MODULAR NEURO-FUZZY NETWORKS: SOLLUTIONS FOR EXPLICIT AND IMPLICIT KNOWLEDGE INTEGRATION

Ciprian-Daniel NEAGU

Department of Computer Science & Eng., "Dunarea de Jos" University of Galati

Abstract: In this paper we propose a unified approach for integrating implicit and explicit knowledge in neurosymbolic systems as a combination of neural and neuro-fuzzy modules. In the developed hybrid system, training data set is used for building neuro-fuzzy modules, and represents implicit domain knowledge. The explicit domain knowledge on the other hand is represented by fuzzy rules, which are directly mapped into equivalent neural structures. The aim of this approach is to improve the abilities of modular neural structures, which are based on incomplete learning data sets, since the knowledge acquired from human experts is taken into account for adapting the general neural architecture. Three methods to combine the explicit and implicit knowledge modules are proposed.

Keywords: neural and neuro-fuzzy integration, modular structure.

1. INTRODUCTION

In recent years, the hybrid neural systems have drawn increasing research interest. This approach has been successfully used in various areas, such as speech/natural language understanding, robotics, medical diagnosis, fault diagnosis of industrial equipment, and financial applications (Becraft et al., 1991; Lin and Lee, 1991; Kosko, 1992; Rocha, 1992; Takagi, 1994; Sima and Cervenka, 1997; Palade, 1999; Wermter and Sun, 2000). The reason for studying hybrid neural systems is based on successful applications of subsymbolic knowledge-based systems, particularly the neuro-fuzzy networks, as well as on the advantages of the symbolic knowledge-based systems. In the hybrid systems, connectionist tools can be interpreted as hardware, and fuzzy logic as software implementation of human reasoning. The modular structure of connectionist implementations of explicit and implicit knowledge can be interpreted as a homogenous inductive and deductive learning and reasoning system.

The fundamental concepts and methods used in our approach are described subsequently: the formalism of the neural fuzzy model MAPI (Rocha 1991; 1992). The context of combining implicit and explicit knowledge in a connectionist implementation is introduced, and three specific methods are presented, based on fuzzy operators, supervised or unsupervised gating networks. Additionally, the steps required in the process of building the proposed system called NEIKES (Neural Explicit and Implicit Knowledge-based Expert System) are described.

The implicit knowledge is defined as a connectionist module-based representation of learning data. The explicit knowledge module of the hybrid system is implemented as a special connectionist structure using hybrid fuzzy neural networks. This kind of implementation is proposed to adjust the performances of implicit knowledge modules. The paper is ended with some conclusions on the subject of neuro-fuzzy approach.

This paper was recomended for publication by V. Mînzu

2. STATE OF THE ART IN NEUROSYMBOLIC INTEGRATION

The last ten years have produced an explosion in the amount of research on both, symbolic and connectionist fields. Symbolic processing is considered as a traditional way in Artificial Intelligence. In connectionist systems, unlike symbolic models, learning plays a central role. The directions of research being done under the banner of these approaches explore both, high-level connectionism (applied to natural language processing or commonsense reasoning (Sun, 1991, 1994) and hybrid systems (as engineering oriented points of view (Khosla and Dillon, 1997; Pal and Mitra, 1992). Connectionist models are powerful tools to process knowledge, so they have been used to build connectionist intelligent systems, mainly for perceptual tasks, where discovering explicit rules does not seem either natural, or direct. Like human beings, the connectionist models rely on learning low-level tasks. Learning by example is not a general solution: it is known that many situations are solved by intelligent entities using explicit rules.

The two approaches can be used in complementary way. This is the premise of the hybrid intelligent systems, which combine connectionist and symbolic features. In such systems, the learner first inserts symbolic information of some sort into a neural network: the learner must use prior knowledge in order to perform well. Once the domain knowledge is put in a neural representation, training examples are used to refine the initial knowledge. Finally, it process the output for a given instance and, using some specific methods (Benitez et al., 1996; 1997)) (Neagu and Palade, 2000; Omlin and Giles, 1996; Palade, 1999), extracts symbolic information from the trained network in order to give some explanations on the computed output and improve understanding of the refined connectionist knowledge. Building hybrid intelligent systems requires an exploration of all these approaches.

The most important effort in such situation is focused to the homogenous implementation of all these methods into connectionist structures, in order to use their capabilities to well perform and provide accurate conclusions using often incomplete and noisy data. In this paper, some approaches to represent, in a neural manner, external fuzzy rules, explicitly acquired from human experts, are proposed. These rules are combined with trained neural structures, built using data sets of the same application domain, in order to improve accuracy of the output. Complex tasks may give raise to local minima; subsequently the “learning by example” paradigm is useful mostly in simple tasks. Our proposed answer to this problem is to combine different connectionist modules solving various subtasks of the main problem.

The connectionist integration of explicit knowledge and learning by example appears to be a natural solution of developing connectionist intelligent systems. The problem to be solved is the uniformity of integration. In order to encourage modularization, explicit and implicit rules should be represented in a neural manner using fuzzy neural networks (FNN, (Buckley and Hayashi, 1995; Fuller, 1999), hybrid neural networks (HNN, Buckley and Hayashi, 1995; Fuller, 1999), and, in a particular approach, standard neural networks (MLP - multilayer perceptron-based structures (Rumelhart, and McClelland, 1986). While fuzzy logic provides the inference mechanism under cognitive uncertainty, neural networks offer the advantages of learning, adaptation, fault-tolerance, parallelism and generalization (Fuller, 1999). The computational process involved in implicit and explicit knowledge acquisition and representation is described next. First definition of a “fuzzy neuron” is provided, based on the understanding of biological neuronal structure, followed by learning mechanisms (for implicit knowledge representation as combined standard and/or fuzzy neural networks), respective fuzzy rule mapping mechanisms (for explicit knowledge representation as hybrid neural networks). This leads to the three steps in a neural computational process (Fuller, 1999): development of fuzzy neural models, models of synaptic connections, which incorporate fuzziness into neural network, and application of learning, respective mapping algorithms to adjust the synaptic weights. The system taken into consideration in the next sections is a multi-input single-output fuzzy system (MISO).

3. KNOWLEDGE REPRESENTATION

The MAPI neuron, proposed by Rocha (Rocha, 1991; Pedrycz and Rocha, 1993; Rocha, 1992), is a useful tool to add another level of programmability in fuzzy reasoning. The combinations of generalized fuzzy computation, the MAPI model and distributed architecture proposed in the next sections, are used as a powerful neurosymbolic processing tool.

3.1. The Implicit Knowledge Module

We define the *implicit knowledge* as the knowledge represented by neural networks, which are created and adapted by a learning algorithm. The representation of implicit knowledge is based on the numerical weights of the connections between neurons. Assuring to integrate implicit knowledge and explicit given fuzzy rules into the same global network, the trained neural networks are hybrid/fuzzy neural networks. A classical MLP would be used to improve the behavior of the implicit knowledge modules (IKMs), as an alternative way to extract fuzzy rules (Benitez et al., 1996; 1997; Neagu and Palade, 1999) from training data sets, and to compare the overall performance of the neurosymbolic system with common approaches.

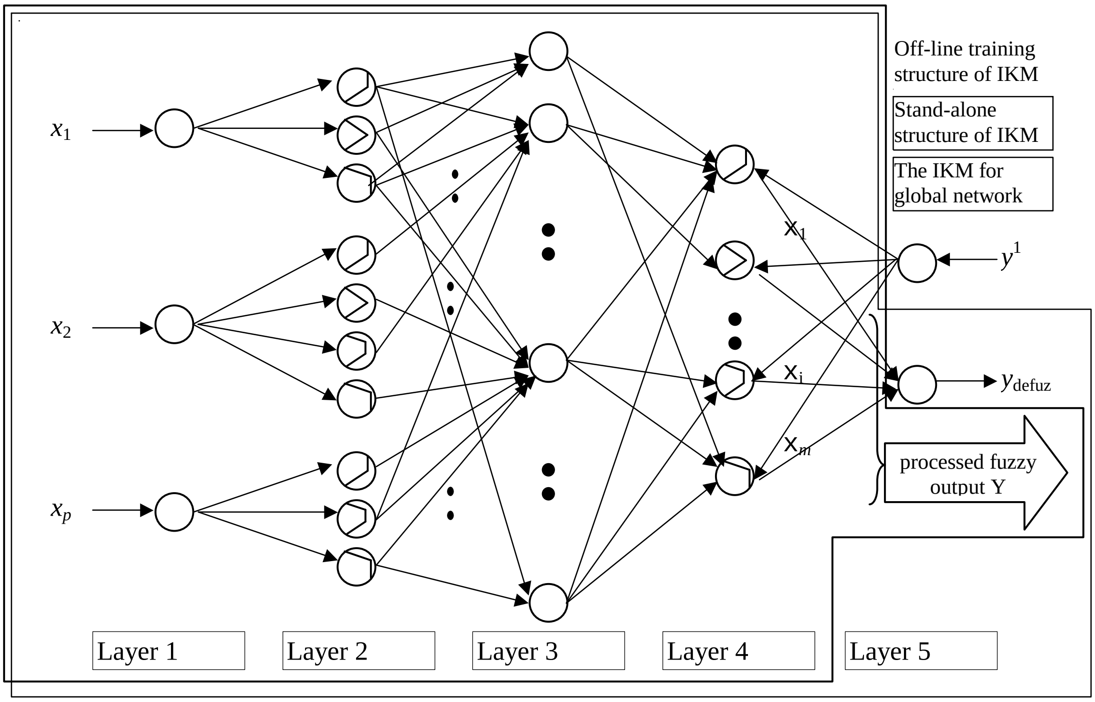

Fig.1. Implicit Knowledge Module implemented as FNN2/HNN.

The IKM is implemented as a multilayered neural structure based on an input layer establishing the inputs to perform the membership degrees of the current values, a fully connected three-layered FNN2 (Fuller, 1999), and a defuzzification layer (figure. 1).

The FNN of type 2 (FNN2) implements IF-THEN fuzzy rules, and is characterized by fuzzy inputs and outputs, and crisp weights. The input nodes of the FNN2 are proposed to be MAPI neurons, parameterized such as to implement given membership functions of the term set of each linguistic input. As a generalized approach, in (Jang and Sun, 1993; Lin and Lee, 1991) the authors proposed similar structures in which the objective is to approximate the shape of membership functions. The objective of the FNN2 as IKM is to learn the fuzzy rules and to implement the dependences between the linguistic output and the fuzzified inputs.

3.2. The Explicit Knowledge Module

We define the *explicit knowledge* as a knowledge base represented by neural networks, which are computationally identical to a fuzzy rules set, and are created by mapping the given fuzzy rules into hybrid neural networks. The fuzzy rule set is described as a discrete fuzzy rule-based system DFRBS (Buckley and Hayashi, 1995). The intrinsic representation of explicit knowledge is based on fuzzy neurons in a MAPI implementation. The numerical weights corresponding to the connections between neurons are computed using Combine Rules First Method: (Buckley and Hayashi, 1995), or Fire Each Rule Method (Buckley and Hayashi, 1995; Fuller, 1999).

4. IMPLICIT AND EXPLICIT KNOWLEDGE INTEGRATION

The introduction of the modular networks into fuzzy systems provides new insights into the integration of explicit and implicit knowledge in a connectionist representation. The modular network is a connectionist architecture that allows each module to exhibit its own "opinion" about the entries, in order to classify or predict the output. Thus, a modular network offers several advantages over a single neural network in terms of learning speed, generalization and representation capabilities (Haykin, 1994; Jacobs et al., 1991; Langari, 1993).

Hence, the idea to represent explicit and implicit knowledge in a connectionist manner is based on the concept of modularity (Haykin, 1994; Jacobs et al., 1991). Modularity may be viewed as a manifestation of the "divide and conquer" principle, which let us solve complex computational tasks by dividing the problem into simpler subtasks and then combining their individual solutions. We used the modular network concept to integrate explicit and implicit knowledge formally defined as follows (a definition adapted from (Hashem, 1997): *A neural network is said to be modular if the computation performed by the network can be decomposed into two or more modules that operate on inputs of the main problem without communicating with each other. The outputs of the modules are mediated by an integrating unit that is not permitted to feed information back to the modules. In particular, the integrating unit decides how the outputs of the modules should be combined to form the final output of the system.*

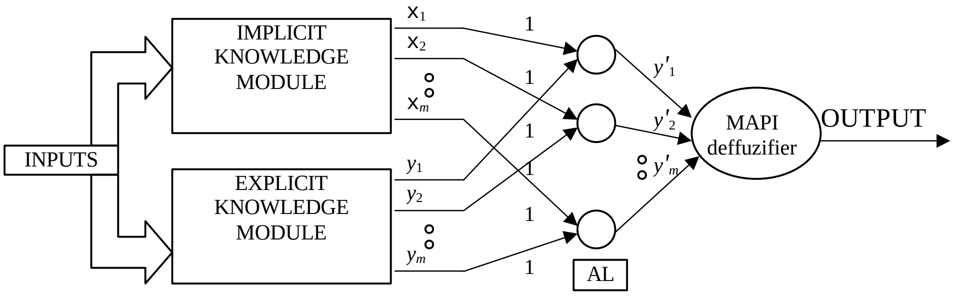

Fig.2. Integration of explicit and implicit knowledge modules in the global network according to FEM strategy.

The main approaches of learning paradigms are involved in a modular network (Hashem, 1997): unsupervised learning allows modules to compete with each other to produce the output, while supervised learning is using an external teacher that supplies the desired target patterns to train different modules.

The global network (GN) of our approach is a modular structure including two different “points of view” about the same problem: the implicit knowledge, implemented by the trained neuro-fuzzy network, and the explicit knowledge, represented by a collection of special neural networks equivalent to some rules proposed by human experts (Neagu and Palade, 1999). First module (IKM) is responsible for generalization, and processing noisy cases, using implicit knowledge, achieved through learning from examples. The second one (EKM) is developed in a top-down manner, using the methods of mapping available explicit rules in hybrid neural structures.

Architectures based on cooperating connectionist modules are proposed to solve integration of explicit and implicit knowledge. In this section, we propose three strategies to combine IKM and EKM in order to build a global hybrid system: Fire Each Module (FEM), Unsupervised-trained Gating Network (UGN), and Supervised-trained Gating Network (SGN). The first strategy is an adapted Fire Each Rule method (Buckley and Hayashi, 1995) in the context of modular networks. The second strategy proposed a competitive-based aggregation of the EKM and IKM outputs, while the third strategy uses a supervised trained layer to process the overall output of the modules.

4.1. Fire Each Module Strategy

The proposed FEM strategy is the simplest mode to integrate IKM and EKM with fuzzy output. A general approach form of this modular structure is proposed in (Neagu and Palade, 1999) and shown in (figure 2). After off-line training phase applied to implicit neuro-fuzzy module, the general output of the system is composed as a T-conorm (Zadeh, 1983) of fuzzy outputs of each module: the four-layered IKM structure for global network and the EKM (implemented using combine rules first or fire each rule method). The system is viewed as equivalent to a set of given fuzzy rules: the overall output is computed using firing (implicit and explicit) rules first method (Buckley and Hayashi, 1995; Fuller, 1999). The method of combining the specific membership degrees provided by both, IKM structure (χ_i values, i=1,2,...,m), and EKM structure (y_i values, i=1,2,...,m), would be done component wise so let:

$$(1)\ y'_i=\text{T-conorm}(\chi_i,y_i),\ i=1,2,...,m,$$

for some aggregating operator, in particular the max fuzzy operator. In the hidden aggregative layer (AL), all the weights are set to one, and the neurons aggregate the specific computed membership degrees χ_i and y_i as implicit, respective explicit opinion about the current output to be described with B_i-th fuzzy term (where the terms set describing the output is $B=\{B_1,...,B_i,...,B_m\}$). Practically, the inputs for the MAPI deffuzifier describe the shape of the fuzzy output. The final neuron is a MAPI device, which computes the crisp value of the output, using, for example, the center of gravity method.

The methodology proposed to build the global network architecture is based on obtaining fuzzy rules, describing the system, which are obtained from a human expert or from a set of examples of its behavior. The methodology consists of:

1. Identification of input and output linguistic variables. The variables are represented by fuzzy sets that are mapped in MAPI units.

2. The IKM is built and train as a five-layered fuzzy neural network (the off-line training structure of IKM).

3. From the hidden part of the IKM, the most relevant rules are extracted, using Relative Rule Strength method, Effect Measure Method, Causal Index Method.

4. We construct a set of possible explicit rules in a given problem with the help of a human expert, using both, external rules and those already extracted at step 3, as the most voted and trusty dependencies

between inputs and output. All these rules are mapped into EKM. Some explicit rules could have just a part of identified inputs in the rule premise, represented as active neurons, while the rest of the input neurons will be set as inactive.

5. The four-layered IKM, and the EKM structures (without the deffuzifier MAPI final neuron) are embedded into the architecture described in figure 2, for which the combining hidden layer AL and the MAPI-based deffuzifier are adapted.

6. After an incremental loop sequence based on steps 2 to 5 (which could be used as a knowledge acquisition procedure), the global network is ready to be used as a classifier or prediction tool.

The incremental loop sequence consisting of steps 2 to 5 could be refined on the basis of combining already given fuzzy rules and training data set as follows. IKM is designed by mapping some external fuzzy rules in the hidden HNN, which further learning with training samples is based on. This way the knowledge is kept at the sub-symbolic level. The main goal of the approach is not just to reduce training period, but also to improve the generalization abilities of the network. The disadvantages consist in both, redistribution of symbolic a priori knowledge (or at least building haloes of initial rules), and necessity of a new refinement of final incorporated knowledge in the resulted network. This strategy follows the variations of *concept support techniques* (Prem et al., 1993; Wermter and Sun, 2000), parameterized by the method used to insert a priori knowledge:

(a) Inserting some rules describing a subset of cases of desired input-output mapping, and learning the training samples (inserted explicit rules play the role of a complement of the training sets in supplying knowledge to the network).

(b) Inserting the symbolic concepts believed to be relevant for the problem solution and training by supporting the relevant concepts.

(c) Inserting explicit rules as in (b), followed by a training phase, in which the used hidden units are different from those designed in first phase.

4.2. Unsupervised-trained Gating Network Strategy

The proposed structure is based on the modular networks paradigm, by considering that the basic configuration consists of two general types of networks: expert networks (implemented by EKM and neuro-fuzzy IKM) and a gating network. A classical modular network considers expert networks competing to learn the training patterns, and the gating network mediating the competition (Haykin, 1994; Jacobs et al., 1991; Langari, 1993). The proposed modular architecture uses neural explicit and implicit knowledge modules, and the gating network for voting the best combination of fuzzy terms computed by expert networks, in order to describe the linguistic output (figure 3).

The EKM and IKM structures are developed and, respectively, trained. The gating network is also trained, with the constrain to have as many output neurons as there are fuzzy terms chosen to describe the linguistic variable Y as the output of global network. The specific membership degrees provided by both, IKM structure (x_i values, i=1,2,...,m), and EKM structure (y_i values, i=1,2,...,m), are aggregated according to the equation (1) by MAPI-based neurons implementing MAX T-conorm (aggregation layer AL). The goal of the learning algorithm for the gating network is to model the distribution of the membership degrees computed by EKM and IKM.

The gating network consists of a single layer of m output neurons (Hashem, 1997), each one having m inputs. The activation function of its output neurons is a *softmax* transformation (Bridle, 1990). The process of gating network training considers that, for each vector $[x'_1, \ldots, x'_m]$ processed by AL, the activation g_i of the i^{th} output neuron is related to the weighted sum of the inputs applied to that neuron. Consequently, the activations of the output neurons in gating network are nonnegative and sum to one:

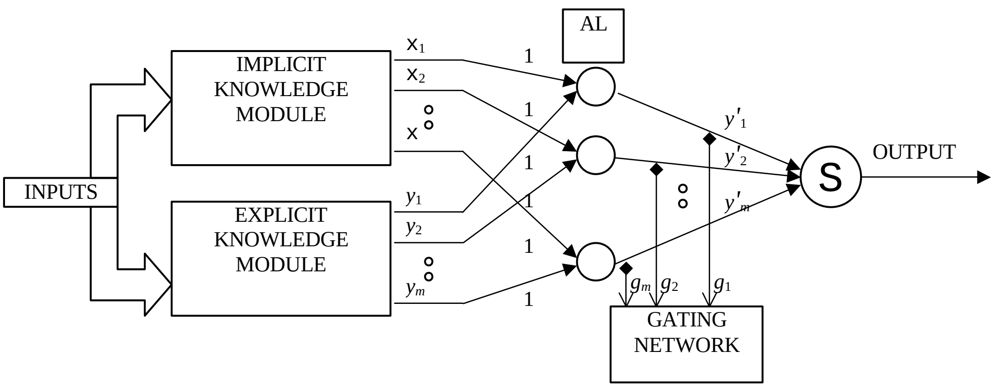

Fig.3. Integration of explicit and implicit modules using an unsupervised-trained gating network (UGN strategy).

$$(2)\ \ 0 \le g_i \le 1 \ \text{and} \ \sum_{i=1}^{m} g_i = 1 .$$

The additional advantage gained by using the gating network is the implicit defuzzification of the overall output of the system:

Proposition 2: Let [x_1, ... , x_p] be the current input of the system and [y'_1, ... , y'_m] be the current output of the aggregation layer and let consider the gating network already trained using the unsupervised *compet* algorithm (Hagan and Beale, 1996). Then the overall output y of the UGN system, computed by a *softmax* transformation is a crisp value representing the defuzzified output of the model.

Proof: Let's consider that the output of the system is computed in respect with the Sugeno model (the consequent part of each rule is described by a linear regression model, (Sugeno and Kang, 1988; Takagi, 1994):

(3) R_i: IF X_1 is A_{i1} AND X_2 is A_{i2} AND ... AND X_p is A_{ip} THEN $y'_i = \sum_{j=1}^{p} b_{ij} x_j$,

where A_{i1}, A_{i2},..., A_{ip} are fuzzy sets having associated matching functions μ_{Ai1}, μ_{Ai2},..., μ_{Aip}, b_{ij} are real-valued parameters, y'_i is the local output of the model due to rule R_i, i=1,2,...,m. The total output of the Sugeno model is a crisp value defined by the weighted average:

$$(4)\ \ y = \frac{\sum_{i=1}^{m} h_i y'_i}{\sum_{i=1}^{m} h_i} .$$

The weight h_i implies the overall truth value of the premise of rule R_i for current input, and is calculated as:

$$(5)\ \ h_i = (\mu_{Ai1}(x_1) \wedge \mu_{Ai2}(x_2) \wedge ... \wedge \mu_{Aip}(x_p)),$$

where ^ is a conjunctive T-norm. The output y described in equation (31) is a crisp value.

Let's now consider the input vector [x_1, ... , x_p] applied to the system described in figure 3. Then, the output of the entire architecture is:

$$(6)\ \ y = \sum_{i=1}^{m} g_i y'_i .$$

The expressions of the current output of Sugeno model and proposed structure are similar: each rule in the Sugeno model could be considered an explicit rule into EKM, or a particular way involving one hidden neuron through the IKM, while the relative weight of the i^{th} neuron in the gating network is:

$$(7)\ \ gi = \frac{h_i}{\sum_{i=1}^{m} h_i} .$$

In essence, the gating network, proposed to combine the outputs of the aggregating layer, acts as a special defuzzifier.

The methodology proposed to build the global network architecture is partially similar to the FEM methodology, and consists of:

1. Steps 1 to 4 are similar to those described for the FEM strategy.

2. The four-layered IKM and the EKM already described structures (without the deffuzifier MAPI final neuron) are embedded into the architecture described in the figure 3, for which the combining hidden layer AL and the deffuzifier MAPI-based unit are adapted.

3. The gating network is trained (*compet* algorithm) using the AL outputs computed for the training data set of the system.

4. After an incremental loop sequence based on the first step (which could be considered as a knowledge acquisition procedure), the global network is ready to be used as a classifier or prediction tool: the final crisp value of the output is computed using the gating network based on the *softmax* transformation, as described in equation (6).

4.3. Supervised-trained Gating Network Strategy

The proposed structure contains expert networks represented by a defined number of EKMs and IKMs solving various sub-problems of the main task, and a supervised trained network mediating their outputs' combination. EKMs represent explicit rules, identified by expert, or refined from a previous knowledge acquisition phase; IKM structures are useful in the overall architecture, because of their generalization and processing noisy data abilities.

After training, different expert networks compute different functions, each of them mapping different regions of the input space. Each defuzzified output of expert networks is considered an input for the final layer. The supervised training process of the final network assures, in fact, a weighted aggregation of expert networks' outputs with respect to their specialization (figure 4).

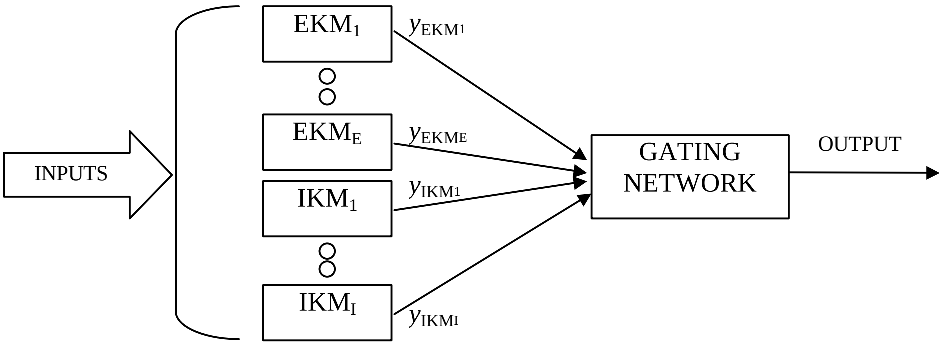

Fig.4. Integration of explicit and implicit modules using a supervised-trained gating network (SGN strategy).

The methodology proposed to build the global network architecture using a supervised-trained gating module consists of the following steps:

1. Identification of input and output linguistic variables. The variables are represented by fuzzy sets that are mapped in MAPI units. The IKM modules are represented as HNN and/or MLP networks. We build and train each IKM as a five-layered fuzzy neural network or as a MLP-based structure, in order to assure for each one a crisp specific output.

2. The most relevant rules are extracted from the IKM structures, using Relative Rule Strength method, Effect Measure Method or Causal Index Method in the case of HNN implementation, respectively using interactive fuzzy operators in the case of MLP implementation.

3. We construct a set of possible explicit rules in the given problem with the help of a human expert, using both, external rules and those already extracted at step 3, as the most voted and trusty dependencies between inputs and output. We map each rule into a specific EKM structure. A MAPI-based defuzzifier as a final layer completes all the EKM structures in order to assure a crisp output for each module.

4. The IKM and EKM structures are embedded into the global architecture (figure 4).

5. The gating network is supervised trained using the EKM and IKM computed outputs, such as the overall network is a combination of expert modules.

6. After an incremental loop sequence based on first four steps (which can be considered as a knowledge acquisition procedure), the global network is ready to be used as a classifier or prediction tool. The final crisp value of the output is computed using the gating network which acts as a classifier of the best combination of each expert network's behavior, describing the overall output for a given vector of input data.

5. CONCLUSIONS AND FUTURE WORK

The proposed structures and methods argue the use of connectionist systems in symbolic processing. Since the presented EKMs were demonstrated to be identical to Discrete Fuzzy Rule-based Systems (Rocha, 1992; Buckley and Hayashi, 1995), the homogenous integration of explicit rules and training data sets permits better cover of the problem domain. In that case, the constraint of the size of neural networks is solved by modularity paradigm. EKMs represent explicit rules identified by expert or refined from IKM structures; IKMs are useful especially for such complex problems described by (noisy) data sets.

The EKM and IKM combination encourages compact solutions for problems described by both, data sets distributed in compact domains in the hyperspace, and isolated data, situated in intersection of compact sub-domains or inhomogeneous intervals. After training, different expert networks compute different functions mapping different regions of the input space.

The different sources of the information explicitly and/or implicitly integrated in the presented modules exhibit the problem of knowledge redundancy in the final structure. The proposed methods, based on explicit and implicit module integration, can operate with redundant knowledge spread in both ways of representation. This redundancy should be clearly minimized. The method to implement redundancy minimization is based on selecting specific data sets from training collection, which are not suitable to verify the implemented rules. The resulted training data sets describe such domains in the hyperspace, which are not covered by the explicit rules. The main disadvantage is that the IKMs are able to generalize just in their domain.

6. REFERENCES

R. Becraft, P. L. Lee, and R. B. Newell, (1991).“Integration of Neural Networks and Expert Systems for Process Fault Diagnosis”, in Procs. Of the International Joint Conference on Artificial Intelligence, pp. 832-837.

J.M. Benitez, A. Blanco, M. Delgado, and I. Requena (1996). “Neural methods for obtaining fuzzy rules”, Mathware Soft Computing, vol.3, pp.371-382.

J.M. Benitez, J.L. Castro, and I. Requena (1997). “Are Artificial Neural Networks Black Boxes?”, IEEE Transactions on Neural Networks, vol.8, no.5, pp. 1157-1164.

J.S. Bridle (1990). “Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters”, in Advances in Neural Information Processing Systems, edited by D.J. Touretzky, Kaufmann, New York, vol. 2, pp. 211-217.

J.J. Buckley and Y. Hayashi (1995). “Neural nets for fuzzy systems”, Fuzzy Sets and Systems, vol. 71, pp. 265-276.

R. Fuller (1999). Introduction to Neuro-Fuzzy Systems, Advances in Soft Computing Series, Springer-Verlag: Berlin.

M. Hagan, and M. Beale (1996). Neural Network Design, PWS: Boston.

S. Haykin (1994). Neural Networks: A Comprehensive Foundation, IEEE Press: New Jersey.

S. Hashem (1997). “Optimal Linear Combinations of Neural Networks”, Neural Networks, vol. 10(4), pp.599-614.

R.A. Jacobs, M.I. Jordan, A.G. Barto (1991). “Task decomposition through competition in a modular connectionist architecture: the what and where vision tasks”, Cognitive Science, vol.15, pp.219-250.

R.J.S. Jang, and C.T. Sun (1993) “Functional equivalence between radial basis function networks and fuzzy inference systems”, IEEE Transactions on Neural Networks, vol. 4(1), pp.156-159.

R. Khosla, and T. S. Dillon (1997). Engineering Hybrid Multiagent Systems, Kluwer.

B. Kosko (1992). Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs: NJ.

R. Langari (1993). “Synthesis of nonlinear control strategies via fuzzy logic”, in Proc. of American Control Conference, pp.1855-1859.

C.-T. Lin, and C.S.G. Lee (1991). “Neural-Network-based Fuzzy Logic Control and Decision System”, IEEE Trans. On Computers, vol. 40(12), pp. 1320-1336.

C.-D. Neagu, and V. Palade (1999). “Fuzzy Computing in a Multi Purpose Neural Network Implementation”, in Proc. of the Int’l Conf. 6th Fuzzy Days in Dortmund, edited by B. Reusch, Springer Verlag, p. 697-700.

C.-D. Neagu, and V. Palade (2000). “An interactive fuzzy operator used in rule extraction from neural networks”, Neural Networks World Journal special issue.

C.W. Omlin and C.L. Giles (1996). “Extraction of Rules from Discrete-Time Recurrent Neural Networks”, Neural Networks, vol. 9, pp.41-52.

S. K. Pal, A. Mitra (1992). “Multilayer Perceptron, Fuzzy Sets, and Classification”, IEEE Transactions on Neural Networks, vol. 3(5), pp. 683-697.

V. Palade (1999). “Hybrid Intelligent Systems for Process Control”, PhD Dissertation, University "Dunarea de Jos" of Galati.

W. Pedrycz, and A.F. da Rocha (1993). “Fuzzy-Set Based Models of Neurons and Knowledge-Based Networks”, IEEE Transactions on Fuzzy Systems, vol. I(5), pp. 254-266.

E. Prem, M. Mackinger, G. Dorffner, G. Porenta, H. Sochor (1993). “Concept Support as a Method for Programming Neural Networks with Symbolic Knowledge”, TR-93-04, OEFAI.

A.F. da Rocha (1991). “The fuzzy neuron: biology and mathematics”, in Proc. of IFSA'91 Brussels, vol.1, pp. 176-179.

A.F. da Rocha (1992). Neural Nets: A Theory for Brains and Machines, Lecture Notes in Artificial Intelligence, Springer-Verlag.

D.E. Rumelhart, J.L. McClelland (1986). Parallel Distributed Processing, Explanations in the Microstructure of Cognition, MIT Press.

J. Sima, and J. Cervenka (1997). “Neural Knowledge Processing in Expert Systems”, TR no. V-735, Institute of Computer Science, Academy of Sciences of the Czech Republic.

M. Sugeno, G.T. Kang (1988). “Structure identification of fuzzy model”, Fuzzy Sets and Systems, vol. 28, pp.15-33.

R. Sun (1991). “Integrating Rules and Connectionism for Robust Reasoning”, TR no. CS-90-154, Waltham, MA, Brandeis University.

R. Sun (1994). “CONSYDERR: A Two Level Hybrid Architecture for Structuring Knowledge for Commonsense Reasoning”, in Procs. of the First International Symposium on Integrating Knowledge and Neural Heuristics, pp. 32-39, Florida, USA.

H. Takagi (1994). “Cooperative system of neural networks and fuzzy logic and its application to consumer products”, in Industrial Applications of Fuzzy Control and Intelligent Systems, edited by J. Yen, R. Langari, L.A. Zadeh, Van Nostrand Reinhold: NY.

S. Wermter, and R. Sun (2000). Hybrid Neural Systems, Springer Verlag: Heidelberg.

L.A. Zadeh (1983). “The role of fuzzy logic in the management of uncertainty in expert systems”, Fuzzy Sets and Systems, vol. 11(3), pp. 199-227.