Deep learning for image analysis quick introduction

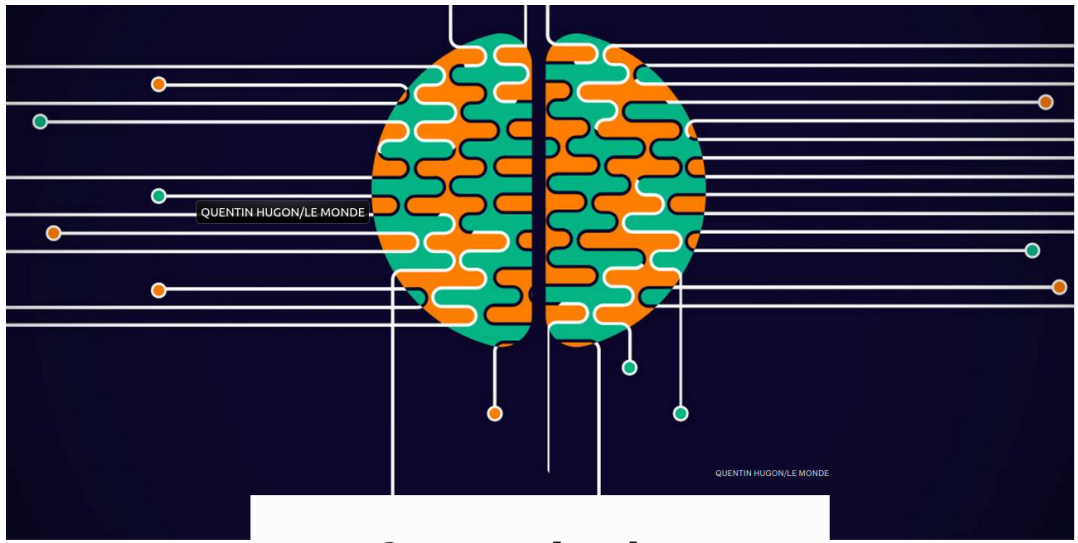
E. Decencière

Mines Paris
PSL Research University
Center for Mathematical Morphology

Contents

Contents

Le Monde, 2015



Comment le « deep learning » révolutionne l'intelligence artificielle

Par Morgane Tual

Nature, 2016


nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO
CONSERVATION
SONGBIRDS À LA CARTE
Illegal harvest of millions of Mediterranean birds
PAGE 452
RESEARCH ETHICS
SAFEGUARD TRANSPARENCY
Don't let openness backfire on individuals
PAGE 459
POPULAR SCIENCE
WHEN GENES GOT 'SELFISH'
Dawkins's calling card 40 years on
PAGE 462
NATUREASIA.COM
28 January 2016
Vol. 529. No. 7587

Le Monde, 2019

Le prix Turing récompense trois pionniers de l'intelligence artificielle (IA)

L'association américaine ACM a remis son prestigieux prix aux chercheurs français, canadien et britannique : Yann LeCun, Yoshua Bengio et Geoffrey Hinton.

Par David Larousserie • Publié le 27 mars 2019 à 11h01 - Mis à jour le 29 mars 2019 à 12h11

Le Figaro, juillet 2017

Pour Elon Musk, l'intelligence artificielle pourrait menacer la civilisation

L'entrepreneur américain, qui a fondé Tesla, a alerté les politiques américains sur la nécessité de réguler l'intelligence artificielle.

Par **Le Figaro**
Publié le 18/07/2017 à 06:00, mis à jour le 18/07/2017 à 11:25

Artificial neural networks and deep learning chronology

- 1958: Perceptron [Rosenblatt, 1958].

Artificial neural networks and deep learning chronology

- 1958: Perceptron [Rosenblatt, 1958].
- 1979: Convolutional neural networks [Fukushima, 1979].

Artificial neural networks and deep learning chronology

- 1958: Perceptron [Rosenblatt, 1958].
- 1979: Convolutional neural networks [Fukushima, 1979].
- 1980's: Backpropagation algorithm [Werbos, 1982, LeCun, 1985].

Artificial neural networks and deep learning chronology

- 1958: Perceptron [Rosenblatt, 1958].
- 1979: Convolutional neural networks [Fukushima, 1979].
- 1980's: Backpropagation algorithm [Werbos, 1982, LeCun, 1985].
- 2006-: Implementations on Graphical Processing Units

Artificial neural networks and deep learning chronology

- 1958: Perceptron [Rosenblatt, 1958].
- 1979: Convolutional neural networks [Fukushima, 1979].
- 1980's: Backpropagation algorithm [Werbos, 1982, LeCun, 1985].
- 2006-: Implementations on Graphical Processing Units
- 2012: Imagenet image classification won by a convolutional neural network [Krizhevsky et al., 2012].

Contents

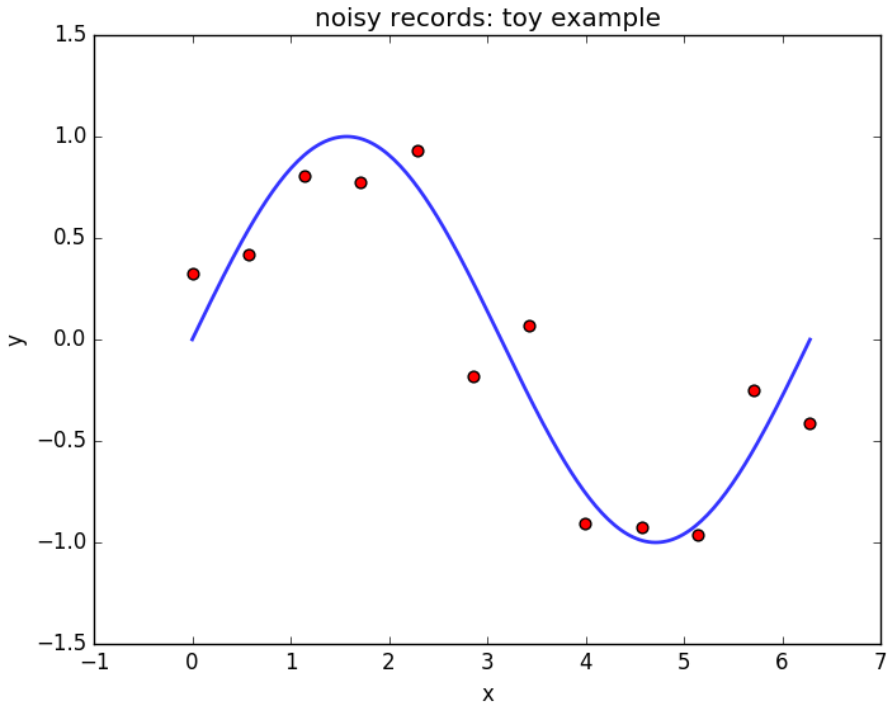
Machine Learning: basic definitions

- Machine Learning aims at predicting some output y from an input (or measurement) x:

$$y = f(x) \tag{1}$$

- Machine Learning aims at finding (learning) f from available data.
- The data that is used to learn f is called **training set**, denoted by $\mathbf{X}$.
- In this general formulation, there is no particular limitation as to the mathematical nature of x and y.

A simple example: polynomial curve fitting[1]



- From a set of measured points (x_i, y_i) (red), we would like to build a model to predict the value y for any given x.
- The true function is $g(x) = \sin(x)$ (displayed in blue).
- The measurements y_i are noisy outputs of that function, i.e.

$$y_i = \sin(x_i) + \epsilon \ , \qquad \epsilon \sim \mathcal{N}(0, 0.2) \tag{2}$$

[1] Example adapted from [Bishop, 2006]

A simple example: polynomial curve fitting

- We use the following polynomial model:

$$\begin{aligned} f(x) &= a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m \\ &= \boldsymbol{\theta}^T \boldsymbol{\phi}(x) \end{aligned} \tag{3}$$

- Parameter vector: $\boldsymbol{\theta} = (a_0, a_1, \ldots, a_m)^T$
- Here, the initial measurement x is a scalar. In our model, we map x to a higher dimensional space:

$$\begin{aligned} \boldsymbol{\phi} : \mathbb{R}^P &\rightarrow \mathbb{R}^Q \\ x &\rightarrow \boldsymbol{\phi}(x) = (1, x, x^2, \ldots, x^m)^T \end{aligned} \tag{4}$$

- The model is linear in the parameters θ and linear in $\boldsymbol{\phi}$, but for $m > 1$, the model is not linear in x.

A simple example: polynomial curve fitting

- One classical approach is to minimize the least squared error between measured and predicted values:

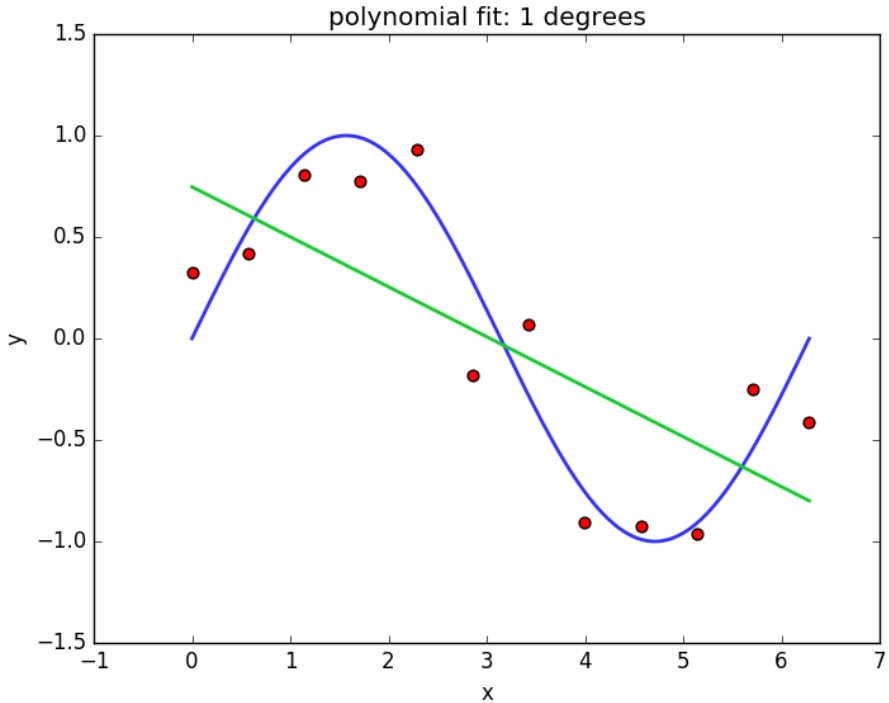
$$\begin{aligned} \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) &= \min_{\boldsymbol{\theta}} \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ &= \min_{\boldsymbol{\theta}} \sum_{i=1}^{N} (y_i - \boldsymbol{\theta}^T \boldsymbol{\phi}(x_i))^2 \end{aligned} \tag{5}$$

- This can be achieved by setting the gradient with respect to $\boldsymbol{\theta}$ to zero:

$$\nabla_{\boldsymbol{\theta}} L = (\frac{\partial L}{\partial a_0}, \frac{\partial L}{\partial a_1}, \ldots, \frac{\partial L}{\partial a_m})^T = 0 \tag{6}$$

- Unlike most optimization problems in this course, this leads to an analytical solution for $\boldsymbol{\theta}$. This is known as **linear regression**. For more details, we refer to [Hastie et al., 2009].

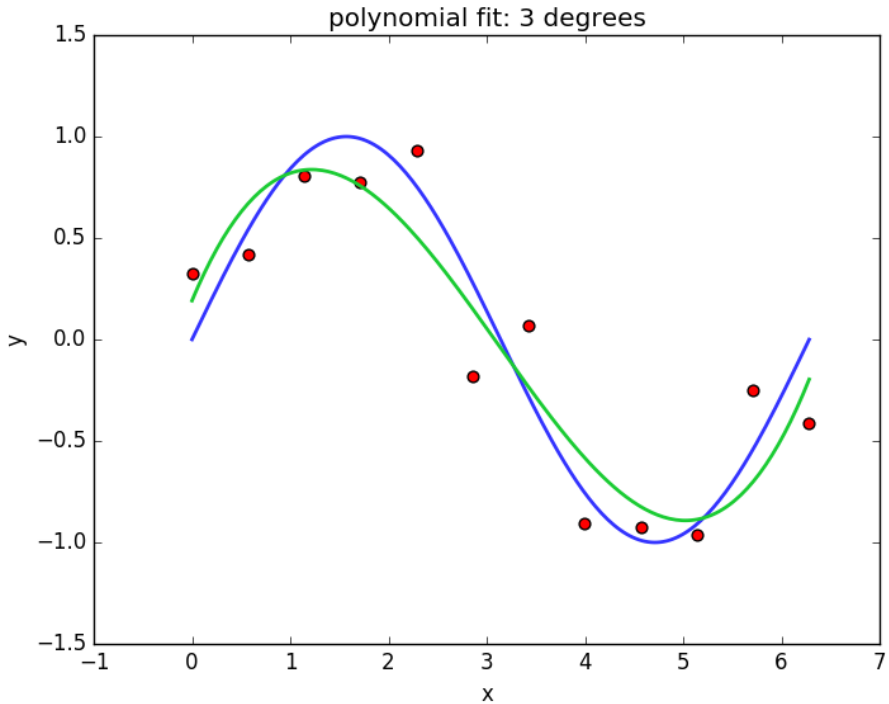
Overfitting and underfitting



$$\|\boldsymbol{\theta}\|^2 = 0.67$$

For $m = 1$, the model is linear in its inputs. The solution is not capable of modeling the measured data points; we get a poor approximation of the original function. The family of functions we have used was not complex enough to model the true data distribution. We also speak of **underfitting**.

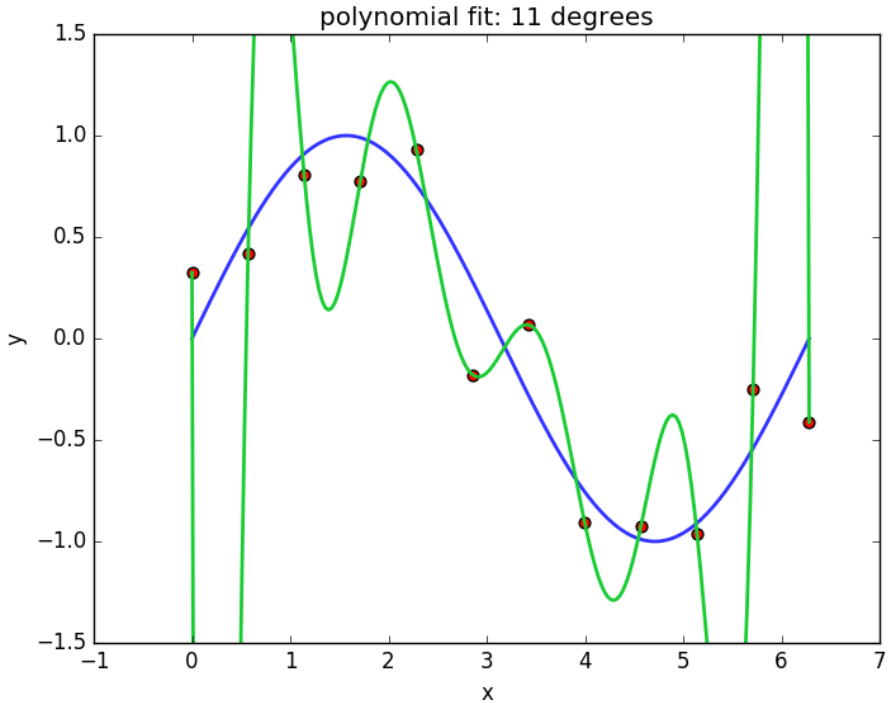
Overfitting and underfitting



$$\|\boldsymbol{\theta}\|^2 = 1.72$$

For $m = 3$, we obtain a solution that seems to be quite right: it is sufficiently complex to model the true data distribution, but not too complex to model the small variations which are due to noise.

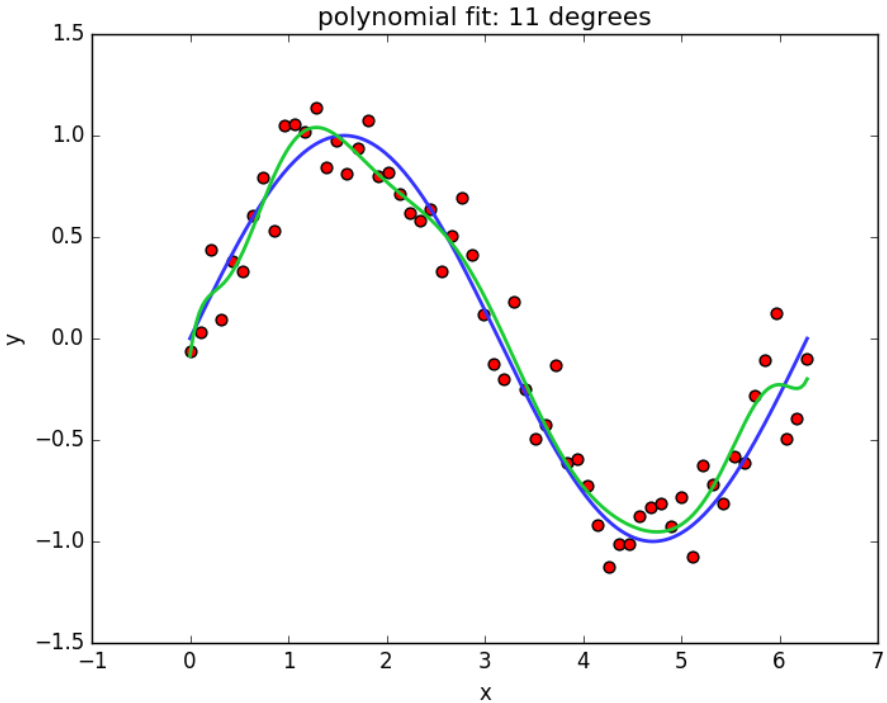
Overfitting and underfitting



$$\|\boldsymbol{\theta}\|^2 \approx 10^7$$

For $m = 11$, we obtain a solution that has zero error (the function passes through every point of the training set). But the coefficients with large absolute values that cancel each other precisely on the training points lead to a highly unstable function. We speak of **overfitting** and **poor generalization**.

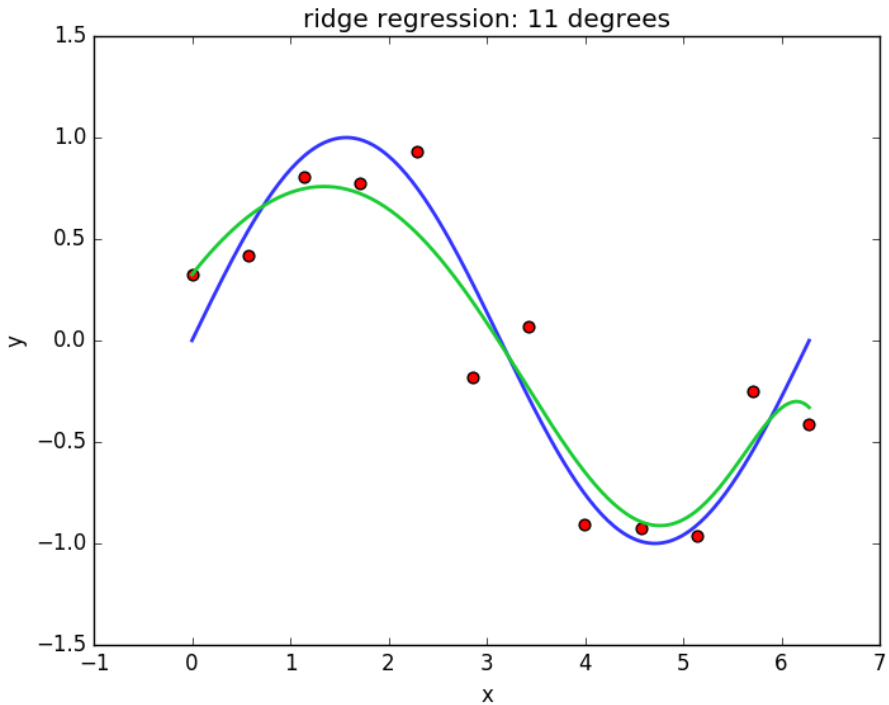
Overfitting and underfitting



$$\|\boldsymbol{\theta}\|^2 = 5647$$

One way of reducing overfitting is to increase the number of samples. Even if the function is complex, it cannot be "too wild", as it has to find a compromise between many training samples. This however implies the annotation (or measurement) of more samples.

Overfitting and underfitting



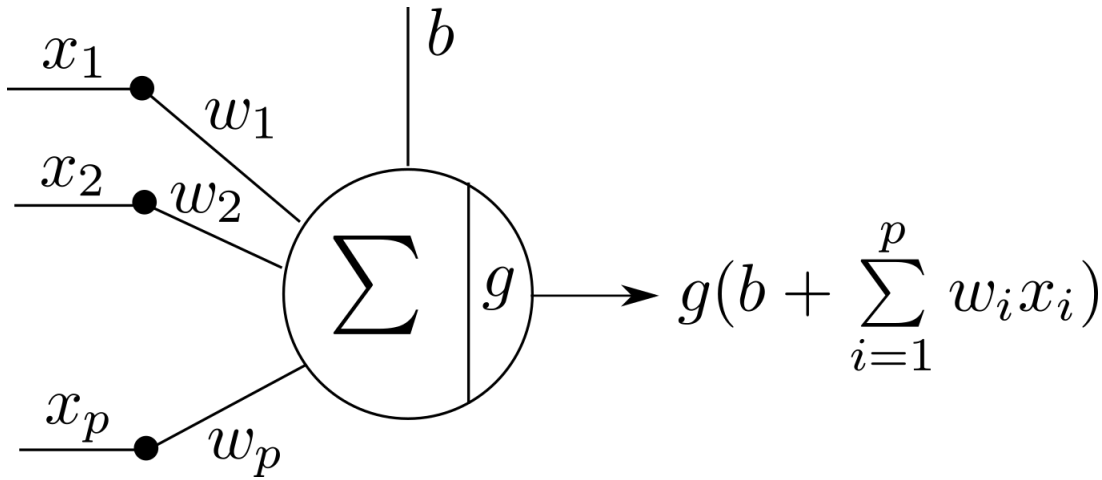
$$\|\boldsymbol{\theta}\|^2 = 0.41$$

Another way of preventing overfitting without increasing the number of samples, is to add a penalization term in the optimization procedure. This is also known as **regularization**:

$$L = \sum_{i=1}^{N} (y_i - \boldsymbol{\theta}^T \boldsymbol{\phi}(x_i))^2 + \lambda \|\boldsymbol{\theta}\|^2 \tag{7}$$

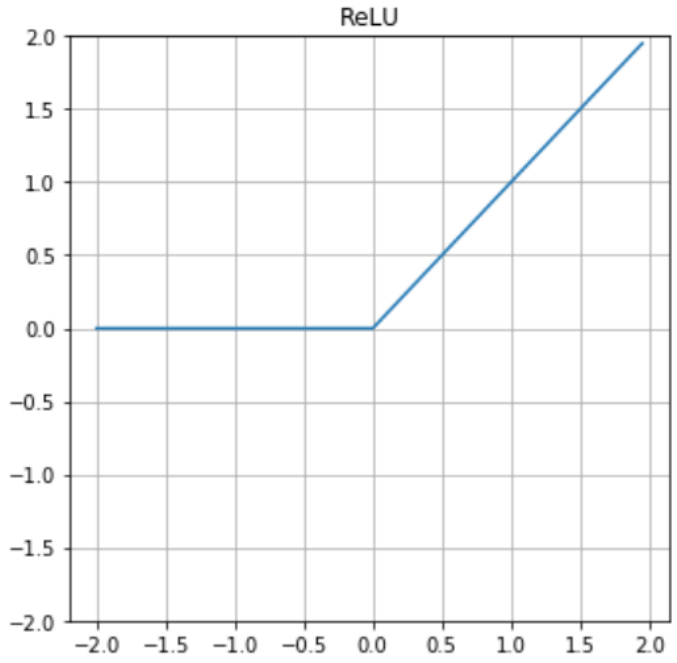
Contents

Artificial neuron



Activation: rectified linear unit (ReLU)

$$\mathrm{g}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

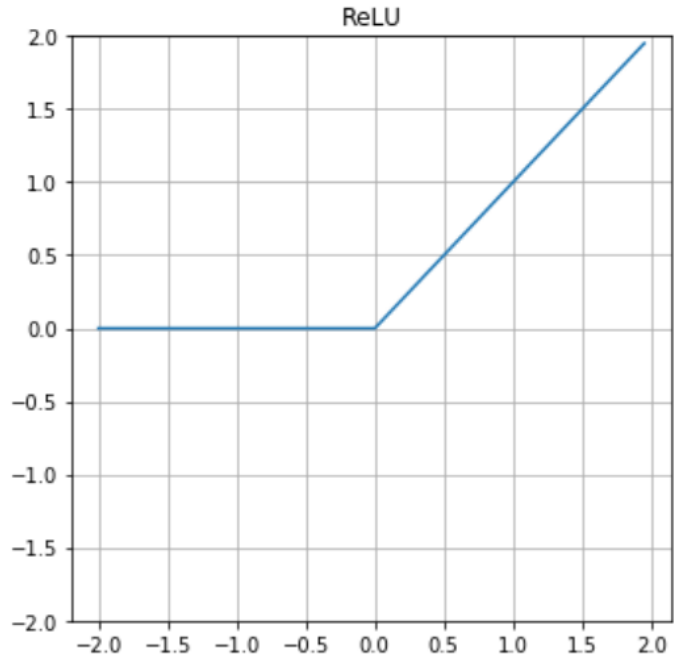


Remarks

+ Usable gradient when activated
+ Fast to compute
+ High abstraction

Activation: rectified linear unit (ReLU)

$$\mathrm{g}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

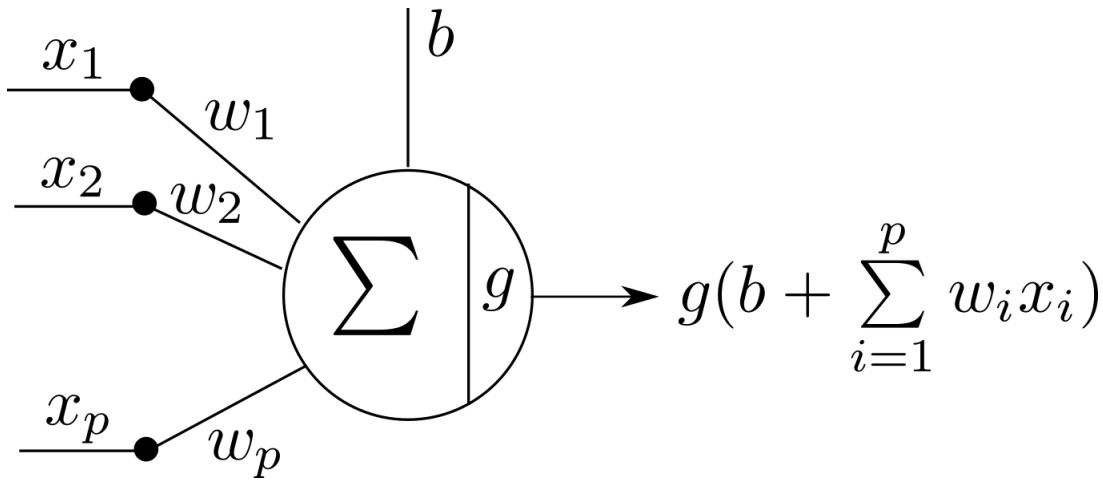


Remarks

+ Usable gradient when activated
+ Fast to compute
+ High abstraction

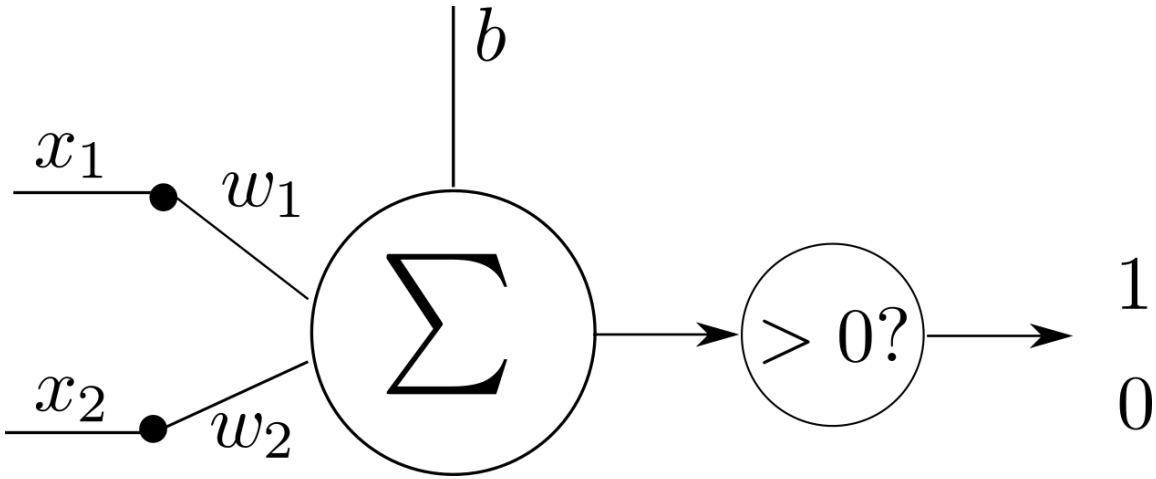
ReLU is the most commonly used activation function.

What can an artificial neuron compute?



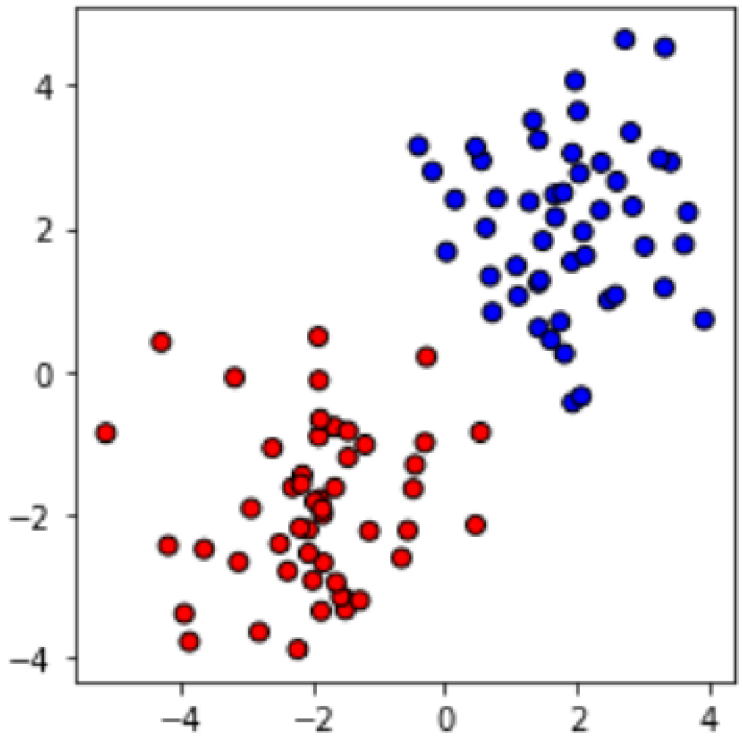
In $\mathbb{R}^p$, $b + \sum_{i=1}^{p} w_i x_i = 0$ corresponds to a hyperplane H. For a given point $\mathbf{x} = \{x_1, \ldots, x_p\}$, decisions are made according to the side of the hyperplane it belongs to.

Example

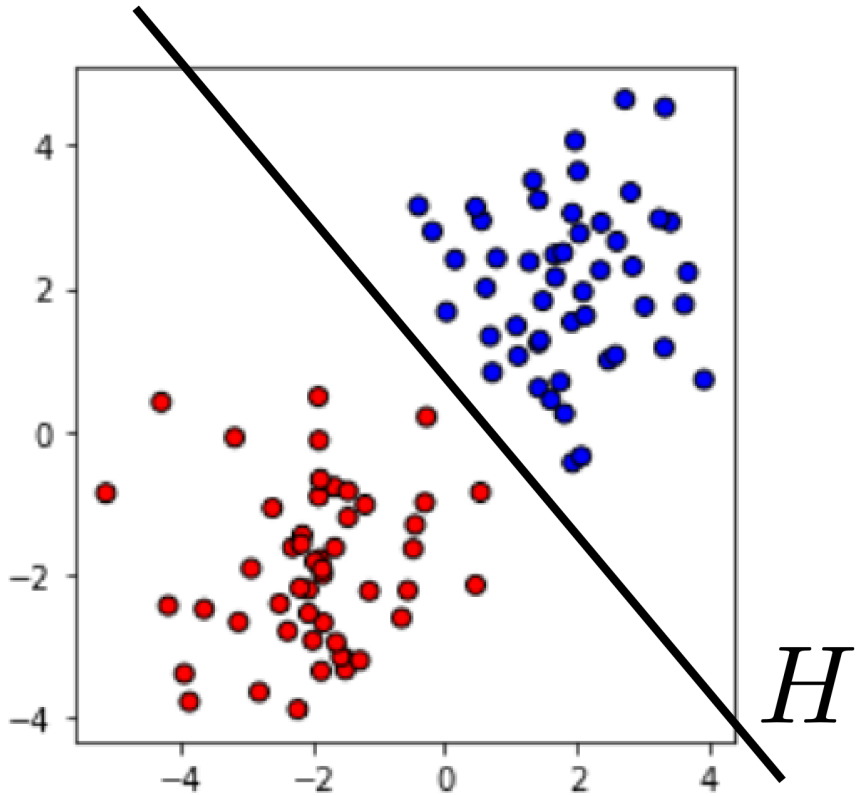


- $p = 2$: 2-dimensional inputs (can be represented on a screen!)
- Activation: binary
- Classification problem

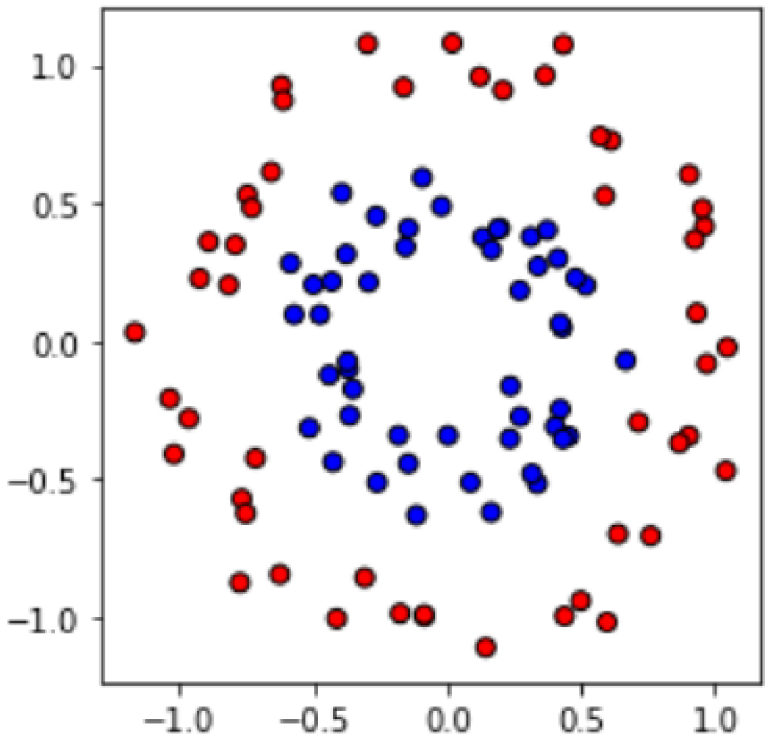
Gaussian clouds



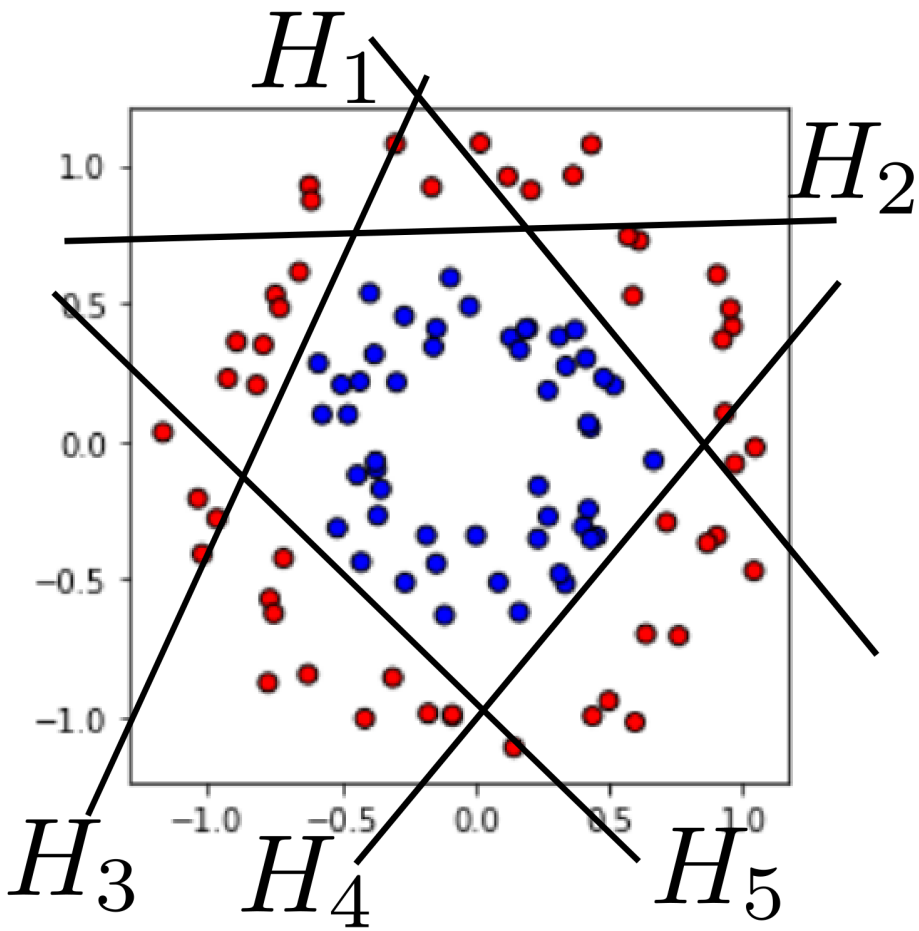
Gaussian clouds



Circles



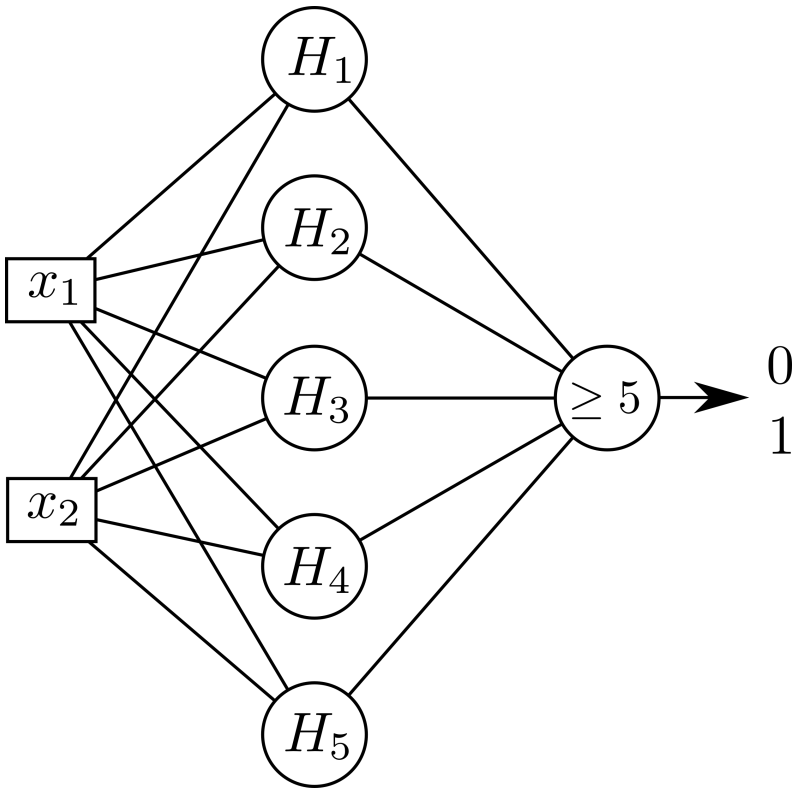
Circles

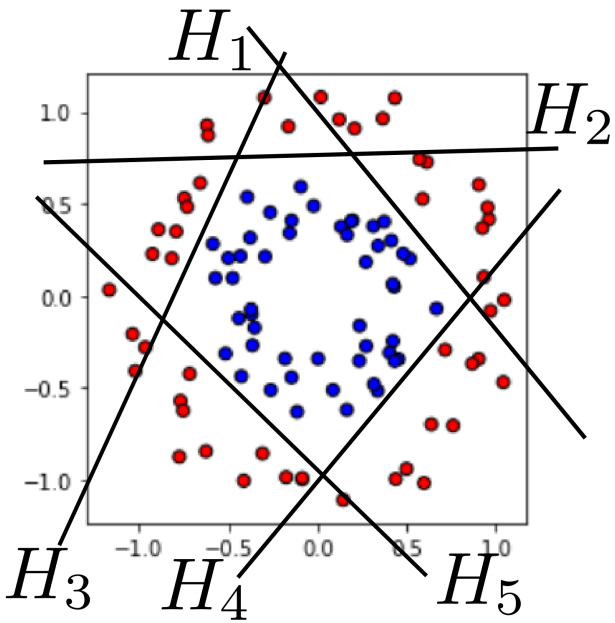


Playing with artificial neural networks

`https://playground.tensorflow.org`

Solution

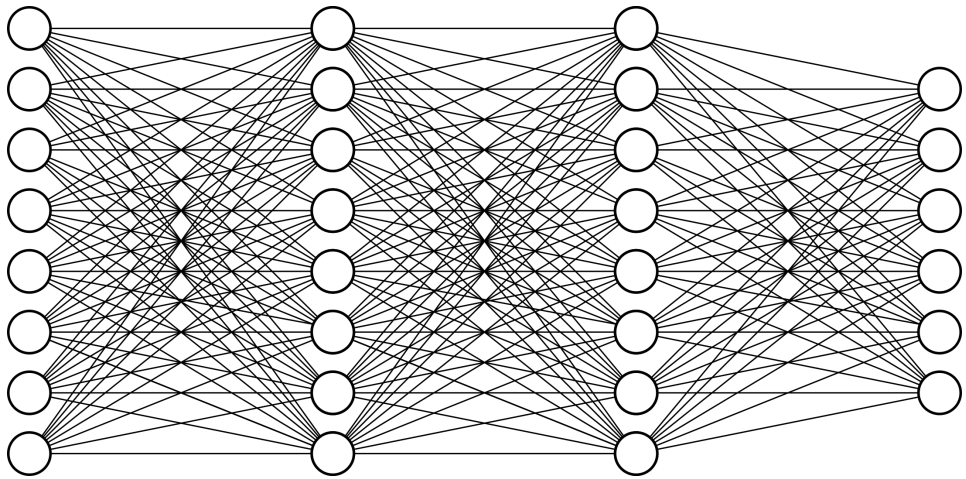




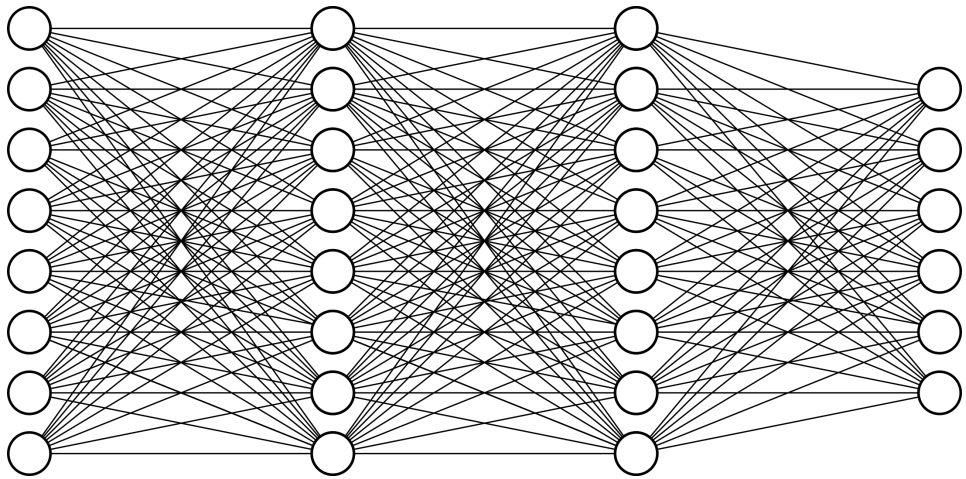
Universal approximation theorem
[Cybenko, 1989, Hornik, 1991]

Any continuous real-valued function of $[0, 1]^p$ can be approximated by an artificial neural network with a single hidden layer.

Multi-layered perceptron



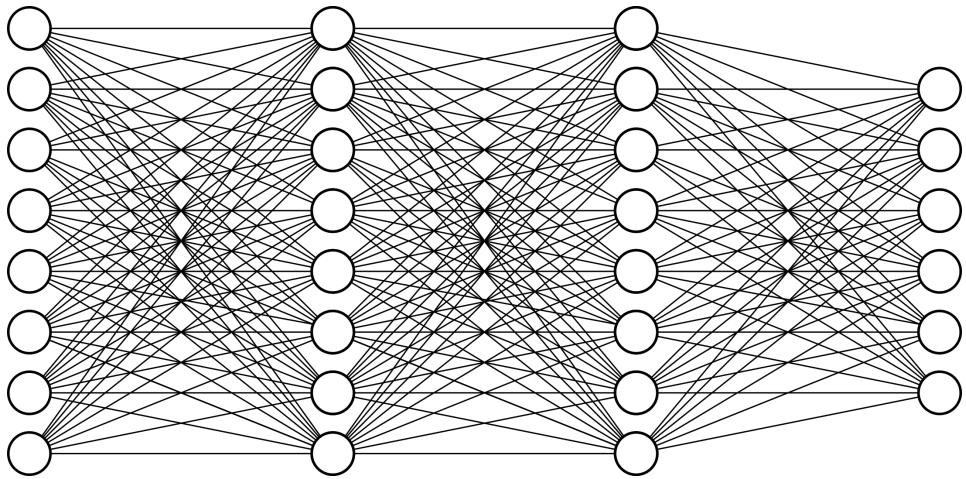
Multi-layered perceptron



Deep learning

- Artificial neural networks with *many* layers.

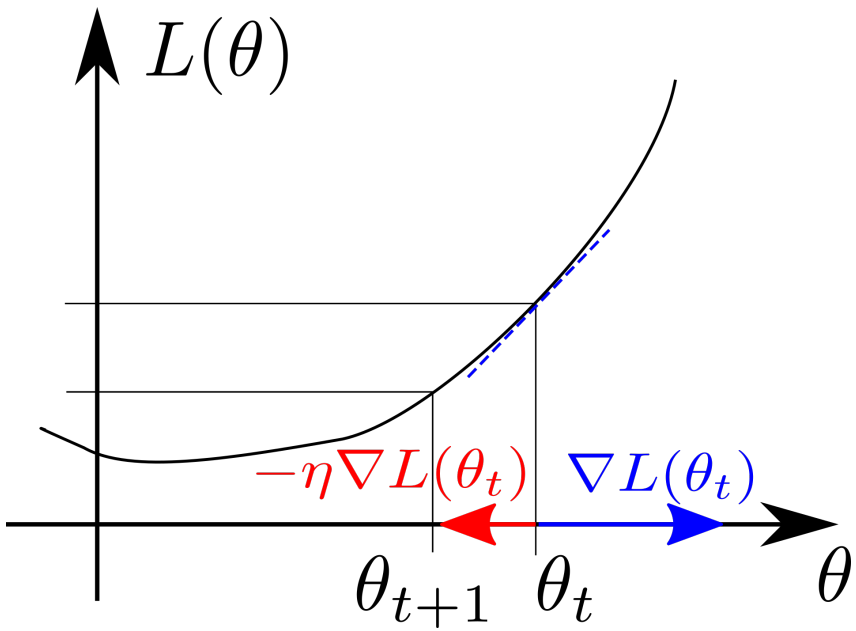
Multi-layered perceptron



Deep learning

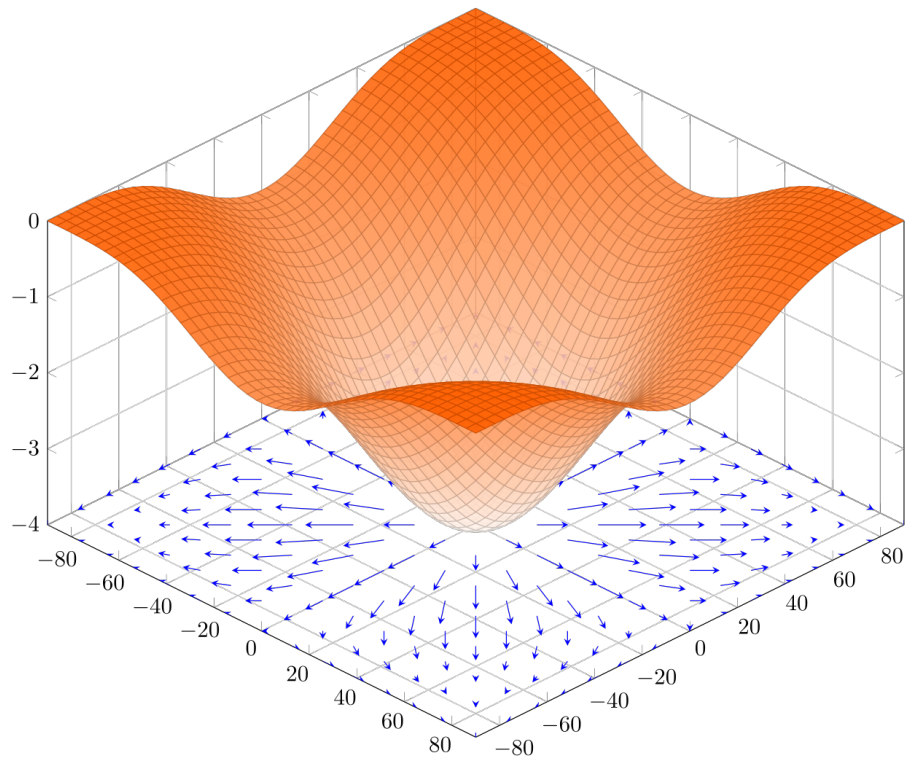
- Artificial neural networks with *many* layers.
- Features get more specific with depth

Training a neural network: gradient descent



$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\nabla L(\boldsymbol{\theta}_t)$$

How to minimize a function?



Definition: gradient

Let L be a derivable function from $\mathbb{R}^n$ into $\mathbb{R}$. Its gradient ∇L is:

$$\nabla L(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial L}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}) \\ \vdots \\ \frac{\partial L}{\partial \boldsymbol{\theta}_n}(\boldsymbol{\theta}) \end{pmatrix}$$

Credits: By MartinThoma, CC0, https://commons.wikimedia.org/

Contents

A picture is worth a thousand words

Grey level values around the left eye of the faun

A picture is worth a thousand words

Grey level values around the left eye of the faun

- Deep learning excels in image analysis.

The role of annotated image databases

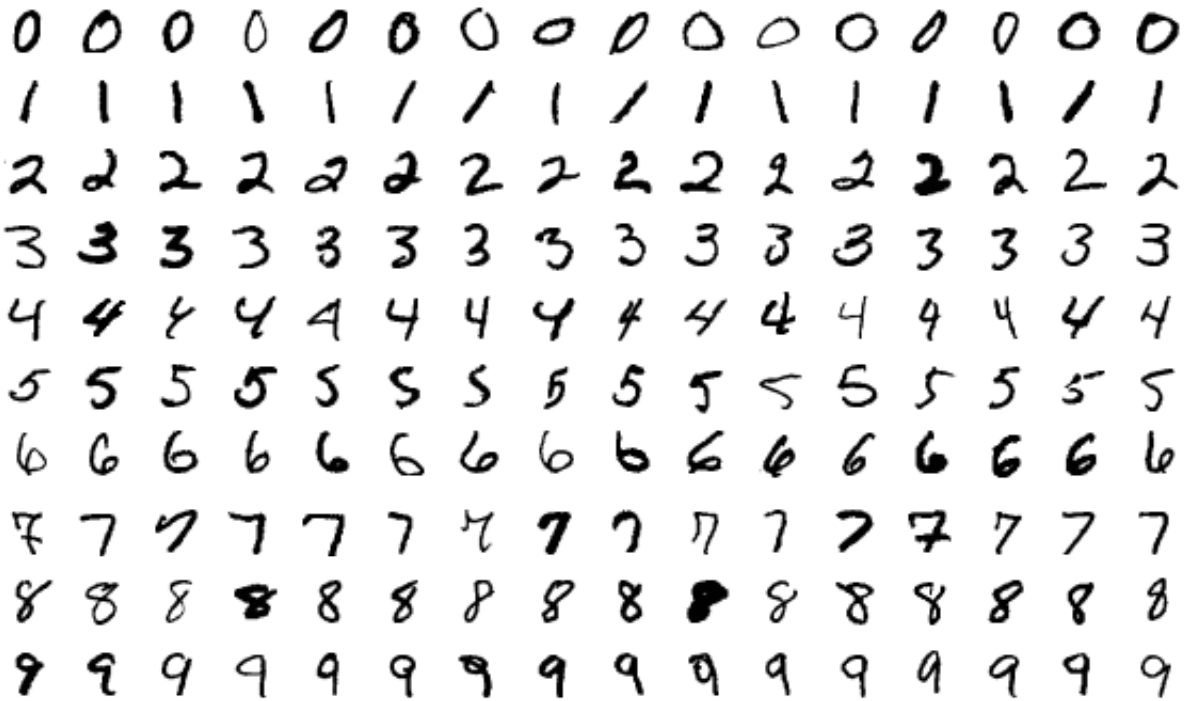
Image databases including *annotations* (typically some kind of high level information) are essential to the development of *supervised* machine learning methods for image analysis.

Annotations

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

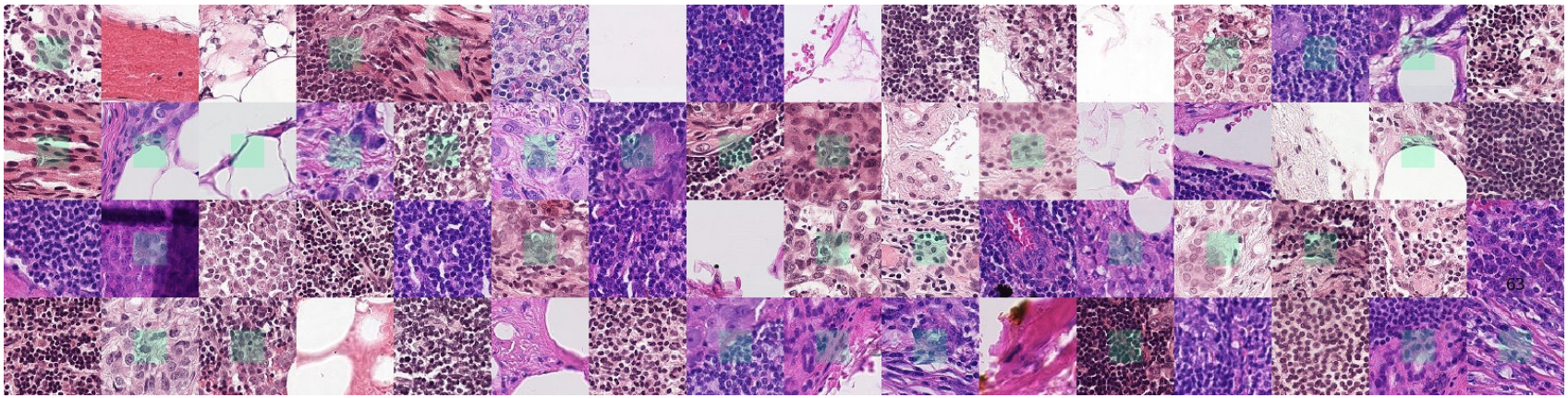
MNIST database [Lecun et al., 1998]

- 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



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

PatchCamelyon database



- $327\,680$ images of size 96×96
- Binary label (tumor tissue in center region or not)
- `https://github.com/basveeling/pcam`

ImageNet project [Russakovsky et al., 2015]

Between 2010 and 2017 ImageNet organized an annual challenge: The ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It represented a breakthrough in the design of image analysis challenges by its size.

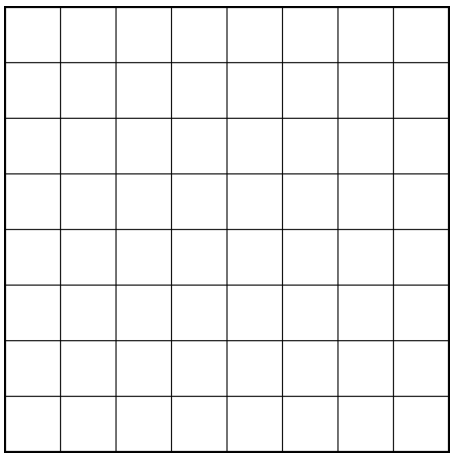
Image classification task

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

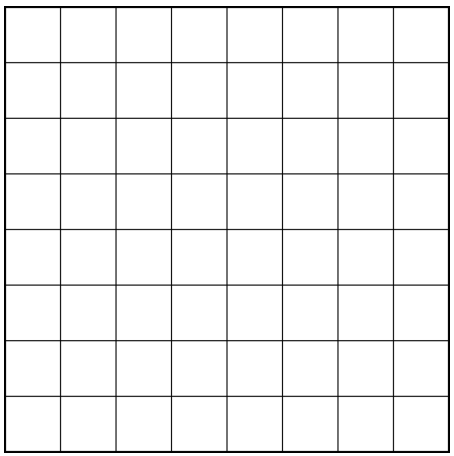
ImageNet projet

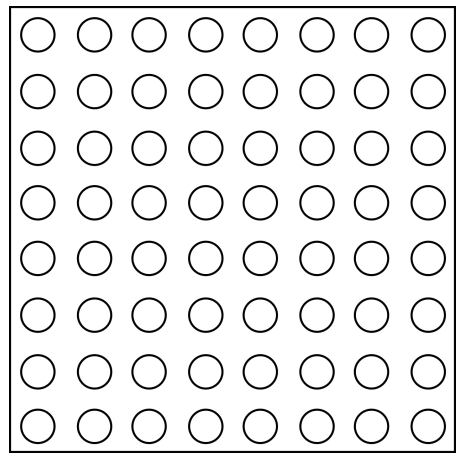
Examples from the *acoustic guitar* class

Layers representation

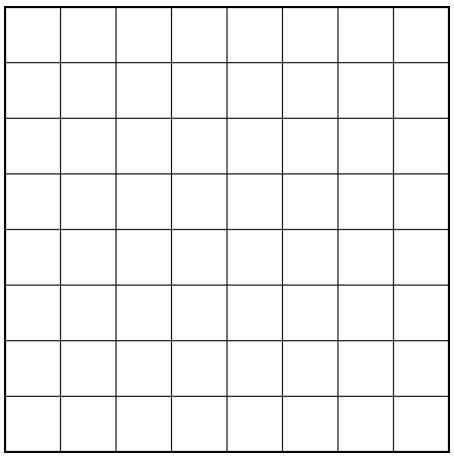


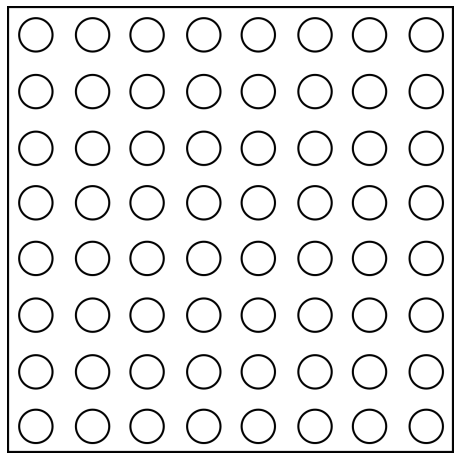
Layers representation

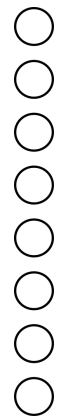




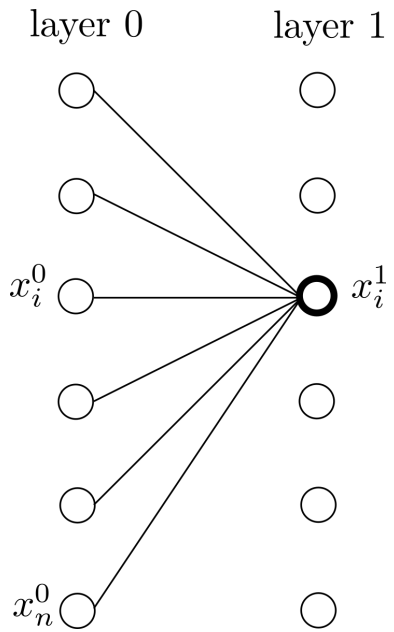
Layers representation





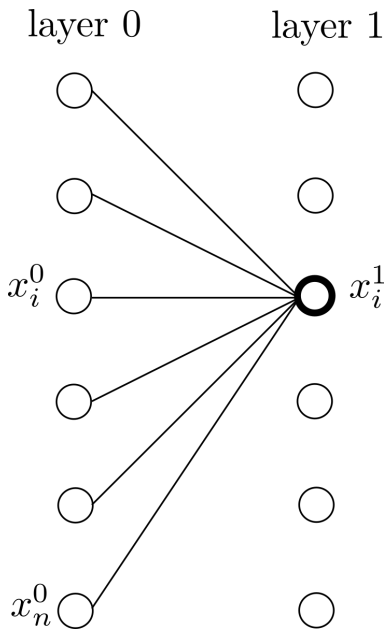


Towards convolutional layers

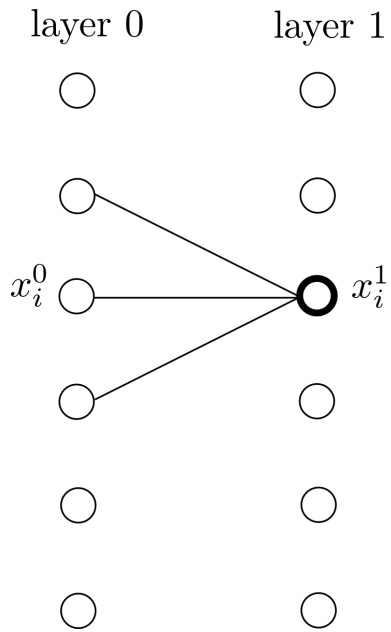


Fully connected layer:
$n(n+1)$ weights

Towards convolutional layers

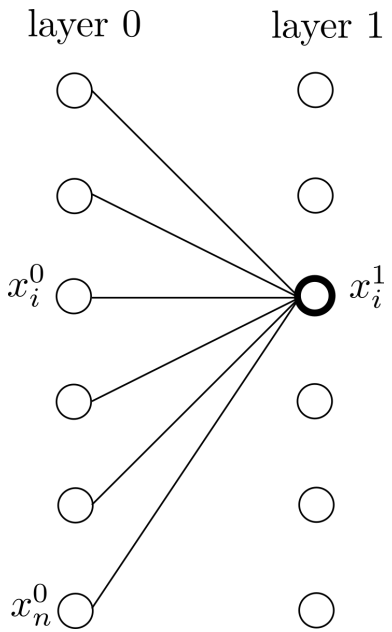


Fully connected layer:
$n(n+1)$ weights

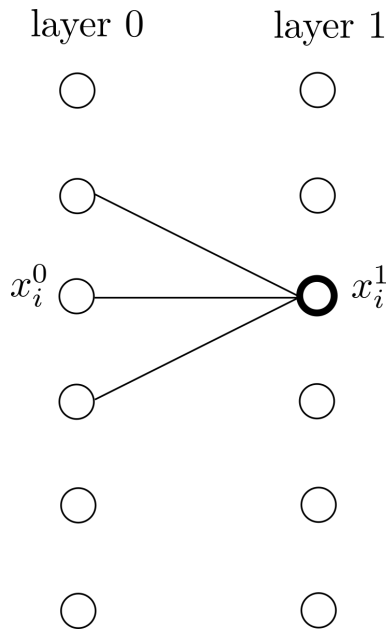


Locally conn. layer:
$n(s+1)$ weights

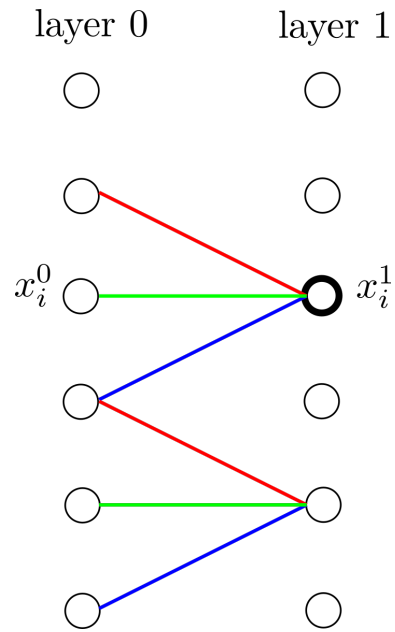
Towards convolutional layers



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



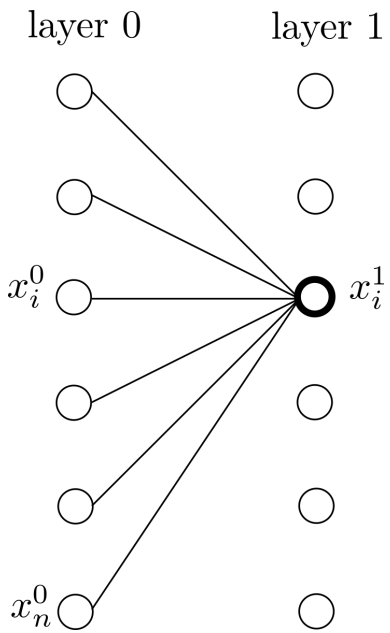
Locally conn. layer:
$n(s+1)$ weights



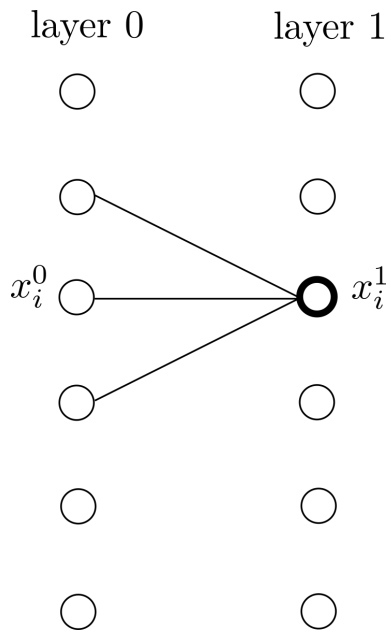
Weight replication: $s+1$ weights.
Convolutional layer.

Towards convolutional layers: some figures

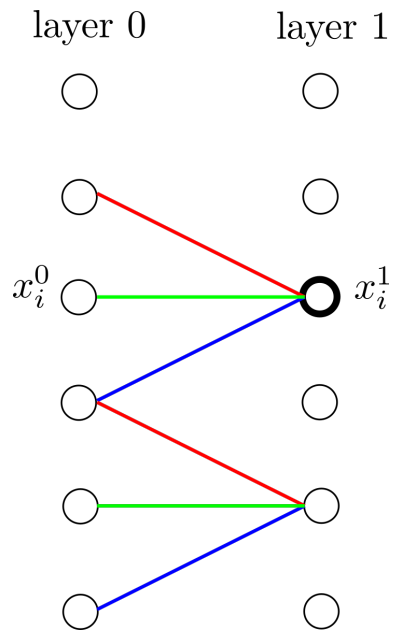
- 3×3 convolutions: $s = 9$
- Toy image: $n = 28 \times 28 = 784$
- Typical image: $n = 1000 \times 1000 = 10^6$



Fully connected layer:
$n(n+1)$ weights
$\approx 6.10^5$
$\approx 10^{12}$

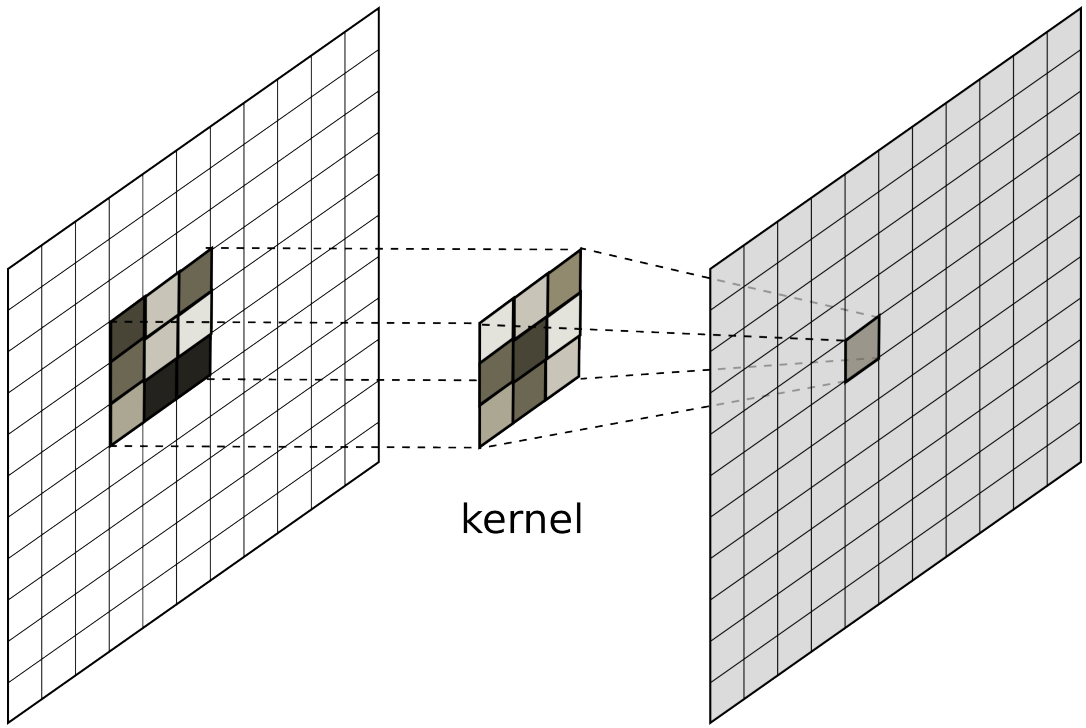


Locally conn. layer:
$n(s+1)$ weights
7840
10^7

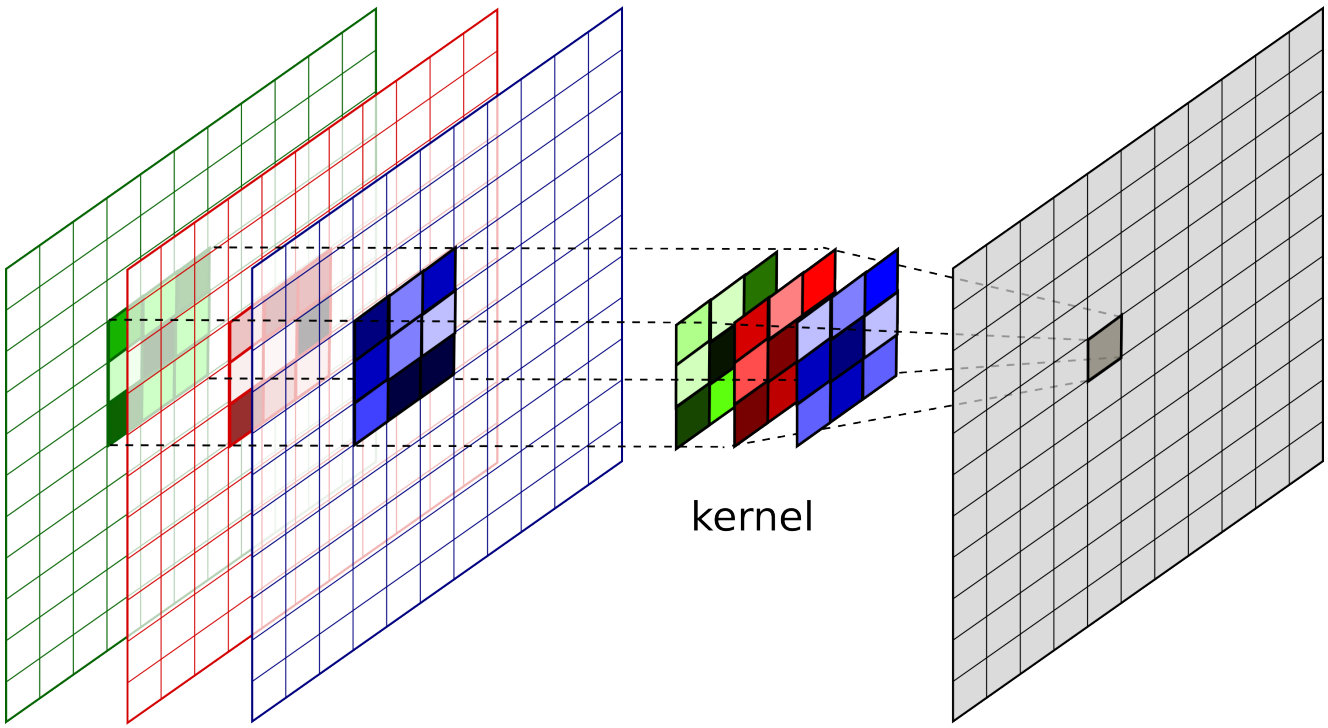


Weight replication: $s+1$ weights.
10
10

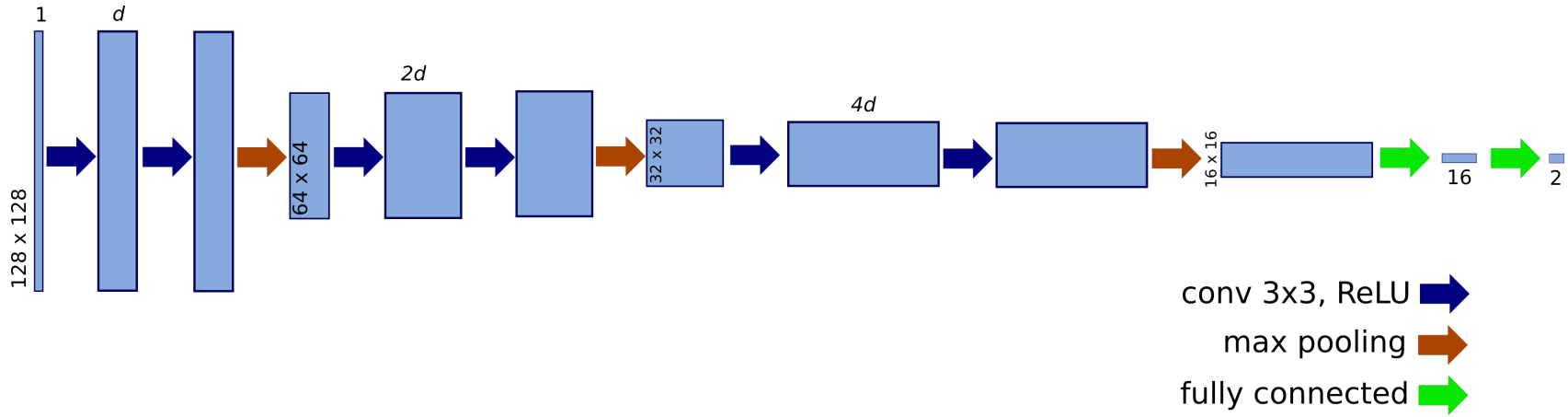
Illustration



Illustration

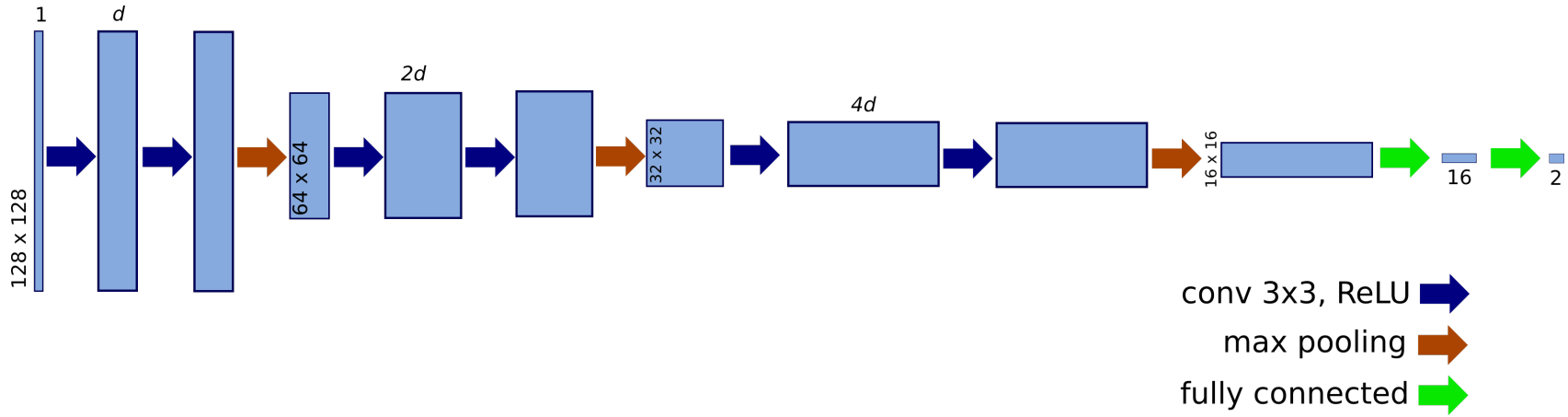


1D representations



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

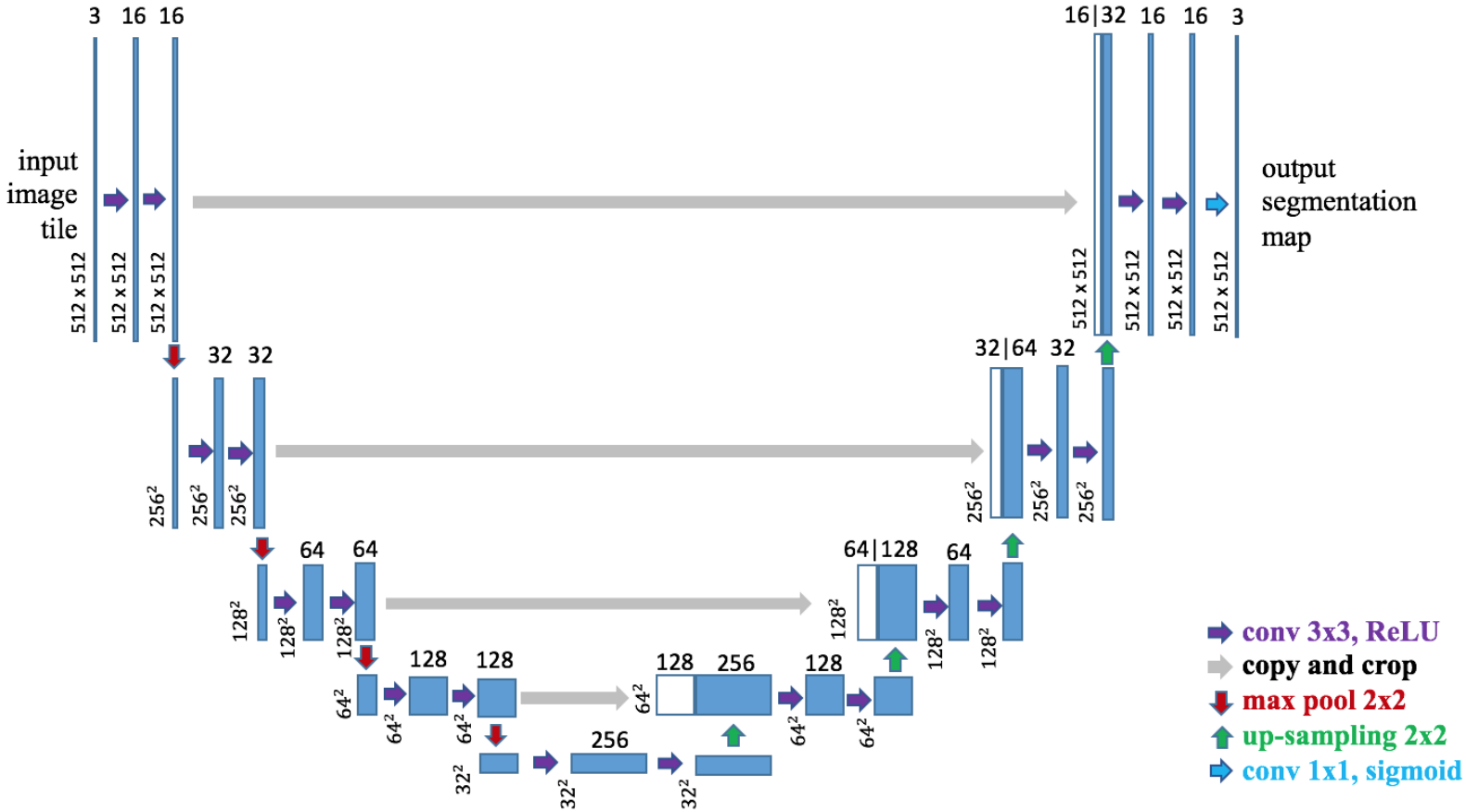
1D representations



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).

U-Net architecture [Ronneberger et al., 2015]



Example: counting cells

image

image_60.png

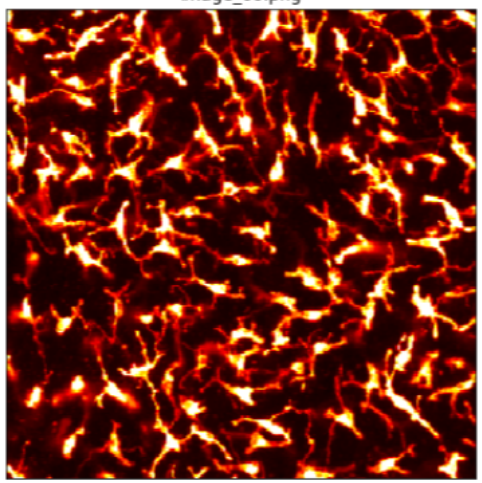
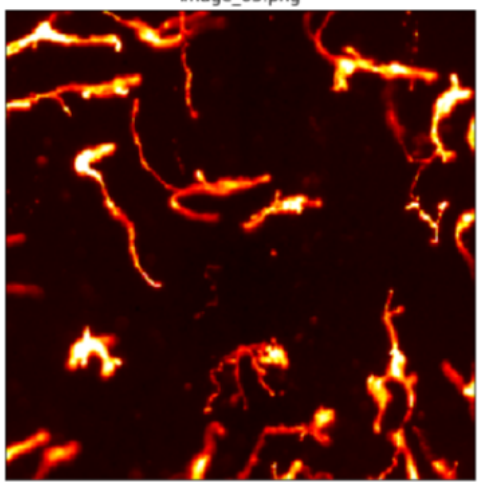
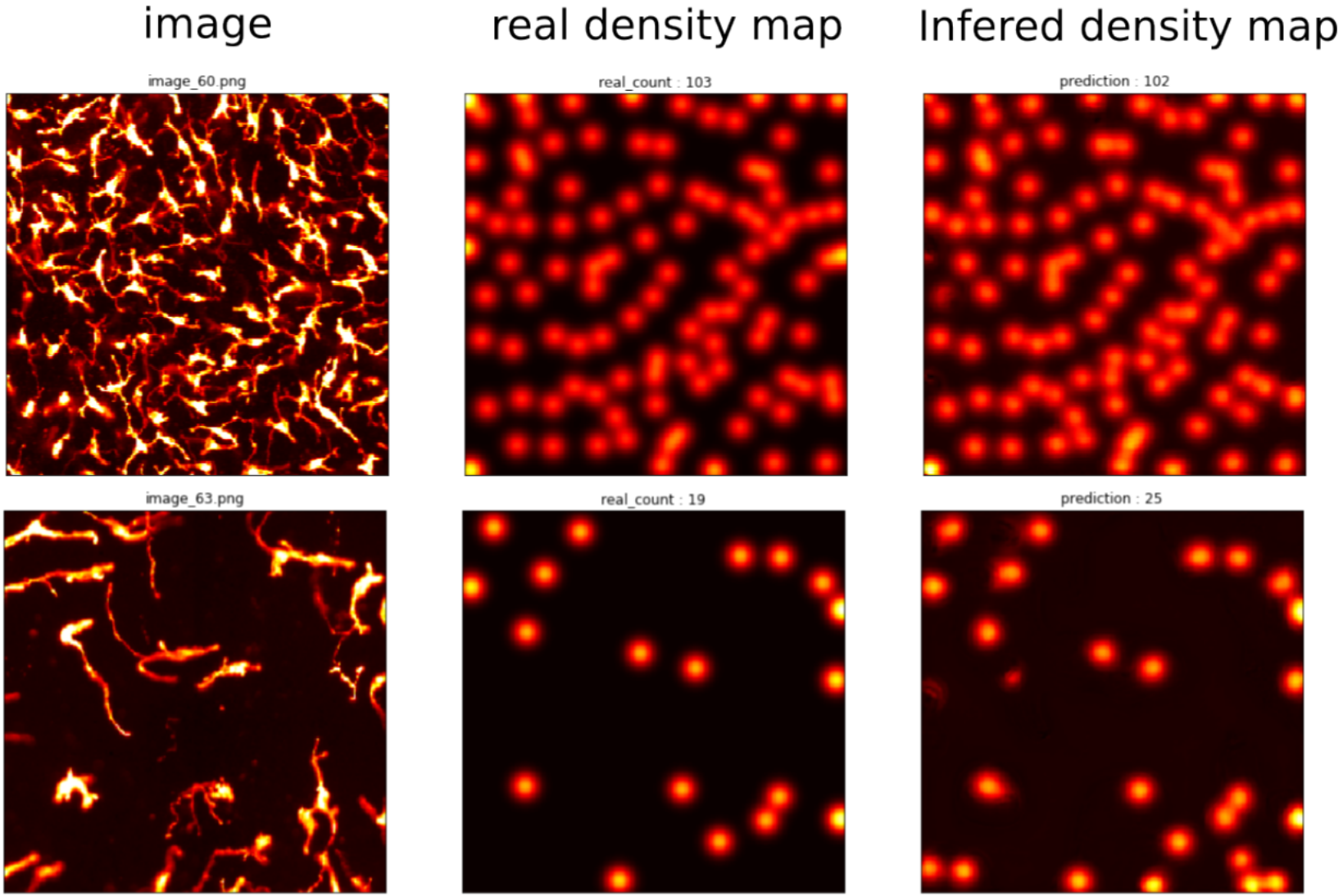


image_63.png



Credits: Tristan Lazard, master thesis. In collaboration with L'Oréal.

Counting cells

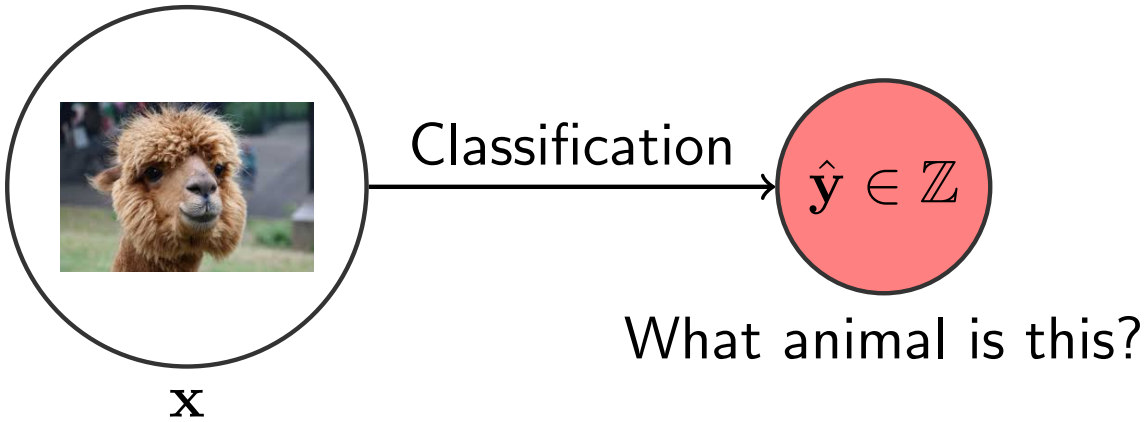


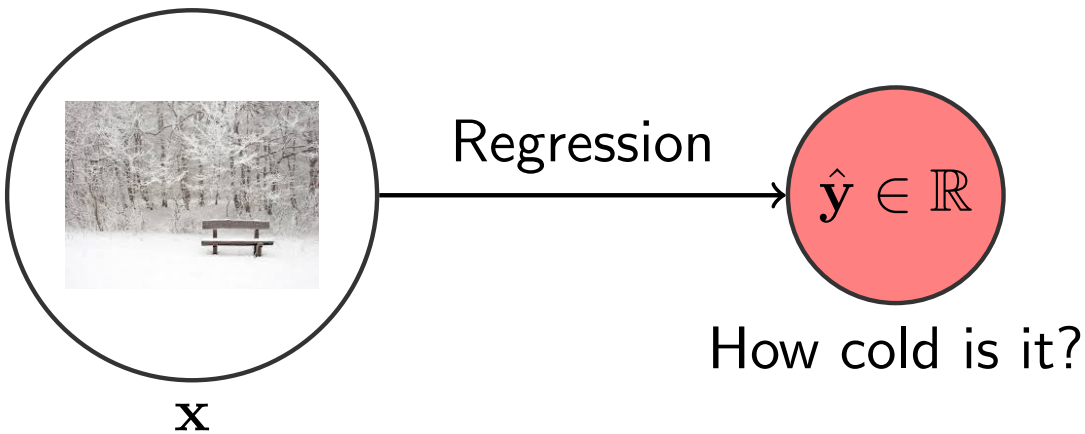
Credits: Tristan Lazard, master thesis. In collaboration with L'Oréal.

Contents

Supervised Learning

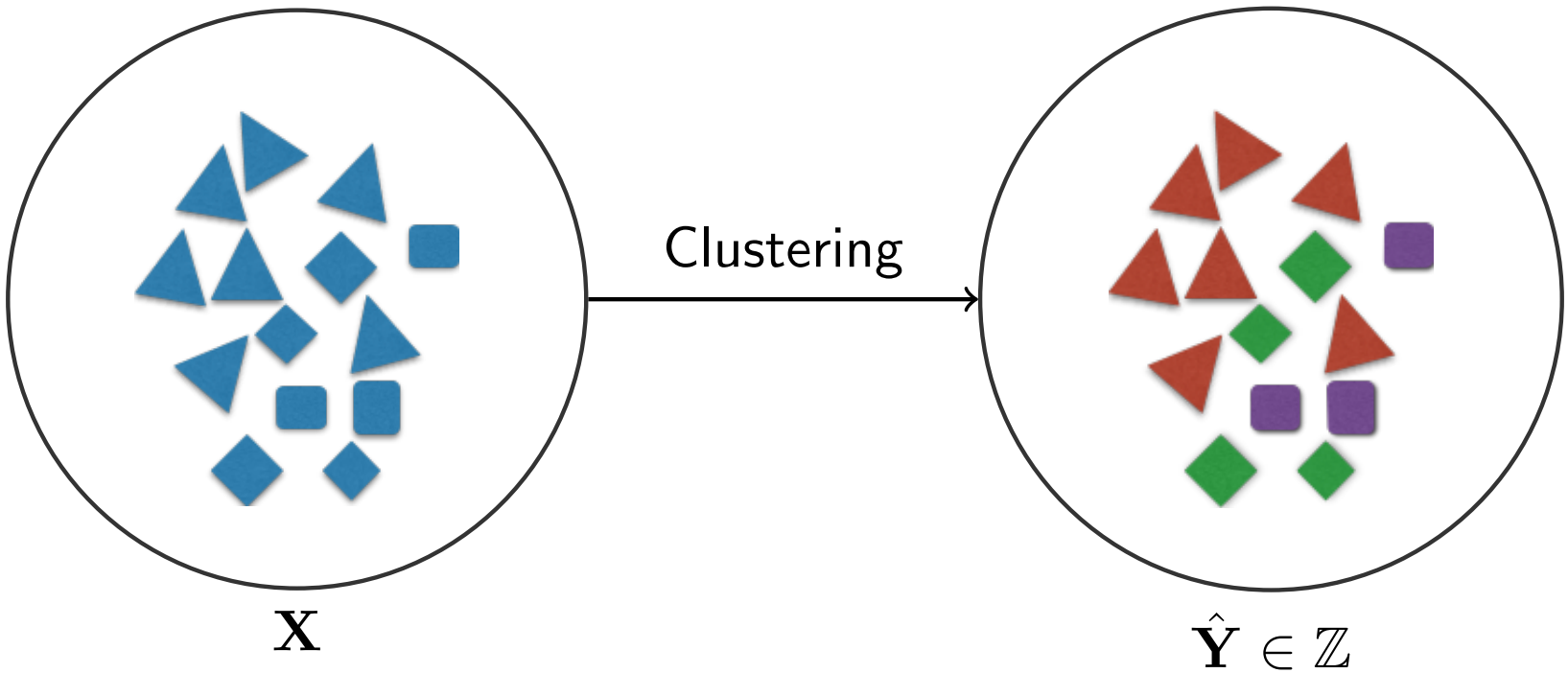
Given a labeled dataset $(\mathbf{X}, \mathbf{Y})$, we would like to learn a mapping from data space to label space.





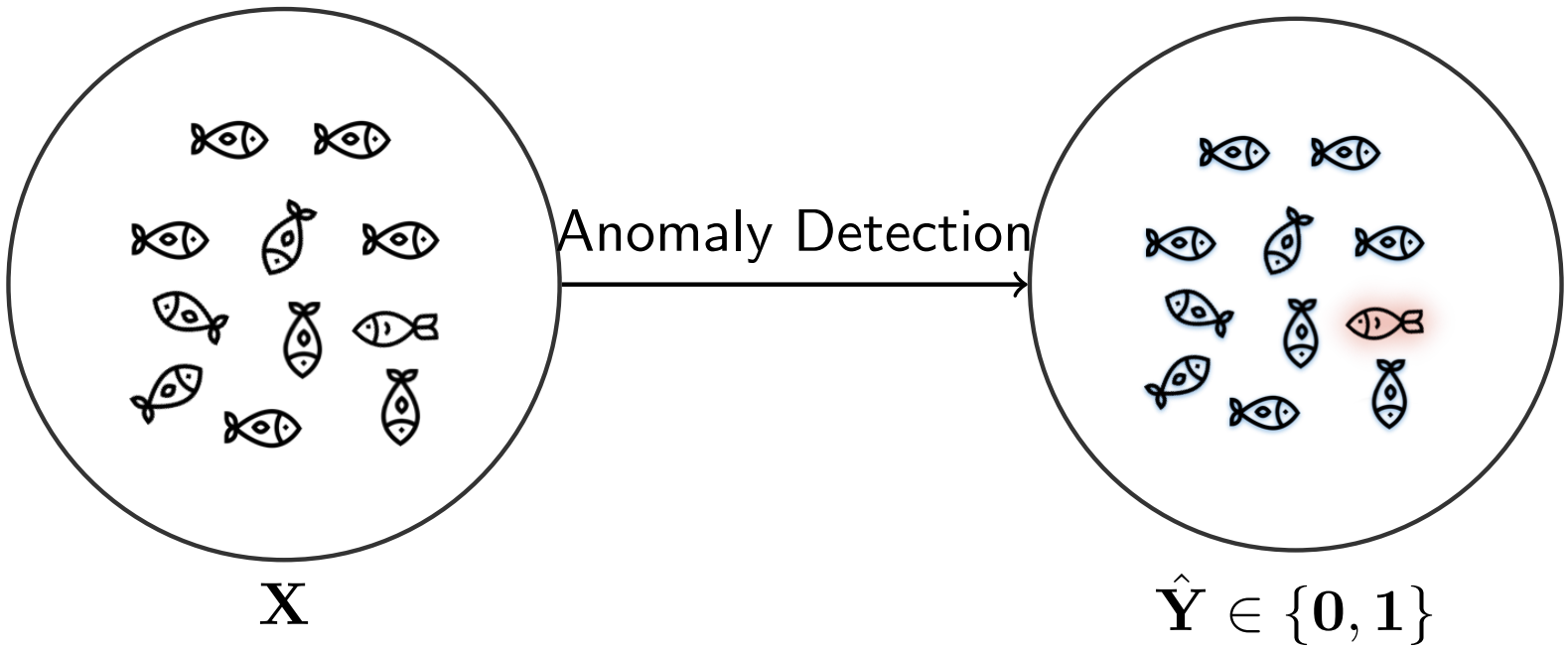
Unsupervised Learning: Clustering

Given an unlabeled dataset $(\mathbf{X})$, we would like to learn: How to group objects into categories?



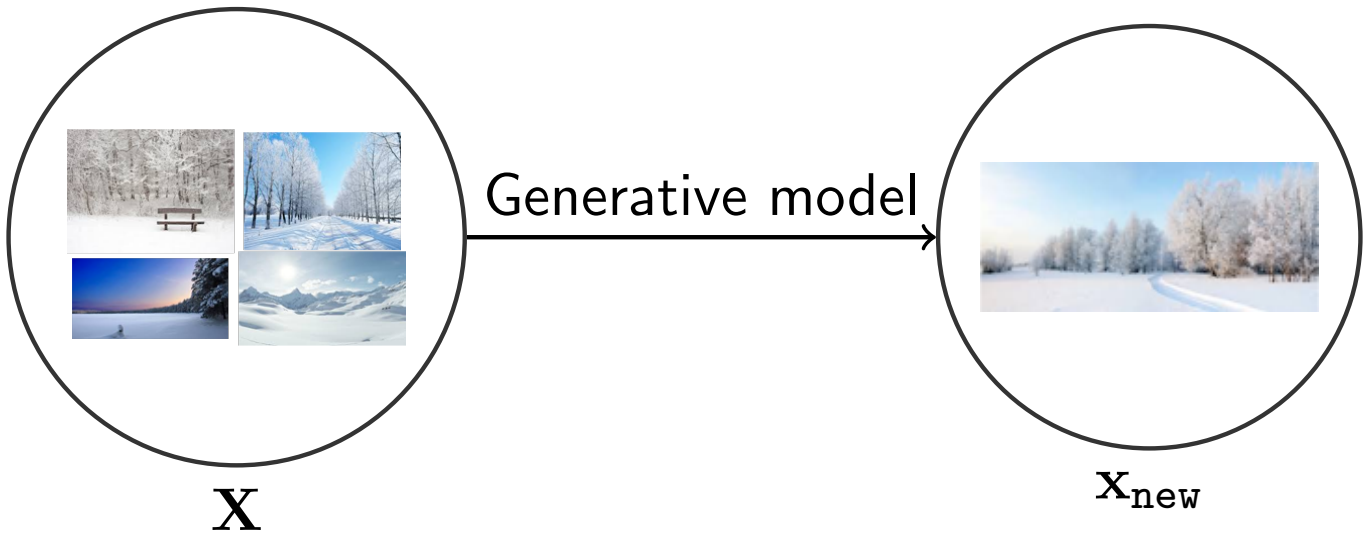
Unsupervised Learning: Anomaly detection

Given an unlabeled dataset $(\mathbf{X})$, we would like to learn: How to identify observations differing significantly from the majority of data?

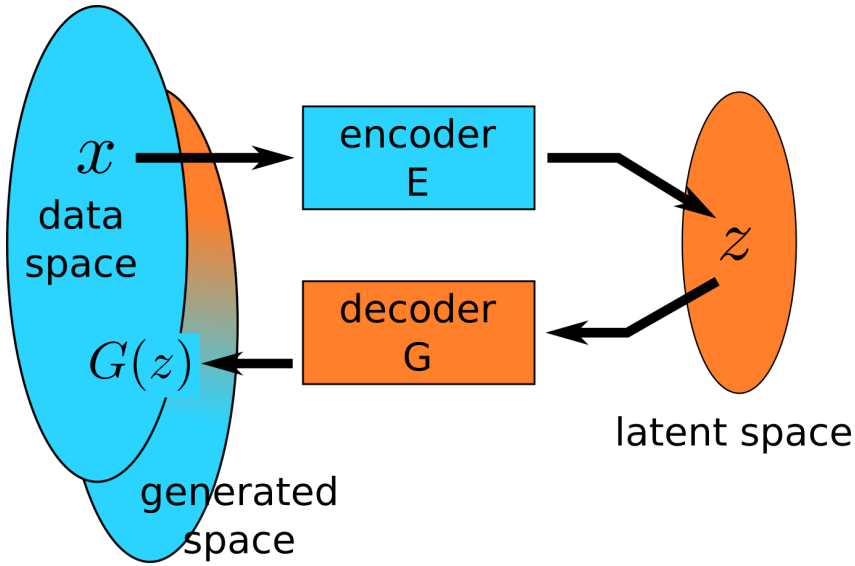


Unsupervised learning: Generative Models

Given an unlabeled dataset $(\mathbf{X})$, we would like to learn: How to generate a new observation from the same distribution (unknown) of dataset?

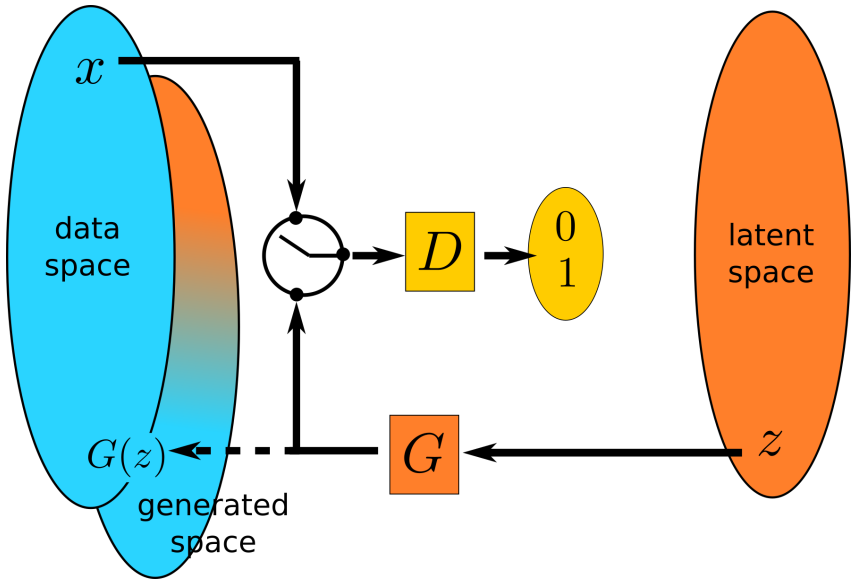


Autoencoders



- Encoder: E; decoder: G; autoencoder: $G \circ E$
- In most applications, the latent space is "smaller" than the data space.
- Objective: $\hat{x}$, i.e. $G \circ E(x)$, "close" to x
- When dealing with images, modern autoencoders use convolutional neural networks

Generative adversarial networks [Goodfellow et al., 2014]



- The discriminator D is optimized so that it correctly classifies images as real (1) or fake (0)
- The decoder or generator G is optimized so that the produced images are classified as real by the discriminator

Value function

$$V(G,D) = \mathbb{E}_{p_{\mathbf{x}}}(log(D(\mathbf{x}))) + \mathbb{E}_{p_z(z)}(log(1 - D(G(z))))$$

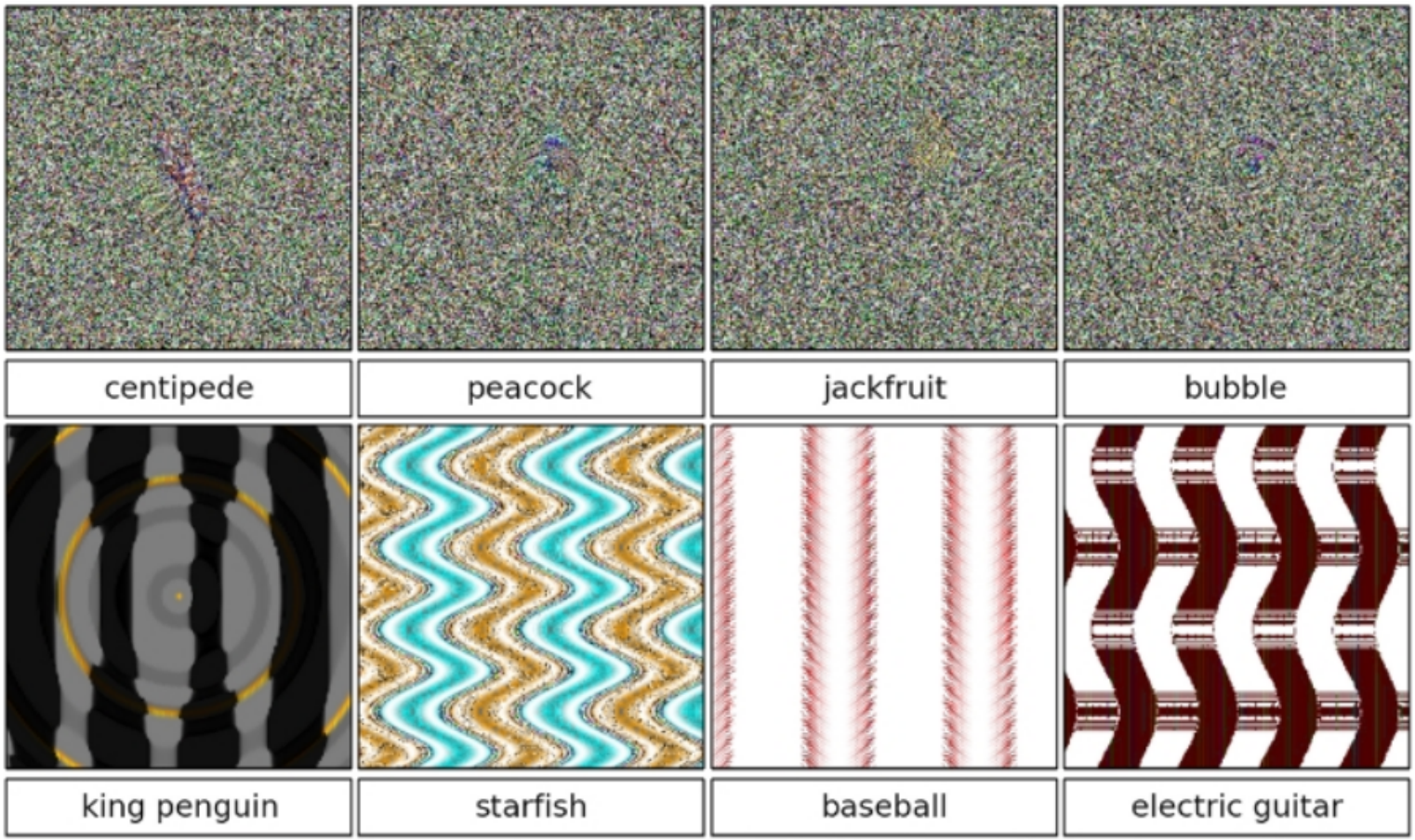
Which face is real?

Contents

Conclusion

- Deep learning allows to learn complex transformations between tensors, thanks to:
  - Smart methods and algorithms.
  - Lots of annotated data.
  - Specialized hardware (for learning).
- Drawbacks:
  - Interpretability problem.
  - Why does deep learning work so well?
  - Deep learning can be easily fooled.
- General artificial intelligence is still far away

ConvNets can be fooled [Nguyen et al., 2015]



References I

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[Cybenko, 1989] Cybenko, G. (1989). Approximations by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:183–192.

[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.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition.

[Hornik, 1991] Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257.

[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 II

[LeCun, 1985] LeCun, Y. (1985). Une procedure d'apprentissage pour reseau a seuil asymmetrique (A learning scheme for asymmetric threshold networks). In *proceedings of Cognitiva 85*.

[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.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, number 9351 in Lecture Notes in Computer Science, pages 234–241. Springer International Publishing.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

References III

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252.

[Werbos, 1982] Werbos, P. J. (1982). Applications of advances in nonlinear sensitivity analysis. In Drenick, R. F. and Kozin, F., editors, *System Modeling and Optimization*, Lecture Notes in Control and Information Sciences, pages 762–770, Berlin, Heidelberg. Springer.