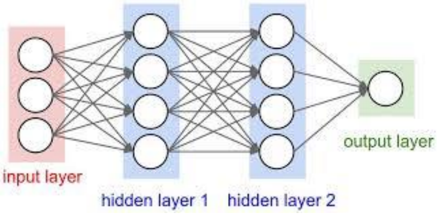
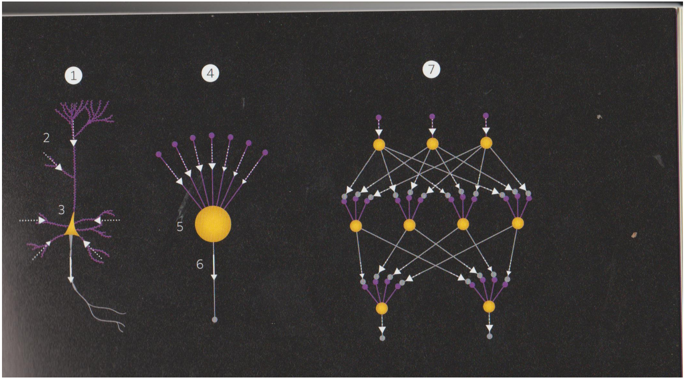
STRUCTURE OF ARTIFICIAL NEURAL NETWORKS



Neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first (input), to the last (output) layer (feed-forward neural networks).
Neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering.

STRUCTURE OF ARTIFICIAL NEURAL NETWORKS



Biological neuron (1)- dendrites (2), cell body (3); Artificial neuron (4)- acumulator (5), connection (6); Artificial neural network (7).

NEURAL NETWORKS AS LEARNING MACHINES

Purpose of a learning machine:
The knowledge acquired from a set of samples can be generalized to arbitrary data.
How to build a learning machine?
Use statistical learning (includes optimization, approximation theory, probability theory,...).

MATHEMATICAL FORMULATION

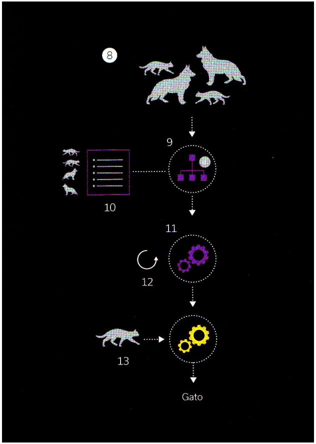
X - input space; Y - output space;
h: - $X \to Y$ (classifier);
Knowing the output of h for a finite set of samples $S \subset X$ (training set), we want to compute its value for every $x \in X$.
The criterium for choosing h is the minimization of a certain risk functional $R(h)$. (probability of getting a wrong answer).
Usually we cannot evaluate the exact risk functional, but only a certain empirical risk functional $R_{emp}(h)$.
Instead $R(h)$ we minimize $R_{emp}(h)$.

NEURAL NETWORKS IN IMAGE PROCESSING

Suppose you want to use an artificial neural network to distinguish images of cats from dogs. In this case, the input set X is the set of all images; the output set Y has only two elements: 0, if the answer is cat; 1, if the answer is dog.

- First you must have a certain set of images (samples) which are recognized as cats or dogs. This is the training set S.
- The neural network is trained by processing these images and comparing the obtained answers with the correct ones.
- Based on this processing the network creates a rule (classifier) to distinguish cats from dogs, which is stored in the artificial neurons.The more samples are used, the more reliable is the rule.
- Each time an arbitrary image enters the network an answer is obtained applying the rule.

NEURAL NETWORKS IN IMAGE PROCESSING



Training process (9-12): Rules are created based on previously classified images. Application (13): the network is tested using new images.

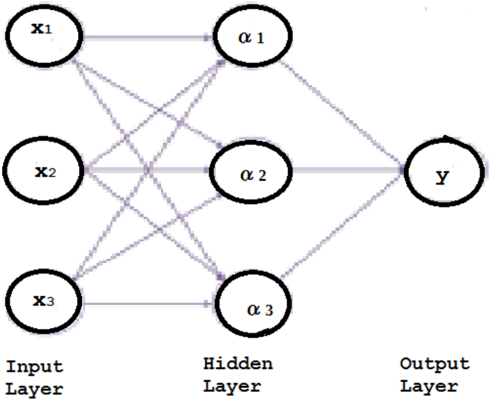
ONE-LAYER PERCEPTRON

The simplest example of learning machine is the one-layer perceptron. In this case, the classifier h has the form

$$h(x) = \begin{cases} 1, & \text{if} \quad (w, x) + b > 0, \\ 0, & \text{otherwise,} \end{cases}$$

where $x \in X$ (input set) and w is a vector of the same dimension as x (weights); $b \in \boldsymbol{R}$(bias); the brackets denote the usual scalar product. The classifier is obtained by computing the weights w_i. This is done during the training of the perceptron.

SCHEME OF A PERCEPTRON



Scheme of a perceptron in the case $n = dim(x) = 3$.

AN APPLICATION TO ECONOMICS

Example

R. Heibrich, M. Keilbach, T. Graepe, P. Bollmann-Sdorra, K. Obermayer, *Neural Networks in Economics*.
http://www.researchgate.net/publication/249900701

Task : evaluate the reliability of bank customers concerning paying back a given loan.

X - set of bank clients; $\mathbf{x} = (x_1, x_2, ..., x_n)$, $\forall x \in X$ (age, sex, income,...)

$Y = \{1, 0\}$ (the answer is 1, if the client is reliable, and 0, if he is not reliable.)

Set of samples (training set): $S = \{(\mathbf{x_t}, y_t)\ \mathbf{x_t} \in X,\ y_t \in Y, t = 1, ..., l\}$

Objective: define $h : X \to Y$ of the type described above, such that h defines the reliability of any customer $x \in X$ (classifier).

HOW TO BUILD THE CLASSIFIER?

Risk functional: $R(h) = \int_{XY} L(y, h(x)) P_{XY} dxdy$,
where L- loss function ,

$$L(y, h(x)) = \begin{cases} 0, \text{ if } y = h(x); \\ 1, \text{ if } y \neq h(x). \end{cases}$$

P_{XY}- probability that a random client is reliable $(y = 1)$ or unreliable $(y = 0)$.
The goal is to minimize R(h).
Empirical risk functional:

$$R_{emp}(h) = \frac{1}{l} \sum_{(\mathbf{x_t}, y_t) \in S} L(y_t, h(\mathbf{x_t})) = \frac{1}{l} \sum_{(\mathbf{x_t}, y_t) \in S} (h(\mathbf{x_t}) - y_t)^2$$

Since we cannot in general evaluate $R(h)$, we replace the minimization of $R(h)$ by the minimization of $R_{emp}(h)$.

HOW TO BUILD THE CLASSIFIER?

h^* - minimizer of R; h_l - minimizer of R_{emp}.
According to the principle of empirical risk minimization as l (cardinal of S) tends to infinity, $|R_{emp}(h) - R(h)| \to 0$, and therefore if l is large enough h_l is a good approximation h^*.
This problem can be solved by different types of neural networks, depending on the form of the classifier h.
We consider here the perceptron.
In this case, we have only one hidden layer. The scheme of the perceptron, as shown above, has 3 layers:

- input layer: $X = (x_1, ..., x_n)$; $\dim(X)=n$.
- hidden layer: $w = (w_1, ..., w_n)$; $\dim(w) = n$.
- output layer: $y \in \{-1, 1\}$;$\dim(Y)=1$ (the output consists only of one number).

HOW TO BUILD THE CLASSIFIER?

Each classifier h is defined by a n-dimensional vector $w = (w_1, ..., w_n)$, $w_i \in \boldsymbol{R}$:

$$h(x) = \begin{cases} 1, & \text{if} \quad (w, x) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

The bias b in this case is 0.

Given a training example $(\mathbf{x_t}, y_t)$ we say that

x_t is classified, if $h(\mathbf{x_t}) = y_t$,

x_t is misclassified, if $h(\mathbf{x_t}) \neq y_t$.

Our goal is that all the samples are classified.

ITERATIVE METHOD

The empirical risk functional $R_{emp}(h)$ is defined as

$$R_{emp}(h) = \frac{1}{l} \sum_{(\mathbf{x_t}, y_t) \in S} (h(\mathbf{x_t}) - y_t)^2.$$

Note: $R_{emp}(h)$ is equal to the percentage of elements of the training set which are misclassified.
The minimization is performed by an iterative process, similar to the gradient method.

ALGORITHM

Pseudo-code of the iterative process:

- Set initial value $w_0 \in \boldsymbol{R}^n$ (randomly chosen);
- While misclassified training examples exist
- compute $h(\mathbf{x_t}, y_t)$;
- If $\mathbf{x_t}$ is misclassified
- then $w_{t+1} = w_t + r(h(\mathbf{x_t}) - y_t)x_t$;
- End If
- End While

Notes:

- r is a real constant, called the learning rate.
- Depending on the properties of the training set, it may happen that the number of misclassified elements never reaches 0; in this case the stopping criterion should be 'while $R_{emp}(h) > \epsilon$,' where ϵ is a given tolerance.

OTHER TYPES OF NEURAL NETWORKS

Besides one-layer perceptron, different kinds of neural networks could be applied.

Multilayer Perceptron

In this case, the classifier has the form:

$$h(\mathbf{x}, \beta, \gamma) = f_2(f_1(\mathbf{x}, \beta), \gamma),$$

where

$$f_1(\mathbf{x}, \beta) = (g_1(\langle \mathbf{x}, \beta_1 \rangle), ..., (g_1(\langle \mathbf{x}, \beta_r \rangle)), \qquad f_2(\mathbf{z}, \gamma) = g_2(\langle \mathbf{z}, \gamma \rangle);$$

here g_1 and g_2 are sigmoidal functions; $\beta_1, ...\beta_r, \gamma$ are vectors that should be optimized.

NEURAL NETWORKS IN ECONOMICS

Radial Basis Functions

$$h(\mathbf{x}, \beta, \gamma, \sigma) = f_2(f_1(\mathbf{x}, \beta, \sigma), \gamma),$$

where

$$f_1(\mathbf{x}, \beta, \sigma) = (g_1(\mathbf{x}, \beta_1, \sigma_1), ..., (g_1(\mathbf{x}, \beta_r, \sigma_r)), \qquad f_2(\mathbf{z}, \gamma) = g_2(\langle \mathbf{z}, \gamma \rangle);$$

here $g_1(x, \beta, \sigma) = \exp\left(-\frac{\|x-\beta\|^2}{2\sigma^2}\right)$; g_2 is a sigmoidal function.
$\beta_1, ...\beta_r, \sigma_1, ..., \sigma_r, \gamma$ are vectors that should be optimized.

In all the different types of neural networks we have to minimize the emipirical risk function using an iterative procedure.

BIBLIOGRAPHY

1. Mark F. Bear, Berry W. Connors, and Michael A. Paradiso (Eds.), *Neuroscience: exploring the brain*, Philadelphia, PA, USA, 2016.
2. Bishop, Christopher, Pattern Recognition and Machine Learning, New York, Springer Verlag, 2006.
3. *Cerebro*, ed. by Fundacao Calouste Gulbenkian, Lisbon, 2019.
4. De Felipe, Javier, and Larry Swanson, *The Beautiful Brain: the Drawings of Ramon y Cajal*, New York, Abbrams and Chronicle Book, 2017.
5. Greenstein, B., and Greenstein, A., *Color Atlas of Neuroscience, Neuroanathomy and Neurophysiology*, Stutgart, George Thieme Verlag, 2000. 2006/08/29/the-discovery-of-the-neuron.
6. R. Heibrich, M. Keilbach, T. Graepe, P. Bollmann-Sdorra, K. Obermayer, *Neural Networks in Economics*, http://www.researchgate.net/publication/249900701.

BIBLIOGRAPHY

7 Mo Constandi, *The Discovery of Neuron*, https://neurophilosophy.wordpress.com/,2006.

8 Netter, F., Craig, J., Perkins, J., Hansen, T., and Koeppen, B., *Atlas Neuroanathomy and Neurophysiology*, Ebook: Icon Costum Communication, 2002.

9 Nieuwenhuys,R., Donkelaar, and Nicholson, C., The Central Nervous System of Vertebrates, Berlin, Spriger,1998.