Uniwersytet Ekonomiczny we Wrocławiu

MINING OF FINANCIAL DATABASES

5. Neural networks - MLP

Jerzy KORCZAK
email :jerzy.korczak@ue.wroc.pl
http://www.korczak-leliwa.pl
http://citi.ae.wroc.pl

History

- Neuron model [McCulloch, Pitts, 1943]
- Learning process [Hebb, 1949]
- PERCEPTRON [Rosenblatt,1958]
  - **Convergence of algorithme of weight adaptation**
- Limitations of PERCEPTRON
  - **erronously claimed by Minsky, Papert, 1969]**
  - **Problem XOR**
- Bolzmann Machine [Hopfield, 1982]
- Back propagation - MLP [Rumelhart, Parker, Le Cun, 1985]
- Self-Adapting Maps [Kohonen, 1980]
- Hopfield networks [Hopfield, 1982]
- … ICANN, IJCNN, ECANN, …

History (2)

Diversification during the 90's:

- Machine learning: mathematical rigor, Bayesian methods, infomation theory, ***support vector machines (now state of the art!)***, ...
- Computational neurosciences: workings of most subsystems of the brain are understood at some level; research ranges from low-level compartmental models of individual neurons to large-scale brain models

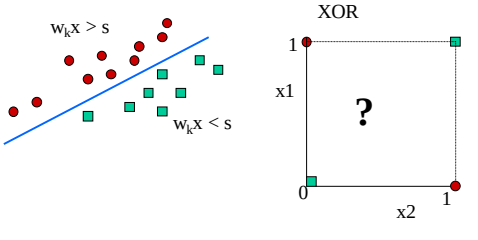
What are neural networks?

- "... systems that are deliberately constructed to make use of some of the organizational principles that are felt to be used in the human brain." — *Anderson*
- "... a neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes." — *DARPA Neural Network Study (1988)*
- "A neural network is an interconnected assembly of simple processing elements, nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns." — *Gurney*

Motivations

- Humans have always wanted to build intelligent machines — stories of golems, automata, mechanical men, and robots
- Computer technology is rapidly approaching the processing power of human brains but organization and function are still huge unknowns
- Biological brains use neurons that are 10^5 or 10^6 times slower than silicon logic gates, yet perform computations better and faster than our best computational algorithms
- (Artificial) Neural Networks are an attempt to harness the massively parallel, distributed computation of biological brains for a variety of purposes

Linearly separable classes

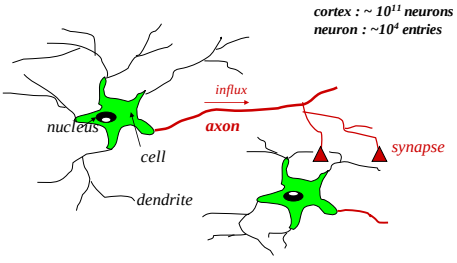


Neural networks: goal and design

- Knowledge about the learning task is given in the form of a set of examples (dataset) called training examples.
- A neural network is specified by:
  - an architecture: a set of neurons and links connecting neurons. Each link has a weight,
  - a neuron model: the information processing unit of the NN,
  - a learning algorithm: used for training the NN by modifying the weights in order to solve the particular learning task correctly on the training examples.

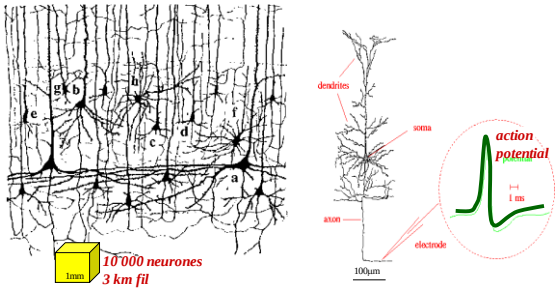
The aim is to obtain a NN that generalizes well, that is, that behaves correctly on new examples of the learning task.

Siimplified schema of biological neural networks

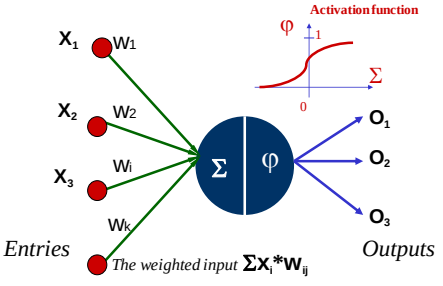


Biological neuron

Signal: action (ang.spike)



Basic unit of neural networks: an artificial neuron

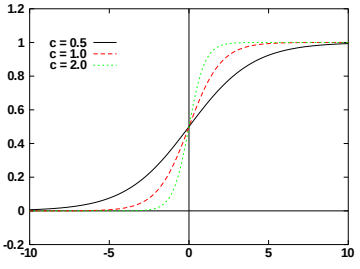


Continuous asymmetric transfer

$\varphi(z) = 1 / (1 + e^{-c \times z})$

sigmoid function

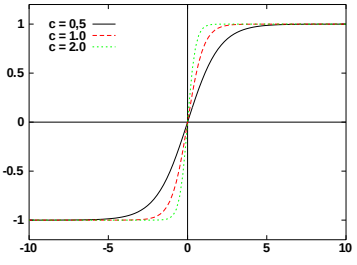
logistic function



Continuous symmetric transfer

$\varphi(z) = (e^{c \times z} - e^{-c \times z}) / (e^{c \times z} + e^{-c \times z})$

tangens hyperbolicus



Dimensions of a Neural Network

- network architectures
- types of neurons
- learning algorithms
- applications

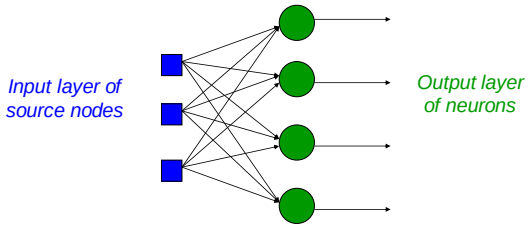
Network architectures

- Three different classes of network architectures:
  - single-layer feed-forward
  - multi-layer feed-forward
  - recurrent

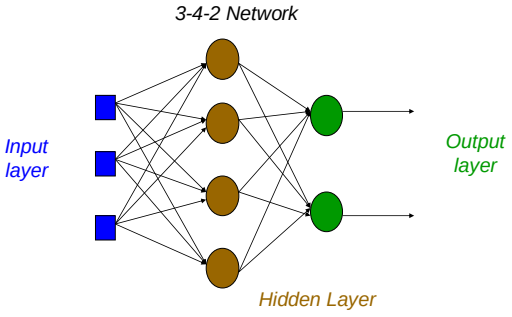
  } *neurons are organized in acyclic layers*

- The architecture of a neural network is linked with the learning algorithm used to train

Single Layer Feed-forward

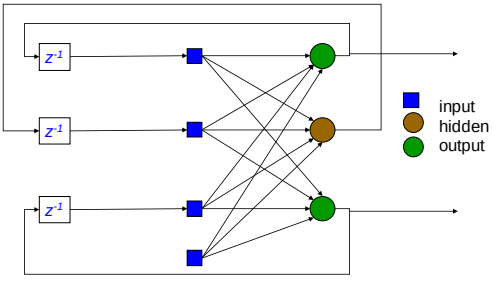


Multi-layer feed-forward



Recurrent network

Recurrent Network with ***hidden neuron***: unit delay operator z^{-1} is used to model a dynamic system



Input Signal and Weights

Input signals

An input may be either a raw / preprocessed signal or image. Alternatively, some specific features can also be used.

If specific features are used as input, their number and selection is crucial and application dependent

Weights

Weights are connected between an input and a summing node. These affect to the summing operation.

The quality of network can be seen from weights

Bias is a constant input with certain weight.

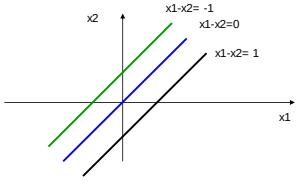
Usually the weights are randomized in the beginning

Bias of a Neuron

- The bias ***b*** has the effect of applying an affine transformation to the weighted sum ***u***

  v = u + b

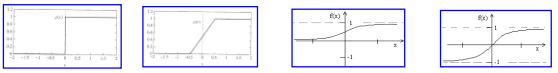
- ***v*** is called **induced field** of the neuron



$u = x1 - x2$

Activation Function φ

There are different activation functions used in different applications. The most common ones are:



Hard-limiter | Piecewise linear | Sigmoid | Hyperbolic tangent

$$\varphi(v)=\begin{cases}1 & \text{if } v \geq 0\\ 0 & \text{if } v<0\end{cases} \qquad \varphi(v)=\begin{cases}1 & \text{if } v \geq 1/2\\ v & \text{if } 1/2 \geq v \geq -1/2\\ 0 & \text{if } v \leq -1/2\end{cases} \qquad \varphi(v)=\frac{1}{1+\exp(-av)} \qquad \varphi(v)=\tanh(v)$$

Neuron Models

- The choice of φ determines the neuron model. Examples:
- step function:

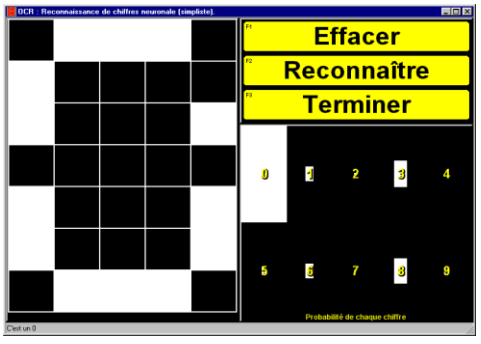
$$\varphi(v)=\begin{cases}a & \text{if } v<c\\ b & \text{if } v>c\end{cases}$$

- ramp function:

$$\varphi(v)=\begin{cases}a & \text{if } v<c\\ b & \text{if } v>d\\ a+((v-c)(b-a)/(d-c)) & \text{otherwise}\end{cases}$$

- sigmoid function:
  with z,x,y parameters $\varphi(v)=z+\frac{1}{1+\exp(-xv+y)}$
- Gaussian function: $\varphi(v)=\frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{1}{2}\left(\frac{v-\mu}{\sigma}\right)^2\right)$

OCR : Character recognition by NN

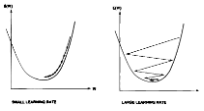


Applications

- Classification:
  - Image recognition
  - Speech recognition
  - Diagnostic
  - Fraud detection
  - …
- Regression:
  - Forecasting (prediction on base of past history)
  - …
- Pattern association:
  - Retrieve an image from corrupted one
  - …
- Clustering:
  - clients profiles
  - disease subtypes
  - …

MLP : Learning algorithm

```
Weight initialisation
Repeat
      for each training pattern
          train on the pattern
            Forward pass
            Backward pass
      end for loop
Until the error is 'acceptably low'
```



Defaults :

- Parameter setting
- Long process of learning
- Choice of network topology

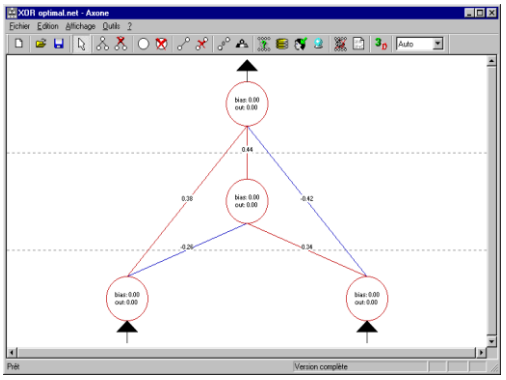
Learning Algorithms

Depend on the network architecture:

- Error correcting learning (perceptron)
- Delta rule (AdaLine, Backprop)
- Competitive Learning (Self Organizing Maps)

Axone : Example XOR

[http://citi.ae.wroc.pl]



Characteristics of 'good' domain of applications

- Problems that are difficult to define explicitly
- Availability of large amount of data
- Noisy data
- Problems that need a fast computing
- No existing algorithmic solutions

Business Applications

Classification, customer research
Forecasting and predicting (stock markets, bankruptcy)
Risk management
Data simplification, data validation, compression
Pattern recognition, biometric authentication
Credit evaluation (Credit Scoring system)

New applications

- Information Retrieval (Web)
- Data Mining
- Multimedia (indexation)

Multi-Layer Perceptron (MLP)

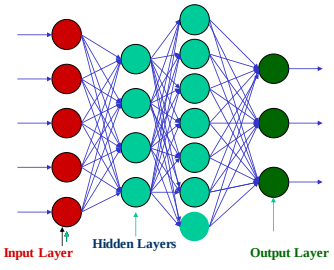
MLP is composed of succesive layers: ***input layer*** (where the data are presented), one or more ***hidden layers***, and ***output layer*** (where the computed outputs are presented).

Learning algorithms of MLP :

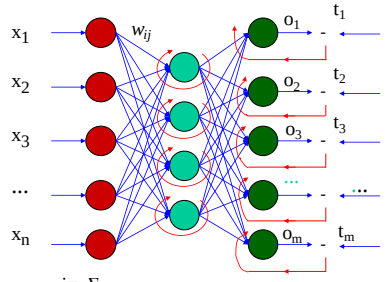
- Gradient Back-propagation, SuperSAB
- Cascade Correlation
- Gonjugate Gradient methods
- Second order methods, …

MLP universal approximators

Multi Layer Perceptron (MLP)



Gradient Backpropagation



$E=1/2\ \Sigma(t_k - o_k)^2$

$yi = \Sigma w_{ij} x_i$

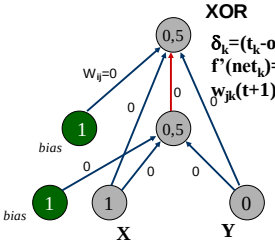
Sigmoid function $\varphi(y)=1/(1+e^{-ky})$ $\varphi'(y) = \varphi(y)(1-\varphi(y))$

Example : Gradient Back-Propagation (GBP) Formulas

XOR

X	Y	XOR(X,Y)
0	0	0
0	1	1
1	0	1
1	1	0

$f(net_k)=1/(1+e^{-net_k})$
$net_j=\Sigma w_{ij} o_i$
$o_j=f(net_j)$

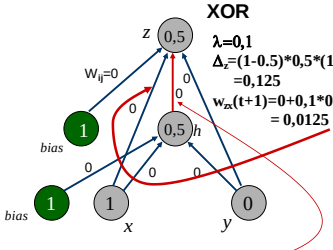


$\delta_k=(t_k-o_k)f'(net_k)$
$f'(net_k)=o_k(1-o_k)$
$w_{jk}(t+1)=w_{jk}(t)+\lambda\Delta_k o_j$

Example : Gradient Back-Propagation (GBP) Learning

XOR

X	Y	XOR(X,Y)
0	0	0
0	1	1
1	0	1
1	1	0



$\lambda=0,1$
$\Delta_z=(1-0.5)*0,5*(1-0.5)=$
$=0,125$
$w_{zx}(t+1)=0+0,1*0,125*1=$
$= 0,0125$

$\Delta_h=f'(net_j)\ \Sigma\Delta_k w_{kj}$
$= 0,5*(1-0,5)*0,125*0,00625= 0,000195$
$w_{hx}=0+0,1*0,000195*1=0,0000195$

Learning: the weights and learning step

$w_{ij}(t+1)= w_{ij}(t) + \lambda * a_j * \Delta w_i$

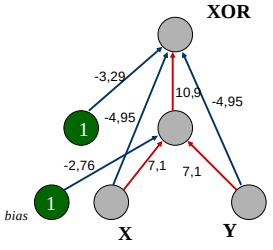
w_{zx}=0,00125
w_{zy}=0
w_{zh}=0,00625
w_{hx}=0,0000195
w_{hy}=0
w_{zbh}=0,0000195
f(net) = 0,507031

λ	iteration
0,1	25496
0,5	3172
3,0	391
4,0	(fails)

Example : Trained MLP

XOR

X	Y	XOR(X,Y)
0	0	0
0	1	1
1	0	1
1	1	0

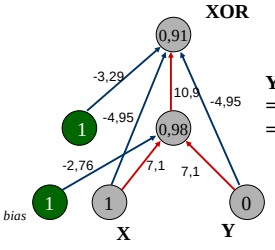


Example : MLP propagation (test)

XOR

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0

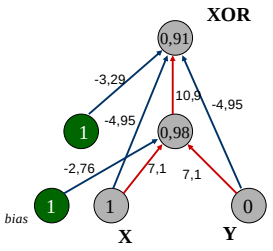
1 * 7,1
1 * -2,76
0 * 7,1
Σxw = 4,34



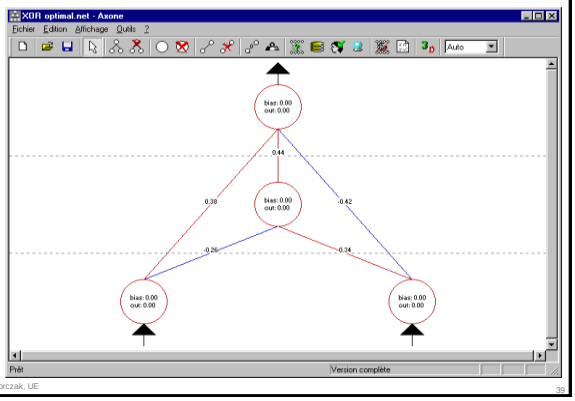
$Y=1/(1+e^{-\Sigma xw})$
$=1/(1+e^{-4,34})$
$=0,98$

Exemple: XOR testing

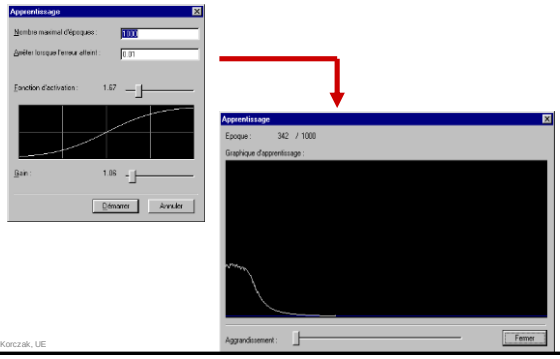
XOR		
X	Y	XOR
0	0	0,08
0	1	0,91
1	0	1,00
1	1	0,10



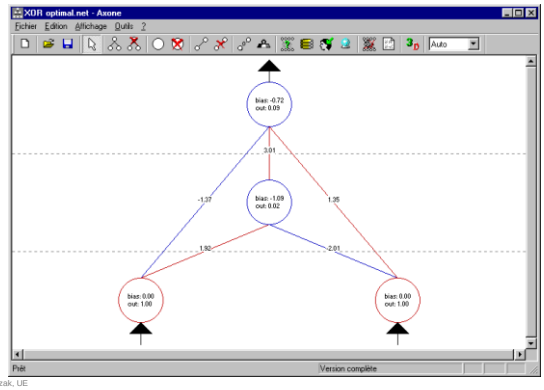
Axone : Example XOR



Axone : Example XOR – learning parameters



Axone : XOR network



Learning : Weight updating and tools

Batch: all possible input signals are evaluated first before any adaptations are done.

On-line: perform an update directly after each input signal

Constant adaptation and decreasing adaptation

Convergence problem

Tools and neural network software: Matlab, Axone, SNNS

Choice of learning step

- Learning step:
  - too small -> 'long' convergence
  - too large -> risk of oscillations
- Heuristics :
  - adjust the step if necessary
    - «manually»
    - according to the shape of error surface
- Approximation:
  - First order : Inertie, SuperSAB, Delta-Bar-Delta, Rprop
  - Second order : QuickProp, Levenberg-Marquard

Training, validation and test sets

- *Training set* is a set of examples used for learning, that is to fit the weights of the classifier.
- *Validation set* is a set of examples to tune the parameters (i.e.architecture) of a classifier.
- *Test set* is a set of examples used only to assess the performance of a classifier.

Estimating of generalization error

- *K-fold cross validation*: training the net k times on k subsets leaving out one for testing.
- *Leave-one-out* cross-validation
- *Bootstrapping*: instead of repeatedly analyzing subsets of the data, you repeatedly analyze subsamples of the data randomly selected with replacement from the full sample.

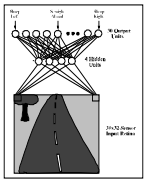
Example: Alvinn

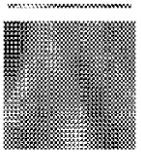
Camera image

Autonomous driving at 70 mph on a public highway

30 outputs for steering
4 hidden units

30x32 pixels as inputs

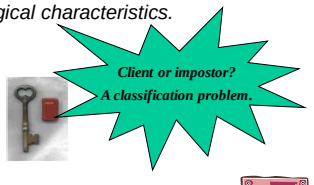


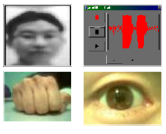


30x32 weights into one out of four hidden unit

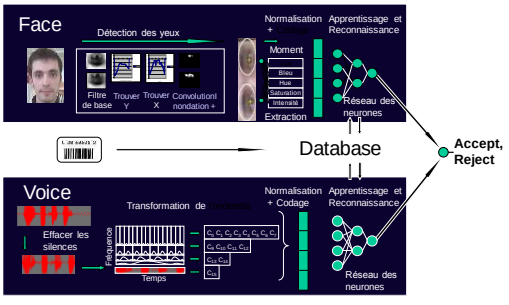
What is Biometric Authentication?

A process of verifying an identity claim using a person's behavioral and physiological characteristics.

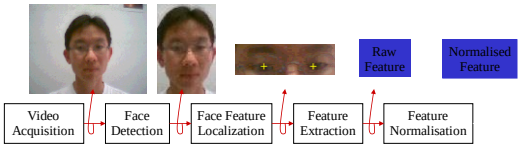




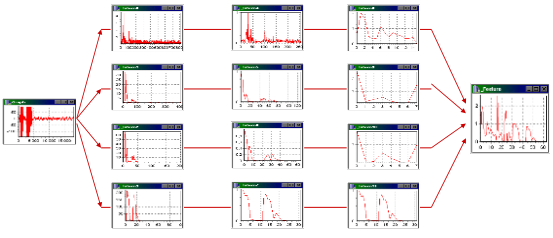
Biometric Authentication



Face Module: Feature Extraction



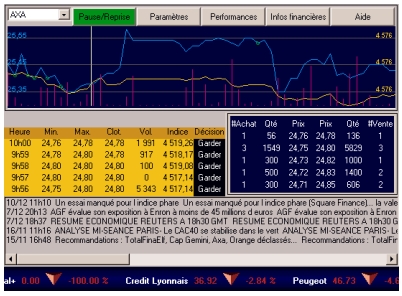
Voice Module: Feature Extraction



Wavelet Transform | Truncate | Sample | Merge

Financial Times Series

- data frequency (every day or every minute),
  - open, high, low, close prices, volume
  - return rate



Learning

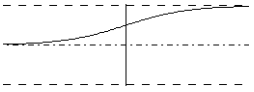
Desired output : $p = f(\text{return rate})$

Error : $(p - y)^2$

General result: $(y - 0.5)(p - 0.5) >= 0$

Back-propagation learning

$\varphi(x) = 1 / (1 + \exp(- B x))$



Testing

After learning on **a training period**, generate signals on **a test period** *(usually equals 1 day or 1 minute)*

Compare these signals with real data

Repeat this process several times on various periods

Results

Interval: 1 day (. – 5 – 1)

# Experiments	Learn Period	Test Period	Accepted
10	5	1	7
10	10	1	8

Interval: 1 minute (. – 10 – 1)

# Experiments	Learn Period	Test Period	Accepted
10	5	1	6
10	10	1	7

Optimization of MLP topology

- Approach by 'try-and-error'
- Ascending approaches: …adding connections and neurons
  - Cascade-Correlation [Fahlman, Lebiere, 1991]
  - Upstart [Frean, 1990]
  - Tiling [Mézard, Nadal, 1989]
- Descending approches: …pruning of connections and neurons
  - during the learning process [Weight Elimination, Weigend,1991]
  - after the learning process [OBD, Brain Surgeon, Le Cun, 1990]
- Evolutionary approaches: genetic connectionism
  - AGWin, Axone [Korczak, 1998]

Knowledge extraction from neural networks

Neural network knowledge is encoded in:

- Network topology
- Activation function
- Connection weights

Objectives :

- Explication of neural network computing
- Software verification
- Software debugging
- Data exploration
- Genaralization improvement
- Induction of scientific theories
- Knowledge acquisition

Bibliography

Bishop C.M., Neural Networks for Pattern Recognition, Oxford Univ., 1995.

Gupta J., Smith K., Neural Network in Business: Techniques and Applications, Idea GR. Pub., 2002.

Haykin S., *Neural Networks: A Comprehensive Foundation*, Prentice, 1999.

Rojas R., *Neural Networks: A Systematic Introduction*, Springer, 1996.

Kohonen T., *Self-Organizing Maps*, Springer, 1997.

Masters T., Practical Neural Network Recipes in C++, Academic Press, 1994.