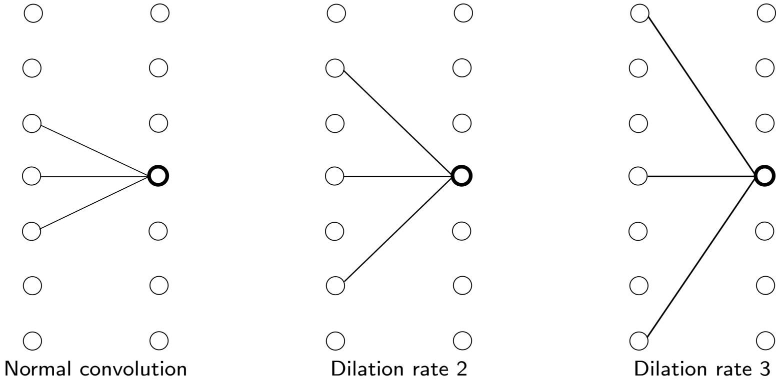
Dilated convolutions are used the increase the size of the receptive field of the network.



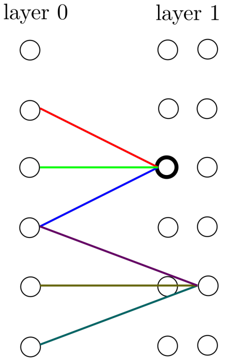
Normal convolution Dilation rate 2

Dilated convolutions

Dilated convolutions are used the increase the size of the receptive field of the network.

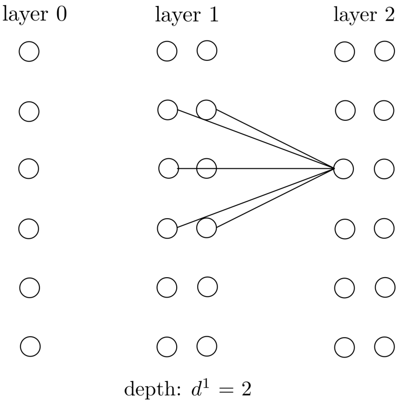


Several filters in the same convolutional layer

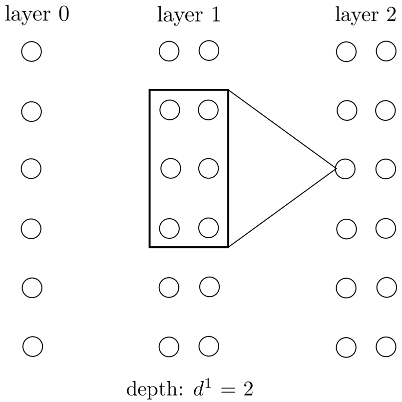


depth: $d^1 = 2$

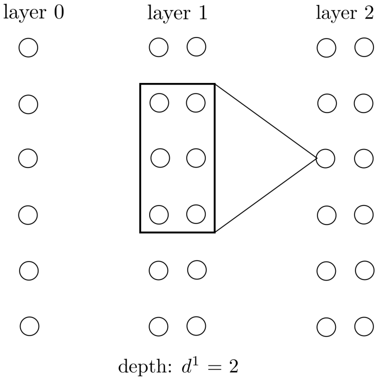
Several filters in the same convolutional layer



Several filters in the same convolutional layer

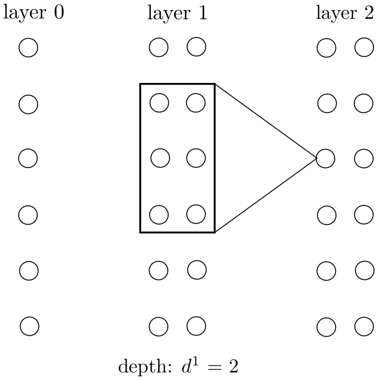


Consequences on the parameter number



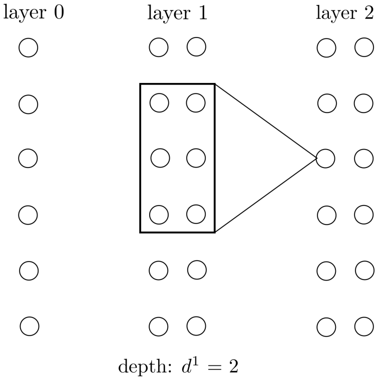
- How many parameters do we have in layer 1?

Consequences on the parameter number



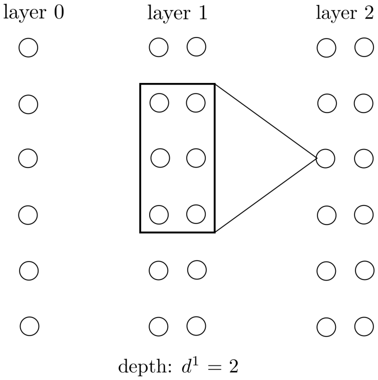
- How many parameters do we have in layer 1?
- $d^1 \times (s + 1)$

Consequences on the parameter number



- How many parameters do we have in layer 1?
- $d^1 \times (s + 1)$
- In layer 2?

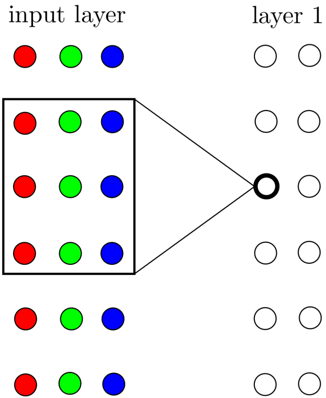
Consequences on the parameter number



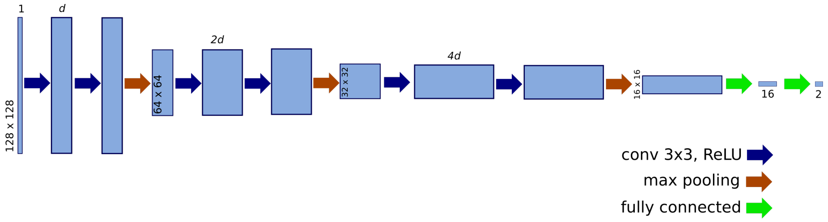
- How many parameters do we have in layer 1?
- $d^1 \times (s+1)$
- In layer 2?
- $d^2 \times (d^1 \times s + 1)$

Multi-valued images

An input image with p channels (for instance a colour image with 3 channels) can be represented by an input layer of depth 3

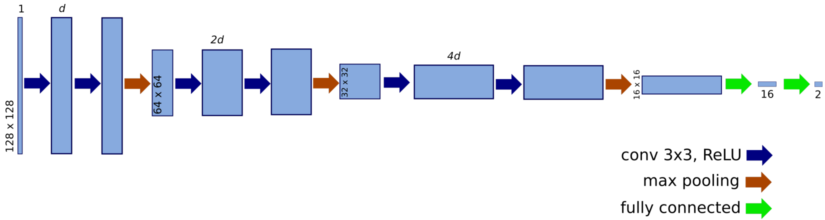


1D representations



Credits: NN is work of Robin Alais et al. Fundus image by Mikael Häggström, used with permission (CC0).

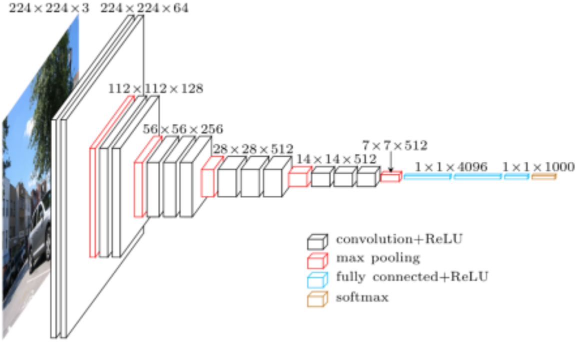
1D representations



This NN was used to estimate the position of the center of the macula on fundus images.

Credits: NN is work of Robin Alais et al. Fundus image by Mikael Häggström, used with permission (CC0).

2D representations



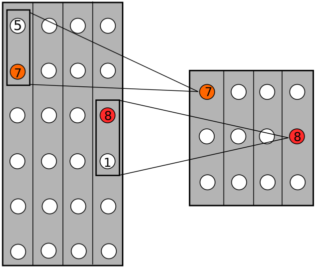
This network is used for image classification tasks.

Credits: VGG16 (From
https://www.cs.toronto.edu/ frossard/post/

Contents

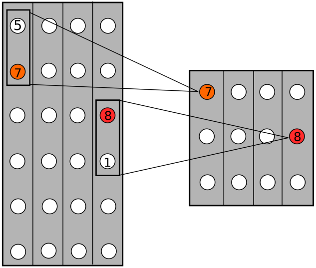
Max pooling

- Convolutional networks often contain subsampling steps. A common way of doing this today is by using *max pooling* layers with stride 2.
- Sampling is only applied along the spatial dimensions, not along the dimension of the filters.



Max pooling

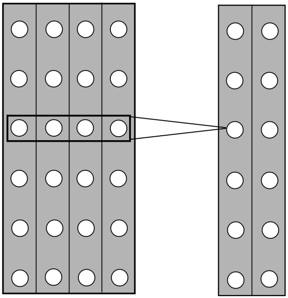
- Convolutional networks often contain subsampling steps. A common way of doing this today is by using *max pooling* layers with stride 2.
- Sampling is only applied along the spatial dimensions, not along the dimension of the filters.



Note however a current trend that consists in using convolutional layers with a stride of 2

Dimension reduction

1×1 convolutions are used to reduce the number of filters - this is called by some authors *dimension reduction*.



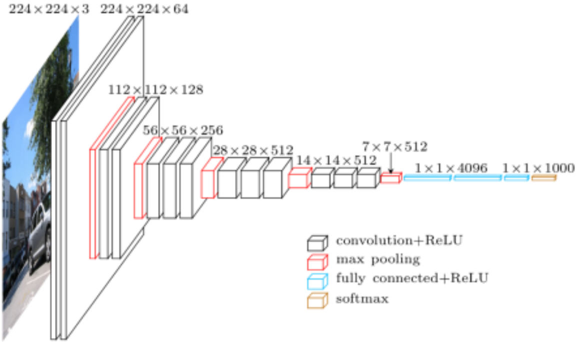
Main components of a convolutional neural network

Many successful architectures, especially for image classification, follow the same pattern:

1. Several iterations of: One or several convolutional layers, with increasing depth, followed by max pooling
2. A few fully connected layers

VGGnet

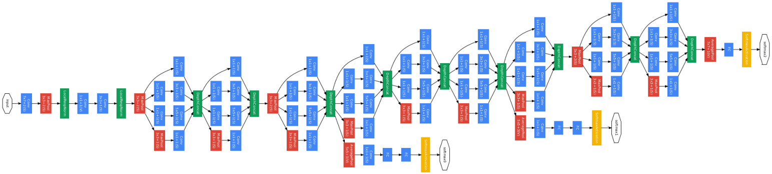
- Proposed by K. Simonyan and A. Zisserman from the University of Oxford [Simonyan and Zisserman, 2014]
- Runner-up in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2014.
- Number of parameters (VGG16): 138 million.



Credits: VGG16 (From https://www.cs.toronto.edu/ frossard/post/

GoogLeNet (a.k.a. Inception v1)

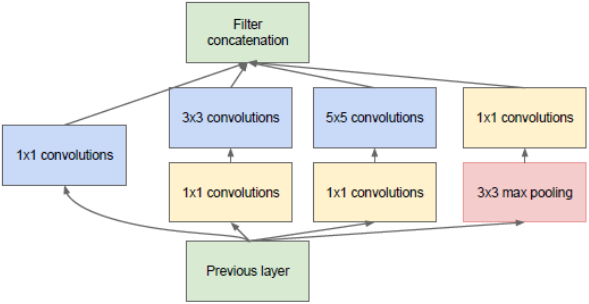
- Winner of the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2014.
- Number of parameters: *only* 5 million.



ResNet won the following year...

Credits: From [Szegedy et al., 2014]

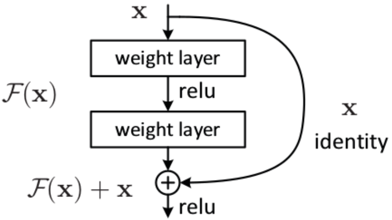
Inception module



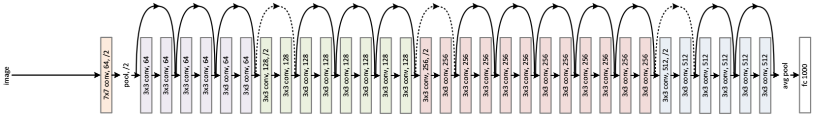
Credits: From [Szegedy et al., 2014]

ResNet

- Winner of the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2015.
- The authors tested up to 1202 layers. They reported no training difficulties, but overfitting [He et al., 2015]



Residual learning block



Residual network with 34 layers

Credits: From [He et al., 2015]

Current trends

- Small convolutions (3×3)
- Dimension reduction using 1×1 convolutions
- Increasing number of layers
- Skip connections

VGG, GoogLeNet and ResNet (and their variants) are still among the most used architectures for image classification and other related tasks.

Some deep learning libraries

Deep learning is a very competitive domain, where code sharing is very common.

- Tensorflow, by Google (Apache licence)
- PyTorch, Torch (Facebook - BSD licence)
- Caffe (Univ. of California, Berkeley - BSD licence)
- Microsoft Cognitive Toolkit (MIT licence)
- MatConvNet (for MatLab users)
- Theano (Montreal Institute for Learning Algorithms; not maintained anymore)

Comments

- Most of these libraries are distributed with very permissive licences
- Most of them use Python as prototyping language
- *Keras* is a very easy to use interface to Tensorflow, Theano and CNTK.

Contents

A revolution in image analysis

- Deep learning has brought an undeniable break-through in image analysis (as in other fields)
- A significant part of research efforts in image analysis today is based on deep learning
- Its applications are ubiquitous

Limitations

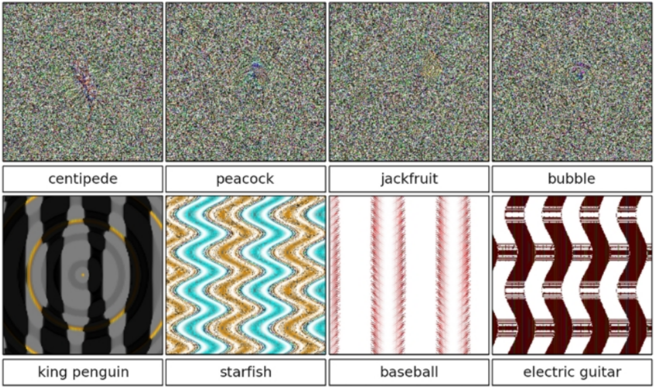
For a deep-learning solution to work, you need:

- Enough annotated data
- A lot of fiddling (different architectures; hyper-parameters; optimization)
- One (or, even better, several) powerful GPUs

Moreover, these models lack interpretability.

ConvNets can be fooled

Deep learning can produce astonishing results [Nguyen et al., 2015]...



References I

[Chellapilla et al., 2006] Chellapilla, K., Puri, S., and Simard, P. (2006). High Performance Convolutional Neural Networks for Document Processing. In *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft.

[Ciresan et al., 2012] Ciresan, D., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2012). Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 2843–2851. Curran Associates, Inc.

[Cireșan et al., 2011] Cireșan, D., Meier, U., Masci, J., and Schmidhuber, J. (2011). A committee of neural networks for traffic sign classification. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1918–1921. IEEE.

[Cireșan et al., 2013] Cireșan, D. C., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2013). Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, pages 411–418. Springer, Berlin, Heidelberg.

[Cireșan et al., 2010] Cireșan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J. (2010). Deep, Big, Simple Neural Nets for Handwritten Digit Recognition. *Neural Computation*, 22(12):3207–3220.

References II

[Everingham et al., 2014] Everingham, M., Eslami, S. M. A., Gool, L. V., Williams, C. K. I., Winn, J., and Zisserman, A. (2014). The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1):98–136.

[Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338.

[Fukushima, 1979] Fukushima, K. (1979). Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position- Neocognitron. *ELECTRON. & COMMUN. JAPAN*, 62(10):11–18.

[Fukushima, 1980] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

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

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

References III

[LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551.

[Lecun et al., 1998] 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.

[Nguyen et al., 2015] Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436.

[Simard et al., 1993] Simard, P., LeCun, Y., and Denker, J. S. (1993). Efficient pattern recognition using a new transformation distance. In *Advances in neural information processing systems*, pages 50–58.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Szegedy et al., 2014] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going Deeper with Convolutions. *arXiv:1409.4842 [cs]*. arXiv: 1409.4842.

References IV

[zhang et al., 2011] zhang, x., thibault, g., and decenciere, e. (2011). Application of the Morphological Ultimate Opening to the Detection of Microaneurysms on Eye Fundus Images from Clinical Databases. In *ICS'13*.

[Zhang et al., 2014] Zhang, X., Thibault, G., Decencière, E., Marcotegui, B., Laÿ, B., Danno, R., Cazuguel, G., Quellec, G., Lamard, M., Massin, P., Chabouis, A., Victor, Z., and Erginay, A. (2014). Exudate detection in color retinal images for mass screening of diabetic retinopathy. *Medical Image Analysis*, 18(7):1026–1043.