ISSN 2278-3091
Volume 8, No.1.2, 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1281.22019.pdf
https://doi.org/10.30534/ijatcse/2019/1281.22019

Model of the Work of the Neurocontroller to Control Fuzzy Data from the Sensors of the Climate Control Subsystem "Smart House"

Oleksii Liashenko[1], Olesia Barkovska[1], Chiai Al-Atroshi[3], Oleh Datsok4, Sergey Liashenko[5]
[1]Department of Electronic Computers, Kharkiv National University of Radio Electronics, Ukraine, oleksii.liashenko@nure.ua
[2]Lecturer, Dept of Information Technology, Lebanese French University, Erbil, KR-Iraq chiai@lfu.edu.krd
[4]Department of Biomedical Engineering, Kharkiv National University of Radio Electronics, Ukraine, oleh.datsok@nure.ua
[5]Department of Life Safety and Law, Petro Vasylenko Kharkiv National Technical University of Agriculture, Ukraine, lyashenkosa05@ukr.net

ABSTRACT

The task control smart home is a complex dynamic problem which can be solved with the use of elements of artificial intelligence. As logic, it is proposed to use the logic of artificial neural networks. The proposed software and hardware implementation of the neurocontroller based on the perceptron.

Key words : Smart house, artificial neural networks, perceptron, climate control, sensor, neurocontroller.

1. INTRODUCTION

The term "Smart House" [1] refers to a system that has the ability to adapt, adapt, and adapt to various changes both at the present time and in the future. For example, this may be an adaptation for a new owner of a house or apartment and its needs. Quite often, such a system can be described as "building ready for change" or "flexible building." Therefore, the term "smart house" requires the highest level of intellectualization of engineering systems.

Realization of the functions of the "smart house" system requires the use of models, methods and means of intellectual processing of data and decision-making on their basis. Of particular practical interest, from the point of view of possible application in the subsystems of the "intellectual home", is the artificial neural networks (ANN), since they allow the software and hardware to process fuzzy and unstructured information from the subsystem of the sensors that provide the registration of changes in the environment of the "smart house". Thus, the use of ANN allows realizing the functions of the system "intellectual home" with different values of speed, complexity and intellectualization. The most common are neural networks that are structured in layers and, depending on their functional purpose, may contain the same type or different types of neurons.

In a one-layered structure with a complete connection, all incoming signals can be received on all neurons. An example of this type is the perceptron - the first artificial neural network in modern terminology. In general, it is the simplest neural structure, which, accordingly, can solve only the simplest tasks associated with the processing of data from the subsystem of sensors in the "smart house". The advantages of such neural structures can be attributed to their software and hardware simplicity and the high speed of the learning algorithm [98], but difficult to implement complex intellectual functions on their basis.

Multilayer artificial neural networks have much greater functionality. In most cases, multilayered perceptron’s are used [2] [10]. The peculiarity of such neural structures is the presence of joints between layers that go from first to next layers. Accordingly, such ANN includes an input layer, which directly inputs signals, hidden and output layers. The number of neurons in the layers may be different, although there are a number of approaches to assess the number of layers and neurons in the layer of artificial neural network [8]. Such ANNs of the type of multilayer perceptron are simple in hardware implementation and can be used in the process of automated design of subsystems of the "intellectual house". At the same time, it is necessary to pay attention to the lack of these structures, namely: the need for special and slow algorithms for their training.

A software version of a multi-layer perceptron in the design process of a "smart house" is a promising method of intellectualizing the functions of such systems.

Of particular note is the Kohonen network [3], which has increased reliability, which ensures its work in the event of failure of one of the neurons. This is a useful property in the process of automated design of "smart houses" especially in the hardware implementation of such an ANN to increase the speed of the subsystem and its reliability [9].
In the process of implementing the system of "smart house" quite often there is a problem of processing images. In such a

situation, the Hopfield network [4] itself has proven itself well, respectively, to solve precisely such or similar problems it should be used in the subsystems of the "smart house". At the same time, Hopfield's network has high-performance and simple hardware implementation [7].

Also, for controlling of dynamic objects, which is a "smart house", ANNs with radial-base functions of activation are used [5].
From the above we can conclude that the most often for the implementation of the functions of "smart house" is worth using an ANN type multilayer perceptron [6].

2. THE STRATEGY OF THE NEUROCONTROLLER TO CONTROL THE CLIMATE CONTROL SUBSYSTEM

The developed neurocontroller must perform the embedded control logic of devices depending on the received signals from the sensors.

In accordance with the hygienic requirements for a microclimate of housing, the temperature in the room should be within 18-20 ^{o}C, and humidity within 30-60%, although at the request of the owner of the room, these parameters can be changed.

Therefore, in any deviation from the comfort zone, it is necessary to adjust the microclimate in the room. Depending on the value of the microclimate, a management strategy was developed, which can be depicted in the form of a scheme depicted in Fig. 1. If necessary, the strategy for climate control in the room can be changed or improved.

From the figure 1 below, it is necessary to activate the humidifier and the heater at a temperature of less than 18 °C and a humidity of less than 30%. When the temperature is in the range from 18 °C to 22 °C and humidity in the limits from 30% to 60% - nothing is necessary to change.

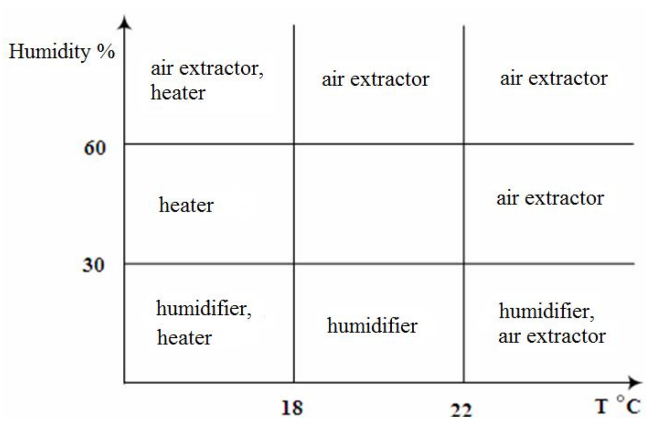

Figure 1: Strategy control “climate control”

At the heart of the system is a microcontroller, which is responsible for working with actuators and sensors. The microcontroller outputs data to the computer screen via a COM port or device displaying a smart house control system. The microcontroller can be programmed for a variety of functions and it is capable of doing simple calculations. In order not to fully download the memory and processor microcontroller, we use a truncated version of the program, which contains all the necessary elements.

So, we will build a model for processing fuzzy data on the basis of an ANN, which will emulate a program based on a network type multi-layer perceptron.
In the general case, the mathematical model of an artificial neuron with n inputs is described by the relation (1):

$$y = F\left(\sum_{i=1}^{n} w_i \times x_i\right) + b, \tag{1}$$

where w_i – weight coefficients; x_i – - input value; b – displacement; F – activation function; y – is the value of the output of the neuron.

In order to describe the structure of ANN, in the process of implementation, the appropriate matrix of bonds is used in the work. The structure of the developed neural network model for the climate control subsystem, in the process of implementing the above strategy (Figure 1), contains 3 layers and a total of 12 neurons.

The constructed ANN has 2 input neurons and 3 output neurons. The number of layers and neurons in layers was determined experimentally so that after learning the error of the results was acceptable. The best option is a network with one inner layer for 4 neurons. Also, in the process of implementation, you need to add 2 balancing neurons.

As a result of ANN training using software received from the network parameters are shown in Figures 2 - 3, and Figure 4 shows a graphical representation of the structural model developed ANN [9, 20].

Net

Neuron ID = 0	*number of function = 2*	*1.0000*
Neuron ID = 1	*number of function = 2*	*1.0000*
Neuron ID = 2	*number of function = 2*	*1.0000*
Neuron ID = 3	*number of function = 3*	*1.0000*
Neuron ID = 4	*number of function = 3*	*1.0000*
Neuron ID = 5	*number of function = 3*	*1.0000*
Neuron ID = 6	*number of function = 3*	*1.0000*
Neuron ID = 7	*number of function = 3*	*1.0000*
Neuron ID = 8	*number of function = 3*	*1.0000*
Neuron ID = 9	*number of function = 1*	*1.0000*
Neuron ID = 10	*number of function = 1*	*1.0000*
Neuron ID = 11	*number of function = 1*	*1.0000*

Figure 2: Parameters of neurons in a developed model based on ANN

Each neuron in the process of implementation of the developed ANN assigned ID number conversion functions and parameter functions.

In general, the structural model can be described as follows ANN tuple:

$$STR\_ANN = \langle ID, N\_Fun, Par, N \rangle, \quad (2)$$

where $ID=id(i)$ – set of ANN identifiers; $N\_Fun=n\_fun(i)$ – set of neuron functions; $Par=par(i)$ – set of neuron parameters; N – the number of neurons in ANN.

n	n	n	o	o	o	o	o	n	n	n	n
n	n	n	o	o	o	o	o	n	n	n	n
n	n	n	o	o	o	o	o	n	n	n	n
i	i	i	n	n	n	n	n	n	o	o	o
i	i	i	n	n	n	n	n	n	o	o	o
i	i	i	n	n	n	n	n	n	o	o	o
i	i	i	n	n	n	n	n	n	o	o	o
i	i	i	n	n	n	n	n	n	o	o	o
n	n	n	n	n	n	n	n	n	o	o	o
n	n	n	i	i	i	i	i	i	n	n	n
n	n	n	i	i	i	i	i	i	n	n	n
n	n	n	i	i	i	i	i	i	n	n	n

Figure 3: Parameters of the matrix of connections between neurons of a model based on ANN

To represent the matrix of connections between neurons, a matrix of size (N + 1) × (N +1) is used. Each element of the matrix can take one of three possible values: "n" - no communication; "o" is the initial rib; "i" is the input edge. An example of an appropriate matrix is shown in Fig. 3

To represent the matrix of the values of the bond weights, the adjacency matrix is used. Each item is a valid value. In the case when the element of this matrix is zero, then the connection is absent. An example of a matrix of adjacency with the values of bond weights obtained as a result of the study of the ANN. Graphical representation of the model structure based on ANN using multilayer perceptron shown in Fig. 4

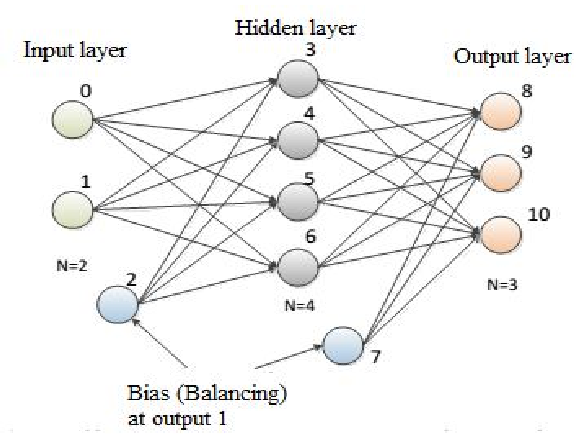

Figure 4: Parameters of neurons in a developed model based on ANN

3. DEVELOPMENT OF THE STRUCTURE AND ALGORITHM OF THE NEUROCONTROLLER

Physical implementation of the neurocontroller involves the use of a microcontroller, which will perform the same program in a loop, so the algorithm of work must be carried out an unlimited number of iterations. Software implementation simulates the work of artificial neural network type multi-layer perceptron.

The algorithm developed in the work includes the steps listed below.

Step 1: Reading input data from the sensors.
Step 2: Initializing variables.
Step 3: i = 0 (the beginning of the cycle).
Step 4: i < netSize ?, yes - go to step 5, no - go to step 17.
Step 5: vin[i] = 0.
Step 6: i <cntIn ?, yes - transition to step 7; no - move to step 8.
Step 7: vout[i] = input[i], transition to step 16.
Step 8: balance[i] == 1?, yes - go to step 9, no - go to step 10.
Step 9: vout[i] = 1, transition to step 16.
Step 10: j = 0.
Step 11: j < netSize ?, yes - switch to step 12, no - go to step 15.
Step 12: type[i][j]==’i’ ?, yes - transition to step 13, no - transition to step 14.
Step 13: vin+=vout[j]*value[j][i].
Step 14: j++, transition to step 11.
Step 15: vout[i] = calc(i).
Step 16: i++, transition to step 4.
Step 17: first_out = netSize – cntOut.
Step 18: i = first_out.
Step 19: i<netSize ?, yes - switch to step 20, no - switch to step 22.
Step 20:output[i-first_out] = vout[i].
Step 21: i++, transition to step 19.
Step 22: output results.

The developed structure of the neurocontroller consists of two interconnected levels of hardware and software implementation.

Hardware implementation (model) includes:
- subsystem of work with executing devices that are responsible for outputting the values of analog signals;
- subsystem of work with sensors, which is responsible for reading digital signals;
- subsystem of the serial port, which is responsible for sending, intermediate and final results to your computer.

Software implementation (model) includes:
- subsystem of working out of input data received from sensors;

- the subsystem of the formation of control signals, which, based on the received results from the neural network, forms teams for the microcontroller;
- subsystem using a neural network that makes data network, launching the network and gets results.

In this system, ANN is implemented in software. Figure 5 shows a block diagram of an algorithm which implements the program type multilayer perceptron ANN.

As noted above, the neurocontroller performs the same program in a loop (Figure 6) and the program execution algorithm is the following:

Step 1: Read the analog signal from the sensors.
Step 2: Analog signals transmitted to digital.
Step 3: Process input data.
Step 4: Enter data for incoming neurons.
Step 5: Run the Neural Network.
Step 6: Read the results from the source neurons.
Step 7: Workout the output.
Step 8: Output the digital signal management devices.
Step 9: Delay, step by step 1.

The operation of the neurocontroller stops after power is turned off.

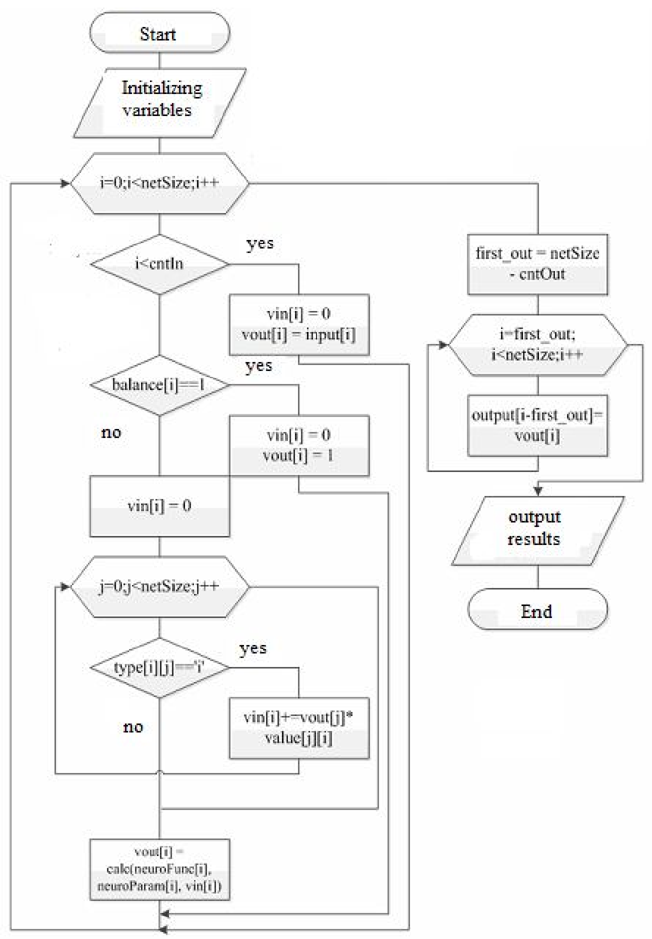

Figure 5: Block diagram of the algorithm of the program, which implements ANN

4. Development of Software Model and Features of Software Implementation of Neurocontroller

The developed software model for the neurocontroller contains a number of modules that are executed once, they include the module of initialization of the network in which the initialization of the weights, connections and parameters of the neurons takes place. Also, a port initialization module is executed once, which sets port status and initial values. Another part of the program executed in the loop and includes a number of modules. In the process of implementation, a modular approach is used, which allows you to quickly modify the system when it is refined.

The software is broken down into a number of individual subtasks such as:
- data readout - processing of data from sensors, checking the function of the sensor, bringing data to the required formats;
- data normalization - organization of data in accordance with the training sample of the network;
- network launch - data entry to the network and data transmission through the network;
- outputting results - obtaining results from the network and outputting them through a serial port;
- formation of control commands - depending on the results of the formation of signals at the output ports;
- delay formation is the creation of a time delay before passing the next cycle of the program

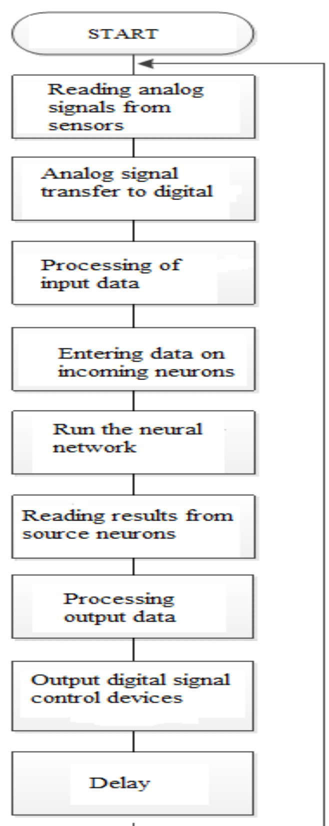

Figure 6: Block diagram of the algorithm of the neurocontroller

5 HARDWARE IMPLEMENTATION OF THE NEUROCONTROLLER FOR THE SUBSYSTEM OF CLIMATE CONTROL

Hardware implementation of the neurocontroller for the climate control subsystem includes a microcontroller, temperature sensor and humidity sensor DHT11, three LEDs and limiting resistors. The microcontroller uses four ports to work with an external circuit, namely: the 2nd port is used to read the signal from the sensor, and the 5th, 6th, 7th ports are used to display the active operation of the devices (heater, fan and humidifier air) signaled by the flash of the corresponding LED (Figure 7).

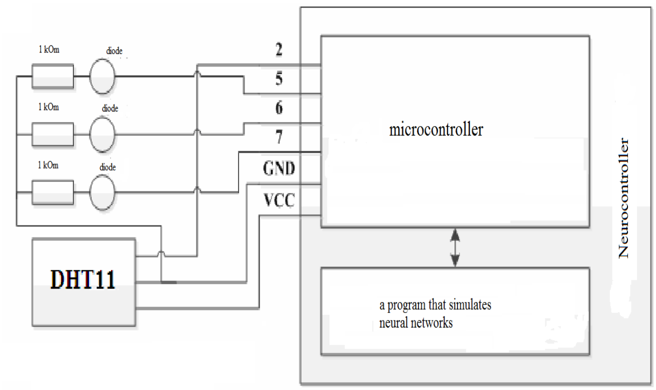

Figure 7: Connection scheme of the subsystem climate control

After launching the neurocontroller for one cycle of the program, the following intermediate and output results were sent through the serial port to the computer, which allows to check the correctness and correctness of the work of the developed neurocontroller.

6. CONCLUSION

In this work a model based on artificial neural networks was developed, which provides processing of fuzzy and unstructured data from the subsystem of the sensors.
The developed model on the basis of ANN and built specialized software enable to build a neurocontroller of the subsystem of climate control "smart house".

REFERENCES

1. Jiang L., Jiang L., Liu D. Y., Yang B., **Smart home research**, in Proc. of the 2004 International Conference on Machine Learning and Cybernetics. – Shanghai, China. – 2004. – V. 2. – pp. 659-663.
2. Divya, P., and V. Manikandan, **AN ENERGY SAVING TECHNIQUE FOR WIRELESS SENSOR NETWORKS** International Journal of Innovative Research in Engineering Science and Technology.
3. Mohamad H. H., **Fundamentals of Artificial Neural Networks.** London : MIT Press, 1995.
4. Kohonen, T. **Self–organization and associative memory**. Series in Information Sciences, vol. 8. Berlin: Springer verlag, 1984.
5. Hopfield J. J., **Neural networks and physical systems with emergent collective computational abilities**, in Proc. of National Academy of Sciences. – 1982. – V. 79, N. 8. – pp. 2554-2558. https://doi.org/10.1073/pnas.79.8.2554
6. Mohammad, Amin Salih, and Moayad Yousif Potrus, **A Method For Compensation Of Tcp Throughput Degrading During Movement Of Mobile Node**, ZANCO Journal of Pure and Applied Sciences 27.6 (2016): 59-68.
7. Porkodi, V., et al, **Survey on White-Box Attacks and Solutions,** Asian Journal of Computer Science and Technology 7.3 (2018): 28-32.
8. Kulandhaivelu, S., Ray, K. K., Jyothi, P., Jayachitra, S., & Rao, V. N. (2011, March), **Experimental evaluation of capacitance value; to self-excite the induction motor operating as generator in wind energy conversion,** In 2011 International Conference on Emerging Trends in Electrical and Computer Technology (pp. 133-140). IEEE.
9. Awasthi, Yogesh, R. P. Agarwal, and B. K. Sharma, **Intellectual Property Right Protection of Browser based Software through Watermarking Techniques,** International Journal of Computer Applications 97.12 (2014). https://doi.org/10.5120/17061-7468
10. Asim Majeed, Rehan Bhana, Shaheed Parvez, **Controlling Energy Consumption by Internet of Things (IoT) Applications,** International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No 1.1, 2019, pp. 8-11. https://doi.org/10.30534/ijatcse/2019/0281.12019