



Neurocomputer Based Complexity Estimator Optimizing A Hybrid Multi Neural Network Structure

Ivan Budnyk, El khier Bouyoucef, Abdennasser Chebira, Kurosh Madani

Image, Signal and Intelligent Systems Laboratory (LISSI / EA 3956), Senart Institute of Technology, University
PARIS XII, Av. Pierre Point, F-77127 Lieusaint, France,
{budnyk; bouyoucef; chebira; madani}@univ-paris12.fr, <http://www.univ-paris12.fr/>

Abstract: *This paper presents application of ZISC© IBM® neurocomputer based approach for estimating task complexity within T-DTS framework. T-DTS (Tree-like Divide To Simplify) is Hybrid Multiple Neural Networks software platform which constructs a neural tree structures of a complex problem following the paradigm “divide” and “conquer”. Complexity estimator modules are the core of this framework. One of them is ZISC© IBM® complexity estimator that has been recently applied to T-DTS. The global aim of this research work is to increase T-DTS performance in terms of generalization and learning abilities. In this paper we demonstrate matchless ZISC© IBM® based neurocomputer complexity estimator effect on database decomposition and searching for optimal T-DTS adjustment of complexity threshold.*

Keywords: *ZISC© IBM® Neurocomputer, T-DTS, Hybrid Multiply Neural Networks, Self Organizing Map – Linear Support Vector Machine Decision Tree, RBF Algorithm*

1. INTRODUCTION

In a very large number of cases dealing with real world dilemmas and applications (system identification, industrial processes, manufacturing regulation, optimization, decision, pattern recognition, systems, plants safety, etc), information is available as data stored in files (databases etc.). So, the efficient data processing becomes a chief condition to solve problems related to above-mentioned areas. In the most of those cases, processing efficiency is closely related to several issues among which are:

- Data nature: including data complexity, data quality and data representative features.
- Processing technique related issues: including model choice, processing complexity and intrinsic processing delay.

Data complexity, frequently related to nonlinearity or subjectivity of data, may affect the processing efficiency. While, data quality (noisy or degraded data), may influence processing success and expected results' quality. Finally, representative features, relating concerning scarcity of pertinent data, could affect processing achievement or resulted precision.

On the other hand, choice or availability of appropriated model describing the behaviour related to data to process is of major importance. Processing

technique or algorithms' complexity (designing, precision, etc.) shapes the processing effectiveness. Intrinsic processing delay or processing time, related to the implementation issues (software or hardware related issues) or processing models parameterization could affect not only processing quality (results' quality) but also the technique's viability to offer an adequate solution for a complex problem. Of course, unfortunately real world and industrial problems are never as comfortable as could be “toy problems”. They are often complex problems with a large number of parameters (which have to be considered). That's why conventional solutions (based on mathematical and analytical models) reach serious limitation for solving this category of dilemmas.

One of the key points on which one can act is the complexity reduction. Complexity reduction could act not only at problem's solution level but also at processing procedure's level. An issue could be model complexity reduction by splitting a complex problem into a set of simpler problems: this leads to “multi-modelling” where a set of simple models is used to sculpt a complex behaviour [1]. Another promising approach to reduce complexity takes advantage from hybridization [2]. Several Artificial Neural Networks (ANN) based approaches were suggested allowing complexity and computing time reduction ([2] to [7]).

This paper deals with a hybrid multi neural network structure and its neuro-processor based complexity estimator, which shapes this structure. We called this structure and the related concept T-DTS (Treelike Divide To Simplify). T-DTS is ANN based data driven treelike multiple model generator with self-organizing and complexity reduction abilities. The main idea of T-DTS is based on “Divide et impera”¹ paradigm (Julius Caesar), transformed here as “Divide To Simplify” (DTS) [8]. The purpose is based on the use of a set of small and specialized mapping neural networks, that we called Neural Network based Models (NNM), supervised by a Supervisor Agent (SA). SA is resulted (obtained) from a tree-like decomposition process splitting the initially complex problem’s (represented by its representative database) feature space into a set of simpler sub-spaces (represented by their representative sub-databases). So, at the leaf’s level of the obtained tree one can find NNM and at its nodes level, one can find Decomposition Agent (DA) controlled by a Complexity Estimator Agent (CEA). Combination of complexity estimation, splitting and learning capabilities confers to the issued intelligent system self-organizing ability.

The paper is organized as follow: Section 2 will present the T-DTS models generator. Section 3 will be dedicated to the proposed neuro-computer based CEA. Section 4 will describe implementation and validation issues. Finally, the last section will conclude the present paper.

2. TREELIKE DIVIDE TO SIMPLIFY MULTIPLE MODELS GENERATOR

As it has been mentioned in introductory section, T-DTS is a data driven ANN based multiple model structure designed to handle processing of complex data by reducing the processing’s complexity. T-DTS and associated algorithm construct a tree-like evolutionary neural architecture automatically.

The T-DTS includes two main operation modes. The first is the learning phase, where T-DTS decomposes the learning database, constructs the Supervisor Agent and provides NNM. The second phase is the operation phase, where the obtained hybrid multi neural network system processes unknown (e.g. unlearned) data. There could be also a pre-processing phase at the beginning, which arranges (prepare) data to be processed. Pre-processing phase could include several steps (conventional or neural stages). Figures 1 gives the general bloc diagram of T-DTS operational steps.

The learning phase is an important phase during which T-DTS performs several key operations:

splitting the learning database into several sub-databases, constructing (dynamically) a SA and building a set of specialized NNM, trained by generated sub-databases, which model different obtained feature sub-spaces.

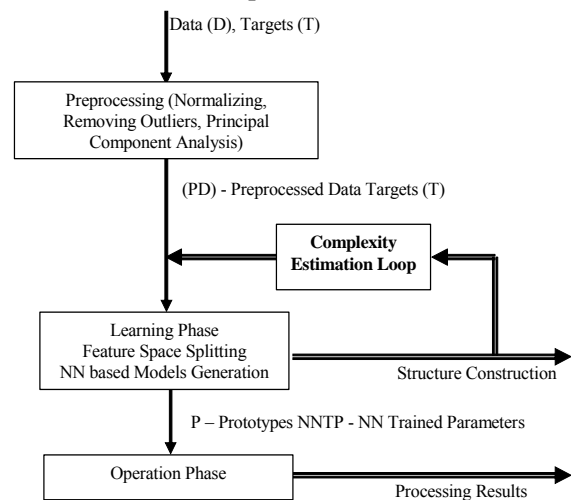


Fig. 1 – General bloc diagram of T-DTS, presenting main operation levels

Figure 2 represents the division and NNM construction process bloc diagrams. As this figure shows, after the learning phase, a set of neural network based models (trained from sub-databases) are available and cover (model) the behaviour region-by-region in the problem’s feature space. In this way, a complex problem is decomposed recursively into a set of simpler sub-problems: the initial feature space is divided into M sub-spaces. For each subspace k , T-DTS constructs a neural based model describing the relations between inputs and outputs. If a neural based model cannot be built for an obtained sub-database, then, a new decomposition will be performed on the concerned sub-space, dividing it into several other sub-spaces.

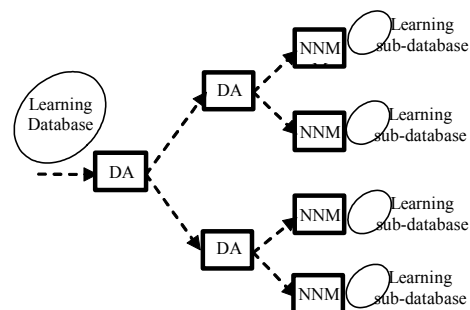


Fig. 2 – General bloc diagram of T-DTS learning phase and its tree-like splitting process

In fact, the learning phase could be considered as a self-organizing model generation process, which leads to a set of ANN based models (or processing

¹ “divide and rule”.

units) managed by Supervisor Agent (SA).

The second operation mode corresponds to the use of the constructed hybrid neural based multi-model system for processing unlearned (work) data. The Operation Phase is depicted by figure 3. The SA, constructed during the learning phase, receives data (unlearned input vector) and classifies that data (pattern) as corresponding to one of the processing subset. Then, the most appropriated NNM is authorized (activated) to process that data.

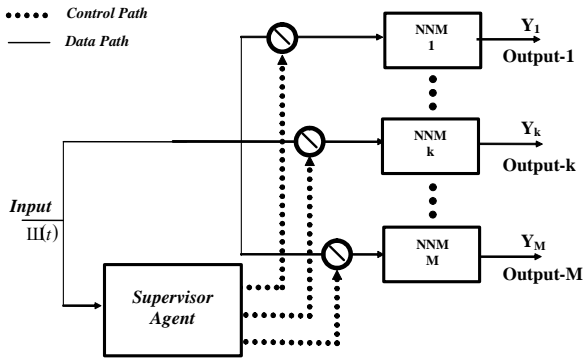


Fig. 3 – General bloc diagram of T-DTS generalization phase

Let $\Psi(t)$ be the system's input ($\Psi(t) \in \mathbb{R}^{n_\Psi}$), a n_Ψ -Dimensional vector, and let $Y_k(t) \in \mathbb{R}^{n_y}$ be the k -th ($k \in \{1, \dots, M\}$) model's output vector of dimension n_y . Let $F_k(\cdot): \mathbb{R}^{n_\Psi} \rightarrow \mathbb{R}^{n_y}$ be the k -th NNM's transfer function. Let $S(\Psi(t), p, \xi) \in B^M$, where $B = \{0, 1\}$, be the Supervisor Agent's output, which may also depend on some additional parameters p and/or on some conditions ξ . p_k represents some particular values of parameter p and ξ_k denotes some particular condition ξ , respectively, obtained from learning phase process for the k -th sub-dataset. Taking into account the above-defined notation, the SA's output could be formalized as relation (1).

$$S(\Psi(t), p, \xi) = (s_1 \ \dots \ s_k \ \dots \ s_M)^T \quad (1)$$

with
$$\begin{cases} s_k = 1 & \text{if } p = p_k \text{ and } \xi = \xi_k \\ s_k = 0 & \text{else} \end{cases}$$

The AS's particular response, corresponding to some particular values of the parameter p and the condition ξ (e.g. $S(\Psi, p_k, C_k)$) will activate the k -th NNM, and so the processing of an unlearned input data conform to parameter p_k and condition C_k will be given by the output of the selected NNM (relation (2)).

$$Y(\Psi, t) = Y_k(t) = F_k(\Psi(t)) \quad (2)$$

It is important to emphasize that the SA uses a Complexity Estimator Agent (e.g. a complexity indicator) in order to handle the splitting process optimizing the generated hybrid multi neural network processing system. So, complexity estimation is among the most important operations performed by T-DTS. The next section introduces the used neuro-processor and describes the CEA based on that neuro-processor.

3. ZISC-036 NEURO-PROCESSOR BASED COMPLEXITY ESTIMATOR AGENT

The goal is to estimate the processing task's difficulty and modify the splitting process in order to handle the whole task more efficiently (e.g. optimizing the generated hybrid multi neural network processing system). Complexity estimation is not an easy subject. Concerning classification task's complexity estimation, most of the used techniques are based on Bayes error calculation (direct, indirectly and non-parametric Bayes error estimation) (from [9] to [15]). In this paper we propose a different slant using a neuro-processor implementing a kernel-functions-like ANN model with evolutionary topology ([16], [17]).

3.1 ZISC-036 NEURO-PROCESSOR

The IBM ZISC-036 ([18], [19]) is a parallel neuro-processor implementing RCE/RBF. Composed of 36 neurons, each chip is capable of performing up to 250 000 recognitions per second. Figure 4 gives the ZISC-036 bloc diagram and an example of input feature space mapping in a 2-D space. This chip is fully cascable which allows the use of as many neurons as the user needs (a PCI board is available with 684 neurons). A neuron is an element, which is able to:

- memorize a prototype (64 components coded on 8 bits), the associated category (14 bits), an influence field (14 bits) and a context (7 bits),
- compute the distance, based on the selected norm (norm L1 or LSUP) between its memorized prototype and the input vector (the distance is coded on fourteen bits),
- compare the computed distance with the influence fields,
- communicate with other neurons (in order to find the minimum distance, category, etc.),
- adjust its influence field (during learning phase).

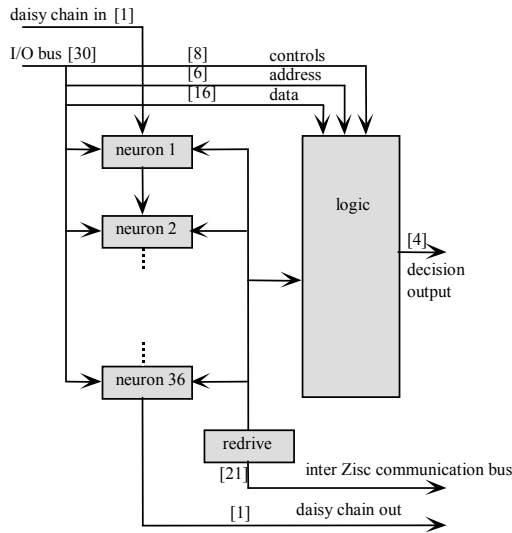


Fig. 4 – IBM ZISC-036 chip’s general bloc diagram

A 16 bit data bus handles input vectors as well as other data transfers (such as category and distance), and chip controls. Within the chip, controlled access to various data in the network is performed through a 6-bit address bus. Controlling the ZISC036 is, by definition, accessing its registers, and requires an address definition via the address bus, and data transfer via the data bus. The inter-ZISC communication bus which is used to connect several devices within the same network, and the decision bus which carries classification information allow the use of the ZISC in a ‘stand alone’ mode. All neurons communicate via the ‘inter-neuron communication bus. This bus is internally driven to allow the connection of several ZISC modules without impact on performance. An efficient protocol allows a true parallel operation of all neurons of the network even during the learning process. Because ZISC is a coprocessor device, it must be controlled by a master (state machine or controller). This can be done by a standard I/O bus. The I/O bus of ZISC036 has been designed to allow a wide variety of attachments from simple state machine interface to standard micro-controllers or buses.

3.2 ZISC-036 NEURO-PROCESSOR BASED COMPLEXITY INDICATOR

In our work we determine a complexity as the amount of computational resource that it takes to solve a classification problem. Thus the *complexity* here is the limited supply of these *resources* (amount of neurons) once *the appropriate program* (classification methods) is supplied. We have to mention also a growing criticism concerning the

term *complexity*, because it has been misused without a proper definition [16]. Our primary study interest is in classification complexity in term of computational difficulty of IBM © ZISC-036 ® hardware to obtain satisfactory learning and generalization rates using RBF algorithm and adjusted initial options [20].

We expect that a more complex problem will involve a more complex ZISC neural network structure. The simplest neural network structure feature is the number n of neurons created during the learning phase. We suppose that a database compounded of a collection of m objects associated to labels or categories is available. To estimate such database complexity we use the ZISC® as a classifier. The simplest neural network structure feature is the number n of neurons created during the learning phase. The following indicator is defined, where n is a parameter that reflects complexity:

$$Q = \frac{n}{m}, m \geq 1, n \geq 0 \tag{3}$$

We suppose that there exists some function $n = g(.)$ that reflects problem complexity. The arguments of this function may be the signal-to-noise ratio, the dimension of the representation space, boundary non-linearity and/or database size. We consider only $g(.)$ function’s variations according to m axis: $g(m)$. We suppose that learning database is free of any incorrect or missing information. On the basis on $g(m)$, a complexity indicator is defined as follow:

$$Q_i(m) = \frac{g_i(m)}{m}, m \geq 1, g_i(m) \geq 0 \tag{4}$$

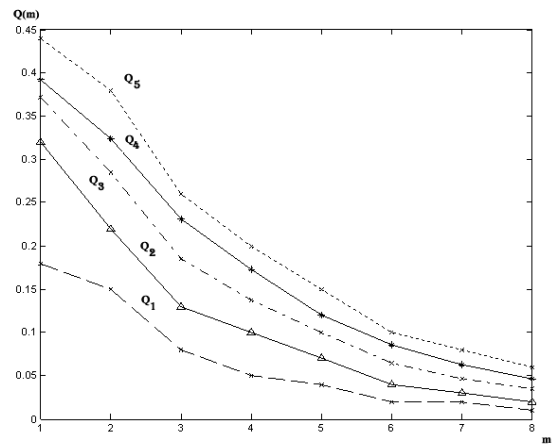


Fig. 5 – ZISC-036 neuro-computer based complexity indicator versus the learning database’s size (m)

We expect that for the same problem, as we enhance m , the problem seem to be less complex: more information reduces problem ambiguity. On the other hand, for problems of different and increasing complexity, Q_i indicator should have a higher value. Figure 5 shows an example of the neuro-computer based complexity indicator's variation versus the learning database's size. 5 different databases with increasing complexity have been considered: Q_1 corresponds to the easiest one and Q_5 to the hardest one. 8 different learning database sizes have been considered (e.g. 8 different m values). The considered databases include 50, 100, 250, 500, 1000, 2500, 5000 and 10000 patterns respectively (corresponding to the indexes 1 to 8 respectively). For each set of parameters, tests are repeated 10 times in order to get statistics and as stated to check the deviations and to get average. Totally, 800 tests have been performed.

As one can remark, the proposed complexity estimator's value decreases with the learning database's size. This means that the considered classification task becomes less complex when enough representative examples are available. On the other hand, the value of the indicators ascends from easiest classification task (e.g. Q_1) to the hardest one (e.g. Q_5). The results of figure 5 show that the proposed complexity estimator is sensitive to the classification task's complexity and behaves conformably to the aforementioned expectations.

4. IMPLEMENTATION AND VALIDATION ISSUES

A mixed implementation, involving both hardware and software modules, of T-DTS system has been realized. The neuron-computer based complexity estimation loop has been implemented using a ZISC-036 based PC board including 16 ZISC-036 chips (developing about 600 to 700 neurons). While, other required functions have been realized as software modules within "MathLab" environment. The left scheme of the figure 6 gives the general implementation's bloc diagram of the realized T-DTS system.

As one can note, T-DTS software architecture includes three main databases:

1. Decomposition methods,
2. ANN based methods,
3. Complexity estimators for decomposition process.

These three databases can be completed, enhanced or customized independently from the main blocs implementing T-DTS functions. In the MatLab based version of T-DTS software Decomposition Agents may operate using Competitive Network (CN), Self Organized Map

(SOM) and Learning Vector Quantization (LVQ) ANNs. From the side of Processing Units, this version implements several ANN models: Learning Vector Quantization (LVQ), Multilayer Perceptron model trained by Levenberg-Marquardt algorithm (MLP-LM), General Regression Neural Network (GRNN), Radial basis function network (RBF), Probabilistic Neural Network (PNN) and Linear Network (LN). Finally, beside the ZISC-036 neuro-processor based complexity estimator several options involving other complexity indicators (described and studied in [23]) are offered:

- MaxStd (Multi-class maximum of the sum of the standard deviations of the data-vector attributes),
- Fisher discriminator,
- Purity algorithm,
- Normalized mean distance,
- Divergence measure,
- Jeffries-Matusita distance,
- Bhattacharyya bound,
- Mahalanobis distance,
- Inter-intra scatter-matrixes' criteria.

For example, SOM-LSVMDT approach described in [21] can be easily implemented in within the T-DTS frame by incorporating LSVMDT (Linear Support Vector Machine Decision Tree) [22]. SOM (Self-Organizing Map) method is already available in DU database.

In the same way, the output result-panel offers to the user several graphic variants. Figure 7 shows the parameterization, and results-display control panels Among the offered possibility, one of the most useful is the option allowing representing a 2D-data representation sorting the decomposed sub-databases and their representative centers conformably to the performed decomposition process. In this representation, the final graphic will show the obtained tree and the obtained clusters. The right picture of figure 6 gives an example of such representation.

In order to evaluate the ZISC complexity estimator's pertinence we have considered the range of benchmark problems described in the [23]. Among the available benchmark problems is "Splice-junction" DNA Sequences classification problem from Genbank 64.1 (ftp site: <ftp://genbank.bio.net>). This considered benchmark deals with DNA sequences' boundaries classification problem: exons-introns boundaries recognition. Exons corresponds to the retained part after DNA sequence splicing and introns is the rejected (spliced out) part. There are 3190 numbers of instances, including 62 attributes which defines DNA sequences. Our ZISC complexity estimator is a leader among the group of available indicators.

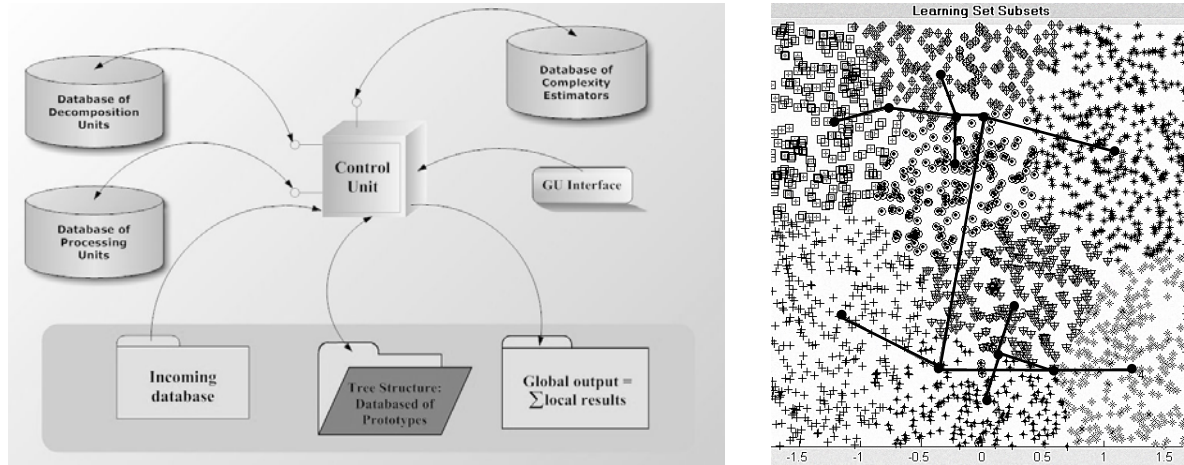


Fig. 6 – General bloc diagram of the T-DTS system’s software architecture (left) and example of learning phase and learning Subsets representation by the T-DTS Graphic User Interface (right)

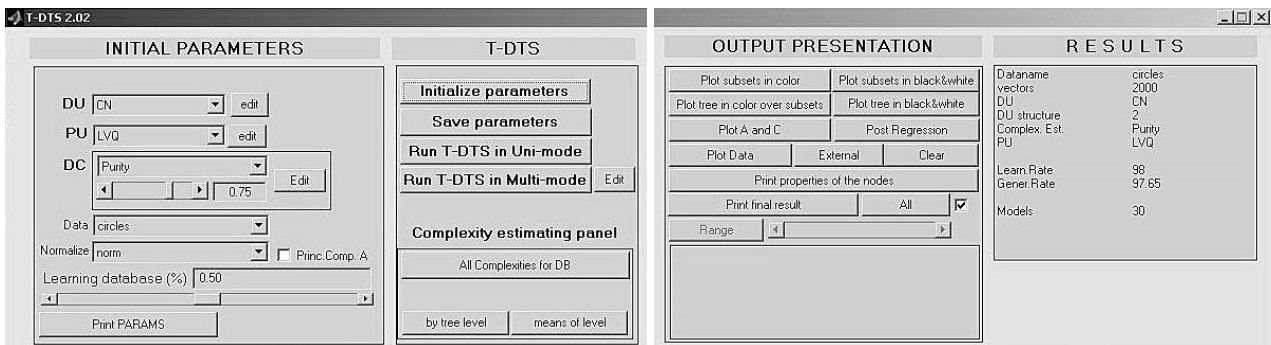


Fig. 7 – Screenshot of Matlab-implementation of T-DTS User Graphic Interface showing parameterization screenshot (left) and results control panel (right)

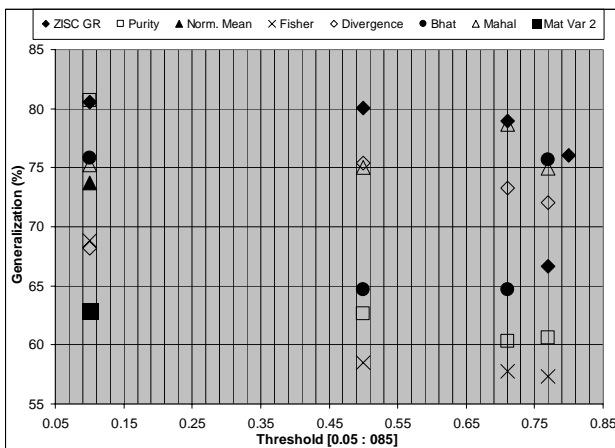


Fig. 8 – DNA sequences classification rate using ZISC-036 based complexity indicator and different other complexity estimators. Numbers of vectors - 1900. Size of learning database 20%. These results have been obtained using DA apply a CN. based approach and NNM are MLP-like ANNs

Summarizing, we can state that ZISC complexity estimator appears as being a general complexity indicator and thus acts more efficiently than the other criteria, because it is less sensitive to the

vector(s) attributes. For example, in Figure 7. for threshold-parameter greater than 0.8, correct T-DTS processing is efficiently performed only by ZISC complexity estimator.

5. CONCLUSION

We have presented new complexity estimator (complexity indicator) extracted from based on kernel-function neural network’s topological structure, taking advantage from evolutionary learning of such ANN model involving the number of connected neurons in the structure.

We have studied and validated the proposed complexity indicator on the basis of an especially designed normalizes synthetic 2D-benchmark problem and show that its ZISC-036 neuro-computer based implementation, embedded in T-DTS framework can reach high generalization rate and benefit from high computational speed due to massively parallel nature of the aforementioned neuron-computer.

A hybrid implementation of T-DTS has then been realized integrating the proposed ZISC-036 based complexity estimator beside other complexity

indicators already embedded in T-DTS. Experimental validation, performed on DNA Sequence pattern classification problem, show that T-DTS with ZISC complexity module takes a leading position remaining independent from tuning parameters (such as learning rate etc.) inherent to the other available indicators. Another appealing advantage of the proposed complexity estimator is related to it's independency from data source's nature as well from distribution between classes: so, the proposed estimator doesn't rely on the database properties.

The future perspective of the work relays self-tuning / self-optimizing skills of T-DTS in order to make T-DTS constructing optimal tree structure without human supervision. We also work on further implementation of additional complexity estimating techniques such as *Collective Entropy* based estimator. This accomplishment will expand database of complexity estimators which play a key role in T-DTS.

6. REFERENCES

- [1] Multiple Model Approaches to Modeling and Control, edited by R. Murray-Smith and T.A. Johansen, Taylor & Francis Publishers, 1997, ISBN 0-7484-0595-X.
- [2] S. Goonatilake and S. Khebbal, "Intelligent Hybrid Systems: Issues, Classification and Future Directions", in Intelligent Hybrid Systems, John Wiley & Sons, pp 1-20, ISBN 0 471 94242 1.
- [3] Krogh A., Vedelsby J.: Neural Network Ensembles, Cross Validation, and Active Learning, in Adv in Neural Inf Processing Syst. 7, The MIT Press, Ed by G. Tesauro, pp 231-238, 1995.
- [4] Sridhar D.V., Bartlett E.B., Seagrave R.C., "An information theoretic approach for combining neural network process models", Neural Networks, Vol. 12, pp 915-926, Pergamon, Elsevier, 1999.
- [5] Jordan M. I. and Xu L., "Convergence Results for the EM Approach to Mixture of Experts Architectures", Neural Networks, Vol. 8, N° 9, pp 1409-1431, Pergamon, Elsevier, 1995.
- [6] Bruske J., Sommer G., Dynamic Cell Structure, Adv in Neural Inf Processing Systems 7, The MIT Press, Ed by G. Tesauro, pp 497-504, 1995.
- [7] Sang K. K. and Niyogi P., Active learning for function approximation, in Neural Information Processing Systems 7, The MIT Press, Ed by G. Tesauro, pp 497-504.
- [8] Madani K., Chebira A., "A Data Analysis Approach Based on a Neural Networks Data Sets Decomposition and it's Hardware Implementation", PKDD 2000, Lyon, France, 2000.
- [9] A. Chernoff, "Estimation of a multivariate density", Annals of the Institute of Statistical Mathematics, vol. 18, pp. 179-189, 1966.
- [10] A. Bhattacharya, "On a measure of divergence between two statistical populations defined by their probability distributions", Bulletin of Calcutta Maths Society, vol. 35, pp. 99-110, 1943.
- [11] J. Lin, "Divergence measures based on the Shannon entropy", IEEE Transactions on Information Theory, 37(1):145-151, 1991.
- [12] K. Matusita, "On the notion of affinity of several distributions and some of its applications", Annals Inst. Statistical Mathematics, 19:181-192, 1967.
- [13] E. Parzen, "On estimation of a probability density function and mode", Annals of Math. Statistics, vol. 33, pp. 1065-1076, 1962.
- [14] W.E. Pierson, "Using boundary methods for estimating class separability", PhD Thesis, Dept of Elec. Eng., Ohio State Univ., 1998.
- [15] A. Kohn, L. G. Nakano, and V. Mani, "A class discriminability measure based on feature space partitioning", Pattern Recognition, 29(5):873-887, 1996.
- [16] I. Budnyk, A. Chebira, K. Madani, ZISC Neural Network Base Indicator for Classification Complexity Estimation, ANNIIP (2007), pp. 38-47.
- [17] Ivan Budnyk, El khier Bouyoucef, Abdennasser Chebira, Kurosh Madani, "A Hybrid Multi-Neural Network Structure Optimization Handled by a Neurocomputer Complexity Estimator", Proceedings of International Conference on Neural Networks and Artificial Intelligence (ICNNAI 2008), ISBN : 978-985-6329-79-4, Minsk, Byelorussia, May 27 – 30, 2008, pp. 310-314.
- [18] Laboratory IBM France: ZISC® 036 Neurons User's Manual. Version 1.2. Component Development (1998).
- [19] G. De Tremiolles, "Contribution to the theoretical study of neuro-mimetic models and to their experimental validation: a panel of industrial applications", *Ph.D. Report*, University of PARIS 12, 1998 (in French)
- [20] D. P. Feldman, J. P. Crutchfield, Measure of statistical complexity. Why? Phys. Lett. (1998) A 238, pp. 244-252.
- [21] I. S. Mehmet, Y. Bingul, K. E. Okan, Classification of Satellite Images by Using Self-organizing Map and Linear Support Vector Machine Decision Tree, 2nd Annual Asian Conference and Exhibition in the field of

GIS, (2003).

- [22] H. Chi, O.K. Ersoy: Support Vector Machine Decision Trees with Rare Event Detection, International Journal for Smart Engineering System Design (2002), Vol. 4, pp. 225-242.E.
- [23] E. Bouyoucef, Contribution à l'étude et la mise en œuvre d'indicateurs quantitatifs et qualitatifs d'estimation de la complexité pour la régulation du processus d'auto organisation d'une structure neuronale modulaire de traitement d'information, PhD Thesis report, University Paris XII, LISSI (2007), in French.



Ivan Budnyk obtained the Masters of Information Control Systems and Technologies in June 2004, from National University of "Kyiv-Mohyla Academy" (NaUKMA), Ukraine. Currently he is working toward his Ph.D. at Image, Signal and Intelligent Systems Laboratory

(LISSI / EA 3956) of PARIS XII University with Prof. Kurosh Madani. His main research interests are Artificial Neural Networks based self-organizing information processing systems and complexity estimation techniques.



Dr. El khier Bouyoucef received his Master of Telecommunication Science degree from UVHC University, Valenciennes, France, in 2003. From October 2003 to October 2007, he has prepared his Ph.D. in Images, Signals and Intelligent Systems Laboratory

(LISSI / EA 3956) of PARIS XII – Val de Marne. He received his Ph.D. in October 2007. From October 2007 to September 2008 he has worked as Post-Doc researcher at LISSI. His research topic and interests deal with complexity estimation techniques and self-organizing ANN based structures for complex patterns classification.



Dr. Abdennasser Chebira received his Ph.D. degree in Electrical Engineering and Computer Sciences from PARIS XI University, Orsay France, in 1994. Since September 1994 he works as Professor Assistant at Senart Institute of Technology of

PARIS XII – Val de Marne University. He is a staff researcher at Images, Signal and Intelligent Systems Laboratory (LISSI / EA 3956) of this University. His current research works concern self-organizing neural network based multi-modeling, hybrid neural based information processing systems; Neural based data fusion and complexity estimation.



Prof. Kurosh Madani received his Ph.D. degree in Electrical Engineering and Computer Sciences from University PARIS XI, Orsay, France, in 1990. From 1989 to 1990, he worked at Institute of Fundamental Electronics (IEF / CNRS) of PARIS XI University. In 1990, he joined Creteil-Senart Institute of Technology of University PARIS XII – Val de Marne, Lieusaint, France, where he worked from 1990 to 1998 as assistant professor. In 1995, he received the DHDR Doctor Habilitate degree (senior research Dr. Hab. degree) from University PARIS XII – Val de Marne. Since 1998 he is Chair Professor in Electrical Engineering of Senart Institute of Technology of University PARIS XII. From 1992 to 2004 he has been head of Intelligence in Instrumentation and Systems Laboratory (I2S / JE 2353) located at Senart Institute of Technology. Since 2005, he is head of one of the three research groups of Image, Signal and Intelligent Systems Laboratory (LISSI / EA 3956) of PARIS XII University. He has worked on both digital and analog implementation of processors arrays for image processing, electro-optical random number generation, and both analog and digital ANN implementation. His current research interests include large ANN structures modeling and implementation, hybrid neural based information processing systems and their software and hardware implementations, design and implementation of real-time neuro-control and neural based fault detection and diagnosis systems. Since 1996 he is a permanent member (elected Academician) of International Informatization Academy. In 1997, he was also elected as Academician of International Academy of Technological Cybernetics.