SLOBODAN BEROŠ, D.Sc.
SAŠA MLADENOVIĆ, B.Eng.
ŠPIRO MATOŠIN, D.Sc.
Fakultet elektrotehnike, strojarstva i brodogradnje
Split, Rudjera Boškovića 6

Traffic Infrastructure
Preliminary Communication
U. D. C.656.053:681.3
Accepted: Sept. 9, 1997
Approved: oct. 28, 1997

NEURO SYSTEM STRUCTURE FOR VEHICLE RECOGNITION AND COUNT IN FLOATING BRIDGE SPECIFIC CONDITIONS

SUMMARY

The paper presents the research of the sophisticated vehicle recognition and count system based on the application of the neural network. The basic elements of neural network and adaptive logic network for object recognition are discussed. The adaptive logic network solution ability based on simple digital circuits as crucial in real-time applications is pointed out. The simulation based on the use of reduced high level noise picture and atree 2.7. software have shown excellent results. The considered and simulated adaptive neural network based system with its good recognition and convergence is a useful real-time solution for vehicle recognition and count in the floating bridge severe conditions.

1. INTRODUCTION

The reason why neural networks applied in solving the pattern recognition problem have found themselves in the centre of attention is the possibility of developing highly parallel systems. Unlike statistical analysis and its mathematical complexity, the neural network processes information by carrying out a number of simpler mathematical operations simultaneously. Quite generally, the neural network can be presented as a machine which simulates the way in which human brain operates, and can be formed by using electronic components or by simulation on a digital computer. The neural network is a massively divided parallel processor with the ability to store and achieve empirical knowledge, simulate human brain in the process of learning, and store the knowledge by synaptic weights links between neurones.

The floating pontoon bridge constructed as replacement for the destroyed Maslenica bridge, of special significance as the only connection from the then isolated Dalmatian region, has motivated, because of its specific problems in war conditions of low and no visibility, the study of the possible application of sophisticated methods in vehicle recognition and identification, in unmanned technical system.

Fuzzy neural system for recognition and count of vehicles on the bridge, presented in flowchart in Figure 1, using an imaging camera, neural network, and fuzzy decision logic, has proven its feasibility in simulation [1].

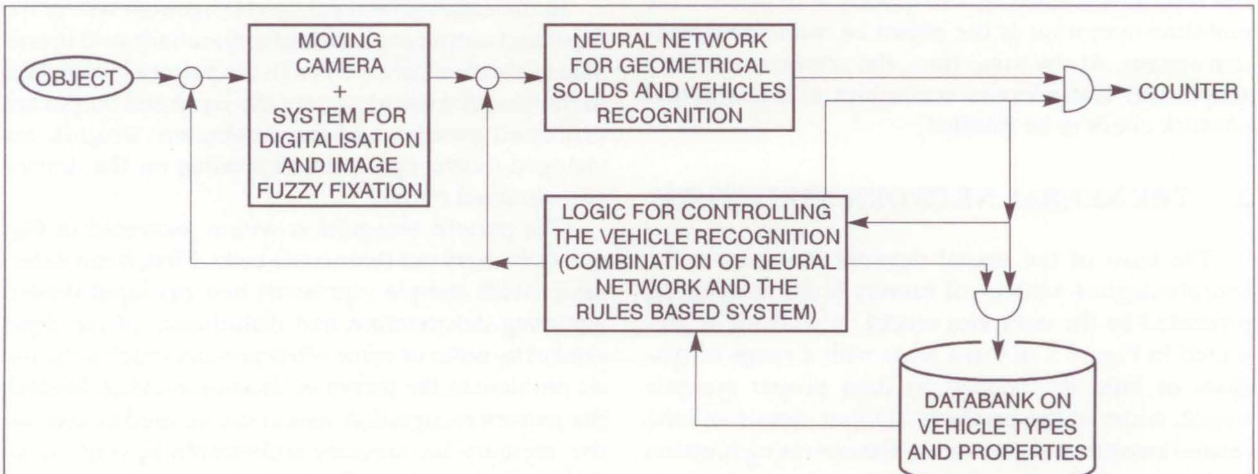


Figure 1 - Fuzzy neural system flowchart

The moving camera and the system for digitalisation and fuzzy fixation make up the input interface converting the actual image of a vehicle into a fuzzy recording in the computer memory. The output from the neural network is not a type of object but the degree of membership within a set of properties, on the basis of which the object recognition control logic classifies the observed object. Solving the vehicle classification problem, due to e.g. many different versions of a make of car, damages, or accessories, and imaging due to e.g. different vehicle velocities, weather conditions, vibrations, or illumination, precisely prove the advantages of neural networks and fuzzy systems because of their reduced high level noise picture. The core of the system is the object recognition control logic, which controls the whole system through feedback and determines when the object should be detected, new object defined, controls learning and extension of the network, as well as discarding of recognised properties. Input into the control logic are fuzzy data and the decision-making process is adequate to the expressing procedure. Due to the fast accurate object detection, the control signals should control the adaptive step, i.e. the activity performed within time unit is increased in proportion with the time of its constant performance. The data bank of vehicles and their properties contains databases of the object properties and their interrelations used by the control logic to determine the object and define a new one or a group of objects. When the control logic is completely realised through neural network, the bank of objects and their properties may be realised within the network itself by defining the strength of links between the nodes [1]. At the same time the complexity and the high price of the projected realisation of such a system need to be taken into consideration, and especially the inadequately long time of decision convergence.

Further research are carried out with the aim of developing a neural system with a network that in object recognition converges so quickly as to provide the real-time operation of the object i.e. vehicle recognition system. At the same time, the requirement for a simple and cost-effective realisation of a fast neural network needs to be fulfilled.

2. THE NEURAL NETWORK STRUCTURE

The basis of the neural network structure is the neurobiological analogy of human brain functioning presented by the neurones model. A neurone is presented in Figure 2 in three parts with a range of synapses or links determined by their proper synaptic weight, adder of the products of input signals and the related synaptic weights and by the operating function for limiting the output signal of the neurone normalised in the form of the interval [0,1] or [-1,1].

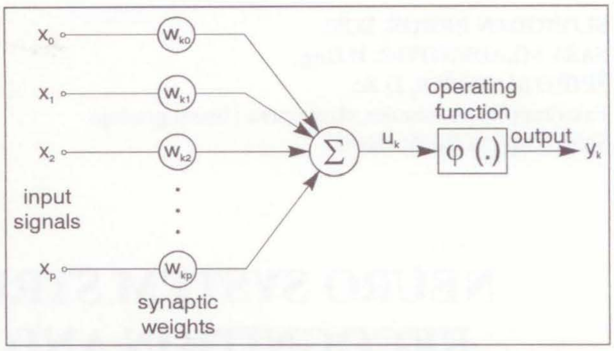


Figure 2 - Model of a non-linear threshold neurone

For input signals $x_1, x_2, ..., x_p$ and synaptic weights $w_{k1}, w_{k2}, ..., w_{kp}$ of the k-neurone with the defined synapse of input x_0 = -1 and related threshold θ_k equal to the synaptic weight $w_{k0} = \theta_k$ output u_k from the adder is defined with $u_k = \sum_{j=0}^{p} w_{kj} \cdot x_j$ and the output signal y_k from the neurone with $y_k = \varphi(v_k)$ where v_k is the level of internal operation $v_k = u_k - \theta_k$.

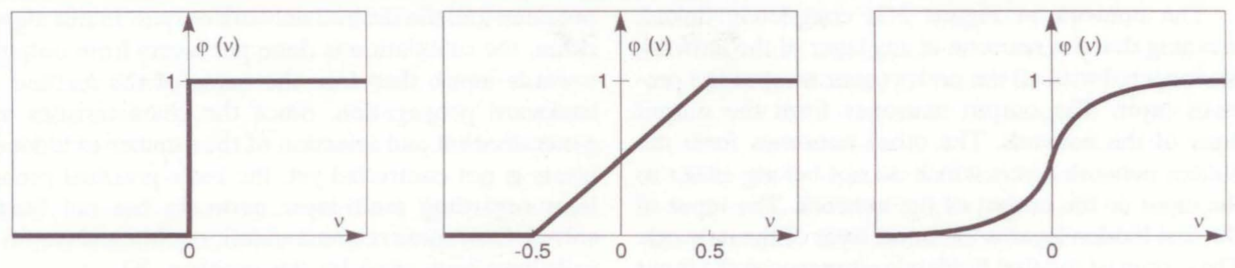
It should be noted that the operating function $\varphi(.)$ stipulates the neurone output thus defining various models.

Figure 3 presents the operating functions and recording for the three most frequently used forms a) McCulloch-Pitts threshold function, b) linear or piecewise function form and c) Sigmund threshold function [2].

The more complex nodes in the network can also have elements with integration per time or some other type of time dependence, as well as mathematical operations more complex than addition used in the presented neurone model. The models of neural networks are determined by network topology, node properties and training or learning rules. These rules determine the starting values of synaptic weights and indicate the changing mode of synaptic weights during operation in order to improve the network properties.

In the conventional pattern recognition system the input and output are analysed sequentially with invariable evaluation parameters. In the pattern recognition in the neural network system the input and output are processed parallel, and the parameters, weights, are changed during operation depending on the desired and obtained output.

The pattern recognition system presented in Figure 4 can carry out three basic tasks. First, it can determine which sample represents best the input symbol assuming deformation and disturbance of the input symbol by noise or some other process which is the basic problem in the theory of decision-making. Second, the pattern recognition system can be used as associative memory i.e. memory addressable by content. In this kind of problem, the desired result is known, and the input symbol is used in order to determine the pro-



a) b) c)

$$\varphi(v)=\begin{cases}1; & v\geq 0\\ 0; & v<0\end{cases}$$

$$\varphi(v)=\begin{cases}1; & v\geq 0.5\\ v; & -0.5>v>0.5\\ 0; & v<-0.5\end{cases}$$

$$\varphi(v)=\frac{1}{1+e^{-av}}$$

$$\varphi(v)=\mathrm{tgh}\left(\frac{v}{2}\right)=\frac{1-e^{-v}}{1+e^{-v}}$$

Figure 3 - Operating functions of the neurone model

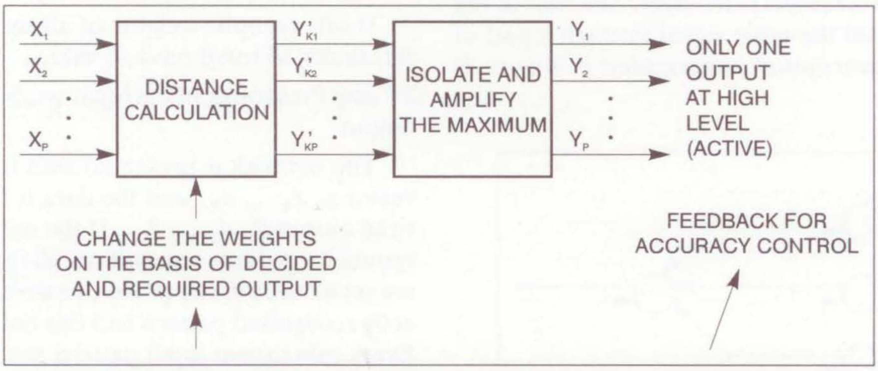


Figure 4 - The basic flowchart of the neurone network recognition system

cess for obtaining the desired result. Third, the vector can be quantised, i.e. N inputs turned into M patterns. The vector quantising is used in transferring the video and audio signal in order to reduce the number of bits necessary for transfer of the original analogue signal without losing important information. In this way, greater amount of information is transferred through the same channel width.

In these tasks, the number of patterns can be precisely determined in advance or limited only by the number of nodes available in the first processing step, i.e. by the number of input nodes.

Multi-layer perceptron is a network with the link forward with one or more layers between input and output. These additional layers contain hidden units or nodes which are not directly linked to the input and output nodes. The multi-layer perceptron with two hidden layers can be presented as in Figure 5. [3],[4].

Although the neural network architecture was known in nature, the knowledge about this architecture could not be used because of the lack of knowledge of the adequate learning rules. In the multi-layer perceptron the learning rules used in single-layer perceptron could not be applied, because of the hidden neurone layers. In the past the multi-layer perceptrons were not used due to the lack of efficient algorithms for their training. Although it cannot be mathematically proven that these algorithms converge as is the case with algorithms used in single-layer perceptrons, they have proven to be good in practice. In the mid-80's the issues change significantly with the appearance of new algorithms for multi-layer perceptron training.

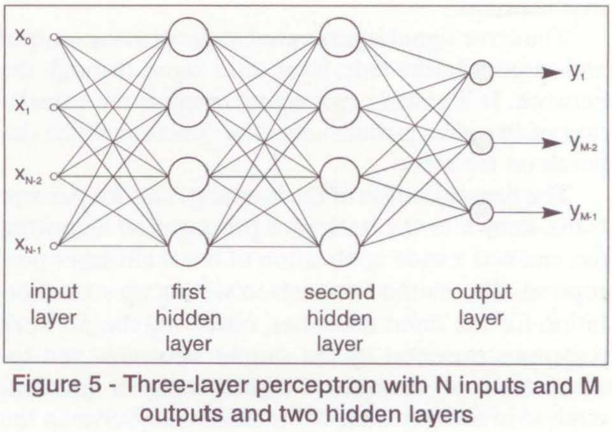


Figure 5 - Three-layer perceptron with N inputs and M outputs and two hidden layers

The network in Figure 5 is completely linked, meaning that the neurone at any layer of the network is connected with all the nodes (neurones) of the previous layer. The output neurones form the output layer of the network. The other neurones form the hidden network layers which do not belong either to the input or the output of the network. The input of the first hidden layer is the input layer of the network. The output of the first hidden layer serves as the input of the next hidden layer, and so on, for the rest of the network. Each hidden or output neurone of the multi-layer perceptron is formed in such a way as to be able to carry out two kinds of calculations; the calculation of the function signal at the output from the neurone and the calculation of the current estimate of the vector gradient which is necessary for the backtracking through the network.

The signal moves forward from left to right through the network, layer after layer. The flow of the function signal and the error signal through a part of the multi-layer perceptron is presented in Figure 6. [5].

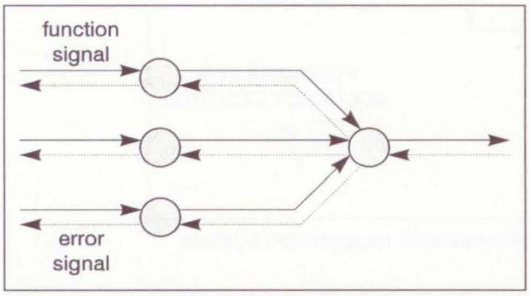


Figure 6 - Flow of the signals through a part of the multi-layer perceptron

The function signal is the input signal, stimulus, which appears at the network input and spreads forward through the network, neurone after neurone, and appears at the network output as the output signal. The output signal is called the function signal because the signal is calculated in each network neurone as the function of the input and the related weights for that neurone.

The error signal is generated at the neurone output and spreads backwards, layer after layer, through the network. It is called error signal because the calculation of its value includes also the function which depends on the error.

The generalisation of the learning rules for perceptrons, known as the backward propagation algorithm has enabled a wide application of the multi-layer perceptron. The method consists in setting up a configuration for the input neurones, observing the network responses provided by the output neurones and recalculation of synaptic weights using a gradient method in order to reduce the difference between the obtained and the desired network output. In this algorithm, the calculation is done per layers from output towards input, therefore the name of the method - backward propagation. Since the characteristics of generalisation and selection of the number of hidden layers is not controlled yet, the basic practical problems regarding multi-layer networks has not been solved. Only some recommendations obtained empirically have been given for this problem. [6].

The backward propagation algorithm is an iterative gradient algorithm with the objective to minimise the mean square error between the desired and the obtained network output. As threshold it uses the Sigmond function $\varphi(v)=\dfrac{1}{1+e^{-(av-\theta)}}$, and is carried out step by step. Pseudo in the backward propagation algorithm is defined by:

1st step Assigning of synaptic weights and threshold

It sets synaptic weights of all the nodes and their thresholds to small random values.

2nd step Presenting of the input symbol and the desired output

The network is presented with the constant input vector $x_0, x_1, \ldots, x_{N\text{-}1}$ and the data is fed about the desired output $d_0, d_1, \ldots, d_{M\text{-}1}$. If the network is used as a system for pattern recognition, all the desired outputs are set at value 0, except the one which matches the exactly recognised pattern and this one is set at value 1. Every subsequent input pattern can be a new one, or the patterns which form the training set can be repeated until the values of synaptic weights become constant.

3rd step Calculation of the current output values

The output $y_0, y_1, \ldots, y_{M\text{-}1}$ is calculated assuming the Sigmond function is used as the threshold function.

4th step Adjustment of weights

The adjustment of weights uses the recursive algorithm starting from the output node, and the calculation is carried out towards the first hidden layer.

The weights are adjusted according to the expression

$$w_{ij}\cdot(t+1)=w_{ij}\cdot(t)+\eta\cdot\delta_j\cdot x_i.$$

For j output node

$$\delta_j=y_j\cdot(1-y_j)\cdot(d_j-y_j).$$

For j internal (hidden) node

$$\delta_j=x_j\cdot(1-x_j)\cdot\sum_k\delta_k\cdot w_{jk}.$$

5th step Go to step 2.

The advantage of using the multi-layer over the single-layer perceptron, is best seen in the examples presented in Figure 7. [7], [8].

It is obvious that the greater the number of hidden layers, the better distinguished decision area is obtained.

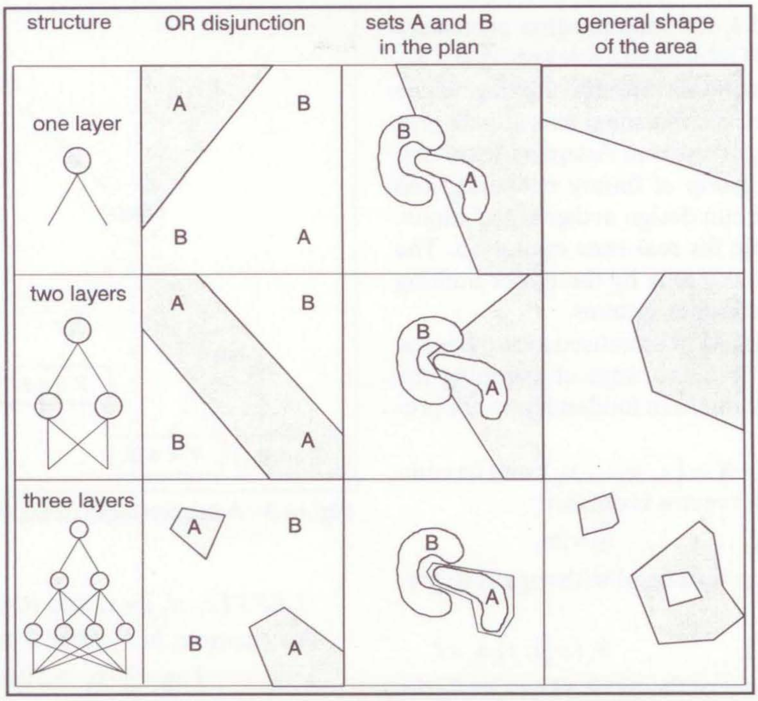


Figure 7 - Examples of decision areas obtained by differently structured perceptron

3. ADAPTIVE LOGIC NETWORK (ALN)

Adaptive logic network is the case of a multi-layer perceptron with connection forward using at the input the threshold nodes, and in the hidden layers and the output layer logic gates AND and OR. All the nodes together form a tree with one Boolean output. In order to explain the operation of the adaptive logic network, the operation model of the multi-layer perceptron has to be known in detail.

3.1. Properties and architecture of the adaptive logic network

The adaptive logic network ALN cannot give at the output the analogue values directly, so in that case some kind of re-coding is necessary. In spite of this limitation, precisely within the network structure there is a property which simplifies the learning control and the circuit realisation. Since the nodes in the network are limited to logic functions and to only two

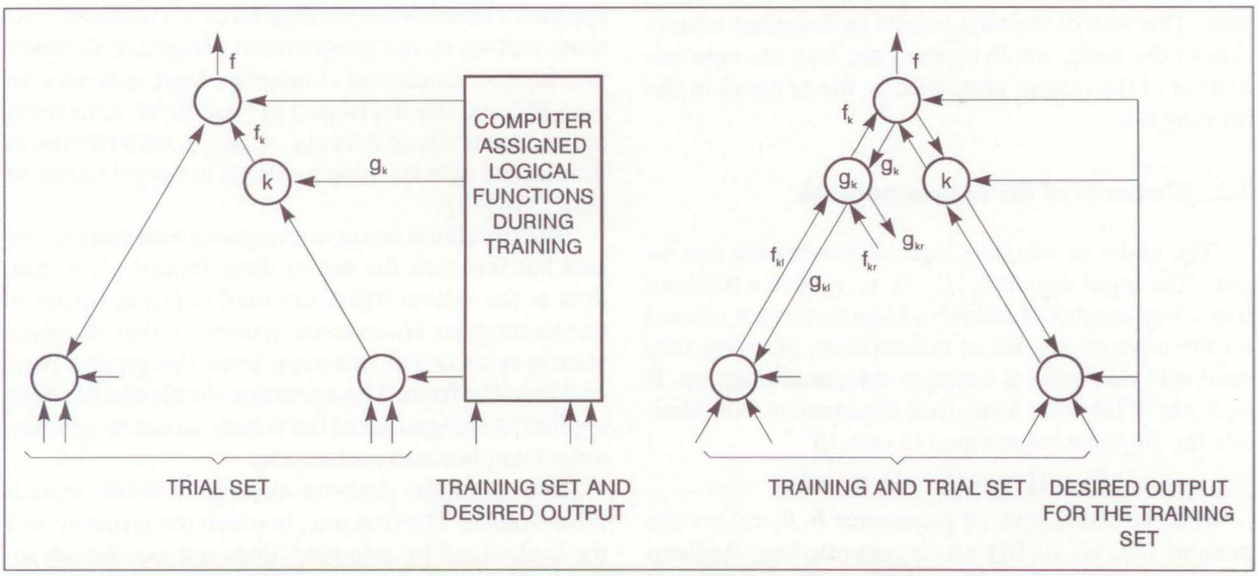


Figure 8 - Two ways of adaptive logic network circuit design a) the computer-aided assignment of functions to nodes, b) the system of adaptive elements responsible for the assignment of logic functions to nodes

possible inputs, 0 and 1, the minimisation procedure and the application of DeMorgan's rules provide a detailed analysis of the network decision making, which is not possible with the conventional neural networks based on multi-layer perceptron. Adaptive logic network belongs to the group of binary networks, and they are suitable for circuit design in digital technique, thus being also suitable for real-time operation. The operating speed is limited only by the upper limiting operation speed of the digital systems.

The circuit design of ALN is realised in two ways as shown in Figure 8, where two ways of assigning the logic functions to ALN nodes in hidden layers are presented [9].

For the input vector $X = \{x_1\ x_2 \dots x_n\}$ and its complement $\overline{X}$, a new input vector is defined

$$Y = X \bigcup \overline{X} = \{y_1\ y_2 \dots y_m\}, \quad m=2n.$$

The output function is defined with regard to previous layers as

$$f_{i,j} = f(f_{i-1,k}, f_{i-1,l}) \qquad k,l \in [1,j], k \neq l$$

where i≥2 the number of the hidden layer, and j the number of nodes of the i-th layer. The number of the hidden layers i as well as the number of nodes j per layer are determined experimentally.

Apart from the operation speed and the logic network learning ability, the important property is the generalisation of such a trained network, based on monitoring the properties of the tree structure of the logic gates with two inputs, theoretically supported by the work [10].

Training of the tree starts from the input into the tree through the leaves, and the output is the node-root. Input variables are elements of the input vector. Training on the set of input vectors along with the given desired output, consists in presenting the input vector in random order together with the desired outputs. This way of training results in assigning a function to the node, which enables the tree the approximation of the output presented to the network in the training set.

3.2. Elements of the neural network

The nodes or adaptive logic elements have two inputs. The input signals $b_1 \mid b_2 = 1\ x_1, x_2$ have a Boolean form. The weights of individual inputs are determined on the basis of one bit of information. Non-linearity used with threshold is a simple threshold function. If b_1, b_2 are of Boolean form, then the element will calculate the Boolean value equal to one, iff

$$(b_1 + 1) \cdot x_1 + (b_2 + 1) \cdot x_2 \geq 2.$$

Four combinations of parameter b_1,b_2 values are possible (00, 11, 10, 01) which generate four Boolean functions of two variables: AND, OR, LEFT and RIGHT, where

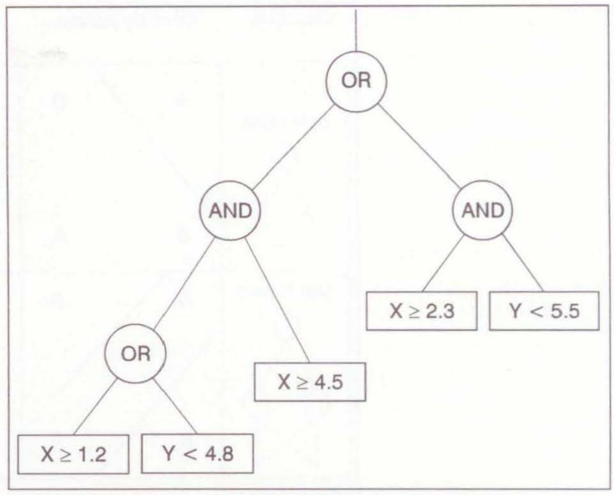


Figure 9 - Adaptive logic network following the training

$$LEFT(x_1, x_2) = x_1 \ and \ RIGHT(x_1, x_2) = x_2.$$

For example, b_1,b_2 value 0 there is a case

$$1 \cdot x_1 + 1 \cdot x_2 \geq 2 \text{ iff } b_1 | b_2 = 1.$$

The functions LEFT and RIGHT provide flexibility in interconnections. After the training has been completed, all the nodes which present the functions LEFT or RIGHT disappear and get replaced by real connections so that the network looks as presented in Figure 9.

4. RESULTS OF THE VEHICLES RECOGNITION AND COUNT SIMULATION

In network simulation of the problem of vehicle recognition using one or more sensors, the software package atree 2.7, a tool for ALN simulation, has been applied. The software package atree is a series of functions written in the programming language C, which provide the simulation of adaptive logic networks using a PC, and was developed by Prof. W.W. Armstrong at the University of Alberta - Canada, with the aim of developing new learning methods in neural networks [11], [12], [13].

The simulated recognition system with several sensors has foreseen the use of three independent cameras as the system input, oriented in (x,y,z) planes of the rectangular co-ordinate system, so that the given vehicle is recorded sideways, from the ground plan, and from the front. The operation simulation has been applied to recognition of the vehicle structure of a passenger car, bus and motorcycle.

Two possible decision-making network models were studied. The first one, in which the resulting vector is obtained by assigning, does not use the advantages of adaptive logic network sufficiently well, does not converge reliably, and was rejected as unsuitable.

The second one, in which the decision on classification category is made on the majority principle i.e. the classification into at least two identical classes defines the vehicle. In order to make the decision, three separate independent structures are used and the result is a reliable convergence.

In the performed experiment, the adaptive logic network used as the input bit-mapped display with 32x32 bits with removed background. The training was completed after about 30 minutes on PC 486, 133 MHz and 32 MB RAM. As example, for the simulation of processing the data obtained by a side camera with shot noise, the Windows operative system programme displays the window as presented in Figure 10.

ALN structure recognition			
	Change #1	Change #2	Change #3
Next	Stop	Back	
Training	Delete	Cancel	% Correct: 89,72
ALN Help topics	About the programme	Quit	%Incorrect: 10,28
The light goes on under the structure recognised by the network			

In the beginning, the adaptive logic network is not trained for vehicle recognition, which can be proven by starting the recognition procedure. The training can be avoided by copying the file already containing the training results. If the recognition procedure is started now, the network will recognise the given vehicle. Each of the structures can be additionally changed or rotated and the recognition started again with the new image. The image recognition with shot noise from 0 to 75% can be started from the highest noise (key Back - Natrag) or the lowest noise (key Next - Naprijed). When the network recognises the structure shown in the left window, the light under the recognised structure goes on. In this way, it is possible to subjectively determine the resistance of the ALN based structure recognition system to noise. To stop the operation, the key Stop (Zaustavi) needs to be pressed, and for quitting the programme, the key Quit (Kraj). The network has been trained with the following parameters:

- tree size: 2048
- number of decision trees: 3
- number of images for each structure: 200
- max. number of training set presentations: 30
- minimum correctness: 99.90%
- maximum turn angle: 5

By changing the training parameters the network properties can be significantly changed. However, there is no mathematical method as yet which would determine the parameters best suited to a certain problem, so that parameters are selected according to experimental experience.

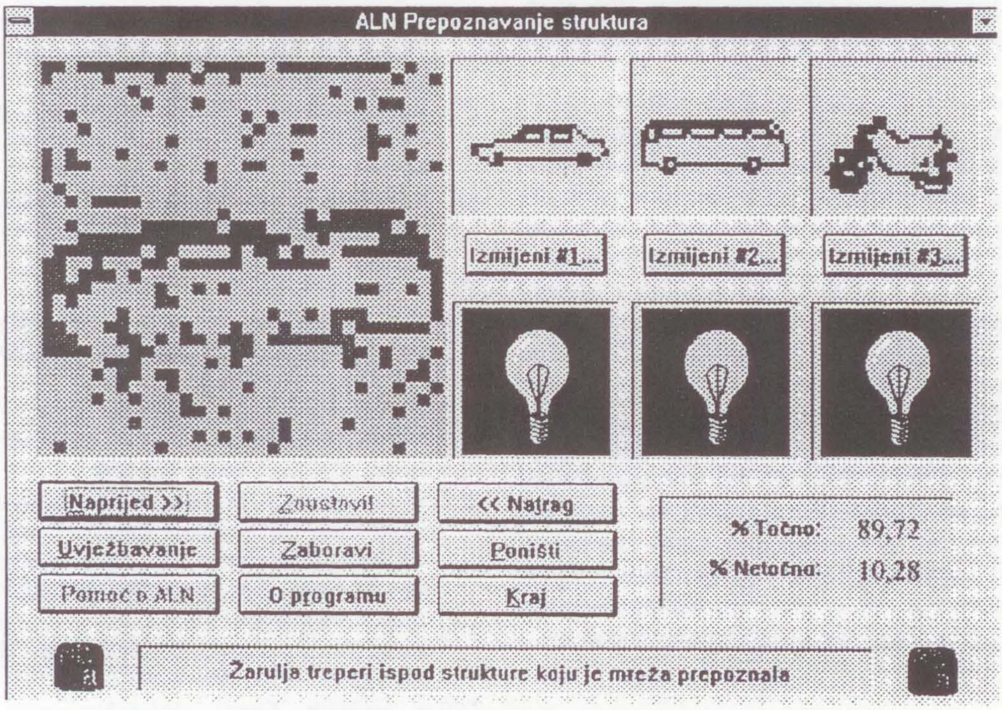


Figure 10 - The window display of the object image affected by noise

5. CONCLUSION

The neural networks whose input uses images recorded from above or from the front of the vehicle, operate in a similar way, but converge more quickly than those that use the side image. This can be explained by the simpler structure which needs to be recognised, i.e. lesser correlation between the object images.

The conventional solution of counting the vehicles, by the most widespread method of induction loop, provides reliable counting of vehicles but indicates a number of limitations [14]. The problem with this approach is that it does not provide any other data on the vehicle, except that the vehicle has passed a control point. The second problem that should not be neglected, is the complexity of installing and maintaining such a system. Moreover, there is the problem of not being able to recognise several lanes. Therefore, if there is a traffic jam on one lane, this information will be lacking because no data are available on the direction of the traffic flow. The application of neural networks could solve the mentioned problems.

By combining several cameras installed in a sequence, data could be obtained on the direction of the traffic, thus providing a timely reaction on those road sections which get unexpectedly overloaded.

When selecting the neural network model the possibility of relatively simple and cost-effective realisation of the selected model needs to be taken into consideration. Furthermore, the selected network should meet the requirements of recognising the vehicles even under conditions of reduced visibility. This problem can be solved by using sensors of several different types.

Our choice is the adaptive logic network, ALN, which, apart from all the mentioned properties, showed high tolerance to shot noise which is to be expected. The same model of ALN can be used in the buildings protection system, access control, and recognition of spotted objects for space guiding purposes.

Furthermore, the possibility for non-linear systems modelling using ALN with input pre-coding should be studied, in order to avoid the limitations of the conventional methods which turn non-linear systems into systems of the second or third order.

Beside all the doubts that may arise regarding operating safety of the neural systems, with suitable design and analysis of the acquired knowledge, fuzzy and ALN systems can be applied in a great number of fields which require high safety of operation [15]. By selecting ALN, the risk of error in making decisions can be reduced to the level of conventional systems.

SAŽETAK

STRUKTURA NEURONSKOG SUSTAVA ZA PREPOZNAVANJE I BROJANJE PROMETALA U SPECIFIČNIM UVJETIMA PLIVAJUĆEG MOSTA

Predstavljeni su rezultati istraživanja struktura sofisticirnog sustava za prepoznavanje i brojenje prometala temeljenog na uporabi neuronske mreže. Istaknuta je realizibilnost i kvaliteta modela višeslojnog perceptrona kao temeljne strukture neuronske mreže za prepoznavanje objekta. Predstavljaju se opće mogućnosti i temelji uporabe a posebice se ističe i diskutira jednostavnost sklopovske realizacije simuliranog modela adaptivne logičke mreže. Uporabom alata atree 2.7. uvježbanom strukturom sustava provodi se simulacija prepoznavanja vozila uz visoku razinu sačmastog šuma slike sa otklonjenom pozadinom. Predložena struktura neuronskog sustava temeljenog na uporabi adaptivne logičke mreže brzo konvergira, zadovoljava postavljene kriterije točnosti prepoznavanja i pokazuje se realizibilnom za rad sustava u realnom vremenu i otežanim uvjetima.

LITERATURE

[1] **I. Deković**, *Prepoznavanje oblika temeljeno na neizrazitoj logici i neuralnim strukturama*. Magistarski rad, FESB, Sveučilište u Splitu, 1994.

[2] **W.S. McCulloch and W. Pitts**, *"A logical calculus of the ideas immanent in neurons activity"*. Bulletin of Mathematical Biophysics, Nr. 5, 1943, pp. 115-133.

[3] **F. Rosenblatt**, *Principles of Neurodynamics*. Washington DC: Spartan Books, 1962.

[4] **B, Widrow**, *Generalization and information Storage in networks of adaline "neurons" In Self-Organizing Systems* pp. 435-461, W.DC Sparta 1962.

[5] **S. Haykin**, *Neural Networks a comprehensive Foundation*, IEEE Computer Society Press, New York 1994.

[6] **J. Hertz, A. Krogh, R.g. Palmer**, *"Introduction to the Theory of Neural Computation"*, Addison-Wesley, Reading, Mass., 1991.

[7] **R.P. Lippmann**, *"An introduction to Computing with Neural Nets"*, IEEE ASSP Magazine, April 1987, pp. 42.

[8] **A. K. Jain, J. Mao**, *"Artificial Neural Networks: A Tutorial"*, IEEE Computer Magazine, March 1996, pp. 31-44.

[9] **W. W. Armstrong** atree 2.7.1995.

[10] **W. W. Armstrong, J. Gecsei**, *Adaption algorithms for Binary Tree Networks*, IEEE Trans. on Systems, Man and Cybernetics, 9, 1979, pp. 276-285.

[11] ftp funic.funet.fi atree.Z., 1996.

[12] Dendronic Decisions Limited, 3624 - 108 Street, Edmond, Alberta, Canada T6J 1B4,

[13] **W. Armstrong, J.-D. Liang, D. Lin, S. Reynolds**, *Experiments Using Parsimonious Adaptive Logic*, Tech. Rept. TR 90-30, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, T6G 2H1,

[14] **Dean A. Pomerlan**, *"Neural Networks for Intelligent Vehicles"*, Neural Networks Applications, IEE Technology update series, 1996, p.p. 927-932.

[15] **Š. Matošin**, *Teorija pontonskog mosta u morskim uvjetima*, Ceste i mostovi, Nr. 5-6, 1995, str. 135-156.