М. Ю. Медведев
А. Е. Кульченко

Neural networks fundamentals in mobile robot control systems

учебное пособие

ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
ЮФУ
РОСТОВ-НА-ДОНУ · ТАГАНРОГ

УДК 004.032.26:004.896(075.8)
ББК 32.97я73
М46

*Печатается по решению кафедры электротехники и мехатроники
Института радиотехнических систем и управления
Южного федерального университета
(протокол № 5 от 17 марта 2020 г.)*

Рецензенты:
ведущий программист ООО «Люксофт Профешнл»
(филиал в г. Санкт-Петербурге), кандидат технических наук
В. А. Крухмалев
профессор кафедры систем автоматического управления
ИРТСУ ЮФУ, доктор технических наук, профессор *А. Р. Гайдук*

Медведев, М. Ю.

М46 Neural networks fundamentals in mobile robot control systems : учебное пособие / М. Ю. Медведев, А. Е. Кульченко ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2020. – 144 с.

ISBN 978-5-9275-3587-3

Включает полное и систематизированное изложение материала по учебной программе курса «Интеллектуальные системы управления роботами». Адресован студентам, обучающимся по программам бакалавриата и магистратуры по специальности «Мехатроника и робототехника» Института радиотехники и систем управления Южного федерального университета. Включает темы, посвященные введению в нейронные сети, их применению, основам обучения нейронных сетей, многослойным нейронным сетям с прямой связью, передовым методам обучения нейронных сетей и варианты индивидуальных упражнений.

УДК 004.032.26:004.896(075.8)
ББК 32.97я73

ISBN 978-5-9275-3587-3

CONTENT

1. LECTURE
INTRODUCTION TO NEURAL NETWORKS

1.1. Application of artificial intelligence in robotics

Nowadays, the automatic control of unmanned mobile objects is more efficient than the remote control. Autopilot is more accurate and faster to make decisions than the driver. Autopilot makes no mistakes (Fig 1.1) [1–3].

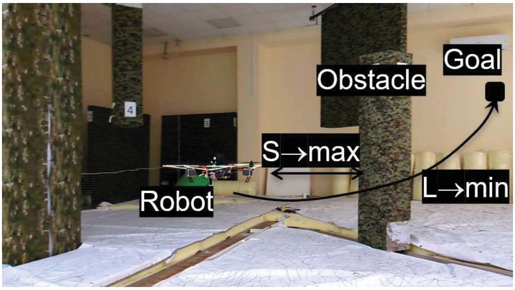
Operation	Remote control accidents	Automatic control accidents
UAV Landing	N	0.1N
Vehicle driving	N	0.6N

Fig 1.1. Autopilots vs remote control

Therefore, the mainstream of robotics is to increase the number of autonomously performed operations.

However, there are **hard problems** for an automatic control (Fig 1.1–1.2). They are driving a traffic, estimation, criteria determination, making a decision under uncertain environments, etc. Characteristic of the hard problems are a dynamical environment, uncertainties, singularities, conflicting criteria, large number of solutions, incorrect statement [4–7].

dynamical environment



conflicting criteria

Fig. 1.2. Hard problems

Intelligent systems are used to solve **hard problems**.

In robotics, an intelligent system is a system that solves the problem of goal-setting, planning and control in a dynamical uncertain environment without the participation of the operator.

1.2. Structure of an intelligent control system of robot

Artificial intelligence solves the following problems (Fig 1.3).

1. The global planning problems are criteria determination, global path planning.
2. The local planning problems are obstacle avoiding, uncertainty of the environment elimination, prediction of environment.
3. The motion control problems are adaptation to model of robot, adaptation to disturbances.

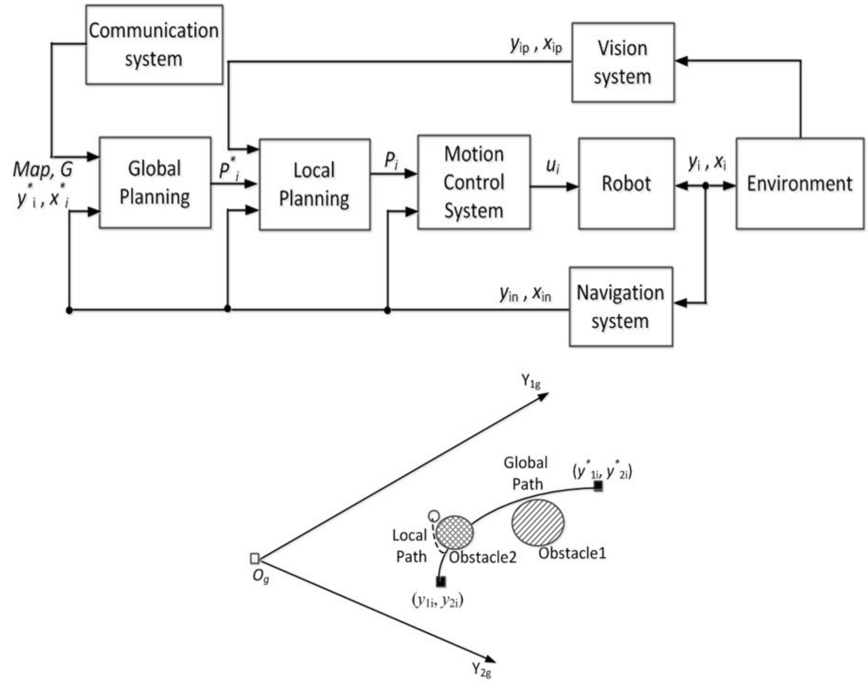


Fig. 1.3. Structure of an intelligent control system of robot

1.3. The artificial intelligence technologies taxonomy

The artificial intelligence approaches are based on different technologies.

1. An expert knowledge base emulates the experience of the experts in the subject area. Examples are diagnostic systems in medicine and engineering. Advantage of an expert knowledge base is unlimited numbers of an expert. Disadvantage is difficulty of the expert coping. Expert knowledge bases are used as recommendation systems.

2. Fuzzy logic emulates a human cognitive function (learning, thinking, reasoning, and adaptation) by uncertain sets. Fuzzy logic uses uncertain notations like “big velocity”, “low temperature”, etc. The main trait of fuzzy logic is uncertain boundaries between fuzzy sets. Fuzzy logic is applied for a decision-making problem.

3. Evolutionary algorithms emulates the mechanisms of natural selection. The most known evolutionary algorithm is a genetic algorithm. Evolutionary algorithms are applied for a global search of the optimal solutions instead of the brute-force searching.

4. Bio-inspired algorithms emulates individual and cooperative behavior of living nature. Well known bio-inspired algorithms are ant algorithms, pigeon algorithms, and bee algorithms.

5. Artificial Neural Network is a mathematical model of a human brain. Neural networks are used as learning systems.

1.4. Morphology of a biological neuron

The intellectual functions of the brain make humans adaptive for handling complex, uncertain, and time-varying environments. The human brain consists of $10^{10} - 10^{11}$ biological neurons. A schematic diagram of **a biological neuron** is shown in figure 1.4.

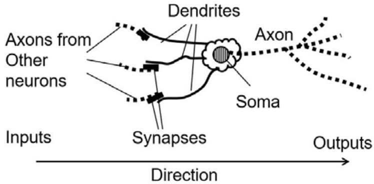


Fig. 1.4. A schematic diagram of a biological neuron

A biological neuron receives about 10 000 inputs (dendrites), processes the inputs, and generates single output. The output (axon) is connecting with about 10 000 dendrites of other biological neurons. Synapses are connections of dendrites of the neuron and axons of other neurons. The input signals are impulses. Dendrites transmit the input impulses to soma. This transmission can be either excitatory or inhibitory. Soma aggregates signals from dendrites and generates output impulses to axon.

1.5. Mathematical model of a biological neuron

A mathematical representation of a biological neuron is shown in figure 1.5.

The inputs of the neuron are signals $x_1 \ldots x_n$ from other neurons, and threshold x_0. Synaptic operation assigns a weight $w_1 \ldots w_n$ to each input $x_1 \ldots x_n$

according to the past experience. Thus, weights $w_1 \ldots w_n$ are memory of the neuron.

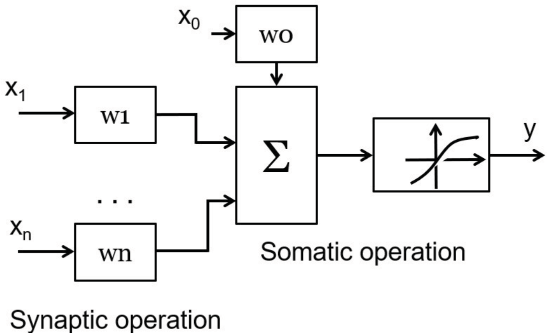


Fig. 1.5. A mathematical representation of a biological neuron

Somatic operation provides aggregation, thresholding, and nonlinear activation. Usually aggregation is a sum operation. Threshold adjusts a sensitivity of the neuron to the level of aggregated signal. Nonlinear activation is nonlinear static function with saturation.

1.6. A neural model for a threshold logic

The first mathematical model of a neuron was proposed by McCulloch and Pitts (1943). This model is a threshold logic element. The element consists of n logic inputs x_i, i=1,2,…,n, and one logic output y.

Output y is modeled as follows:

$$y = \begin{cases} 1, if \sum_{i=1}^{n} w_i x_i \geq w_0, \\ -1, if \sum_{i=1}^{n} w_i x_i < w_0. \end{cases} \tag{1.1}$$

The schematic representation of a threshold logic element is shown in figure 1.6.

A threshold logic element executes logic operations AND, OR, NOT. Figure 1.7 shows these operations.

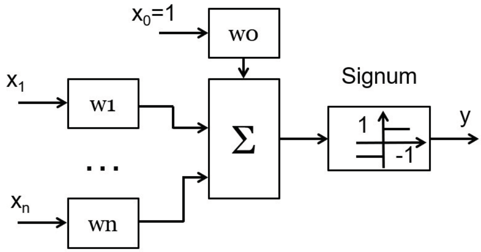
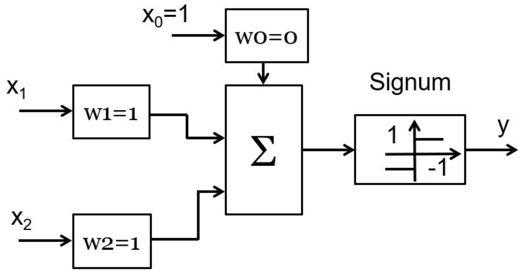
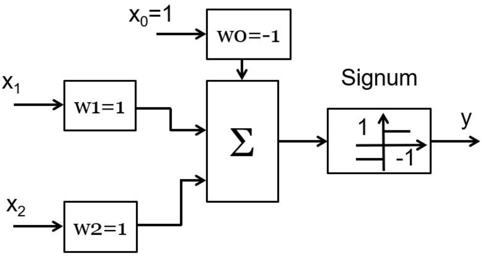


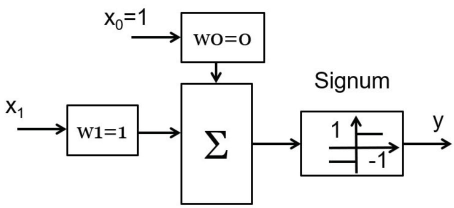
Fig. 1.6. A mathematical model of a threshold logic element



OR



AND



NOT

Fig. 1.7. A threshold logic implementation of OR, AND, and NOT

1.7. A neural threshold logic synthesis

Let introduce the notion of an augmented vector of synaptic weights w_a. *Augmented vector* w_a is defined as follows:

$$w_a = [w_0 \quad w_1 \quad \cdots \quad w_n]^T. \qquad (1.2)$$

Augmented vector of neural inputs x_a is defined as follows (x_0=1):

$$x_a = [x_0 \quad x_1 \quad \cdots \quad x_n]^T. \qquad (1.3)$$

Thus, we can write the neural signals using the notion of the augmented vectors as

$$y = signum(w_a^T x_a). \qquad (1.4)$$

Expression (1.4) can be described by a binary logic. Let us consider the augmented vectors x_a= [1 x_1 x_2 x_3] and w_a= [-1 3 4 5]. The output signal y is

$$\begin{aligned} y &= signum([-1 \quad 3 \quad 4 \quad 5][1 \quad x_1 \quad x_2 \quad x_3]^T) = \\ &= signum(-1 + 3x_1 + 4x_2 + 5x_3). \end{aligned} \qquad (1.5)$$

Let us the inputs are logic variables xi∈{-1, 1}. Then the truth table is

x_1	-1	1	-1	1	-1	1	-1	1
x_2	-1	-1	1	1	-1	-1	1	1
x_3	-1	-1	-1	-1	1	1	1	1
y	-1	-1	-1	1	-1	1	1	1

From the truth table we obtain

$$y = x_1(x_2\bar{x}_3 + \bar{x}_2 x_3) + x_2 x_3. \qquad (1.6)$$

Let us consider the following logic function

$$y = x_1(\bar{x}_2 + \bar{x}_3). \qquad (1.7)$$

Output y=-1 for both $x_1x_2x_3$ and not($x_1x_2x_3$). Therefore

$$w_1 + w_2 + w_3 < w_0, and - w_1 - w_2 - w_3 < w_0$$

The last inequalities are conflicting. Thus, no weights and threshold values can satisfy them. Function (1.7) cannot be realized by a single threshold element.

A switching function that can be realized by a single threshold element is called a threshold function. A threshold function is also called a linearly separable function (fig 1.8). It means that hyper plane $w_a{}^{\mathrm{T}} x_a$ divides all values of a threshold function in the next way. All the true points are on one side of the hyperplane, and the false points are on the other side of the hyper plane.

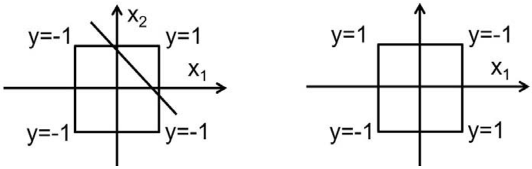


Fig. 1.8. Functions: a) a linearly separable function; b) a linearly non-separable function (XOR)

Realization of a linearly non-separable function requires network of threshold elements (fig 1.8). Decomposition of the non-threshold function is way to synthesis such network. Any given switching function may be realized by a two-layered threshold network, as shown in figure 1.9. The intermediate variables $z_1, z_2, ..., z_m$, may be computed by

$$z_i = signum\left(\sum_{j=1}^{n} w_{ij}^1 x_j\right). \tag{1.8}$$

The output of the network is computed by

$$y = signum\left(\sum_{i=1}^{m} w_i^2 z_i\right) = signum\left(\sum_{i=1}^{m} w_i^2 \left(signum\left(\sum_{j=1}^{n} w_{ij}^1 x_j\right)\right)\right). \tag{1.9}$$

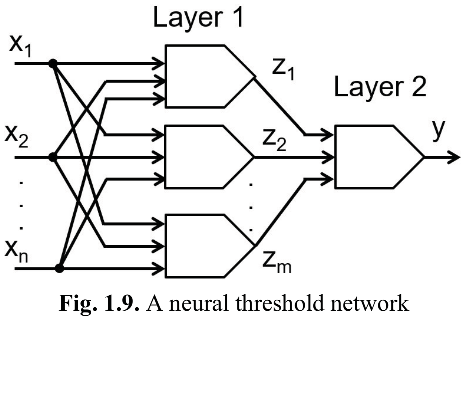


Fig. 1.9. A neural threshold network

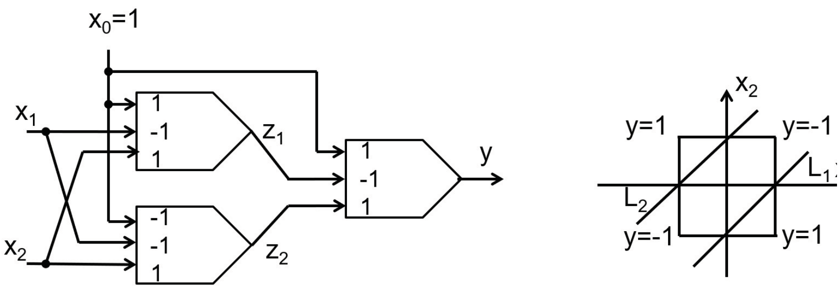
Realization of **XOR** using a two-layered neural network is shown in figure 1.10 (a)

$$\begin{gathered} y = signum(1 - z_1 + z_2) = \\ = signum\big(1 - signum(1 - x_1 + x_2) + signum(-1 - x_1 + x_2)\big). \end{gathered} \tag{1.10}$$

As shown in figure 1.10 (b) the two-layered neural network separates plane (x1, x2) using two discriminant lines which are defined by

$$\begin{gathered} L_1: 1 - x_1 + x_2 = 0, \\ L_1: -1 - x_1 + x_2 = 0. \end{gathered} \tag{1.11}$$

The region between the two lines **L**1 and **L2**, **y = -1**, while in the regions outside these two lines, **y = 1**.



a b

Fig. 1.10. Functions: a) XOR function; b) Two discriminant lines for XOR function

1.8. Problems

1. What are the advantages of **automatic control** over **remote control**?
2. What are the traits of the **hard problems**?
3. Draw the structure of **the intelligent robot control system**. What are the problems solved by **artificial intelligence**?
4. Give the taxonomy of **the artificial intelligence technologies**.
5. Give a schematic diagram of a **biological neuron**. Explain the functions of **dendrites, axon, synapses, and soma**.
6. Give a **mathematical representation** of a biological neuron. Explain the **synaptic operation** and the **somatic operation**.

7. Give a structure and mathematical model of the **threshold logic element**. Illustrate execution of logic operations **AND, OR, NOT** by the **threshold logic element**.

8. What is **a linearly separable function**? Give a geometric interpretation.

9. Give realization of **XOR** using a two-layered neural network.

Practical training 1

1.9. Task for practical training 1

1. Aim of the practical training is study of learning algorithm for an Adaline with sigmoidal function. A diagram of the learning scheme is presented in figure 1.11.

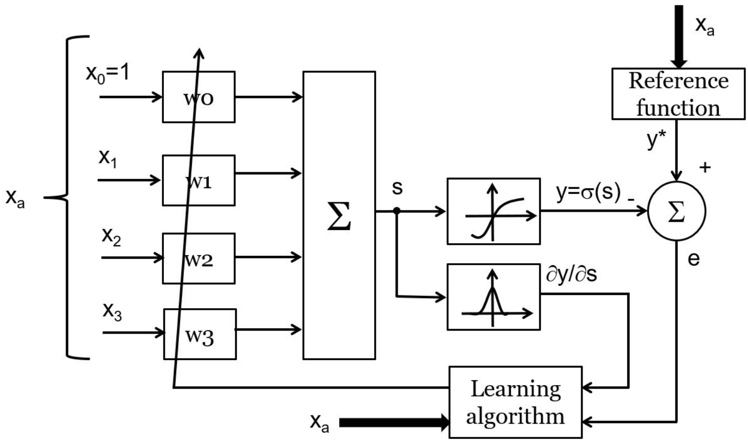


Fig. 1.11. A diagram of the learning scheme

The learning algorithm for an Adaline with sigmoidal function is described as follows.

$$w_a(k+1) = w_a(k) + 2\mu e(k)\frac{\partial y(s)}{\partial s}x_a(k); \tag{1.12}$$

$$y(s) = tanh(qs) = \frac{e^{qx} - e^{-qx}}{e^{qx} + e^{-qx}}; \tag{1.13}$$