Aspects of Integration of Explicit and Implicit Knowledge in Connectionist Expert Systems

Ciprian-Daniel Neagu
Mircea Negoita
Vasile Palade

Department of Applied Informatics
"Dunarea de Jos" University of Galati
Romania
{dneagu, mnegoita, pvasile}@cs.ugal.ro

Abstract

A unified approach for integrating explicit and implicit knowledge in connectionist expert systems is proposed. The explicit knowledge is represented by discrete fuzzy rules, which are directly mapped into an equivalent Multi Purpose Neural Network based on MAPI neuron (Rocha 1992). Learning result is a refinement process of data sets, which is represented in a module (or combination of modules) of classical feedforward structures incorporating implicit fuzzy rules. The combination of explicit and implicit knowledge modules is viewed as an iterative process in knowledge acquisition and refinement.

1. Introduction

The difficulties arising from extracting knowledge from trained neural networks and to integrate implicit (connectionist) knowledge with symbolic expert systems are starting points in the direction of developing neural expert systems. In these systems, connectionist tools are interpreted as hardware and fuzzy logic as software implementation of human reasoning. In that way, a modular structure of connectionist implementations of explicit and implicit knowledge could be interpreted as a homogenous system combining inductive and deductive learning and reasoning.

A family of connectionist tools is developed as a collection of fuzzy processing operators to model logic oriented computation of fuzzy sets. The MAPI neuron, proposed by (Rocha, 1992), is a useful tool to add another level of programmability in fuzzy reasoning. The combinations of generalized fuzzy computation, the expanded MAPI model and specific distributed architecture, are used as a powerful fuzzy-connectionist processing tool.

In this paper, a connectionist model based on the MAPI formal neuron is proposed to represent multi-premises fuzzy rules. In section 2, the formal neuron MAPI and fuzzy operators implemented using MAPI structure are shortly presented. In section 3, the theoretical aspects of mapping fuzzy rules in Multi Purpose Neural Networks (MPNN) used as Hybrid Fuzzy Neural Networks (HFNN) and a generalized structure of MPNN equivalent with a discrete multi-premises fuzzy rule base (DFRB), based on the fuzzy processing abilities of MAPI neuron, are discussed. The section 4 identifies methods of combining explicit and implicit knowledge connectionist modules and emphasizes aspects of knowledge acquisition and redundancy. Results from a financial test application are reported in section 5. The paper is ending with some conclusions and ideas on future work.

2. Multi Purpose Neural Networks

Multi Purpose Neural Networks (introduced by Rocha, 1992) define a formal language to support complex chemical processing involved in different types of learning and reasoning. The capabilities of these connectionist tools to perform fuzzy computing on fuzzy signals define Multi Purpose Neural Networks based on MAPI neuron model (Rocha, 1992) as Hybrid Fuzzy Neural Networks (Buckley and Hayashi, 1995).

2.1. The MAPI Neuron

The Artificial Neuron N was proposed by (Rocha 1992) and developed under bio-chemical evidences of neuro-phisiology, as so called the MAPI neuron:

(1) $N=\{\{W_p\}, W_0, T, R, C, \Theta, \{\alpha,g\}\}$, where:

- $\{W_p\}$ is the family of pre-synaptic inputs conveyed over N by all its n pre-synaptic axons;
- W_0 is the output code of N;
- T is the family of transmitters used by N to exchange messages with other neurons;
- R is the family of receptors to bind the transmitter released by the pre-synaptic neurons. The strength s_i of the synapsis with the i-th pre-synaptic neuron is:

0-7803-5871-6/99/$10.00

(2) $s_i = M(t) \wedge M(r) * \mu(t,r)^a \, v_0$ and the post-synaptic activity v_i is evaluated as (3) $v_i = w_i \circ s_i$. M(t) is the size of functional pool of the transmitter t at the pre-synaptic cell n_i. M(r) is the amount of r available to bind t. $\mu(t,r)$is the affinity of t^r binding. v_0 is the standard electromagnetic variation triggered by this binding. ^, *, ^{a} , ° are T-norms or T-co-norms;

- Θ is the function used to aggregate the actual pre-synaptic activity: (4) $a_n = \overset{n}{\underset{i=1}{\Theta}} (w_i \circ s_i)$;
- $\{\alpha, g\}$ is a family of thresholds and encoding functions defined as:

$$(5)\ w = \begin{cases} w_l, \text{if } a_n < \alpha_1 \\ w_u, \text{if } a_n \geq \alpha_2 \\ g(a_n), \text{otherwise} \end{cases} ;$$

- C is the set of controllers, activated by $t_i \wedge r_i$>> c_i, $t_i \in T_p$, $r_i \in R$, $c_i \in C$. T_p is the set of presynaptic transmitters. Each c_i exercises actions over N itself and over other neurons.

The formal neuron introduced exhibits the capabilities of a multipurpose processing device, since it is able to handle different types of numerical calculations. This is in contrast with the simple processing capability of the classic neuron introduced by McCulloch and Pitts, 1943. The structures presented in section 3 use this neural model in order to perform fuzzy computing by choosing various forms of the parameters given above.

2.2. Fuzzy Logic and MAPI Neuron

The extended version of Modus Ponens is proposed in (Zadeh, 1983):

(6) IF X_1 is A_1 ^ ... ^ X_j is A_j THEN Y is B

$\underline{(X_1 \text{ is } A'_1) \wedge ... \wedge (X_j \text{ is } A'_j)}$

Y is B'

This process is performed in four steps:

- Matching (the compatibility σ between A' and A): in the MAPI case, the t^r coupling measures the matching between the incoming information and the prototypical knowledge encoded in R.
- The aggregation function is a triangular norm or even a non-monotonic aggregation based on geometric mean: (7) $\sigma_a = \overset{n}{\underset{i=1}{\Theta}} (\sigma_i)$.
- Projection: the compatibility σ_c of the consequent is obtained as function of the aggregated value σ_a :

(8) $\sigma_c = g(\sigma_a)$, where σ_c measures the compatibility of (Y is B') with (Y is B).

- Inverse-Matching and Defuzzification: in MAPI context are performed at the axonic terminals. The inverse matching can be described by any triangular norm or co-norm assigned to axonic processing.

2.3. Neural Fuzzy Operators

Arising from the junction of fuzzy sets and neurocomputation (Pedrycz 1993, Pedrycz and Rocha 1993, Buckley and Hayashi 1995), Artificial Neural Networks performing fuzzy computation are heterogeneous parallel structures aimed at logic-based processing of fuzzy data with fuzzy neural functions. The aggregation and projection are performed by generalized aggregative OR and AND neurons. The standard implementation of fuzzy set connectives involves triangular norms or co-norms. Generally:

(9) $y = \overset{j}{\underset{i=1}{\textbf{T-conorm}}} [x_i \ \textbf{T-norm} \ w_i]$.

(10) $y = \overset{j}{\underset{i=1}{\textbf{T-norm}}} [x_i \ \textbf{T-conorm} \ w_i]$,

where x_i and w_i are the inputs and the weights of the MAPI neuron implementing fuzzy operators. The neural implementation is an equivalent structure, which uses the method of combining rules first (Buckley and Hayashi, 1995).

3. Explicit Knowledge Representation Using MPNN

Explicit knowledge representation using connectionist tools is viewed as a subject of mapping explicit rules into MPNN architecture (used as HFNNs). The aim of this strategy is to obtain hybrid fuzzy neural structures, equivalent to Discrete Fuzzy Rule Based Systems.

3.1. Reasoning with a Fuzzy Rule Base

Let be considered a single rule with three antecedents described as:

(11) IF X is A AND Y is B AND U is C THEN Z is D.

A, B, C, D are fuzzy sets having associated matching functions μ_A, μ_B, μ_C, μ_D. Let the matching function $\mu_A(\xi)$ be described by a vector X of size Nx, so that:

(12) $x_i=\mu_A(\xi)$, if $\alpha_i<\xi\leq\alpha_{i+1}$, i=1,2,.., Nx-1.

Thus, the fuzzy set A is:

(13) $A=\begin{bmatrix} x_1 & \cdots & x_{Nx} \end{bmatrix}$.

Similarly, fuzzy sets B, C and D are described in discrete forms as follows:

(14) $B=\begin{bmatrix} y_1 & \cdots & y_{Ny} \end{bmatrix}$, $y_i=\mu_B(\psi)$, if $\beta_i<\psi\leq\beta_{i+1}$, i=1,2,.., Ny-1

(15) $C=\begin{bmatrix} u_1 & \cdots & u_{Nu} \end{bmatrix}$, $u_i=\mu_C(\upsilon)$, if $\gamma_i<\upsilon\leq\gamma_{i+1}$, i=1,2,.., Nu-1

(16) $D=\begin{bmatrix} z_1 & \cdots & z_{Nz} \end{bmatrix}$, $z_i=\mu_D(\zeta)$, if $\delta_i<\zeta\leq\delta_{i+1}$, i=1,2,.., Nz-1.

The fuzzy relation:

(17) $R: A\times B\times C\times D\rightarrow[0,1]$,

(18) $\mu_R(x,y,u,z)=(\mu_A(\xi)\wedge\mu_B(\psi)\wedge\mu_C(\upsilon))\Gamma\mu_D(\zeta)$

defines the implication according to (6), where ^ is a conjunctive T-norm and Γ is an associative T-norm; so that given A', B', C':

(19) $D'=(A'\wedge B'\wedge C')\circ R$, where ° is usually interpreted as max-Γ operator.

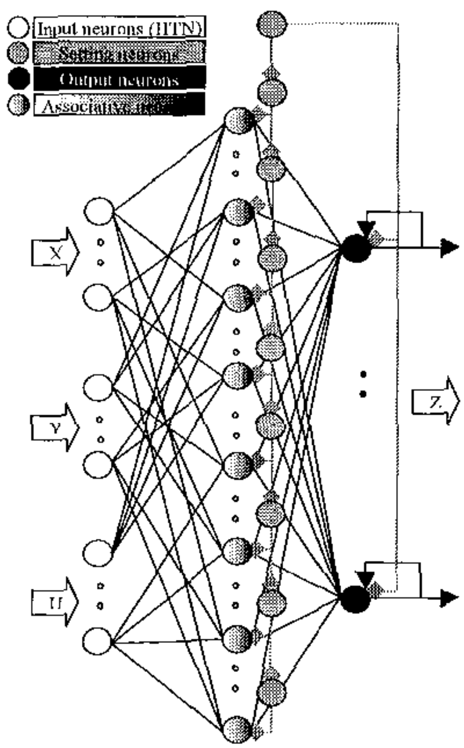


Figure 1. MPNN equivalent to a rule with three premises.

3.2. Mapping Fuzzy Rules into MPNN

The implementation of an explicit multi-premises rule presented in (11) into an equivalent MPNN structure using MAPI neurons with fuzzy abilities (as described above) is shown in figure 1. The vectors are considered of size N_x, N_y, N_z, respectively.

In these conditions, $D'=(A'\wedge B'\wedge C')\circ R$, where inverse matching is implemented by the axonic encoding at the output neurons (Rocha, 1992).

4. Integration of Explicit and Implicit Knowledge in Connectionist Systems

Symbolic processing is considered as a traditional way in Artificial Intelligence; unlike symbolic models, learning plays a central role in connectionist structures. A combination of both approaches is already the subject of research in hybrid systems. Viewing connectionist models as a powerful tool to process knowledge, it is straightforward to try to build connectionist intelligent systems, mainly applied to perceptual tasks, where discovering explicit rules does not seem either natural, or direct. Like humans, connectionist models rely on learning low-level tasks. Learning by example is not a general solution: it cannot be stressed for exhibiting any intelligent behavior. Many situations are solved by intelligent entities using explicit rules.

Complex tasks may give raise to local minima, that it is, learning by example paradigm is useful in small tasks. One answer is to combine different connectionist modules solving various subtasks of a 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: explicit and implicit rules should be represented in a neural manner: in this case, using MPNN as HFNN to encourage the modularization.

Architectures based on cooperating connectionist modules are proposed to solve integration of explicit and implicit knowledge. First module is developed in a top-down manner, using the methods of mapping available explicit rules in neural structures, as described above. The second one is responsible for unmanageable cases of implicit knowledge, achieved using learning by example paradigm. Since the discrete fuzzy inputs and outputs are considered common for all rules three strategies used to combine explicit knowledge module (EKM) and implicit knowledge module (IKM) are identified:

1) EKM is firstly designed by mapping fuzzy rules in MPNN, on which further learning with training samples is based. This way the knowledge is kept at least sub-symbolic. The main goal is not just to reduce training period, but mainly 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) with the difference of 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 learning by supporting the relevant concepts during training.

c) Inserting explicit rules as in 1b followed by a training phase in which the used hidden units are different from them designed in first phase. The resulted structure is exhibited in figure 2.

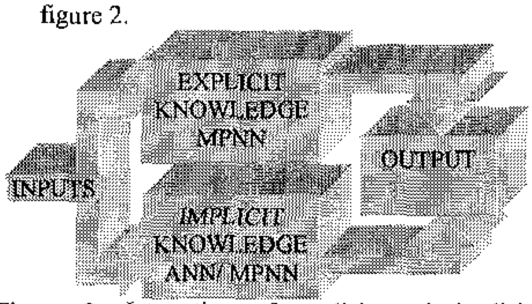


Figure 2. Integration of explicit and implicit knowledge modules in training network.

2) All sub-modules implementing explicit rules are firstly combined in a global module: EKM (a case similar to firing rules first method, as in Buckley and Hayashi, 1995) and a learning by example IKM is generated. The final output is obtained following a process of training a third gating network with two global outputs controlling the outputs of EKM and IKM (figure 3).

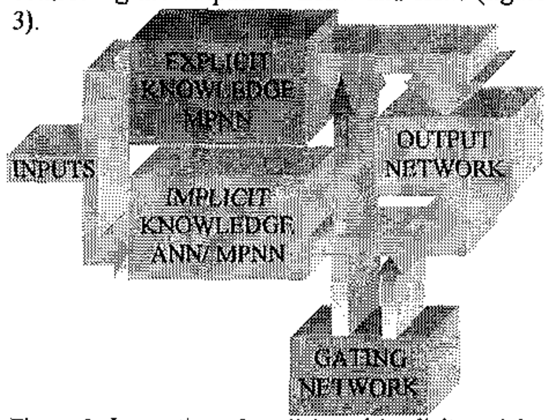


Figure 3. Integration of explicit and implicit modules using a gating network.

The design and training method of the gating network and the aggregative output network follow the principles of hybrid learning presented in (Langari and Wang, 1996).

This method of explicit and implicit knowledge integration is useful in complex problems solving, which are simply breaking down in sub-problems, a major part of them easily defined as symbolic (fuzzy) rules; the IKM is used to generalize and to characterize the domain of solutions explicitly unidentified.

3) The global architecture combines the explicit and implicit sub-modules using a gating network, which mediates the competition of all involved expert networks (figure 4) as a network based on firing different (explicit and implicit) rules first.

The resulted structure is a modular network: expert networks are represented by a defined number of EKMs and IKMs solving various sub-problems of the main task, and the gating network mediates the competition. EKMs represent explicit rules identified by expert or refined from a previous extracting rules from IKM phase; IKMs are useful especially for such complex problems having solutions distributed not only in a compact intervals in hyper-space, but isolated, situated in intersection of compact sub-domains or inhomogeneous intervals. After training, different expert networks compute different functions mapping different regions of input space. The weights introduced to realize the global network are obtained using a competitive algorithm. The number of output neurons of gating network is indicated by the number of expert networks, and the activations of the output neurons must be nonnegative and sum to one (Langari and Wang, 1996).

The structures and methods presented argue the use of connectionist systems in symbolic processing. Since the EKMs presented are demonstrated to be equal to Discrete Fuzzy Rule Systems, the homogenous integration of explicit rules and data training sets comes with the advantage of a well cover of the problem domain. In that case, the constraint of neural networks dimension is solved by modularity paradigm.

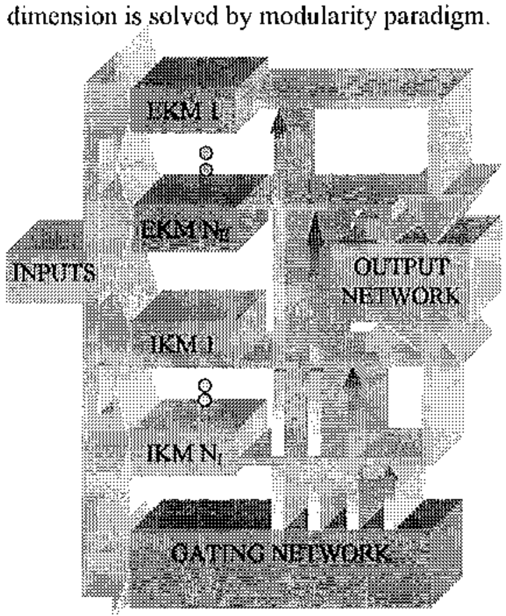


Figure 4. Basic configuration of integrated network based on firing explicit and implicit rules first.

The different sources of the information explicitly and implicitly integrated in presented modules exhibit the problem of knowledge redundancy in the final structure. The integration strategy presented in 4.1) is not suitable for this problem: explicit knowledge is viewed as a starting point of learning by example. The strategy based on concept support indicates the most efficient compact domain in verifying the majority of data patterns. The last two methods, based on explicit and implicit module integration are open to process, without any validations, redundant knowledge. If the application is oriented to identify the answers as clear as possible, it is preferable to minimize the redundancy. The method to implement redundancy minimization is based on selection just the 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 IKMs obtained by learning data are able to generalize just in their domain.

In the situation of a non-validation process of implicit knowledge, the consequences of maximization of the redundancy of data in resulted structure are: the advantage of the possibility to generalize on the full inputs space, the advantage to refine more covering explicit rules from IKM, subjects to define new concentrated EKMs and the disadvantage of a much more complicated structure than is necessary.

5. An Example in the Financial Domain

The capabilities of MPNN to perform fuzzy computing (Rocha, 1992, Rocha and Yager, 1992, Pedrycz and Rocha, 1993) and the methods of integrating explicit and implicit knowledge are tested in a financial application developed to solve a portfolio problem. The neural reasoning engine is accorded to multiple premises fuzzy rules, which model a portfolio evaluation process, taken from the current Romanian financial context.

Three explicit three-premise fuzzy rules were used. The explicit rules describe criteria for the portfolio value to be classified as negative, zero or positive and were obtained by consulting a financial expert. In the application, triangular membership functions are used for primary fuzzy sets. All of these rules cover a limited part of the input space. The knowledge base mapped in the EKM is:

(21) IF (X is LOW_1) AND (Y is $HIGH_2$) AND (U is LOW_3) THEN (Z is PO)

(22) IF (X is MED_1) AND (Y is MED_2) AND (U is MED_3) THEN (Z is ZE)

(23) IF (X is $HIGH_1$) AND (Y is $HIGH_2$) AND (U is $HIGH_3$) THEN (Z is NE)

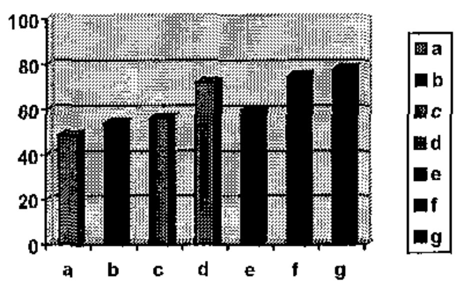
where, for example, X is USD/1000ROL. The data sets are obtained from the currency fluctuation in financial years 1990-1998.

The proposed methods of integrating explicit and implicit rules in a connectionist structure were tested using the same rules and implicit information. The concept supported by the network in the first case was the currency fluctuation differences in an internal political crisis. After training, the following values (Prem et al, 1993) were computed for comparison (figure 5):

✓ the percentage of cases of the whole data set that trained IKM classified correctly;

✓ the percentage of cases of the whole data set that given rules classified correctly;

✓ the percentage of cases of the whole data set the final network classified correctly: cases 1a,b,c, 2, 3.



F

Figure 5. Preliminary results of integrating explicit and implicit knowledge in a connectionist structure. The prediction rate of a regularly trained IKM (value a) is compared to the results obtained by rules connectionist structure EKM (value b), and integrated networks presented in 1a (value c), 1b (value d), 1c (value e), 2 (value f), 3 (value g).

6. Conclusions and Future Work

This paper described a neural approach of fuzzy reasoning using a neural structure based on generalized MAPI neuron, and some strategies to integrate explicit and implicit knowledge in connectionist systems. The developed MPNN structure is equivalent with a discrete multi-premises fuzzy rule base. Using fuzzy capabilities of the MAPI neuron, one of the main advantages of mapping fuzzy rules in MPNN structures is developing of a programmable neuro-fuzzy tool. The raison of implementing fuzzy rules using MPNN consists in both, the necessity of combining explicit and implicit knowledge in a natural manner, and the property of MAPI neurons to be implemented as fuzzy processors (Rocha, 1992, Pedrycz et al, 1995).

Future research will focus on using neural MAPI operators and MPNN structures for knowledge acquisition, in order to develop expert systems with particular applications in the financial area. This aim supposes to clarify the problems of combining neural modules using fuzzy gateway based on MAPI model and to obtain some theoretical reasons for the results obtained using different integration strategies. Also the research will be focused on identifying methods to extract fuzzy rules from trained neural networks and to describe strategies of using neural expert systems in knowledge acquisition, particularly in financial forecasting.

References

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

Langari, R., and Wang, L., (1996), Fuzzy models, modular networks, and hybrid learning, Fuzzy Sets and Systems, 79, pp. 141-150, 1996

Pedrycz, W., (1993), Fuzzy Neural Networks and Neurocomputations, Fuzzy Sets and Systems, 56, pp.1-28, 1993

Pedrycz, W., Rocha, A.F., (1993), Fuzzy Set Based Models of Neurons and Knowledge Based Networks, IEEE Transactions on Fuzzy Systems, Vol.1, no.5, pp.254-266, 1993

Pedrycz, W., Poskar, C.H., Czezowski, P., (1995), A Reconfigurable Fuzzy Neural Network with In-Situ Learning, Technical Report UMECExx3/1995, University of Manitoba

Prem, E., Mackinger, M., Dorffner, G., (1993), Concept Support as a Method for Programming Neural Networks with Symbolic Knowledge, Technical Report OEFAI04/1993, Austrian Research Institute for Artificial Intelligence

Rocha, A.F., (1992), Neural Nets – a Theory for Brains and Machines, Springer-Verlag, 1992

Rocha, A.F., Yager, R.R., (1992), Neural nets and fuzzy logic, in Intelligent Hybrid Systems, CRC Press Inc, 1992

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