A Cooperative Approach To Integrating Expert Systems with Neural Networks

Pavol Frič, Jozef Šefčík and Vladimír Vojtek

Department of Computer Science and Engineering, Faculty of Electrical Engineering, Bratislava, Slovakia

Integrating expert systems and other information and knowledge processing technologies, e.g. neural network approach, represents a great challenge for the knowledge-based systems technology itself. This paper presents a brief introduction into knowledge-based systems technology state of the art and highlights also the basics of the neural networks technology. The problem solving aspects and principles of both mentioned technologies are compared and discussed more in detail. Proposed expert systems and neural networks integration method is described in the paper. A short discussion on results achieved is provided also in the paper.

Keywords: expert systems, neural networks, knowledge representation, cooperation

1. Motivation

Knowledge-based systems technology was reported to have achieved a broad practical applicability. A large number of knowledge-based systems applications characterised by significant variability in problem domains developed during the last years was also reported. Hence the technology of knowledge-based systems is considered to have reached its limits. These applicability limits are mostly represented by currently applied inference schemes and knowledge representation principles. Further qualitative evolution of knowledge-based technology can be seen only in integration with other knowledge and information processing technologies. Therefore the integration effort becomes more apparent in the current research directions.

Most of up to date reported integration activities are oriented towards integrating knowledge-based systems with database systems. Such interconnection is already commercially available in several environments for development of knowledge-based systems. In the presented paper we will focus our attention on integration of knowledge-based systems with other technologies, namely with neural networks.

Neural network represents an alternative approach to knowledge representation and utilisation in the problem solving process. In the paper we will discuss the differences between both, the mentioned knowledge representation and knowledge processing approaches, stressing the problem solving and knowledge representation aspect. We will analyse the possibilities of integrating knowledge-based systems with neural networks and we will highlight possible consequences of such integration for the further development and applicability of both technologies.

Expert systems and neural networks in the past were considered to be different domains for scientific research. There has been done not so much in exploring the possibilities of their interaction and cooperation in the process of cooperative problem solving. As it is apparent from the description of both the technologies given below, in many aspects, expert systems and neural networks can be seen complementary in a sense, that drawbacks in one approach can be compensated by the advantages of the composite approach. In the presented paper we discuss an experiment in integrating expert system with neural network to participate in a cooperative solution of a common problem.

2. Knowledge representation using the Expert systems approach

Expert system is a typical representative of the knowledge-based systems technology approach that is based on utilisation of explicit knowledge. Expert systems from the historical view represent the first approach to modelling the human expert behaviour within the problem solving process. The applicability of expert systems was from their early beginning restricted to clearly defined problem domains, where specific approaches to knowledge representation and reasoning can be defined and applied.

Expert systems are capable of handling and exploiting explicit knowledge using symbolic manipulation to simulate expert reasoning. Expert systems can be classified according to knowledge and reasoning schemes applied into two categories: the first generation expert systems and the second generation expert systems. **The first generation expert systems** were based on the massive usage of heuristic knowledge represented in most cases by the rule formalism. The method of heuristic classification was widely used as the problem solving scheme. Heuristic knowledge provides the interconnection of observable characteristics (symptoms) with the solutions available in the knowledge base. This type of knowledge is reported to be the so-called shallow knowledge. A variety of models for handling uncertainity in data and represented knowledge has also been developed to support the heuristic classification approach.

The second generation expert systems are based on the more sophisticated knowledge structures and reasoning schemes. They rely on massive exploitation of causal knowledge, which can be combined with heuristic knowledge used by the first generation expert systems. Such knowledge is referred to as the so-called deep knowledge. The second generation expert systems are able to deal with meta-knowledge, i.e. the knowledge how to handle the knowledge represented in the knowledge base.

Several knowledge representation formalisms are used in providing a framework for representing the required knowledge in the knowledge base. **Frame formalism** [Minsky:1975] and its extensions are used in representing stereotypical situations, e.g. representing structures of knowledge belonging to the same object/entity in the problem domain. **Scripts** [Schank:1976] can be used for representing knowledge about actions in known situations. **Semantic networks** are applicable in representing relations between entities in the problem domain. **Rule formalism** [Brownston:1985] is widely used for representing heuristical knowledge of situation-action type.

The applicability of expert systems is restricted to problem domains, where the knowledge and the reasoning performed can be captured by structures and inference principles of specific knowledge representation formalism, or the combination of selected knowledge representation formalisms. Problem domains, where processing of incomplete and noisy information is required, real-time applications, or problem domains that have to be resistent to failure or damage of part of the processing system, are risky and not very suitable for using the expert system technology.

Expert systems reasoning process is inherently sequential. The problem of the reasoning efficiency arises, when the knowledge base grows over a certain limit. Even the parallelisation of inference engines and the distribution of the problem solving process is the subject of intensive research, the basic philosophy of expert systems reasoning still remains sequential. Therefore, no qualitative breakthrough will probably arise.

When limits of expert systems technology are reached, another problem solving technology has to be applied, or has to be integrated with knowledge-based systems technology. The neural network approach sems to be very promising.

3. Knowledge representation using the Neural networks approach

Neural networks are based on the model massive parallel distributed processing, which is believed to be the model of human brain information processing. The main idea applied in neural networks approach to divide available knowledge into many independent "knowledge islands" and to precisely define the way of interaction between them. Information processing is considered to be achieved by the interaction of a huge amount of simple processing elements (neurons), interconnected by connections of excitation and inhibition types (e.g. [Rumelhart:1986], [McClelland:1986]).

Every processing element (neuron) in the neural network is described by its internal state or activity, that is dependent on processing element inputs and its internal characteristics. Processing elements can represent fragments of captured knowledge, i.e. they can be used as short-term or long-term memory. Short-term memory is defined by internal state (or activation) of processing element — it represents the status of the neural network in the process of problem solving (analogy of expert systems working memory can be seen).

Long-term memory is defined by weighed activation interconnections of processing elements contained in the neural network. Such weighted interconnection represents a fragment of knowledge incorporated in the neural network. This knowledge (i.e. weighed interconnections) can be entered explicitly by the "knowledge engineer", or it can be generated in the neural network learning process.

The ability of improving the own problem solving behavior (i.e. learning from own experience) is an important advantage of neural networks over expert systems. Knowledge acquisition for expert systems is a long-term process, where the knowledge engineer has to acquire and explicitly state the knowledge that has to be incorporated in the knowledge base.

Neural networks have a capability of own behavior self improving. The knowledge acquisition and transfer process can be simplified to the selection of the set of learning patterns. These patterns are described by inputs with corresponding outputs and can be used for adjusting the weights of processing element interconnections, for creating new interconnections, or removing the existing interconnections. The goal of such learning process is to achieve equivalence between required neural network behavior and its real behavior.

Learning methods can be classified into two categories: supervised learning methods and unsupervised learning methods. **Supervised learning** is a process that incorporates an external teacher and/or global information. This technique also incorporates the decision when to stop the learning process, the decision about frequecy and duration of the learning process based on learning patterns, processing of error information. **Unsupervised learning** (self-organization process) is a process that incorporates no external teacher. Unsupervised learning self-organizes presented data and discovers its emergent collective properties. Important characteristics is the convergence and stability of the learning process.

There are two basic approaches to knowledge representation in neural networks: local and distributed. In the local approach every knowledge fragment is represented by a single processing element [Feldman:1986]. In the distributed representation each fragment of knowledge is represented by a pattern of activity distributed through a set of processing elements, where each processing element may be a component of more activity patterns.

Several models are for reasoning available in neural networks. The **Interactive Activation and Competition Model** (IAC) is based on the existence of more processing elements organised into a number of competitive pools. Interconnections between the processing elements in a pool are of inhibition (exclusive) type, interconnections between the processing elements belonging to different pools are of excitation (supporting) type. The competition is based on lowering the activation of other, less activated processing elements in the same pool, and activating the related elements in other pools.

The IAC network can be seen as distributed knowledge base, where processing elements in each pool represent exclusive attributes. IAC network can be considered to be an associative memory, which identifies the "most similar" known pattern to a given situation to be recognised.

Linear associative memory (LAM) is a two-layer, heteroassociative, interpolative pattern matcher that learns offline, operates in discrete time. Modifications of LAM (e.g. Optimal linear associative memory) are useful as storage medium, they can also be used as a novelty filter that provides a dimension-by-dimension comparison of an input vector to all stored vectors. Self-organizing feature maps [Kohonen:1981] automatically determines the k-best reference vectors from a sufficiently large set of data points in n-dimension.

Backpropagation Neural Network (BNN) consists of several processing elements (neurons) layers. Basic layers in this model are the input layer and the output layer. These layers can be separated by more internal (hidden) layers. Interconnections are allowed only between elements on neighbouring levels, neurons on the same layer are not interconnected. Backpropagation network is able to learn, using the set of selected inputs and outputs (supervised learning).

Neural networks can be easily implemented on sequential machines by software simulations. Conventional computers can virtually implement any neural network. The only problem is the long duration of software simulation needed for a large parallel system.

The practical applicability of neural networks is very wide. Suitable problem domains can be identified by the usage of parallel processing of large amount of information, dealing with incomplete and noisy information. Learning paradigm is also an important argument for applying the neural network approach. From the user point of view, the explanation and validation of results generated by a neural network is not sufficient. Hence, other knowledge-based approaches can be applied for this purpose.

4. An approach for integrating Expert systems and Neural networks

The idea of integrating expert systems with neural networks is not new. There are several approaches already reported that deal with the problem of integration of expert systems with neural networks, e.g. [Kasabov:1992], [Ultsch:1992]. As the knowledge representation and reasoning principles of neural networks and expert systems differs significantly, no tight integration at the level of basic reasoning and knowledge representation elements (e.g. neurons, frames, rules) is possible. The only possible approach for this integration is to enhance the capabilities of expert system by adding a module or more modules to neural network. Such extension can be accomplished according to the problem solving paradigm applied in the reasoning process. Several possibilities are available:

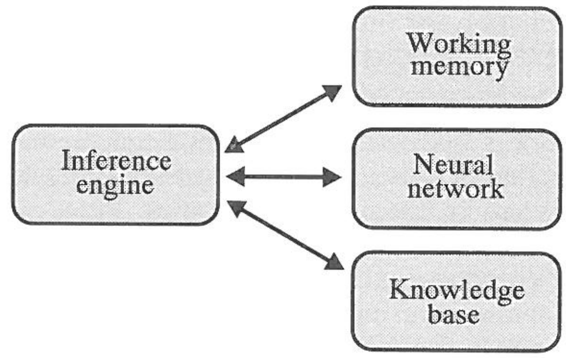

Fig. 1.

– to extend the capabilities of expert system by adding a separate neural network module explicitly activated from the expert system's inference engine (Fig. 1). Such approach is applicable, when there exists an explicitly stated subproblem that can not be solved by using techniques provided by expert systems technology, or where the neural network approach is more suitable (e.g. by replacing a large amount of rules representing required knowledge). Such subproblem, if identified in the problem solving process, can be solved by explicit activation of the neural network, where the results obtained by such activation can be utilised in further expert system reasoning.

– to incorporate the neural network in a set of cooperating agents within the blackboard architecture. In the above approach a neural network module can be added to a set of already existing knowledge sources (Fig. 3), or it can be used as extension of the module controlling the activation of knowledge sources (Fig. 2). When extending the control module, more flexible conflict resolution strategy can be represented and obtained. When incorporating a new knowledge source, additional problem solving capabilities can be provided to the system.

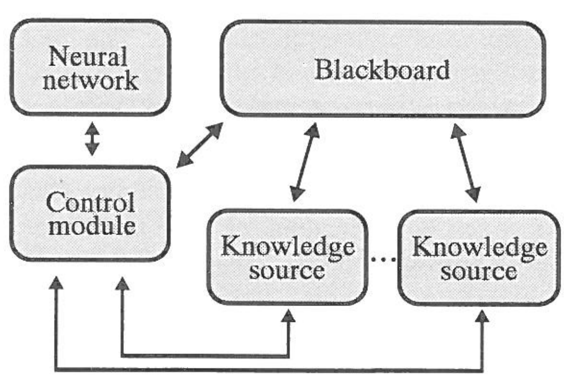

Fig. 2.

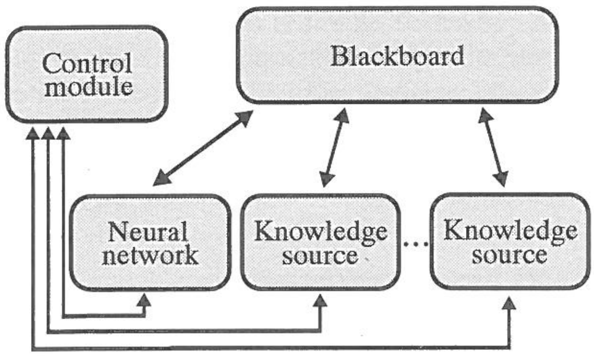

Fig. 3.

The integration approach illustrated in Fig.1 is the simplest one. We decided to verify the cooperation paradigm using this model of cooperation, since our development environment enables to incorparate this paradigm without spending much effort. In our future work we plan to implement an environment allowing cooperative integration of expert systems and neural networks, based on ideas illustrated in Fig.2 and Fig.3.

In the experiment of implementation of an integrated expert system — neural network environment we enhanced an existing experimental expert system environment KEX, that was developed at the Dept. of Comput. Sci. and Eng. at Slovak Technical University [Bielikova:1992]. The KEX knowledge representation formalism is based on rules and frames. For the development of a prototype of expert system enhanced with neural network we utilised the highly modular KEX environment architecture and flexible knowledge representation formalism. Due to the above features, KEX provided a cost-effective approach to the planned enhancement of an existing expert system development shell. For the above purpose, KEX architecture was enriched by a new separate neural network module.

As we intented only to demonstrate the value and practical applicability of incorporating neural network within an expert system and to study the principles of such integration, we did not find any use of implementing all available neural network types. For the purpose of experiments described later in this paper, we have implemented only the backpropagation model.

When extending the expert system with special neural network module activated explicitly by the inference engine, the interface and activation principles have to be precisely defined. As the knowledge representation formalism KEX/L provided no support for interfacing neural networks, a special class representing neural network characteristics was introduced. The mentioned neural network class enables the definition of attributes describing specific features of a particular neural network, incorporated in the problem solving environment. Defined neural network characteristics to be represented by class attributes are the following:

- INPUTS – an attribute that represents neural network inputs. As neural network may have more inputs, this attribute references a frame, the attributes of which represent particular inputs of the neural network.
- OUTPUTS – an attribute that represents neural network outputs. As neural network may have more outputs, this attribute references a frame, the attributes of which represent particular outputs of the neural network.
- WEIGHTS – an attribute that references the external file where neural network parameters are stored.
- NEURAL_NETWORK_TYPE – an attribute defining the type of neural network. According to neural network type, a specific procedure is applied when the network is activated.

The above class provides a general framework for interfacing neural networks. When developing a specific application using a particular neural network, a frame representing the particular neural network activation has to be defined, i.e. created as a neural network class instance. Such approach enables us to incorporate, within a particular application, more neural networks, that can be activated separately. This idea of incorporating several distinct neural networks in one problem solving environment raises the flexibility and applicability of the proposed integrated environment.

The incorporation of neural network in the problem solving process can be achieved by using various approaches. Neural network activation can be achieved through demons bounded with frame attributes values, or by explicit activation represented by an action incorporated in the expert system. Immediately before the neural network module is activated, attributes representing particular neural network inputs have to have values assigned. For this purpose, demons can be associated with attributes representing neural network inputs. Demons can be used for representation of knowledge of the pre-activation phase, the way of acquiring and preparing required data for neural network activation and for monitoring the defined value range of input data.

The control of the problem solving process is transferred to the neural network module. Immediately after its activation, it accesses the input values and network characteristics. According to neural network characteristics the module is parametrised for computing output values for the particular neural network activation. The output values are transferred to expert system's working memory, i.e. the attribute representing neural network outputs. The problem solving process control is given back to the expert system.

When assigning values to attributes representing neural network, outputs can be activated. These demons can be utilised for checking output values for integrity and consistency. They represent the post-activation reasoning knowledge: e.g. the way of incorporating the activation results for the purpose of further reasoning. Output values are imported into the working memory and reasoning proceeds in the expert system.

Neural network activation is achieved by a specific action contained in the set of KEX actions. The process control tranfer is highlighted by following text describing the implementation of such action in PROLOG:

```
action_neural_network_activation(Filename,
InputFrame, OutputFrame):-
   take_parameters(Filename, InputFrame, Inputs,
   Weights, Err),
         % acquiring neural network parameters, input
         data and connection
         % weights
   ifthen(not Err,
      !,
      activate_propagation(Inputs, Weights, Outputs),
         % activating the propagation of neural network
         inputs to neural
         % network outputs
      insert_output(Outputs, OutputFrame),
         % exporting ouput data to expert system working
         memory – attribute
         % values
      insert_data_for_explanation(Filename, Inputs,
      Outputs)
         % saving data to protocol
   ).
```

In the presented example the InputFrame and OutputFrame represent inputs and outputs for the neural network activation. These frames can be accessed and processed by the expert system. The structure of the mentioned frames is illustrated by the following example:

```
FRAME NeuralNetworkActivation1
  INSTANCE_OF NeuralNetworkActivation
  ATTRIBUTE Inputs
    VALUE InputFrame1
  ATTRIBUTE Outputs
    VALUE OutputFrame1
  ATTRIBUTE Weights
    VALUE confuguration_file
  ATTRIBUTE Neural_Network_Type
    VALUE backpropagation
FRAME InputFrame1
  INSTANCE_OF InputFrame
  ATTRIBUTE Input1
    VALUE InputValue1
  ATTRIBUTE Input2
    VALUE InputValue2
  ...
FRAME OutputFrame1
  ATTRIBUTE Output1
    VALUE OutputValue1
  ATTRIBUTE Output2
    VALUE OutputValue2
  ...
```

First implementation of the proposed extension of the KEX environment, i.e. the neural network module, was accomplished in Arity/Prolog, the same language as the KEX environment is implemented in. For efficiency reasons, a neural network learning module was implemented as a separate part of KEX environment in C language. Such approach allowed a very fast implementation of the neural network extension and its incorporation into the existing environment. We are aware of low computational efficiency of the neural network module, therefore reimplementation in C language is planned in the near future.

For the examples described below, the neural network is implemented as a backpropagation network with two intermediate layers. The learnig process for the presented examples was based on gradient learnig method. Correction of weighed interconection values was calculated for each input pattern. The structure of the main learning cycle can be schematically described by the following algorithm:

```
repeat
  OK := true;
  for ii := 1 to NumberOfPaterns do begin
    PaternError := PropagatePatern(ii);
    if AcceptableError < PatternError then begin
      BackPropagatePatern(ii);
      OK := false;
    end;
  end;
  Increment(NumberOfCycle);
until OK or NumberOfCycle > MaximalNumberOfCy-
cles;
```

MaximalNumberOfCycles is used to secure the finite number of learning cycle executions in the case of having an inconsistent set of input learning patterns.

In the first implementation we have not tackled the problem of explanation of the reasoning process performed by the neural network. Explanation of neural network activation is a separate problem. Currently, we have incorporated into the explanation engine only the explanation of the activation state of the neural network and the transfer of results back to the expert system environment. It would be possible to explain also the states and results of intermediate levels, but until now we cannot give any meaningful interpretation and utilisation of presented data. The above problem is a great challenge for the future and will require further experiments.

5. Results achieved

When demonstrating the applicability of the presented approach we have selected a set of problem domains, upon which we validated the proposed approach. Two main applications have been developed so far:

- prediction of possible behavior of a known person when reacting on a joke concerning himself
- prediction of the influence of surrounding working environment on human creative feelings

Behavior prediction

Human behavior is a complex activity that is influenced by a large number of internal and external factors. After having consulted this problem with experts — a group of psychologist, we proposed a problem solving model that has two levels of reasoning: identification of personality type (introvert, extrovert, sanguinic, choleric) and adjusting the expected behavior according to a particlular situation and psychical state (context) of the person being observed.

There are more psychological tests currently available to identify the personality type. This problem is sufficiently solved by psychology. According to a set of acquired personality characteristics and knowledge incorporated in the knowledge base, expert system can solve this diagnostic problem by using the model of heuristic classification. Frame formalism and rule formalism are suitable for representing this kind of required knowledge and providing the problem supporting solving scheme.

The complex human behavior cannot be described by such simple stereotypes. It can be significantly influenced by other characteristics, like current psychical state of the person being observed, problems in the family and psychical stress etc. We have identified a huge amount of such characteristics in the knowledge acquisition process. Even after detailed sessions with experts, integrative influence of all characteristics on the actual behavior could not be described unambigiously and consistently. Psychology does not provide any model for simple solution to this problem. Reasoning based on incomplete knowledge and information has to be applied.

We have proposed to represent the acquired knowledge by using a huge set of rules. Each rule should represent one of the observed heuristics. Soon, serious problems arised:

- interaction among individual rules due to the problem complexity cannot be foreseen
- there was no way of being sure that all relationships between characteristics and behavior were captured

At the end we had to revise our decision to represent this type of knowledge by rule formalism. We decided to solve the problem of adjusting the predicted behavior by applying the neural network approach and by making use of the self learning capability of neural network.

The input to the neural network is defined as identified personality type and available information about the current personality background (stress, problems etc.). All identified characteristics were defined as neural network inputs. The outputs from the neural network were defined as possible behavior adjustments. These outputs are used in the expert system to adjust the predicted personality type behavior.

For knowledge acquisition and representation we utilised the learning algorithm of the backpropagation model. For the first experiment about 100 learning patterns were provided. Even using such limited set of patterns, the results achieved were sufficient enough for the demonstration purpose. Resulting neural network activated by expert system was able to adjust the predicted behavior according to additional information and was able to provide rationale prediction for the reaction of a particular person in the situation being modelled. The results achieved were in about 90% coincidence with the results predicted by experts cooperating in the developed application.

Interior design evaluation

Another application of cooperative expert system — neural network integration being developed — is the application for assistance in interior design and evaluation. The purpose of the system is to assist in designing the working place interiers to initiate and support creative feelings. This project was a common research with a team of psychologist and its goal is to implement an expert module as a component in the support system design. The problem is analogous to the example described before — the use of neural network is necessary to combine a large number of characteristics and their influence on basic characteristics of the analysed working environment.

The way of expert system — neural network cooperation — was very similar to the previous experiment. The expert system initialised the problem solving process by acquiring all characteristics of the room interior. It performed the first combination of characteristics, resulting in hypothesis about the influence of each interior component. Such a set of hypotheses provides the activation of information for the neural network. The integration of these hypotheses into a plausible evaluation was accomplished by the neural network. Expert system then generated an explanation of the resulting evaluation.

The evaluation system was built as a separate module. The development team of psychologist was impressed by the results achieved during a relatively short time period — 3 months. Even working with a limited set of learning patterns (about 350), the behavior of the developed system reached the abilities of a common psychologist.

6. Conclusions

In the work described in this paper we verified the usefulness and practical applicability of integration of expert system and neural network technology for cooperative solving of problems. We have implemented a simple extension of existing expert system environment and highlighted its capability to face problems, that are beyond the capabilities of current expert systems technology.

We are aware that this is only a first step in the integration process. Its applicability and usefulness for solving problems and the reasoning process is evident. In the work that has been done so far we have identified the following problems to be solved:

- more models of neural networks to support the problem solving process have to be implemented and combined in the problem solving process
- the explanation of neural network activation requires more elaboration, where new explanation principles specific to neural networks have to be proposed
- the cooperative approach seems to be more powerful, so we plan to implement a distributed environment that will allow the integration of neural network as an independent knowledge source and its incorporation into the distributed knowledge-based environment
- to elaborate the approach, where neural network is the driving element of the reasoning

process. In examples mentioned so far, expert system was the main reasoning element, neural network was considered to be only some extension of expert system.

References

ANDERSON J. A., ROSENFELD E. (1988) Neurocomputing: Foundations of Research, MIT Press, Cambridge, Mass.

BIELIKOVA M., FRIČ P., GALBAVY M., VOJTEK V. (1992) KEX: An Environment for Development of Expert Systems, Proc. of the *"Information Technology Interface" ITI'92*, Pula, pp. 153–158

BROWNSTON L., FARRELL R., KANT E., MARTIN N. (1985) Programming Expert Systems in OPS5: An Introduction to Rule–Based Programming, Addison–Wesley

FELDMAN J. A. (1986): Neural Representation of Conceptual Knowledge, Tech.Re,. TR189, Dept. of Comput.Sci., Univ. of Rochester, Rochester NY

HINTON G. E., MCCLELLAND J. L., RUMELHART D. E. (1986) Distributed Representations, in Rumelhart D.E., McClelland J.L. (eds.): *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol 1: Foundations*, MIT Press, Cambridge, Mass.

KASABOV N. K., PESTOK S. (1992) Neural Networks and Logic Programming: Hybrid Models and its Applicability to Building Expert Systems, *Proc. of the ECAI'92 Conference*, pp. 287–288

KOHONEN T. (1981) Automatic Formation of Topological Maps in Self-Reorganizing System, in Oja E. Simula O (eds.) Proc. of 2–nd Scandinavian Conf. on Image Analysis, pp. 214–220

MCCLELLAND J. L., RUMELHART D. E. (1986) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, *Vol 2: Psychological and Biological Models*, MIT Press, Cambridge, Mass.

MINSKY M. (1975) A Framework for Representing Knowledge, in P.H.Winston: *The Psychology of Computer Vision, McGraw–Hill*, pp. 211–277

RUMELHART D. E., MCCLALLAND J. L. (1986) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, *Vol 1: Foundations*, MIT Press, Cambridge, Mass.

SCHANK R. C. (1976) The Role of Memory in Language Processing, in C.N.Cofer (ed.): *The Structure of Human Memory*

SIMPSON P. K. (1990) Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations, Pergamon Press, San Diego

ULTSCH A. (1992) Self-Organizing Neural Network for Knowledge Acquisition, *Proc. of the ECAI'92 Conference*, pp. 208–210

Received: February, 1994
Accepted: June, 1994

Contact address:
Department of Computer Science and Engineering
Faculty of Electrical Engineering
Slovak Technical University
Ilkovičova 3, 812 19 Bratislava, Slovakia

PAVOL FRIČ received the M.S. and Ph.D. degree in computer science from the Slovak Technical University in Bratislava. From 1986 he is with the Dept. of Comput. Sci. and Eng. at the Slovak Technical university as Assistant Professor. His research areas include artificial intelligence, expert systems and software engineering.

JOZEF ŠEFČIK received the M.S. in computer science from the Slovak Technical University Bratislava. Currently he is a Ph.D. student at the Dept. of Comput. Sci. and Eng. at the Slovak Technical University. He is involved in research projects concerning neural networks and expert systems.

VLADIMÍR VOJTEK Vladimir Vojtek received the M.S. and Ph.D. degree in computer science from the Slovak Technical University in Bratislava. From 1980 he is with the Dept. of Computer Sci. and Eng. at the Slovak Technical University, currently as Professor. His research interests are software engineering, artificial intelligence and neural networks.