ISSN: 1327-2314

International Journal of Knowledge-Based Intelligent Engineering Systems

Volume 7 Number 1, January 2003

CONTENTS

Submitted Papers

Emergence in Combined System Structure of Rough Set Theory and Neural Networks

Yasser Hassan, Ayahiko Niimi, and Eiichiro Tazaki
Department of Control and System Engineering,
Toin University of Yokohama,
1614 Kurogane-cho, Aoba-ku, Yokohama 225-8502,
Japan.

Abstract
AI has made great strides in computational problem solving using explicitly represented knowledge extracted from the task. If we continue to use explicitly represented knowledge exclusively for computational problem solving, we may never computationally accomplish a level of problem solving performance equal to humans. The need for more effective methods to generate and maintain other global nonfunctional properties suggests an approach analogous to those of natural processes in biological systems, social behavior, and economic systems in generating emergence properties. Emergence system allows the constraints of the task to be represented more naturally and permits only pertinent task specific knowledge to emerge in the course of solving the problem. The paper describes some basics of emergence system and its implementation in the combination system of Rough Set Theory and Artificial Neural Networks. We will present the demonstrations and guidelines on how to exploit emergence intelligence to extend the problem solving capabilities of this combination.

1. Introduction

Knowledge discovery in database (KDD) has been defined as "The nontrivial extraction of implicit, previously unknown, and potentially useful information from data" [10, 14]. In recent years numerous successful applications of rough set methods for knowledge discovery in database have been developed [2, 4, 6, 10, 11, 13, 14, 15]. In another direction there has been a rapid development in our understanding of the detailed mechanisms underlying the emergence of intelligent behavior [1, 3, 4, 7, 12].

The paper is primarily concerned with identifying and analyzing of some well-defined types of emergence that occur in the combination of rough set theory with artificial neural networks [10].

In this method, the behavior of the overall system emerges from the interactions of the quasi-independent computational agents or neurons. Each agent contains the entire specification for its behavior, which includes interactions between it and its computational environment and other agents. Thus unlike traditional systems modeling [7], there is no overall controlling entity or programs that orders or otherwise constrains the interaction between agents.

The main issue tackled in this paper is auto-adaptation occurred in this method that generates the behaviors from the study of local interactions between agents and the environment. Some key issues associated with the understanding and representation of such emerging behaviors in multi-agent systems are introduced.

The paper is structured as follows. We propose to begin in section 2 with a brief introduction to Emergence System. Then we recall basic rough set preliminaries in Section 3. In fourth section we present the concept of incorporating rough set methods into construction of the neural networks by using so called rough neuron. The demonstrations and guidelines of emergence properties in the combination system are described in Section 5. Section 6 show how to exploit emergence properties to extend the problem solving capabilities in the combination of rough set theory and artificial neural networks. Some applications are presented in Section 7. The paper concludes in Section 8 with directions for further research on the considered topics.

2. Emergence System

We may be able to say exactly what 'emerges' in a particular case than we generally can in the case of real-world systems. Emergence is generally understood to be a process that leads to the appearance of structure not directly described by the defining constraints and instantaneous forces that control a system. We can define emergence system [3, 7, 12] as: 'system behavior that comes out of the interaction of many participants' or 'local interactions creating global properties'. Some of the most engaging and perplexing natural phenomena are those in which highly structured collective behavior emerges over time from the interaction of simple subsystems.

Using emergence intelligence allows the removal of explicit knowledge that is a natural consequence of the problem solving process interacting with the task environment. By allowing the task environment to be an integral component of the problem-solving algorithm [1], all the natural constraints, including those too subtle for the knowledge engineer to extract, are available to the algorithm and emerge at appropriate moments while solving problems.

The studying of emergence properties in mathematically well-defined systems may be particularly useful in constructing a topology of emergence. Emergence as the existence of properties of a system is not possessed by any of its parts [3]. This, of course, is so ubiquitous a phenomenon that it's not deeply interesting. It probably will help to focus on a few core examples of emergence systems [7]:

(A) The game of Life: High-level patterns and structure emerge from simple low-level rules (Cellular Automata).

(B) Connectionist networks: High-level "cognitive" behavior emerges from simple interactions between dumb threshold logic units (Neural Networks).

(C) Evolution: Intelligence and many other interesting properties emerge over the course of evolution by genetic recombination and mutation operators (Genetic Algorithm).

3. Rough Set Theory

The underlying ideas of rough set theory, proposed by Zdzislaw Pawlak in the early 1980's [8], have been developed into a manifold theory for the purpose of data and knowledge analysis, due to a systematic growth of interest in this theory in scientific community. The primary goal of rough set theory has been outlined as a classificatory analysis of data [9].

The main paradigm of rough set theory states that the universe of known objects is assumed to be only source of knowledge about a domain specified by our needs. Data based reasoning is then concerned with the analysis of dependencies between features labeling known cases with values from some pre-defined domain. Let us represent any sample of known data as an information system IS = (U, A), whose columns are labeled by attributes, rows are labeled by objects of interest and entries of the table are attributes values, where U is non-empty finite set of objects called universe, and A is non-empty finite set of attributes.

Decision table DT = (U, A $\cup$ {d}) is a special form of information system, where A is called condition attributes, and $d \notin A$ is called decision attribute. If V_a be the value set for attribute $a \in A$ called the domain of a, then attribute $a \in A$ is a map;

$$a: U \rightarrow V_a, \forall a \in A.$$

Table1 has an example of decision table where the set of attributes A = {Sex, Clinical Stage}, values of decision attribute is $V_{infection}$={Yes, No}, and U={P1, P2,…,P10}.

Let $X \subseteq U$ be a set of objects and $B \subseteq A$ be a set of attributes, the indiscernibly relation can be defined as:

$$I(B) = \{(x, y) \in U \times U: a(x) = a(y), \forall a \in B\}.$$

Objects x, y satisfying the relation I(B) are indiscernible by attributes from B, I(B) is reflexive, symmetric, and transitive.

Table 1: an example.

U	Sex	Clinical Stage	Infection
P1	F	T	Yes
P2	M	T	Yes
P3	F	B	No
P4	M	T	Yes
P5	M	T	Yes
P6	M	T	No
P7	F	T	Yes
P8	F	B	No
P9	M	B	No
P10	M	T	No

An order pair AS = (U, I(B)) is called an approximation space. According to I(B), we can define two crisp sets $\underline{B}X$ and $\overline{B}X$ called lower and upper approximation of the set of objects X in the approximation space AS as:

$$\underline{B}X = \{x \in U: I_B(x) \subseteq X\}, \text{ and}$$

$$\overline{B}X = \{x \in U: I_B(x) \cap X \neq \phi\}.$$

$\underline{B}X$ consists of all objects of U that can be with certainty classified as elements of set X given the knowledge represented by attributes from B, and $\overline{B}X$ consists of all objects that can be possibly classified as elements of set X employing the knowledge represented by attributes from B. The difference BNB(X) = ($\overline{B}X$ - $\underline{B}X$) is called boundary of X, which contains all objects that cannot be classified either to X or complement of X given knowledge B. Pawlak [8] defined a rough set to be a family of subsets of a universe that has the same lower and upper approximations. From **Table 1**, the upper and lower approximation of the decision attribute "infection" can be formatted as follows:

$$\underline{X}_{\inf ection=yes} = \{P1, P7\}$$

$$\overline{X}_{\inf ection=yes} = \{P1, P2, P4, P5, P6, P7, P10\}$$

$$\underline{X}_{\inf ection=no} = \{P3, P8, P9\}$$

$$\overline{X}_{\inf ection=no} = \{P2, P3, P4, P5, P6, P8, P9, P10\}$$

Decision rules can be perceived as data patterns, which represent relationship between attributes values of a classification system. If DT = (U, A $\cup$ {d}) is a decision table and V= $\cup\{v_a: a \in A\} \cup v_d$, is a set of values for attributes, then the decision rule is a logical form: IF α THEN β, or can be written [11] as (α, β), where α is a condition part of the rule, it is a conjunction of selectors: for nominal attributes take the form: ($a_1=v_1$ AND ….AND $a_n=v_n$) and for numerical attributes take the form: $v_1 < a < v_2$, and β is the decision part: ($d = v_d$), it usually describes the predicted class. We can define the set of rules as:

$$\text{Rule\_Set} = \{(\alpha_i, \beta_i): i=1,\ldots,k\}.$$

There exist several measurements in order to evaluate the decision rule. The classification accuracy and coverage of rule r are defined as follows [10, 11, 13, 14]:

$$Acc(r) = \frac{|\sup(r) \cap D|}{|\sup(r)|}$$

$$Cov(r) = \frac{|\sup(r) \cap D|}{|D|}$$

where |A| is the cardinality of a set A, Acc(r) is the classification accuracy of the rule r, Cov(r) is the coverage of the rule r, sup(r) is the number of cases that match the condition part of rule r, and |D| is the number of cases that match the decision part of rule r. It is clear that Acc(r) and Cov (r) belong to the interval [0,1].

4. Combination of Rough Set Theory and Neural Networks

This section is an attempt to summarize an approach aimed at connecting rough set theory with artificial neural networks. Artificial neural network in its most general form is attempted to produce systems that work in a similar way to biological nervous systems. The nature of connections in neural networks and data exchange through the connections depend upon the application.

Driven by the idea of decomposing the set of all objects into three parts: the lower approximation, the boundary region and the upper approximation with respect to a given set X of objects in the decision table, Ligras [10] introduced the idea of rough neuron to construct network called Rough Neural Network.

Each rough neuron r is a pair, one for the upper bound called upper neuron $\bar{r}$ and another for lower bound called lower neuron $\underline{r}$. Those two neurons can exchange information between each other and between other rough or conventional neurons; so rough neural network consists of both conventional and rough neurons.

The outputs of a rough neuron r depending on a pair of neurons: lower neuron $\underline{r}$ and upper neuron $\bar{r}$, are calculated using formula:

$$output_{\bar{r}} = \max(f(input_{\bar{r}}), f(input_{\underline{r}}))$$

$$output_{\underline{r}} = \min(f(input_{\bar{r}}), f(input_{\underline{r}}))$$

where f stands for any transfer function, for example: sigmoid function, which takes the form:

$$f(x) = \frac{1}{1+e^{-\beta x}}$$

where β is the coefficient called gain, which determines the slope of the function.

The connections between conventional neurons in rough neural network are made as in usual case. While connection between rough neuron and conventional one is made as connecting lower neuron $\underline{r}$ and upper neuron $\bar{r}$ separately. Two rough neurons in the network can be connected to each other using either two or four connections. A rough neuron r is said to be fully connected to rough neuron s, if $\underline{r}$ and $\bar{r}$ are connected to both $\underline{s}$ and $\bar{s}$. If there exist two connections only from neuron r to neuron s, then the two neurons are partially connected. If a rough neuron r excites the activity of neuron s (i.e. increase in the output of r will result the increase in the output of s), then we connect only $\bar{s}$ with $\bar{r}$ and $\underline{s}$ with $\underline{r}$. In the opposite situation, if r inhibits the activity of s (i.e. increase in the output of r corresponds to the decrease in the output of s) we connect only $\underline{s}$ with $\bar{r}$ and $\bar{s}$ with $\underline{r}$.

If two rough neurons are partially connected, then the excitatory or inhibitory nature of the connection is determined dynamically by polling the connection weights. If a partial connection from a rough neuron r to another rough neuron s is assumed to be excitatory and weights of both the connections are negative, then the connection from r to s is changed from excitatory to inhibitory. On the other hand, if r is assumed to have as inhibitory partial connection to s and weights of both the connections are positive, then the connection from r to s is changed from inhibitory to excitatory.

Now we will call each neuron and link in the network as "agent". From a collection of agents obeying explicit instructions (weight-modification and signal propagation algorithms), learning and pattern-recognition emerge. The agent receiving this information can describe objects using its own attributes. In this way a decision table is created and the receiving agent can extract approximate description of concept.

If the agent j (conventional or rough neuron) connects to agent i (conventional or rough neuron), then the collected weighted input of agent i is calculated as:

$$input_i = \sum \omega_{ij} \times output_j$$

where ω_{ij} is the connection weight between agents i and j.

Let the network here is represented by a set of agents Ag={ag_1,...,ag_p}. Any agent from Ag is equipped with an information system $IS_{ag} = (U_{ag}, A_{ag})$ where U_{ag} is a set of objects and A_{ag} is a set of attributes associated with agent ag. The decision table is a pair $DT_{ag} = (U_{ag}, A_{ag} \cup \{d_{ag}\})$ for any agent ag∈Ag where d_{ag} is the local decision attribute. The lower and upper approximations of any concept X defining by agent ag ∈ Ag with respect to condition attributes of DT_{ag} describe the vagueness in understanding of X by agents from Ag. Every agent is autonomous in the sense it is not under the control of a supervisor: all its decisions are derived from embodied rules depending only upon local information accessible to the agent

The agents differ in their learned behavior, and their consequential experience and performance. The learning process for the network is based on any general learning scheme, so the weights in the network are adjusted according to the general equation:

$$\omega_{ji}^{new} = \omega_{ji}^{old} + \alpha(t).g(input_i)$$

where g is any transfer function, $\alpha(t)$ is a learning factor, which starts with a high value at the beginning of the training process and is gradually reduced as a function of time. E.g. the weights adjusted according to a simple bachpropagation-learning scheme take the form:

$$\omega_{ji}^{new} = \omega_{ji}^{old} + \alpha.err_i.f'(input_i)$$

where f' is the derivative of sigmoid function, α is the learning coefficient and err_i is an error for agent i. Due to the properties of sigmoid function, calculation of $f'(x) = f(x).(1 - f(x))$ is easy.

Let N(ag) be a function, which determines the set of immediate neighbors agents of the agent ag $\in$ Ag. The function N can be defined as:

N(ag) = {ag_1: $ag_1 \in$ Ag and ag_1 is immediate neighbor of ag}.

Any agent ag from the network can use function N to determine which agent will interact with it.

The communication between agents is provided by mapping E(ag, N(ag)) such that:

For x $\in U_{ag}$, the value E(ag, N(ag))(x) $\in U_{N(ag)}$.

According to function E each agent can send or receive information through immediate neighbor agents. Perturbation and undirected communications can be beneficial to the success and performance of emergence system. The information can only be transmitted through sequences of immediate neighbor communications. The undirected communications and absence of complete information permits rough neural network that sometimes but not always succeed in satisfying its goals.

Figure1 shows an example of rough neural network, which consists of the input layer, which the pattern is presented, distributes the pattern throughout the net, and propagates the pattern down their connections to the middle layers (one or more hidden layers). The pattern is modified by weights associated with each connection. The agents in the hidden layer pass on the pattern in an appropriate manner, again modified by weight connections, to evoke the desired response in the output layer. The operations are natural in a sense that they correspond to different views on global approximation space represented by the network. These new approximation spaces can be used for better description of concepts.

Now let we give small example to show how to use rough neuron. **Table 2** has example medical data, where the attribute values take the form of rough pattern (minimum and maximum values). The data consists of 6 objects each of which has values for 3 attributes { total PSA, PSA density, PSA TZ density} and one decision {Patho} with value set {0,1}. "0" means not infected and "1" means infected.

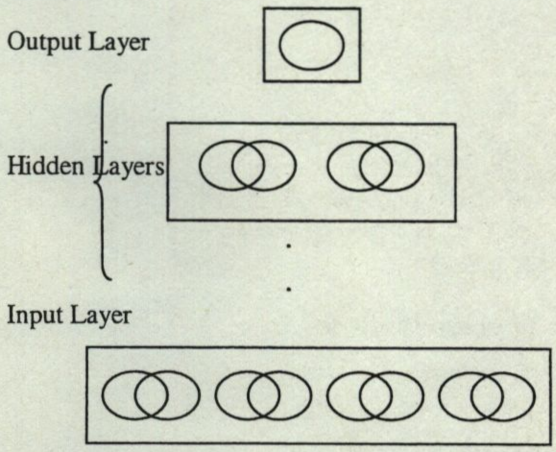

Figure1: An example of Rough Neural Network

The constructed network is shown in **Figure2**. We use here 3 rough neurons as input layer corresponding to 3 condition attributes. One hidden layer has 2 rough neuron and one conventional neuron in output layer for decision attribute "Patho". The connections between rough neuron in this network are taken as full connection.

5. Guidelines for Emergence System in Combination of Rough Sets and Neural Networks

We provide some guidelines for directed introduction of emergence properties into the combination of rough set theory and artificial neural networks. In this system the interaction of the dynamic representation and non-positional interpretation provides some innate emergence properties that assist in the acquisition of solutions [1, 7]. Those properties emerge not because they were designed into the neural network itself, but because the dynamics of the method determine them to be useful or necessary for success. This combination represents emergence technology on two levels. First, in the learning process itself, the ability to recognize the pattern-set (embodied in the connectional topology and weights) emerges from the interactions of agents (neurons and links). Second, once the net is trained, the appropriate pattern at the output layer emerges from the interactions between agents in the static network [7]. Four characteristics of the system are observed to define the emergence properties:

i. No agent controls globally the dynamic of all the

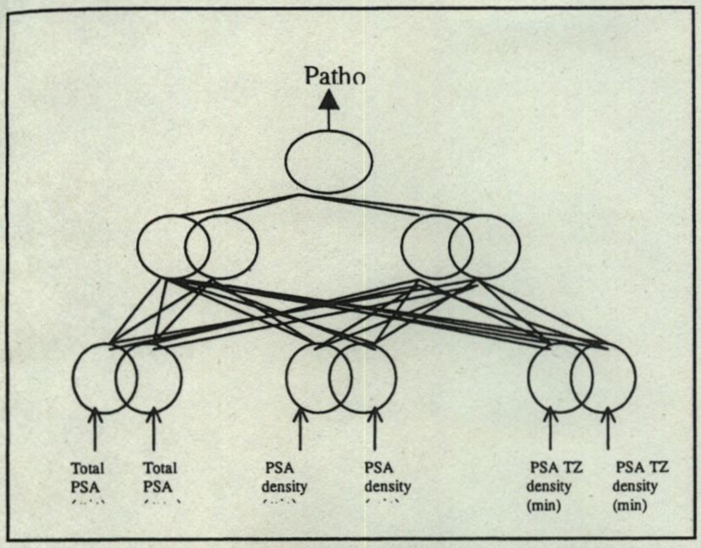

Figure 2: The rough neural network model constructed on example data in table2.

system. The agents are limited and they are unaware of some parts of the global system. So each agent has a local environment. Each participant can neither read nor write directly to agents, in other word, the system can make use of neither global visibility nor central control.

ii. The agents act and modify locally this environment. Each agent has only a partial view of others and the environment, in which it's immersed, and in the absence of global control, every agent must be able to communicate with its immediate neighbors without depending on the knowledge of the overall network topology.

iii. Interaction is a basis mechanism for agents, and the system considers a result as emerging from exchanges between agents. Each agent can communicate directly only with a number of immediate neighbors that is less than the total number of agents in the system, which called local interactions.

iv. Emergence of global solutions is adaptation of agent's behavior to knowledge and environment. This hypothesis can be overcome in considering each component as a whole system, in which the sub-components are using the same self-organization method. Each neuron consists of a pair of lower neuron and upper neuron. If we consider the rough neuron as whole system, so lower and upper neurons are considered as sub-component and each can use the same self-organization method.

The network here is a group of agents, none of which can deal with a difficulty alone, but only do so when each cooperates.

6. Exploiting Emergence Intelligence in Rough Neural Network

We will describe example of modification to the system that harness inherent dynamics for emergence intelligence in problem solving.

Two kinds of dynamic evolution could be considered: modification of the structure by re-organization of the acquaintances network or modification of behavior. The goal here is to demonstrate the self-organization of the rough neural network. Consequently, rough neural network to be used in an emergence way and can perform its roles, a duplication system is required. We will define a duplicate operator that an agent can use to duplicate itself. In the same manner, removable operator can be defined where the agent has the ability to die and delete itself from the network structure.

This idea was first discussed in [5] as adaptation of neural networks structure but in different way, where this process was under control of overall system error. But in our combination system, this process is local for agent and no global control exist, so each agent has the ability to produce new agent and also remove itself from the system under local control only. To define local control for each agent, we assign a fitness value for each agent. This fitness value is not only depend on agent performance, but also on how this agent "better" against other agents. Let define a fitness function in a simple form as summation of two terms: first term is the performance of an agent, it is the average between the input and the output values of this agent, and second term for measure how this agent better against other agents in the network. So the fitness function can take the form:

$$F_{ag} = \alpha_1 . V + \alpha_2 . B,$$

Where F_{ag} is a fitness function for agent ag $\in$ Ag, α_1 and α_2 are parameters, V is the average of input and output values for agent ag $\in$ Ag, and B is how the agent ag "better" against other agents in Ag.

Depending on the value of fitness function F_{ag}, the neuron can be spilt into two using duplicate operator, i.e. it produces another neuron to the exactly same interconnection of the network as its parent neuron's attributes are inherited. If a neuron does not form the correct interconnections between other neurons or it is a redundant in the network, then it will die. We will limit this property to hidden layers only, and input layer and output layer are fixed. We can say that rough neural network is not designed but evolved. From generation to generation, the system learns its structure through interactions with its environment.

By embedding this modification within an ongoing evolutionary scenario and by allowing processes of agent/environment interaction to take place within each agent's lifetime, we can obtain performance, which is reliably good.

7. Application

The purpose of the experiments described in this section is not to propose better methods for solving the particular problem, while our new model of rough neural network provides better results. But the model used in the experiment was too simplistic to make any concrete recommendations. Instead this section tries to verify the emergence properties in this combination.

The neural networks have shown to be more effective than the existing methods for estimation of many applications, so we choose a data contains information about the representation of students from California State Univeristy taking the equivalent of a 15-unit course load, and the task is to predict the volume of students for year 2000 using data about this volume from the last years. The input to the rough neural network model consists of rough pattern, i.e. upper and lower bounds of yearly volumes of students. We divided the data into sections each section for three years, and take the upper and lower bound of values that exist in each section. The data begin from 1991 until 2000. So the sections were determined as follows:

Section1: 1991-1993,

Section2: 1994-1996,

Section3: 1997-1999.

For comparing the results, we construct three models for neural networks: first model for rough neural network with modification that described in this paper, second model for standard rough neural network as defined in [10], as well as model for conventional neural network. For first and second models (proposed model and standard rough neural network), The network has three input rough neurons, each of which for one section, and one hidden layer with eight rough neurons. Since the output is a unique value, the output layer used one conventional neuron. The important difference in rough neural network approach is that they take as inputs the upper and lower bounds for attributes. So in fact this network has twice the number of neurons as compared to the conventional one.

For conventional neural network, the inputs are the average value for each section of data. The model consists of three input neurons, one hidden layer with eight neurons, and one neuron in output layer. Now we mention the discussion of experimental result with three models.

Initially, for each network, the connections are assigned somewhat random weights. The training set of input is presented to the network several times.

Figures 3, 4, and 5 show the average reduction error of global output during training process every 1000 generations for each model of neural networks. From the figures, we observe that the average error of new model of neural network is going more natural than standard rough neural network and conventional neural network.

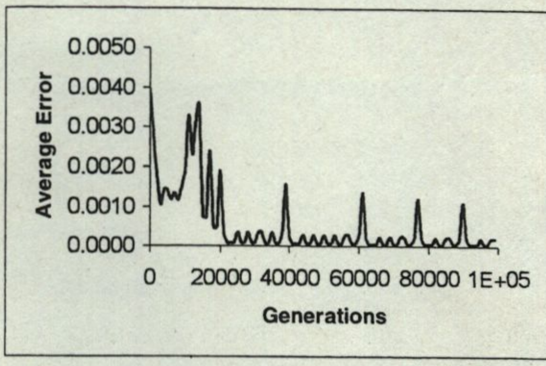

Figure 3: The average error through 100,000 generations produced by the proposed new rough neural network.

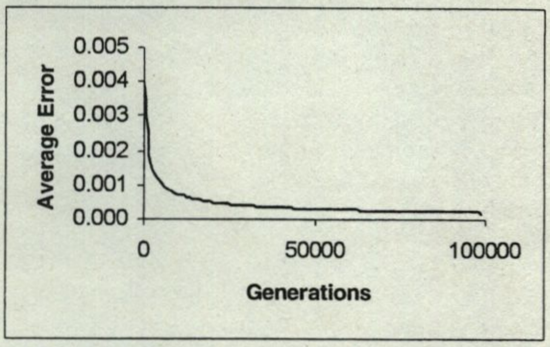

Figure 4: The average error through 100,000 generations produced by standard rough neural network.

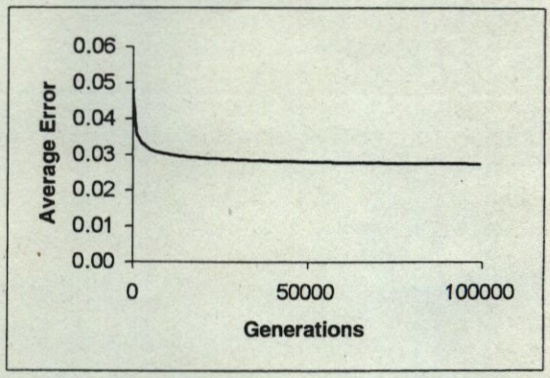

Figure 5: The average error through 100,000 generations produced by conventional neural network model.

Table 3 shows the comparison between average errors of each neural network model through all generations. From the table we observe that second model (standard rough neural network) has best average error and it is very close to the average error of our new model as well as they are better than conventional neural network. **Table 4** shows the maximum error through three models of neural networks through 100,000 generation, our proposed model and standard rough neural network have the same value of maximum error and it is better than the error value of conventional one. **Table 5** shows the minimum error values through 100,000 generations for each model of neural networks. From the table we observe that our proposed model provides very good result, where the value produced from our new model is better than values of standard rough neural network and conventional neural network.

Table 3: Average error through 100,000 generations for proposed model of rough neural network, standard rough neural network, and conventional neural network.

Proposed Model	Standard Rough Neural Network Model	Conventional Neural Network Model
0.00055	0.00044	0.02866

Table 4: Maximum error through 100,000 generations for proposed model of rough neural network, standard rough neural network, and conventional neural network.

Proposed Model	Standard Rough Neural Network Model	Conventional Neural Network Model
0.00397	0.00397	0.04804

Table 5: Minimum error through 100,000 generations for proposed model of rough neural network, standard rough neural network, and conventional neural network.

Proposed Model	Standard Rough Neural Network Model	Conventional Neural Network Model
0.000046	0.000203	0.027472

To see more in our proposed model, we observe some emergence properties in this model. From generation to generation, the group of agents interacts with each other. Each agent has the ability to produce itself with the same connection and also has the ability to die and delete itself from the network structure. Through the interactions between agents, the weights of connections, i.e. the attributes of agents are modified. In this experiment for new model of rough neural network, we find through 100,000 generations, the largest using one of duplicate operator is the agent number 12, which it uses as 31% through overall agents, and for removable operator neurons number 15 and 16 use it as the same rate 31% against all agents in the network.

Regarding the application we introduced, the combination of rough set theory and artificial neural network represents emergence computation in the strict sense.

8. Conclusions

In recent years, an approach termed emergence computation has gained popularity in a variety of fields. This paper thus provides a "big-picture" story about the way in which the development of complex intelligent behaviors might involve evolutionary processes, learning processes, agent/environment interaction, and representation development. The learning method provided by rough set forms a bridge between the neural network paradigms on the one hand and the representation list paradigm on the other.

We begin in this paper with introduction to emergence system and illustrate what are the properties of emergence system with some examples of existing emergence systems. Followed that, we summarized the approach of combining rough set theory with artificial neural networks, which called rough neural network and give new description to this combination from rough set view. In next part of this paper, we illustrate the emergence properties that exist in rough neural network, and show how to exploit emergence properties to extend the problem solving capabilities in the combination of rough set theory and artificial neural networks. Where we add two new operators: duplicate and removable operators, and define new function to assign fitness value for each agent in the network. By using these new modifications, rough neural network can perform its roles and be used in emergence way. In last part of the paper, we describe in details some experiments with real life data from California State Univeristy. The task of experiment is to predict the volume of student using data about the volumes from last years. We compare between conventional neural network, rough neural network, and new model of rough neural network. We need to continue this direction of research where we will extend this idea to be in general case for rough set theory when it combines with any other method.

Reference

[1] Angeline, Peter J., Genetic Programming and Emergent Intelligence, in Kinnear, Jr., K.E. (Edi.), Advances in Genetic Programming, Mit Press, 1994.

[2] Dong, J.Z., Zhong, N., and Ohsuga, S., Rule Discovery by Probabilistic Rough Induction, Japanese Soc. Artificial Intelligence, 2000, pp 274-286.

[3] Gotts, N., Emergent phenomena in large sparse random arrys of Conway's 'Game of Life', International Journal of System Science, vol. 31 no. 7, pp 873-894, 2000.

[4] Hassan, Y., and Tazaki, E., Emergence in Combined System Structure of Rough Set Theory and Genetic Programming, Submitted to KES International Conference, 2001.

[5] Lee, T., Structure Level Adaptation for Artificial Neural Network, Kluwer Academic Publishers, 1991.

[6] Nakayama, H., Hattori, Y., and Ishii, R., Rule Extraction based on Rough Set Theory and its Application to Medical data Analysis, IEEE SMC'99 Conference Proceedings, 1999.

[7] Olson, R., and Sequeira, R., Emergent computation and the modeling and management of ecological systems, Computers and Electronics in Agriculture 12, pp 183-209, 1995.

[8] Pawlak, Z., Rough Sets, International Journal of Computer and Information Science, Vol. 11, No. 5, 1982.

[9] Pawlak, Z., Rough Classification, International Journal of Man-Machine studies 20, 1984, pp 469-483.

[10] Polkowski, L., and Showron, A., Rough sets in Knowledge Discovery, Physica Nerlag, 1998.

[11] Polkowski, L., Tsumoto, S., and Lin, Y., Rough Set Methods and Applicatons, Physica Nerlag, 2000.

[12] Thornton, C., Brave mobots use representation: Emergence of Representation in Fight-or-Flight learning, Minds and Machines 7: pp 475-494,1997.

[13] Tsumoto, S., and Tanaka, H., Discovery of Approximate Medical Knowledge based on Rough Set Model, Principles of Data Mining and Knowledge Discovery. Second European Symposium, PKDD'98, pp 468-76,1998

[14] Ziarako, W., Ed. Rough Sets, Fuzzy Sets, and Knowledge Discovery, Springer Verlag, Berlin, 1994.

[15] Ziarako, W., Discovery Through Rough Set Theory, Communications of the ACM: Vol. 42. No. 11, 1999, pp 54-57.

Table 2: an example of rough values in decision table.

Objects	Total PSA (min)	Total PSA (max)	PSA density (min)	PSA density (max)	PSA TZ density (min)	PSA TZ density (max)	Patho
P1	2.07	3.84	0.07	0.13	0.29	0.53	1
P2	4	24	0.22	0.07	0.13	0.13	0
P3	6	20	0.4	0.07	0.13	0.13	1
P4	4	24	0.14	0.11	0.21	0.89	0
P5	2.07	3.84	0.29	0.07	0.13	0.13	0
P6	4	24	0.22	0.11	0.21	0.29	1