The Paradigm Model of The Artificial Neural Networks and Its Evolution up to The Expert Level Representation

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

Abstract: - **The paper proves the necessity of a global model of the paradigm of the artificial neural networks (ANN). The authors introduce the evolution of the ANN paradigm model from single-layered to a three-layered model on a block level and also with hierarchical ANN. The introduced model is supported by a due mathematical formalism.**

Keywords - **neural networks, neural networks paradigm model, object neural network, cluster of expert neural networks, hierarchical neural network.**

I. Introduction

The recent years are marked with a real boom of the research in the field of the artificial neural networks (ANNs). This is due to their place in the paradigm of the scientific knowledge - between the mathematical formalism and the artificial intelligence, because it is based on concepts and techniques from both of them.

The existing references in the area are either specialized (the scientific periodicals) or they are popularizations and textbooks. Still the number of publications which cover both aspects equally well is negligible.

The authors offer the paradigm model of ANNs. This is an attempt to unite the theoretical base with the practical approach to study and use this tool. The model is introduced also in the aspect of the ways of the possible realizations; 'self-modeling' of the ANNs is mathematically proved.

II. The Artificial Neural Networks Paradigm Model - from The User Task to The Object Neural Network Parameters

Modularity appears to be an important principle in the architecture of vertebrate nervous systems [1], e. g. the hierarchical representations of the information in the cortical visual areas [2]. In them the highly complex computation performed by the visual system is broken down into pieces, just like any good engineer would do when designing a complex system according to [2].

According to [1, 3-4] the use of a modular hierarchical approach offers many attractive features. The present paper is a result of a previous research [5-6]. If we treat the ANN paradigm model (NNPM) as a 'black box' (Fig. 1) we expect to obtain the constructive parameters of the object neural network (ONN) as an output from the NNPM if we feed the model with the user task data. In this way the NNPM must output the constructive parameters of the ONN which correspond to the user task which defines the NNPM input data. The model consists of clusters and the clusters consist of modules (experts).

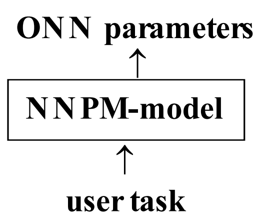

Fig. 1. Formulation

The single-layer NNPM model (Fig. 2) is the simplest NNPM model. Here the authors accept that the clusters and the modules in them have equal initial probabilities for activity. The single-layer NNPM is the basis for the more sophisticated multi-layer NNPM models.

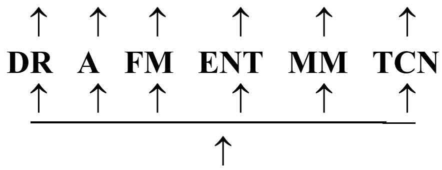

Fig. 2. Single-layer NNPM

The meaning of the clusters is as follows: DR - Data Representation, A - Application, FM - ONN Functional Methods, ENT - Elementary Network Type, MM - Mathematical Methods, TCN - Type of the Connections in the ONN.

The multi-layer models are obtained by the particularization of the single-layer model (from left to right).

The fist two clusters in the bilayer NNPM model in Fig. 3 are these which are nearest to the user requirements; the cluster A depends directly on the user task and the cluster DR depends implicitly on this task (the definition of these

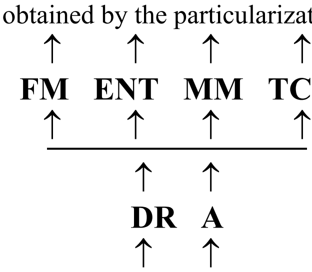

Fig. 3. Bilayer NNPM

[1] 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
[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] 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
[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

clusters is an obligation of the ONN designer).

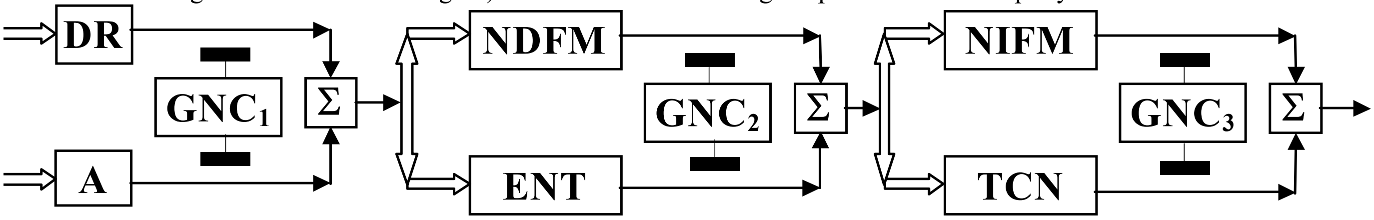

Fig. 5. Multilayer Hierarchical Neural Network

The model consists of clusters and the clusters in their turn of modules or experts (Fig. 4). The purpose of the clusters follows below.

The ***input clusters*** reflect in an optimal manner the user task:1) ***A*** handles the information related with the ONN Application. This expert is responsible for such features of the application like ‘how’, ‘where’, ‘when’ and ‘what’. These specifications cover the user task. 2) ***DR*** is connected with the spaces of the input- and the output vector Data Representation. This expert also depends on the user task data, but it may be additionally defined by the ONN designer.

The ***output clusters*** define in an unique way the ONN parameters:1) ***NIFM*** outputs the possible Network-Independent Functional Methods (e. g. fuzzy

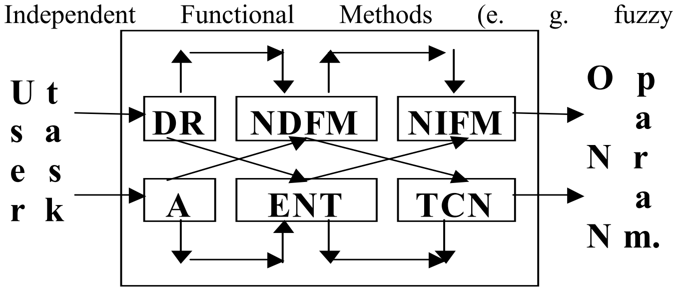

Fig. 4. Neural Network Paradigm Model

methods, methods for optimization, statistical methods, type of the analysis, etc.); 2) ***TCN*** defines the Type of the Connections in the Network (the topological structure).

The ***medial layer clusters*** follow basically [7] and they define the common properties of the ONN: 1) ***NDFM*** is the Network-Dependent Functional Methods cluster. It is important to note that the outputs from cluster ***A*** define the task type expert in cluster ***NDFM*** in an unambiguous way; other important experts in cluster ***NDFM*** are the type, the rule and the algorithm of learning; 2) ***ENT*** is the Elementary Network Type cluster. This cluster depends significantly on the outputs from cluster ***DR***.

III. NNPM: Possible Approaches

The authors offer the following possible realizations of NNPM: chips and out-of-computer storage, associative memories, multi-layer artificial hierarchical neural networks (MHNN) and inference engine-based expert systems (IEBES).

Fig. 5 introduces a triple-layer MHNN model realized with a hierarchical neural network. GNC_i stands for the gating network of the clusters of the *i*-th level, Σ - for the output of the level.

Fig. 6. presents the exemplary structure of a cluster in the paradigm model. In it the abbreviations E1, …, E4 stand for the different expert numbers the concrete meanings of which are given in the following below explanations; ***GN*** stands for the gating network of the experts in the cluster. Next follows the most detailed precision of the described model.

The NNPM model consists of the following units: 22 experts/modules, 6 gating networks (GN from Fig. 6) of the experts in every cluster, 3 layers and 3 GNCs (from Fig. 5) on every level. All the units are made of optimally tuned and trained ANNs; 1) *TCN* is the cluster of the *Type of the Connections in the Network*. It consists of the following experts: ***AdM*** - Adaptive Model, ***AsM*** - Associative Model, ***MM*** - Multi-layer Model, ***SLM*** - Single Layer Model; 2) *NIFM* are the *Network-Independent Functional Methods* with experts: ***FM*** - Fuzzy Methods, ***MO*** - Methods for Optimization, ***SM*** - Statistical Methods, ***TA*** - Type of the Analysis; 3) *ENT* is the *Elementary Network Type* with experts: ***HN*** - Hopfield Networks, ***P*** - perceptron, ***RBFN*** - radial-basis function networks, ***SOM*** - self-organizing maps; 4) *NDFM* are the *Network-Dependent Functional Methods* with experts: ***TL*** - type of learning, ***TLA*** -type of the learning algorithm, ***TLR*** - type of the learning rule, ***TT*** - type of the task; 5) *DR* is the *Data Representation* cluster with experts: ***CR*** -continuous representation, ***DR2*** - digital representation, ***QR*** - quantized representation; 6) *A* is the *Application* cluster with experts: ***PA*** - Place of the Application (‘where’), ***NS*** - Network Specialization (‘what’), ***CA*** - Conditions of the Application (‘how’ and ‘when’).

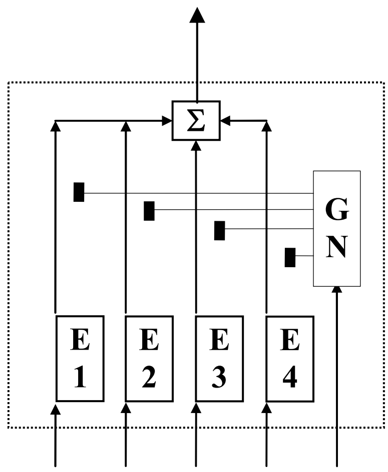

Fig. 6. A Model Cluster

IV. MHNN: Mathematical Proof

The mathematical proof is based on a generation of training couples from a set of embedded regressions [8]. The approach is probabilistic and the following rules are actual for each layer: 1) **x** (the input vector of dimension *p*) is selected at random for some prior distribution; 2) the i-th cluster is chosen from the distribution $P(i|\mathbf{x})$ which is the probability of the i-th rule given the input vector $\mathbf{x}$; 3) the j-th expert / rule is chosen from the distribution $P(j|\mathbf{x}, i)$ for an input $\mathbf{x}$ to the cluster; 4) $\mathbf{d}$ (the desired response vector of dimension q) is generated in accordance with the regression:

$$\mathbf{d} = \mathbf{F}_{ji}(\mathbf{x}) + \varepsilon \tag{1}$$

where $\mathbf{F}_{ji}$ is a deterministic vector-valued non-linear function of the vector argument $\mathbf{x}$; $\boldsymbol{\varepsilon}$ is a random vector with a zero average and Gaussian distribution.

The dynamics of NNPM is determined by the variable l - the log-likelihood probability to obtain definite values of the output vector $\mathbf{y}$:

$$l = \ln \sum_{i=1}^{K} g_i \cdot \sum_{j=1}^{L} g_{j|i} \cdot \exp\left(-\tfrac{1}{2} \left\| \mathbf{d} - \mathbf{y_{ji}} \right\|^2 \right) \tag{2}$$

where g is the activation function of the i-th output neuron of ***GNC***, $g_{j/i}$ is the activation of the j-th output neuron in $\boldsymbol{GN_i}$, $\mathbf{y}_{ji}$ is the output vector of the j-th expert in the i-th cluster.

The considerations below will be fulfilled if we accept the following probabilistic interpretation of the parameters g_i, $g_{j/i}$ and $\mathbf{y}_{ji}$: 1) g_i and $g_{j/i}$ are the conditional apriori probabilities to generate the current training image {$\mathbf{x}$, $\mathbf{d}$}; 2) $\mathbf{y}_{ji}$ are the conditional vectors of the mathematical average for a multiargumental Gaussian distribution.

It is easy to evaluate the values of g_i and $g_{j/i}$ by maximization of l if we know the following weighted sums of the inputs to the output neurons in the gating networks: 1) u_i is the weighted sum of the inputs to the i-th output neuron in ***GNC***; 2) $u_{j/i}$ is the weighted sum of the inputs to the j-th output neuron in $\boldsymbol{Gn_i}$. Now g_i and $g_{j/i}$ are the normalized exponential transformations of u_i and $u_{j/i}$.

Finally we introduce the following two conditional aposteriori probabilities to generate $\mathbf{d}$: 1) h_i is the conditional aposteriori probability to generate $\mathbf{d}$ from the i-th cluster $\left(\sum h_i = 1, \forall i\right)$; 2) $h_{i/i}$ is the conditional aposteriori probability to generate $\mathbf{d}$ from the j-the expert in the i-th cluster $\left(\sum h_{j|i} = 1,\ \forall i \forall j\right)$, then

$$\partial l / \partial u_i = h_i - g_i,\ \forall i \tag{3}$$

$$\partial l / \partial u_{j|i} = h_i \cdot (h_{j|i} - g_{j|i}),\ \forall i \forall j \tag{4}$$

$$\partial l / \partial \mathbf{y}_{ji} = h_i \cdot h_{j|i} \cdot (\mathbf{d} - \mathbf{y}_{ji}), \forall i \forall j \tag{5}$$

These partial derivatives comprise the criteria to setup ***GNC***, $\boldsymbol{GN_i}$ for the i-th cluster and for the j-th expert in the i-th cluster. The last two partial derivatives share the common factor, namely, the aposteriori probability h_i. This means that the experts are tied to each other. Consequently, the experts within a cluster tend to learn similar mappings early in the training process. However, when the probabilities associated with a cluster to which the experts belong assume larger values later in the training process, they start to specialize in what they learn. Thus MHNN tends to evolve in a coarse-to-fine structural way. This is important because it implies that a deep hierarchical network is naturally robust with respect to the overfitting problem [9].

Now it is easy to determine the following below sensitivity factors, even with iterative definitions:

$$\frac{\partial l}{\partial \mathbf{a}_i} = \frac{\partial l}{\partial u_i} \cdot \frac{\partial u_i}{\partial \mathbf{a}_i} \tag{6}$$

$$\frac{\partial l}{\partial \mathbf{c}_{j|i}} = \frac{\partial l}{\partial u_{j|i}} \cdot \frac{\partial u_{j|i}}{\partial \mathbf{c}_{j|i}} \tag{7}$$

$$\frac{\partial l}{\partial \mathbf{w}_{ji}^{(m)}} = \frac{\partial l}{\partial \mathbf{y}_{ji}^{(m)}} \cdot \frac{\partial \mathbf{y}_{ji}^{(m)}}{\partial \mathbf{w}_{ji}^{(m)}} \tag{8}$$

where $\mathbf{a_i}$ is the synaptic weight vector for the i-th output neuron in ***GNC***, $\mathbf{c_{j|i}}$ is the synaptic weight vector for the j-th output neuron of the GN in the i-th cluster, $\mathbf{w}_{ji}^{(m)}$ is the synaptic weight vector of the m-th output neuron in the j-th expert for the i-th cluster.

V. CONCLUSIONS

The paper introduces NNPM. The presented model possesses both fundamental theoretical and applicational properties in a way which makes the research of the ANNs easier and more practical. It is a triple-layer model of an expert type and the experts are united in clusters. This approach leads to grouping the clusters in three different groups. The research proves that the NNPM approach guarantees great enhancements when modeling and learning ANNs both theoretically and in practice. The most important consequence is that the effective practical application is based on the organized efforts not only of concrete scientific explorers, but also of working groups and scientific enterprises.

REFERENCES

[1] J. C. Houk and A. G. Barto, “Distributed Sensorimotor Learning”, *Tutorials in Motor Behavior II*, G. E. Stelmach, J. Requin, eds., Amsterdam, Elsevier, pp. 71-100, 1992.

[2] D. C. Van Essen, C. H. Anderson and D. J. Felleman, “Information Processing in The Primate Visual System: An Integrated Systems Perspective”, *Science* 255, pp. 419-423, 1992.

[3] M. I. Jordan, “Recent Developments in Supervised Learning, Tutorial Notes”, *Int. Joint Conf. on Neural Networks*, Baltimore, MD, 1992.

[4] R. A. Jacobs, M. I. Jordan and A. G. Barto, “Task Decomposition Through Competition in A Modular Connectionist Architecture: The What and Where Vision Tasks”, *Cognitive Science* 15, pp. 219-250, 1991.

[5] S. Koynov, “Artificial Neural Networks - Background,

Evolutionary Models and History", *Problems of Engineering Cybernetics and Robotics*, V. 48, Sofia, Bulgaria, pp. 36-44, 1999.
[6] 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.
[7] A. Jain, "Artificial Neural Networks: A Tutorial", *IEEE Computer*, pp. 31-44, March, 1996.
[8] R. Jacobs and M. Jordan, "A Competitive Modular Connectionist Architecture", R. P. Lippmann, J. E. Moody, D. J. Touretzky, (eds.), *Advances in Neural Information Processing Systems*, San Mateo, CA, Morgan Kaufmann, 3, pp. 767-773, 1991.
[9] R. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, "Adaptive Mixtures of Local Experts", *Neural Computation*, 3, pp. 79-87, 1991.