Multi-Layer Models and Learning of The Artificial Neural Networks

Stefan L. Koynov[1], Chavdar D. Korsemov[2], Hristo I. Toshev[3] and Leoneed M. Kirilov[4]

Abstract:—.**The multi-layer model is introduced as a pyramidal-type hierarchical model with its properties and limitations. Based on it interpretations are presented concerning different features of the learning process for the artificial neural networks including examples of bad cases for modeling with multi-layer models. The paper proves that multi-layer models provide the most relevant description of these features.**

Keywords—.**artificial neural networks, learning, hierarchical systems, pyramidal hierarchies, multi-layer models.**

I. Introduction

The presented report is a result from a research done by the authors in previous papers [1-3]. It is centered at the learning process of the artificial neural networks (ANN), the goal being the ease of the ANN design process. The theoretical basis for this goal are the hierarchically-based multi-layer models corresponding to the different levels of importance for the layers in the model. The hierarchical models possess features that make them unique in the design process and the multi-layer models can be described mathematically.

The paper consists of three parts.

The first part presents the artificial neural networks (ANN) models methodology that concerns the design and the learning of ANN. It starts with the 'anatomy' of creating a multi-layer model on the basis of pyramidal hierarchies with their basic properties and limitations.

The second part introduces concrete multi-layer models for the learning process. These models interprete the architecture in the ANN learning, the links between the learning paradigms and the learning rules and also the multi-layer models of the relations between the learning rules and the learning algorithms of ANN.

The third part presents the multi-layer-model mathematical formalism based on evolutionary and genetic approaches to multi-layer models [3].

[1] Stefan L. Koynov is with the Institute of Information Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev str., bl. 29A, 1113 Sofia, Bulgaria, E-mail: baruch@iinf.bas.bg

[2] Chavdar D. Korsemov is with the Institute of Information Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev str., bl. 29A, 1113 Sofia, Bulgaria, E-mail: korsemoviinf.bas.bg

[3] Hristo I. Toshev is with the Institute of Information Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev str., bl. 29A, 1113 Sofia, Bulgaria, E-mail: toshev@iinf.bas.bg

[4] Leoneed M. Kirilov is with the Institute of Information Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev str., bl. 29A, 1113 Sofia, Bulgaria, E-mail: lkirilov@iinf.bas.bg

II. The ANN Design Methodology

The design sequence for creating a multi-layer model naturally includes the following steps: 1) a multi-layer model del is guaranteed by an additional and specific information (e. g. the electronic enterprise model is related to the ANN design

This process can greatly be simplified if we introduce the concept of the hierarchies of a pyramidal type with their properties and limitations as it follows:

1) Hierarchical models. The authors state that multi-layer models are hierarchical models in which the layers reflect the direction of an increase or a decrease the degree of importance in some sense or some feature [2-3]. Such models are designed to model complex hierarchies where the levels of different layers present the different hierarchical levels: this is the core of the idea to compare multi-layer models and hierarchies.

Multi-layer models also can be viewed as topographic cross-sections of the horizontal layers in a hierarchical pyramid where every layer is an analog of a horizontal structure in the hierarchy: the lowest layer is the periphery and the highest layer is the core, v. v.

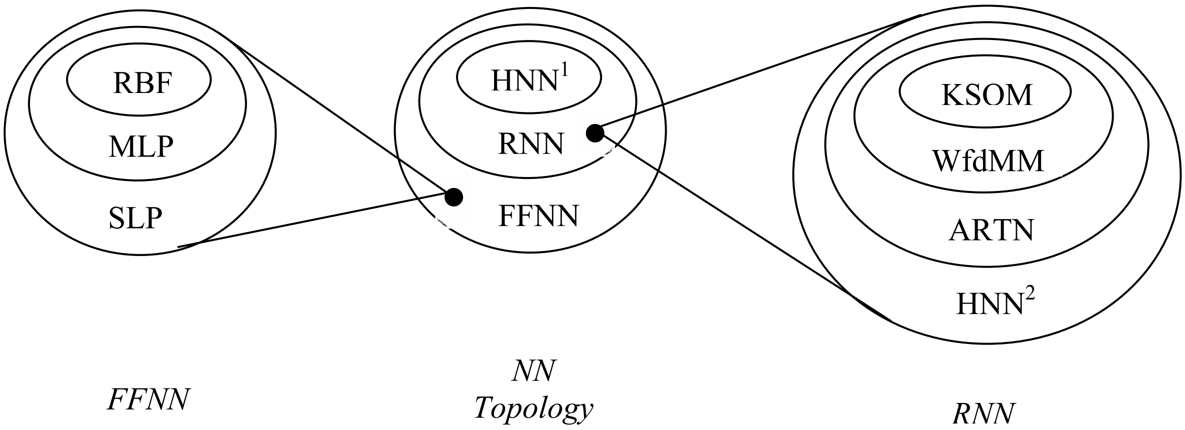
In this way multi-layer models are oriented in a most natural way to model complex pyramidal structures.

The close relationship between multi-layer models and the evolutionary approaches makes such models fit to model different structures in the living creatures.

The great complexity in the design process of ANN makes multi-layer models convenient for effective analyses of such networks.

2) Limitations of multi-layer models. The item cites objects that are inconvenient for multi-layer models because they have no hierarchical or a pyramidal structure:

1. *Tables*.
2. *Unilevel objects in a hierarchical system*.
3. *Matrixes*. Tables represent non-hierarchical structures in a non-mathematical form and matrixes are their mathematical equivalent.
4. *Unidimensional coordinates*. They differ from one another only in their mathematical meanings (numbers) that depend on the number domains. So unidimensional coordinates are objects exclusively of mathematical transforms and they have no distinct hierarchical orderings.
5. *Competitive learning rules* (supervised and unsupervised). They have no definite hierarchical structure with regard to competitive- and ART-network architectures. This explains why they are convenient examples of ineffective multi-layer models (see [4] and Table 1).



Radial Basis Functions (RBF), Multi-layer Perceptron (MLP), Single-Layer Perceptron (SLP), Hybrid Neural Networks (HNN^1), Recurrent Neural Networks (RNN), FeedForward Neural Networks (FFNN), Kohonen's Self-Organizing Map (KSOM), Willshaw - von der Malsburg Map (WfdMM), Adaptive Resonance Theory Network (ARTN), Hebbian Neural Network (HNN^2)

Fig. 1. ANN architectures

It is possible to interpret such non-hierarchical structures with multi-layer models but it can be shown that such interpretations are not effective (they are not included due to limitations from the size of this paper).

III. Multi-Layer Models and Learning of ANN

The following presentation reveals the basic learning aspects for the ANN like their architectures, learning paradigms, learning rules and learning algorithms. The architecture 'pulls' the learning rule and the learning algorithm that are to a greater extent inherent to the idea of the ANN learning process; the learning rules and the learning algorithms are most directly linked with the applied mathematical formalism and their interpretation with multi-layer models has its specific features and peculiarities.

The ANN design starts with the selection of some standard prototype of the ANN architecture which may be upgraded in some way or other or even it may be combined with some other standard ANN architecture. Following [4] Fig. 1 shows different multi-layer models of the feedforward and recurrent ANN and also their place in the overall architectural model of the ANN.

Chapter II proves that the learning rule is the first and most important feature in the ANN learning process since it is most closely connected with the learning paradigm. According to the authors this fact explains why the models of the links between the paradigms and the rules in Fig. 2 resemble to a great extent the scheme in Fig. 1 with the ANN architectures. Both figures present *star-like* multi-layer models because small areas in the central model are expanded to separate multi-layer models.

This is not the case in Fig. 3 which is not a star-like multi-layer model. According to some authors (e. g. [4]) the ANN architectures occupy the place between the learning rules and the learning algorithms. This is a possible explanation for the absence of links between the different multi-layer models in Fig. 3.

Let us return to chapter II, item 2 which postulates that tables are not convenient objects for modelling with multi-layer models and also to item 3 which states that cases with competitive learning rule – supervised and unsupervised do not possess distinct hierarchies with regard to competitive and ART-network architectures. These concepts are illustrated by the following below Table I (LVQ stands for Learning Vector Quantization and VQ for Vector Quantization).

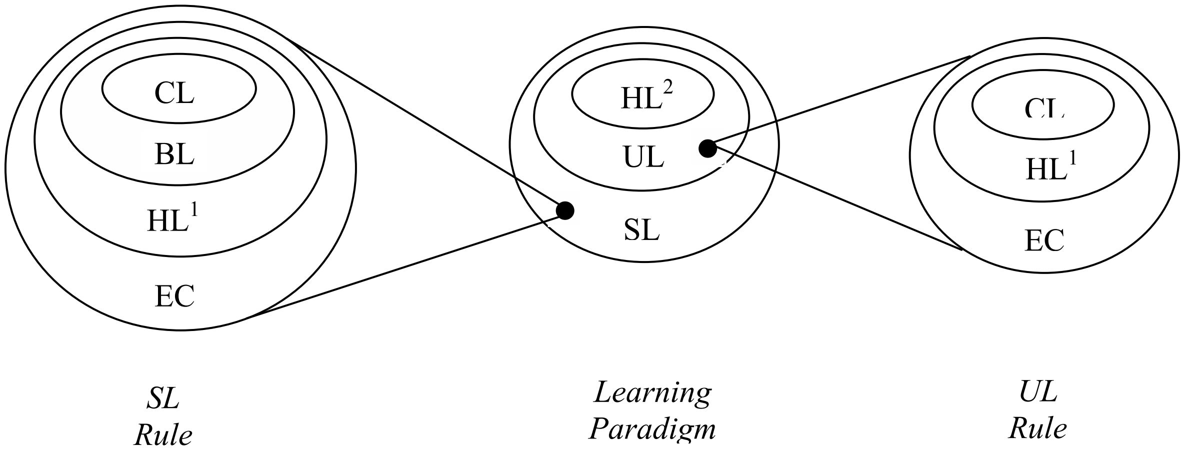
TABLE I
INCONVENIENT EXAMPLES FOR MULTI-LAYER MODELS IN THE ANN LEARNING PROCESS

ARCHI-TECTURE	Super-vised	Unsu-pervised	LEARNING PARADIGM
Compe-titive	LVQ	VQ	LEARNING
ART net-work	ART map	ART-rea-lizations	ALGOITHM

IV. Mathematical Foundation of Multi-Layer Models

The following material is according to [3]. The authors introduce the mathematical description of the multi-layer model such that the penalty functions may be classified in three types: 1) *inside* any concrete layer, 2) *between* any two layers in a single multi-layer model, 3) *between any two multi-layer models*:

$$\mathrm{eval}\left(\overline{X}\right)=f\left(\overline{X}\right)+\sum_{l=1}^{L}a_l\left[\lambda(t)\sum_{j=1}^{m}f_j^2\left(\overline{X}\right)\right]^l \qquad (1)$$



Competitive Learning (CL), Boltzmann Learning (BL), Hebbian Learning (HL[1]), Error Correction (EC), Hybrid Learning (HL[2]), Unsupervised Learning (UL), Supervised Learning (SL)

Fig. 2. ANN paradigms and rules

where:

$\text{eval}(\overline{X})$ — feasible and unfeasible solutions if $\overline{X} \in F$ is the optimal solution of the general non-linear programming model with continuous variables;

$f(\overline{X})$ — goal function for optimization;

$\lambda(t)$ — updated every generation t in the following way [5]:

$$\lambda(t+1) = \begin{cases} (1/\beta_1)\cdot\lambda(t), \text{ if } \overline{B}(i) \in F \text{ for all } t-k+1 \le i \le t \\ \beta_2 \cdot \lambda(t), \text{ if } \overline{B}(i) \in S - F \text{ for all } t-k+1 \le i \le t \\ \lambda(t), \quad \text{else} \end{cases}$$

$f_j(\overline{X})$ — constraint violation measure for the *j*-th constraint such that [6]:

$$f_j(\overline{X}) = \begin{cases} \max\{0, g_j(\overline{X})\}, \text{if } 1 \le j \le q \\ |h_j(\overline{X})|, \quad \text{if } q+1 \le j \le m \end{cases}$$

Here $g_j(\overline{X}) \le 0, j = 1,...,q$ and $h_j(\overline{X}) = 0, j = q+1,...,m$ is a set of additional constraints $m \ge 0$ the intersection of which with the search space S defines the feasible set F.

l — indicator of the constraint type with upper bound $L = \{2|3\}$:

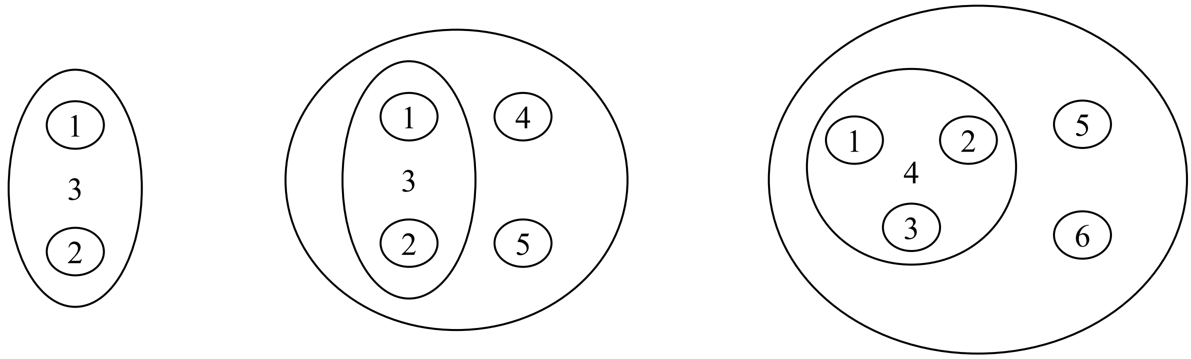
$$l = \begin{cases} 1: \text{insidea given layer (the lowest constraint level);} \\ 2: \text{betweentwo layers inside a given multilayer model} \\ \quad \text{(the moddle constraint level);} \\ 3: \text{betweentwo multilayer mod els (the highest const -} \\ \quad \text{raint level).} \end{cases}$$

a_l — coefficient array reflecting the weights of the different constraint levels in the formula. It is adjusted heuristically.

V. Conclusions

The paper is centered at the learning process of the ANN, the goal being the ease of the ANN design process. The theoretical basis for this goal are the hierarchically-based multi-layer models corresponding to the different levels of importance for the layers in the model. Hierarchical structures of a pyramidal type interpreted with multi-layer models are introduced with their advantages and limitations. Hierarchical systems are complemented in a natural way but not overlapped by structures without hierarchies. Both types of organizations arrange and represent *in different ways* objects from a *common* domain. The multi-layer models provide the most relevant description of the ensemble of features of the ANN learning process.

It is possible to interpret non-hierarchical structures with multi-layer models but it can be shown that such interpretations are not effective (not included due to limitations from the size of this paper).



a) RBF Hybrid *b) Unsupervised* *c) Supervised*

(a): error-correction learning rule (1), competitive learning rule (2), RBF learning algorithm (3); (b): principal component analysis learning algorithm (1), associative memory learning algorithm (2), Hebbian learning rule (3), error-correction learning rule ↔ Sammon's projection learning algorithm (4), competitive learning rule ↔ Kohonen's SOM learning algorithm (5); (c): perceptron learning algorithm (1), back-propagation learning algorithm (2), Adaline, Madaline learning algorithm (3), error-correction learning rule (4), Boltzmann's learning rule ↔ Boltzmann's learning algorithm (5), Hebbian learning rule ↔ linear discriminant analysis learning algorithm (6).

Fig.3. ANN rules and algorithms

REFERENCES

[1] S. Koynov, Ch. Korsemov, and H. Toshev, "The State-Of-The-Art In Modeling The Artificial Neural Networks And The Evolutionary Computation", *Proc. of the XXXVI Int. Sci. Conf. "Energy and Information Systems and Technologies (EIST 2001)"*, Bitola, Macedonia, pp. 680-685, 7-8 June 2001.

[2] S. Koynov, Ch. Korsemov, and H. Toshev, "The Artificial Neural Networks Design Model", *Proc. of the XXXVI Int. Sci. Conf. "Energy and Information Systems and Technologies (EIST 2001)"*, Bitola, Macedonia, pp. 686-691, 7-8 June 2001.

[3] S. Koynov, Ch. Korsemov, and H. Toshev, "A Generalized Evolutionary CAE Model to Design Artificial Neural Networks", *Proc. of the XXXVI Int. Sci. Conf. "Energy and Information Systems and Technologies (EIST 2001)"*, Bitola, Macedonia, pp.692-697, 7-8 June 2001.

[4] A. Jain, "Artificial Neural Networks: A Tutorial", *IEEE Computer*, pp. 31-44, March, 1996.

[5] C. Bean, and A. B. Hadj-Alouane, *A dual genetic algorithm for bounded integer programs*, Department of Industrial and Operations Engineering, The University of Michigan, TR 92-53, 1992.

[6] Zb. Michalewicz, "The significance of the evaluation function in evolutionary algorithms", *Evolutionary algorithms*, L. D. Davis, K. De Jong, M. D. Vose, L. D. Whitley (eds.), Springer, pp-151-166, 1999.

[7] E. Falkenauer, "Applying genetic algorithms to real-world problems", *Evolutionary algorithms*, L. D. Davis, K. De Jong, M. D. Vose, L. D. Whitley (eds.), Springer, pp-65-88, 1999.

[8] S. Haykin, *Neural networks*. Englewood Cliffs, Macmillan Publishing Co., 696 p., 1994.