Emergent Model Based On Hybrid Rough Sets Systems

Yasser Hassan[*]
Eiichiro Tazaki[†]
Department of Control and System Engineering,
Toin University of Yokohama,
1614 Kurogane-cho, Aoba-ku, Yokohama 225-8502, Japan

The development of many knowledge discovery methods provided us with a good foundation for building hybrid systems based on rough set theory for knowledge discovery in a database. The need for more effective methods of generating and maintaining global nonfunctional properties suggests an approach analogous to those of natural processes in generating emergent properties. An emergent model 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. This paper describes some basics of emergent phenomena and its implementation in the hybrid system of rough sets combined with other methods. Further, demonstrations and guidelines are presented on how to exploit emergent intelligence and extend the problem-solving capabilities of these hybrid systems.

1. Introduction

In the ensuing years, we have witnessed a systematic and worldwide growth of interest in rough sets and their applications for knowledge discovery in databases [1–3]. In another direction, there has been a rapid development in our understanding of the detailed mechanisms underlying the emergence of intelligent behavior [4, 5].

In [6] we introduced a hybrid system of rough sets and genetic programming, and in [7] we presented the hybrid system of rough sets and cellular automata. In this paper, we present a general model of hybrid rough sets systems and apply this idea to the case of hybrid rough sets and neural networks, which is presented in [8]. In the hybrid systems of rough set theory, the behavior of the overall system emerges from the interactions of the quasi-independent computational components or *agents*. Each agent contains the entire specification for its behavior, including interactions between it and its computational environment and

[*]Electronic mail address: y_fouad@hotmail.com.
[†]Electronic mail address: tazaki@intlab.toin.ac.jp.

other agents. These agents cooperate to make local teams that are able to understand and apply any of the given local languages. Local teams are merged into a *scheme* (e.g., assembling scheme or organizational scheme) whose aim is to construct a global solution to a given specification. Members of a scheme are local assembling teams that are fused from some local teams of agents. Thus unlike traditional systems modeling [9], there is no overall controlling entity that orders or otherwise constrains the interaction between system components.

This paper is structured as follows. We propose the formalism necessary to describe notions of rough set theory in section 2. Section 3 provides a brief introduction to emergent systems. Section 4 presents attempts at using rough set theory as the base method for general hybrid systems. In Section 5, rough sets are combined with neural networks into one hybrid model. The paper concludes with section 6.

2. Rough set theory

In this section, we recall some notions related to information systems and rough sets. The rough set concept was introduced by Pawlak in [10]. One of its essential merits is its direct application to classification problems [11] founded on the assumption that each object is associated with some information (e.g., data or knowledge). Objects characterized by the same information are said to be *indiscernible* in the view of available data. This induces the indiscernibly relation (equivalence relation) which is the mathematical base of rough set theory.

Knowledge representation in rough sets is done *via* an information system. An information system is a pair $IS = (U, C)$, where U is a nonempty finite set of objects called the *universe*, and $C = \{c_1, \ldots, c_m\}$ is a nonempty finite set of mappings $c : U \to v_c$; each $c \in C$ is called a *condition attribute*. A *decision table* $DT = (U, C \cup D)$ is a special case of an information system introduced in rough set theory as a tool for representing data, where the major different feature is the distinguished attribute D, $(D \cap C = \phi)$ which is called the *decision attribute*.

Let $I \subseteq U \times U$ denote the indiscernibly relation on U which can be defined as:

$$I = \{(x, y) \in U \times U : b(x) = b(y), \forall b \in B\}, \tag{1}$$

where $B \subseteq C$ is a set of attributes. Objects x and y satisfying the relation I are indiscernible by attributes from B.

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

$$\begin{aligned} \underline{B}X &= \{x \in U : I_B(x) \subseteq X\} \\ \overline{B}X &= \{x \in U : I_B(x) \cap X \neq \phi\}, \end{aligned} \tag{2}$$

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

Table 1. An example of a decision table.

where $I_B(x)$ denotes the set of all objects indiscernible with x, that is, the equivalence class which is determined by x. $\underline{B}X$ consists of all objects that can be with certainty classified as elements of set X, and $\overline{B}X$ consists of all objects that can be possibly classified as elements of set X. The difference $BN_B(X) = (\overline{B}X - \underline{B}X)$ is called the *boundary* of X, which contains all objects that cannot be classified either to X or the complement of X. In [10] Pawlak defined a rough set to be a family of subsets of a universe that have the same lower and upper approximations.

Table 1 shows a decision table where the set of attributes C = {Sex, Clinical Stage}, the values of the decision attribute are $V_{\text{infection}}$ = {Yes, No}, and the set of objects is $U = \{P1, P2, \ldots, P10\}$.

From Table 1, the upper and lower approximations of the decision attribute "infection" can be formatted as follows:

$$\underline{X}_{(\text{infection=yes})} = \{P1, P7\}$$
$$\overline{X}_{(\text{infection=yes})} = \{P1, P2, P4, P5, P7, P10\}$$
$$\underline{X}_{(\text{infection=no})} = \{P3, P8, P9\}$$
$$\overline{X}_{(\text{infection=no})} = \{P2, P3, P4, P5, P6, P8, P9, P10\}.$$

The decision table is called *consistent* provided that a functional dependency between the set of condition attributes and decision attributes is fulfilled, otherwise the table is called *inconsistent*. That is, the table is inconsistent when it contains rows, which for equal values of condition attributes C, there are different values of decision D. It is possible to determine the range of consistency in data as [12]:

$$\text{Consist}_B(X) = \frac{\sum_j |\underline{B}X_j|}{|U|} \tag{3}$$

where $|A|$ is the cardinality of a set A.

One of the problems related to the practical application of rough set methods is whether the whole set of attributes is necessary and if not,

how to determine the simplified and still sufficient subset of attributes equivalent to the original. The rejected attributes are redundant since their removal cannot worsen the classification. There are usually several such subsets of attributes and those, which are minimal with respect to inclusion, are called *reduct*.

Decision rules can be perceived as data patterns, which represent the relationship between attribute values in the classification system. If $V = \cup\{v_c : c \in C\} \cup v_D$ is a set of values for attributes, then the decision rule is a logical form: IF$[(c_1 = v_1)\& \ldots \&(c_n = v_n)]$ THEN $(D = v_D)$. The decision rule is true in the decision table if $\{u \in U : c_1(u) = v_1, \ldots, c_n(u) = v_n\} \subseteq \{u \in U : D(u) = v_D\}$.

The classification accuracy and coverage of a rule r are defined as follows [3]:

$$\mathrm{Acc}(r) = \frac{|\mathrm{sup}(r) \cap S|}{|\mathrm{sup}(r)|} \tag{4}$$

$$\mathrm{Cov}(r) = \frac{|\mathrm{sup}(r) \cap S|}{|S|} \tag{5}$$

where Acc(r) is the classification accuracy of the rule r, Cov(r) is the rule coverage of r, sup(r) is the number of cases that match the condition part of the rule r, and $|S|$ is the number of cases that match the decision part of the rule r.

We are also interested in the measurements of the set of rules [13]. The simple strength of a set of rules is defined as:

$$\mathrm{strength}(X_j, u_t) = \frac{|\mathrm{MRul}(X_j, u_t)|}{|\mathrm{Rul}(X_j)|}, \tag{6}$$

where $u_t \notin U$ is a tested object, $\mathrm{Rul}(X_j)$ is the set of all rules for decision class X_j, and $\mathrm{MRul}(X_j, u_t) \subseteq \mathrm{Rul}(X_j)$ is a set of rules that matches the tested object u_t for decision class X_j.

3. Emergent phenomena model

Emergence of a system, in a broad sense, is said to be the properties or behavior of a system that cannot easily be predicated from its internal properties [14]. From an intuitive point of view, emergence is used as a name for the creation of new structures and properties, in other words, "the whole is more than the sum of its parts [15]." Emergent systems [4] are philosophically and methodologically different from traditional approaches. The simple definition of an emergent system can be formulated as: System behavior that comes out of the interaction of many participants. The emergent model replaces the traditional high-level control scheme of functional modularization of behavior generating modules [9]. The characteristics of emergent models are examined. Instead of

high-level computation of behavior, the bottom-up self-organization of simpler components produces a variety of behaviors depending on the interaction of the component or agent with its environment. The structure coupling of the system's component and its environment is this source of the behavior, and not just this component's control system alone [9].

Emergence as the existence of properties in a system is not possessed by any of its parts [4]. This, of course, is so ubiquitous a phenomenon that it is not deeply predicated from the system's parts. This phenomenon can be observed in a few core examples of "emergence" such as the following.

- *The Game of Life*. High-level patterns and structures emerge from simple low-level rules.
- *Connectionist networks*. High-level "cognitive" behavior emerges from simple interactions between dumb threshold logic units.
- *Evolution intelligence*. This and many other interesting properties emerge over the course of evolution by genetic recombination and mutation operators.

We construct an emergent model in this paper; this needs a set of rules, which are satisfied where we apply this model of emergence [15]. The following rules together can be considered as a test for emergence.

1. The environment and system are complex enough to allow the mechanisms of self-organization to occur in the situated interaction of system parts in the environment.
2. The system is self-replicating.
3. The system has been constructed by describing local elementary interactions between its components. There should be sufficient interaction to have these agents constructing shared semantics by virtue of their situated interactions with the world and each other.

These three points describe our rules for generally diagnosing emergence in a system. Some of the above points deserve further elaboration, or indeed invite debate. The emergent model has these characteristics.

- Direct communication only happens between local neighbors.
- There is no central control.
- Individual participant components are not able to view the state of the system from a global perspective.
- The environment has sufficient initial richness and structure to allow for embodied emergent classification of that environment/system coupling.

Let us now consider the mathematical foundation of emergence and emergent properties. If we have the model A, we use $\pi(A)$ to denote

the amount of computation required to simulate a system, and arrive at a prediction of the given phenomenon. Let $u(A)$ be the amount of computation required to arrive at the result by performing a creative analysis depending on our understanding of the symmetries of the system that help us to deduce the future state whilst circumventing most of the previously-required computation.

Let us consider an idealized means of prediction in that complex system, requiring an amount of computation $u_{\text{opt}}(A)$ based on perfect understanding [9]. Now, the related measure of understanding is given by:

$$\lambda = \frac{u_{\text{opt}}(A) - u(A)}{\pi(A)}. \tag{7}$$

Then, to consider a system as emergent, two conditions should be held:

1. $u(A) \geq \pi(A)$
2. $\lambda = 0$.

Condition 1 means that the global result of the emergent model is not equal to the sum of its parts. Therefore, the amount of computation required to simulate the system is less than the actual amount, which we predicate depending on the analysis of all the parts of the system. Condition 2 happens when $u_{\text{opt}}(A) = u(A)$, that is, it gives us the necessary condition of satisfying condition 1, where $u(A)$ should be the optimal when it is considered. Global nonfunctional properties of a system, including those needed to fulfill mission requirements and performance, often arise through the interactions among components of the system. Global properties that prevail for a system as a whole, but cannot or do not exist within individual components of that system, are called *emergent* properties [9]. To focus on these emergent properties, let us start out with a family of structures $\{S_i\}$, $i \in J$ (some index set, finite or infinite). Then we apply our observation mechanisms O to obtain the properties of the structures S_i as $O(S_i)$. Next we assign to the S_i a family of interactions q using the properties registered under the observation. Hence, we can get a new structure as follows:

$$S = R(S_i, O(S_i), q), \qquad i \in J \tag{8}$$

where R stands for the result of the construction process. Here, S is a second-order structure, which is induced from the first-order structures S_i. The interactions may be caused by structure themselves or imposed by external factors. At each level of construction, new behaviors or new properties may emerge, giving room for new interactions, and hence each level is necessary in order to get the previous level's properties. Therefore, the Nth order structure is defined as follows:

$$S^N = R(S^{N-1}_{i_{N-1}}, O^{N-1}, q^{N-1}), \qquad i_{N-1} \in J_{N-1} \tag{9}$$

where N means the Nth order and i_{N-1} means the ith structure of the $N-1$ order. From this, we can introduce the definition of emergent properties as: E is an emergent property of the system S^n if and only if:

$$\mathrm{E} \in O(S^n) \text{ and } \mathrm{E} \notin O^n(s^{n-1}_{i_{n-1}}) \text{ for all } i_{n-1}. \tag{10}$$

That is, the property E is an emergent property if and only if it is observable in phase S^n, not below.

4. Hybrid rough sets systems

This section is an attempt to generalize an approach aimed at connecting rough set theory with hybrid systems. Hybrid rough set methods are inherently distributed in character, often involve unreliable information and untrustworthy participants including those outside the administrative control of the system, may be implemented on platforms that are dynamically changing and not fully known, and may involve components or agents that have been compromised.

To discuss general rough hybrid systems, we will consider that the system is composed of parts or a set of agents $Ag = \{ag_1, \ldots, ag_p\}$, where $p > 0$. Any agent from Ag is equipped with an information system $IS_{ag} = (U_{ag}, C_{ag})$ where U_{ag} is a set of objects and C_{ag} is a set of attributes associated with agent ag. The decision table of the agent ag is a pair $DT_{ag} = (U_{ag}, C_{ag} \cup D_{ag})$ for any agent $ag \in Ag$ where D_{ag} is the local decision attribute for agent ag. The lower and upper approximations of the set of objects X with respect to condition attributes of DT_{ag} describe the vagueness in understanding of X using agents from the set Ag. Each agent is autonomous in the sense that 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 result of agents cooperating to solve a given problem will be a scheme of agents.

The team of agents of the hybrid rough system is involved in the following activities: convert their specifications into specifications for their children using local languages, manage the negotiation processes leading to the emergence of assembling schemes, and assemble a product from parts submitted by children.

Let $L(ag)$ be a function that determines the set of immediate neighbors for each agent ag. The function L can be defined as:

$$L(ag) = \{ag_i : ag_i \in Ag \text{ and } ag_i \text{ is an immediate neighbor of } ag\}. \tag{11}$$

L provides the selection technique used in the hybrid rough system, for example, in the system of combining rough sets and genetic programming [6], each individual (agent) uses the function L to select another

individual to interact with, such as in the crossover operation. Communication between agents can be defined as mapping

$$M(ag, L(ag)) : \text{ for } x \in U_{ag}, \text{ the value } M(ag, L(ag))(x) \in U_{L(ag)}. \quad (12)$$

According to M, each agent from the hybrid system can send or receive information with other agents. The information can only be transmitted through sequences of immediate neighbor communications. The undirected communications and absence of complete information permit these hybrid rough set systems to sometimes, but not always, succeed in satisfying their goals.

One of the proposals from rough hybrid systems is the decomposition scheme of an agent into simpler parts. Further, we will discuss how each agent can be divided into subcomponents and the relation between them. From an abstract point of view, this approach is actually about establishing for any two objects (agent and subagent) a degree in which one of them is a "part" of the other. We will use the notion of a rough inclusion function [16], which gives for any two entities of discourse a degree in which one of them is a part of the other. Rough inclusion is an extension of the rough membership function, which is defined by Pawlak in [10]. Rough inclusion is parameterized by a real parameter r in the interval $[0, 1]$. Therefore, the predicate $X\mu_r Y$ means that X is a part of Y in degree r.

We define rough inclusion μ_r by letting

$$\mu_r(X, Y) = \frac{|X \cap Y|}{|X|} \quad (13)$$

in the case $X \neq \phi$ and $\mu_r(\phi, Y) = 1$.

Rough inclusion satisfies the following conditions; $\mu(X, X) = 1$ for any object X, $\mu(X, Y) = 1$ implies that $\mu(Z, Y) \geq \mu(Z, X)$ for any triple X, Y, Z, and there is an η such that $\mu(\eta, X) = 1$ for any object X.

Let us define the relation part(μ) on the set of agents Ag using rough inclusion μ as follows:

$$X \text{ part}(\mu) Y \Leftrightarrow \mu(X, Y) = 1 \text{ and } \mu(Y, X) < 1. \quad (14)$$

The relation part(μ) is a nonreflexive and transitive relation of the set Ag. The formula x part(μ)y reads: x is a part of y and satisfies the axioms:

- X part(μ)$Y \Rightarrow$ non(Y part(μ)X) for any pair X, Y.
- X part(μ)Y and Y part(μ)$Z \Rightarrow X$ part(μ)Z for any triple X, Y, Z.

5. Hybrid rough sets and neural networks

This section is an attempt to present an approach aimed at connecting rough set theory with artificial neural networks [8]. First, we will use what is called the "rough neuron" [17] to integrate rough sets into the

structure of neural networks. Section 5.1 describes the use of rough sets as a preprocessing step for determining the important features used as input for the network. Following that, a new algorithm for neural network pruning is presented. By setting the neurons in the form of a decision table, rough sets can determine the need for deleting one or more neurons from the network. Finally, we present the guidelines and demonstrations of an emergent model in this hybrid system.

5.1 The reduction of input features

Rough set theory provides tools for expressing inexact dependencies within data. A minimum description length principle (MDL-principle) gives us the reason for using rough sets to reduce input features in a hybrid system. It states, generally speaking, that rules of the most simplified construction which (almost) preserve consistency with data are likely to classify unseen objects with the lowest risk of error [16]. Therefore, to enable classifying more objects with high accuracy, it needs to neglect features that are the source of redundant information, that is, to use what is called a "reduct of attributes." A reduct [18] is a subset of attributes such that it is enough to consider only the features that belong to this subset and still have the same amount of information.

The input data to the model will be quantized first, that is, the features defining the problem should be identified and labeled. If the input data x_i is given as real numbers in the preprocessing stage, one has to divide the data into distinct sets and introduce a new logical input variable s_k such that:

$$\text{IF}(x_i \in X_{i,j}) \text{ THEN } (s_k = \text{label}(x_i) = T). \tag{15}$$

Let us define what is called the "weighted information system" as $WIS = (U, C, W)$, where U is the set of objects, C is the set of attributes, and W is the set of weights related to the set of attributes C connected with each object. The weighted the decision table takes the form $WDT = (U, C \cup D, W)$. The values of the set of weights W are determined according to the interactions between objects, and the interactions between objects and their environment. If object p_1 interacts with object p_2, then the weight values for attribute i related to object p_1 is modified according to the formula:

$$\Delta w_i = \begin{cases} \alpha, & d_{o_1} = d_{o_2} \\ -\alpha, & d_{o_1} \neq d_{o_2}. \end{cases} \tag{16}$$

The value of α is determined according to the similarity of attribute i between these two objects:

$$\alpha = \begin{cases} \tau & i(p_1) = i(p_2) \\ -\tau & i(p_1) \neq i(p_2), \end{cases} \tag{17}$$

where τ is a constant. The weighted information system is a specific

information table and, therefore, all concepts of the rough set analysis can be adapted to it [7].

The algorithm for the reduction of attributes based on the idea of a weighted decision table [7] can take the form:

1. *Input*: Decision table $DT = (U, C \cup D)$
2. *Output*: Reduct of attribute R
3. *Process*:
   (a) Convert the decision table DT into the weighted decision table WDT by adding the distinguished set W, that is, values associated with the attribute values and interactions between objects.
   (b) While (other objects exist to interact with) do
      i. Interact object i with object j, $i = \{1, 2, \ldots, n\}$, $j = \{1, 2, \ldots, n\}$
      ii. For ($a = 1$ to n)
         A. Modify the weight value associated with attribute a and object i
   (c) $R = \phi$
   (d) For ($a = 1$ to n)

$$\text{If} \left(\sum_{j=1}^{n} \omega_{ij} > 0 \right) \text{ then } R = R \cup \{a\}.$$

Finally, after we determine the set of reducts for the input features, we construct the model of rough neural networks by removing from the input vector those attributes not included in any reduct set.

5.2 Rough neuron

Driven by the idea of decomposing the set of all objects into two parts: the lower approximation and the upper approximation with respect to a given set X of objects, Lingras introduced in [17] the idea of a rough neuron to construct what is called the "rough neural network." Each rough neuron r is a pair, one for the upper bound value, called the upper neuron $\overline{r}$; and another one for the lower bound value, called the lower neuron $\underline{r}$. Those two neurons can exchange information between each other and between other rough (conventional) neurons.

Rough neurons will be mainly used in the construction of this hybrid system. They can be used as input neurons when the input value will be a rough value (upper and lower bound). In the same way, a rough neuron can be used in the hidden layer where it has provided better results than conventional neurons [8, 13].

The outputs of a rough neuron r are calculated using the formulas:

$$\begin{aligned} \text{output}_{\overline{r}} &= \max(g(\text{input}_{\overline{r}}, g(\text{input}_{\underline{r}})) \\ \text{output}_{\underline{r}} &= \min(g(\text{input}_{\overline{r}}), g(\text{input}_{\underline{r}})) \end{aligned} \tag{18}$$

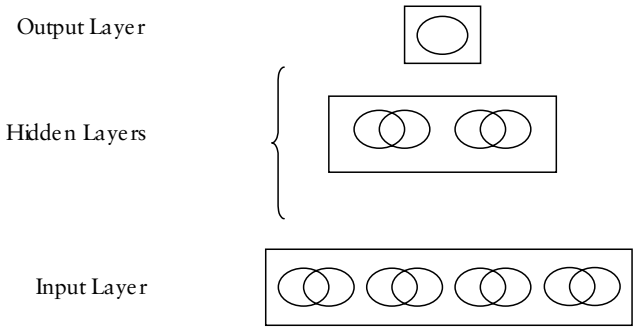

Figure 1. An example of a rough neural network.

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

$$f(x) = \frac{1}{1 + e^{-\beta x}} \tag{19}$$

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

Figure 1 shows an example of a rough neural network that consists of the input layer, which once the pattern is presented, distributes the pattern throughout the net, and propagates the pattern down its connections to the middle layers (one or more hidden layers). The neurons 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.

Connections between conventional neurons in rough neural networks are made as in the usual case, that is, a single connection. Connections between rough neurons and conventional ones are made by connecting lower neuron $\underline{r}$ and upper neuron $\overline{r}$ separately [8]. Two rough neurons in the networks can be connected to each other using either two or four connections. A rough neuron r is said to be *fully connected* to a rough neuron s, if $\underline{r}$ and $\overline{r}$ are connected to both $\underline{s}$ and $\overline{s}$. If only two connections exist from neuron r to neuron s, then the two neurons are called *partially connected*. If a rough neuron r excites the activity of rough neuron s (i.e., increasing the output of r will increase the output of s), then we connect only $\overline{s}$ with $\overline{r}$ and $\underline{s}$ with $\underline{r}$. In the opposite situation, if r inhibits the activity of s (i.e., increasing the output of r decreases the output of s), then we connect only $\overline{s}$ with $\underline{r}$ and $\underline{s}$ with $\overline{r}$.

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

$$\text{input}_i = \sum \omega_{ij} \times \text{output}_j \tag{20}$$

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

The learning process for the network is based on the general learning scheme, so the weights in the network are adjusted according to the general equation:

$$\omega_{ji}^{\text{new}} = \omega_{ji}^{\text{old}} + \alpha(t) \; . \; g(\text{input}_i) \tag{21}$$

where g is any transfer function and $\alpha(t)$ 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. The most popular learning algorithm is the back-propagation algorithm. In this algorithm weights are adjusted according to the formula:

$$\omega_{ji}^{\text{new}} = \omega_{ji}^{\text{old}} + \alpha \, \text{err}_i \; . \; f'(\text{input}_i) \tag{22}$$

where f' is the derivative of the sigmoid function, α is the learning coefficient that remains constant during the learning, and err_i is an error for neuron i.

5.3 The structure adaptation of the network

Removing redundant neurons and their connections from the network can reduce its complexity. Rough sets can be used to determine if the network structure is sufficient for processing its function and whether or not there are some redundant neurons [8]. The adaptation is regarded as successful and will be executed if the new structure (offspring) improves its parent's performance by a certain margin.

Let us define a neural network space N as a collection of formal neural network models [19]. A reduct operator over the neural network space is a mapping that transforms a neural network $P_i \in N$ into a neural network $P_j \in N$.

Since our goal is to find and eliminate as many unnecessary network neurons as possible, it is important that we identify the effect of removing neuron connections on the output of the network. Let us denote the weights of the connections between input and hidden units by w_i^h, $i = 1, 2, \ldots, n$; $h = 1, 2, \ldots, H$ and the weights of the connections between the hidden and output units by w_h^o, $h = 1, 2, \ldots, H$; $o = 1, 2, \ldots, m$ where H is the number of hidden units, n is the dimensionality of the input patterns, and m is the number of output units. The output of the network [8] is:

$$Y_o = f\left(\sum_{h=1}^{H} w_h^o f\left(\sum_{i=1}^{n} x_i w_i^h\right)\right), \tag{23}$$

where f is a sigmoid function.

Let Y_o be considered as a function of a single variable corresponding to the connection between input unit i and hidden unit h. The derivative

of Y_o with respect to the weights of the network is:

$$\frac{\partial Y_o}{\partial w_i^h} = Y_o \times (1 - Y_o) \times w_h^o \times x_i \times \left(1 - f\left(\sum_{i=1}^{n} x_i w_i^h\right)\right) f\left(\sum_{i=1}^{n} x_i w_i^h\right) \tag{24}$$

$$\frac{\partial Y_o}{\partial w_h^o} = Y_o \times (1 - Y_o) \times f\left(\sum_{i=1}^{n} x_i w_i^h\right). \tag{25}$$

By the mean value theorem, and considering Y_o as a function of a single variable corresponding to the connection between input unit i and hidden unit h, we have

$$Y_o(w) = Y_o(w_i^h) + \frac{\partial Y_o(w_i^h + \delta(w - w_i^h))}{\partial w_i^h} \times (w - w_i^h) \tag{26}$$

where $0 < \delta < 1$. When we set w equal to zero, we obtain

$$\left|Y_o(0) - Y_o(w_i^h)\right| \leq \left|w_i^h \times \frac{\partial Y_o(w_i^h)}{\partial w_i^h}\right|. \tag{27}$$

Assume that the sigmoid function belongs to the interval $[-0.5, 0.5]$, it then follows that

$$\left|\frac{\partial Y_i(w)}{\partial w_i^h}\right| \leq \frac{|w_h^o x_i|}{8}. \tag{28}$$

From equations (27) and (28), then

$$|Y_o(0) - Y_o(w_i^h)| \leq \frac{|w_i^h w_h^o x_i|}{8}. \tag{29}$$

This inequality gives us an upper bound of the change in the output of the network when the weight w_i^h is eliminated. If $x_i \in [0, 1]$ then equation (27) takes the form

$$|Y_o(0) - Y_o(w_i^h)| \leq \frac{|w_i^h w_h^o|}{8}. \tag{30}$$

Similarly, by considering Y_o as a function of a single variable v that corresponds to the connection between hidden unit h and output unit o, we have

$$\left|\frac{\partial Y_i(v)}{\partial w_h^o}\right| \leq \frac{1}{8}. \tag{31}$$

Hence, the change in the output of the network after the weight v_h^o has been eliminated is bounded by

$$|Y_i(0) - Y_i(w_h^o)| \leq \frac{|w_h^o|}{8}. \tag{32}$$

Equations (29) and (31) show the maximum error occurring in the network if a connection is removed from any layer in the model.

We now describe the algorithm for reducing the redundant neurons from the network.

1. Classify the output value of each neuron to the one cutoff from the logical value set V as

$$\text{IF } o_i \in [a, b] \text{ THEN } v_k = \text{label}([a, b]) = \text{cutoff}_k.$$

2. Replace the value o_i with the new logical value v_k.
3. Construct a new decision table $MDT = (U, V \cup D)$ where MDT is induced from the regular decision table DT by replacing the set of attributes C with the set of logical values V.
4. Apply the reduct algorithm described previously to the decision table MDT to get a set of reducts.
5. Remove any neuron and its connections that does not exist in any of the reducts.

5.4 Guidelines for emergent phenomena

The previously discussed characteristics of the system are also observed to define the emergent properties. In this system, the interaction of the dynamic representation and nonpositional interpretation provides some innate emergent properties that assist in the acquisition of solutions [15]. A global function emerges from the system unless this function is not explicitly dictated in each of its components. In fact, each component is always looking for maintaining a cooperative situation with other components in this hybrid system. 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 hybrid system represents an emergent model on some levels. First, the determination of the redundant features emerges from the interactions between input features. Second, 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). Third, once the net is trained, the appropriate pattern at the output layer emerges from the interactions between agents (neurons and links) in the static network. In addition, the adaptation of the network structure emerges from the interactions between neurons in the rough set environment.

The underlying strategy of emergent systems is that each neuron should act independently to contribute to global objectives in a way that, if sufficiently many other participating neurons acted with similar objectives, the satisfaction of the global objectives would be assured. Each neuron must not contribute directly to the objectives based on the

resources that it has access to, but must anticipate the needs and potential contributions of its neighbors. Information that must be transferred among neurons to fulfill the mission goals can be communicated whenever it is available, but additional synchronization and communication to support the structure of the algorithm itself is not needed. Thus, emergent models generally have lower communication costs than conventional distributed models.

Two kinds of dynamic evolution could be considered in this hybrid system: modification of the structure by re-organization of the acquaintances network, and modification of behavior. The goal here is to demonstrate the self-organization of the rough neural network. Consequently, rough neural networks need to be used in an emergent way and perform their roles; therefore, a duplication system is required. We will define a duplication operator that an agent can use. In the same manner, a removable operator can be defined. This idea was first discussed in [19] as an adaptation of a neural network structure but in a way different from our approach: this process was under the control of an overall system error. Nevertheless, in our hybrid system, the adaptation process is local for the agent and no global control exists. Therefore, each agent has the ability to produce new agents and it can remove itself from the system only under local control. To define local control for each agent, we assign a fitness value for each agent or neuron. This fitness value not only depends on individual performance, but also on how this agent performs "better" than other agents. Let us define a fitness function in a simple form as the summation of two terms. The first term is the performance of an agent, it is the average between the input and the output values of this agent. The second term of the fitness function is for measuring how this agent is better than other agents in the network. Therefore, the fitness function can take the form:

$$F_{ag} = \alpha_1 \,.\, V + \alpha_2 \,.\, B, \tag{33}$$

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 (neuron) $ag \in Ag$, and B is how the agent ag is better than other agents in Ag. In the first experiment using data from California State University, we set V as the average of input and output values for the neuron. That is, $V_i = (\text{output}_i + \text{input}_i)/2$. But for the medical dataset we change this to be how the weight vector values that connect to this neuron are modified after some number of iterations, and the highest value is the best one (it has the best value of V). That is, $V_i = \overrightarrow{W}_{i\text{new}} - \overrightarrow{W}_{i\text{old}}$, where $\overrightarrow{W}_{i\text{new}}$ is the weight vector that connects the lower layer to neuron i after some number of iterations from $\overrightarrow{W}_{i\text{old}}$.

Depending on the value of the fitness function F_{ag}, the agent or neuron ag can be spilt into two neurons using a duplication operator, that

is, it produces another neuron that interconnects exactly the same as its parent because the attributes are inherited. If a neuron does not form the correct interconnections between other neurons or it is redundant in the network, then it will die. We will limit this property to the hidden layers only; the input and output layers will be fixed. We can say that this hybrid system 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 the processes of agent/environment interaction to take place within each agent's lifetime, we can reliably obtain good performance.

Some additional characteristics of the hybrid system are observed to define the emergent properties in this system such as the following.

- No agent globally controls the dynamics of the entire system. The agents are limited and are unaware of some parts of the global system. Therefore, each agent has a local environment. Each participant can neither read nor write directly to agents, in other words, the system can make use of neither global visibility nor central control.
- The agents act and modify this environment locally. Each agent has only a partial view of other agents and of the environment, in which it is immersed, and in the absence of global control, every agent must be able to communicate with its immediate neighbors without depending on knowledge of the overall system topology.
- 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.

The network here is a group of agents, none of which can deal with a difficulty alone, but only do so when each cooperates. Emergence in our model is not something unpredictable. The carving of a process in a set of neurons gives a basis in terms of expected results and then some events can be predicated, as they become a goal for the process. The resulting design emerges from the various competencies of the members of the team and is something unpredictable as to content, but not as an expected goal and then event.

6. Experiments

If one is modeling phenomena, biological processes, or social systems that involve emergent properties, performance may not be a major concern. 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 networks does provide better results. Instead, this section tries to verify the emergent properties in this hybrid

system. We will show the results for some different applications; namely, a California State Univeristy dataset and two medical datasets. For comparing the results, we construct three models of neural networks. Model 1 is a conventional neural network, Model 2 is a standard rough neural network as defined in [17], and Model 3 is a rough neural network with modifications described in this paper.

6.1 California State Univeristy dataset

The data contains information about the representation of students from California State Univeristy taking the equivalent of a 15-unit course load. The task is to predict the volume of students for year 2000 using data from previous years. The input to the rough neural network model consists of the rough pattern, that is, the upper and lower bounds of annual attendance. The data are divided into three-year sections and the upper and lower bound of values that exist in each section is taken. The data is for years 1991 to 2000. The data sections are: section 1, 1991–1993; section 2, 1994–1996; and section 3, 1997–1999.

For Models 2 and 3 (standard rough neural network and proposed model), there are three rough input neurons, one for each section, and one hidden layer with eight rough neurons. Since the output is a unique value, the output layer uses one conventional neuron. For the conventional neural network of Model 1, the inputs are the average values for each section of data. The model consists of three conventional input neurons, one hidden layer with eight conventional neurons, and one conventional neuron in the output layer.

Now we will discuss the experimental results from these three models. Initially, for each network, the connections are assigned somewhat random weights. The input training set is presented to the network several times. Figures 2, 3, and 4 show the average global output reduction error during the training process, taken every 1000 generations for each neural network model. From the figures, we observe that the average error of the proposed Model 3 appears more natural than the standard rough neural network Model 2 and the conventional neural network Model 1.

Table 2 compares the average errors for each neural network model through all generations. From the table we observe that Model 2 (standard rough neural network) has the best average error and it is very close to the average error of our new model. Results for both Models 2 and 3 are better than the results of the conventional neural network. Table 3 shows the maximum error of the three models through 100,000 generations. Our proposed Model 3 and the standard rough neural network Model 2 have the same maximum error value, which is better than the error value of the conventional neural network. Table 4 shows the minimum error values through 100,000 generations for each

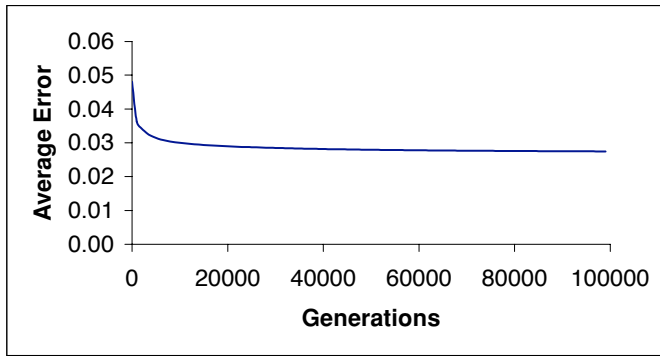

Figure 2. The average error through 100,000 generations produced by Model 3.

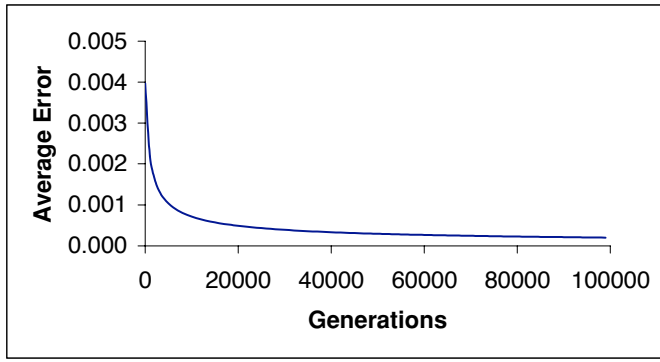

Figure 3. The average error through 100,000 generations produced by the standard rough neural network Model 2.

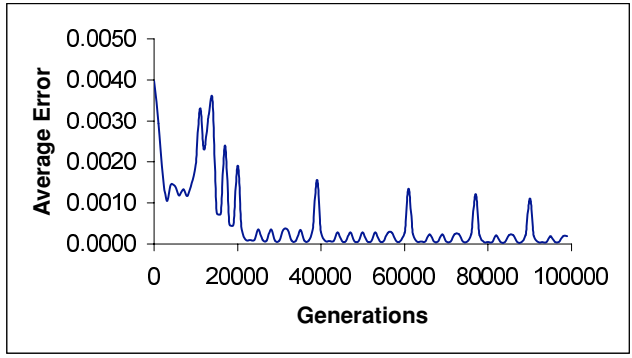

Figure 4. The average error through 100,000 generations produced by the conventional neural network Model 1.

Model 3	Model 2	Model 1
0.00055	0.00044	0.02866

Table 2. Average error through 100,000 generations.

Model 3	Model 2	Model 1
0.00397	0.00397	0.04804

Table 3. Maximum error through 100,000 generations.

Model 3	Model 2	Model 1
0.00005	0.00020	0.02747

Table 4. Minimum error through 100,000 generations.

model. From the table we observe that our proposed Model 3 provides very good results, the error value produced is better than the values of Models 1 and 2.

Some emergence can be observed in our proposed model. From generation to generation, the group of agents or neurons interact with each other. Each agent has the ability to produce itself and the ability to die and delete itself from the network structure. Through the interactions between agents, the weights of connections; that is, the attributes of agents, are modified. In this experiment for the new Model 3, we find through 100,000 generations, that agent number 6 used the duplication operator the most at 31%. For the removable operator, neuron number 5 used it at the same rate of 31% against all agents in the network.

6.2 Meningitis dataset

We now report results of experiments on a meningitis dataset. This data was colleted at the Medical Research Institute, Tokyo Medical and Dental University. We modified this data to be used with our proposed model, so we set up the attribute values as rough patterns (upper and lower bound values).

For Models 2 and 3 (standard rough neural network and proposed model), there are 17 rough input neurons, one for each attribute, and one hidden layer with 51 rough neurons. Since the output is a unique value, the output layer has one conventional neuron. For the conventional neural network, Model 1 consists of 17 conventional input neurons, one hidden layer with 51 conventional neurons, and one conventional neuron in the output layer.

Initially, the connection weights are assigned somewhat random values. The input training set is presented to the network several times. Figure 5 shows the average reduction of global output errors during the training process that are taken every 1000 generations for each model.

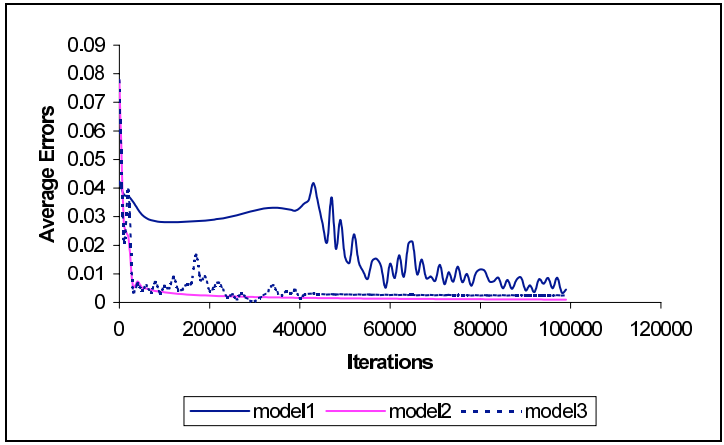

Figure 5. The average error through 100,000 generations for Models 1, 2, and 3.

Model 3	Model 2	Model 1
0.00466	0.00304	0.02050

Table 5. Average error through 100,000 generations.

Model 3	Model 2	Model 1
0.07770	0.07770	0.04234

Table 6. Maximum error through 100,000 generations.

Model 3	Model 2	Model 1
0.000348	0.00099	0.00042

Table 7. Minimum error through 100,000 generations.

From the plots, we observe that the average error of the proposed model is close to zero and better than both the standard rough and conventional neural networks.

Table 5 shows the comparison between average errors of each neural network model through all generations. From the table we observe that Model 2 (standard rough neural network) has the best average error. Table 6 shows the maximum error of the three models through 100,000 generations. The standard rough neural network has the same value as the proposed model, and the conventional neural network Model 1 has the best maximum error value. Table 7 shows the minimum error values through 100,000 generations for each model. From the table we observe that the proposed Model 3 has the best value.

We observe some emergence in this model. Each neuron has the ability to produce itself with the same connection and it has the ability to die and delete itself from the network structure. In this experiment, for Model 3, we find through 100,000 generations that neuron number 7 used the duplication operator the most at 21%. For the removable operator, neuron number 43 used it the most at 22% against all agents in the network.

6.3 Prostate cancer datasets

We report results of experiments on two medical datasets: prostate cancer dataset PSA and biopsy dataset. The datasets were obtained from the Department of Urology and Pathology, Kitasato University School of Medicine, Japan. More details about the data are given in [20, 21].

6.3.1 Prostate cancer dataset PSA

This dataset contains 178 male patients treated with a radical retropubic prostatectomy. This data is used to predict the pathological outcome and early biochemical failure following radical prostatectomy in Japanese males. In this data, pathological results (e.g., maximum cancer length in biopsy specimen and number of biopsy cores with cancer) were added to standard clinical features (e.g., serum PSA, clinical stage, primary and secondary Gleason grade, and Gleason sum in biopsy specimens).

The data consists of eight condition attributes and four decision attributes as in Table 8. In our experiment, the whole dataset was partitioned randomly into a training set ($n = 89$) and a testing set ($n = 89$). The testing set was not seen by the network during the training phase and is only used for testing the generalization of neural networks after they are trained. The problem is nontrivial and difficult to solve, as the dataset is complex and relatively small.

The target of the data is to predict the combination of four decision attributes together by classifying data according to condition attributes. The error is calculated as:

$$E = \sum W_k E_k, \tag{34}$$

where E_k is the number of misclassified patients divided by the total number of patients for decision attribute k. W_k is the decision weight for decision attribute k. The weight values for decision attributes are as follows: $W_{\mathrm{ECE}} = 3/22$, $W_{\mathrm{SV}} = 5/22$, $W_{\mathrm{Node}} = 8/22$, and $W_{\mathrm{Margin}} = 6/22$.

In the three neural network models, the output layer contains four conventional neurons corresponding to the decision attributes of the dataset. The input layer in Model 1 has eight conventional neurons, each of which corresponds to one input attribute. In the rough neural network of Model 2, the input layer contains one conventional neuron,

Attribute name	Description
Condition attributes	
Age	The patient's age.
IMX_PSA	Prostate specific antigen in serum.
Clin_stage	Clinical stage (finding of digital rectal examination).
Pos_core	Positive core (number of biopsy specimens with cancer).
Max_ca_length	Maximum cancer length in biopsy specimens.
P_GI_Core	Primary grade of cancer in biopsy specimens.
S_GI_Core	Secondary grade of cancer in biopsy specimens.
G_S_Core	P. GI core + S. GI core.
Decision attributes	
ECE	Extra capsular extension in surgical specimens.
SV	Seminal vesicle invasion in surgical specimens.
Nodes	Lymph node metastasis in surgical specimens.
Margin	Margin in surgical specimens.

Table 8. The attribute descriptions of the PSA dataset.

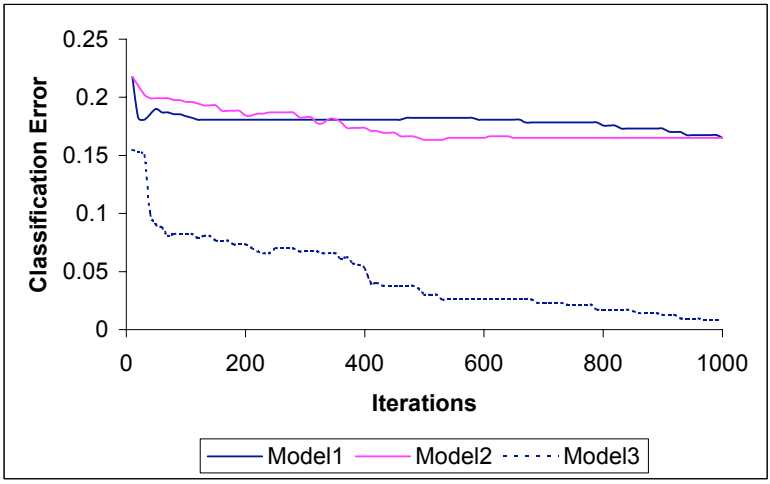

Figure 6. The rate of classification error for three models of neural networks: Model 1: standard neural network; Model 2: rough neural network; and Model 3: proposed model.

which corresponds to input attribute "age" and seven rough neurons. With respect to the new Model 3, the structure of the input layer is determined according to the result of a preprocessing rough set step. Model 3 ignores the values of attribute "age" where it is not included in all reduct sets.

Figure 6 shows the rate of global classification error with the training data for the three neural network models. From the result, we observe

Attribute name	Description
Age	The patient's age.
Total PSA	Prostate specific antigen.
Free PSA	Free prostate specific antigen.
Free total ratio	Free PSA / Total PSA.
PSA density	Total PSA / Gland volume.
PSATZ	Total PSA / Transitional zone volume in prostate.
Gland volume	Prostate size.
TZ volume	Transitional zone in prostate.
DRE	Digital rectal examination.

Table 9. The attribute descriptions of the biopsy dataset.

that the maximum errors are almost the same for all three models. For Model 1 (conventional neural network) and Model 2 (standard rough neural network), the maximum error is 0.2176; and for Model 3 (proposed model) it is 0.1547. The convergence for Model 1 starts after 940 iterations, while the convergence for Model 2 starts at time 540. For our proposed Model 3, the convergence happened after 970 iterations.

6.3.2 Biopsy dataset

The second dataset contains 246 patients; we randomly divide the dataset into a training set ($n = 123$) and a testing set ($n = 123$).

The description of the attributes is shown in Table 9. The decision attribute is the pathological result: 0 means no cancer and 1 means cancer exists.

The output layers of all models contain one conventional neuron. The input layer in the conventional neural network Model 1 has nine conventional neurons, each of which corresponds to one input attribute. In the rough neural network Model 2, the input layer contains one conventional neuron for attribute "age". In Model 3, we find that four attributes will be not included in any of the reduct sets; namely, Total PSA, Free PSA, PSATZ density, and age. Therefore, we will ignore one attribute only because removing more than one attribute from this data is not acceptable from the medical domain. Then, we will omit the attribute age.

Figure 7 shows the rate of global classification error with the training data for the three neural network models. Table 10 has information about the classification error rate for the training and testing data with respect to each model for both datasets. We observe that the best classification error for the training and testing data is obtained from the proposed method. Additionally, the error of the rough neural network Model 2 is better than the conventional neural network Model 1.

Figure 8 illustrates the difference in the classification error arising from the use of a pruning algorithm to reduce the size of the network.

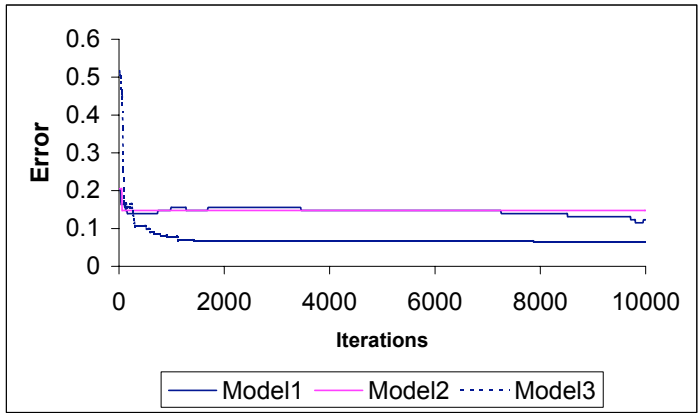

Figure 7. The rate of classification error for the three neural network models.

Model	PSA dataset		Biopsy dataset	
	Training	Testing	Training	Testing
1. Standard neural network	0.1609	0.1899	0.1229	0.1557
2. Conventional rough neural network	0.1649	0.1742	0.1475	0.1475
3. Proposed rough neural network	0.0082	0.0787	0.0656	0.0574

Table 10. The result of the three neural network models with the two medical datasets.

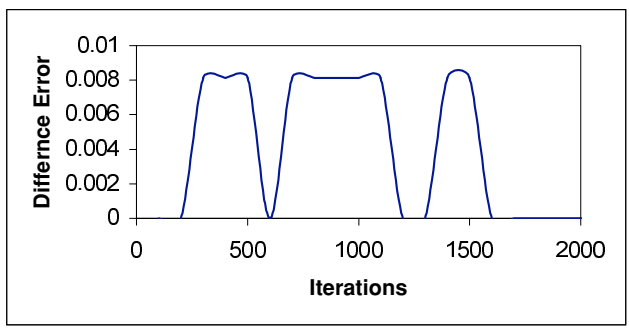

Figure 8. The difference error is caused by pruning the network.

We observe from the figure that the maximum error is 0.008 when one or more connections are removed from the network. This indicates that the reducing algorithm did not affect the global error of the network very much.

The results showed high sensitivity and specificity rates in predicting biopsy result and recurrence after radical prostatectomy. The proposed Model 3 showed the highest accuracy for predicting the pathological

stage after radical surgery and biopsy results. As a new strategy, this model is essential for diagnosis and treatment selection of prostate cancer. Therefore, the proposed model represents a new, good method for classification and decision making algorithms. These results suggest that our method can be considered as a promising tool for extracting laws from experimental datasets and its performance is fully comparable with the performance of other systems. The application we introduced; that is, the combination of rough set theory and artificial neural networks, represents emergent computation in the strict sense.

7. Conclusions

In recent years, an approach called "emergent computation" has gained popularity in a variety of fields. The concept of emergence has been focused as a result of research in nonlinear dynamics, artificial life, complex systems, and behavior-based systems. This paper thus provides a 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 the rough set forms a bridge between neural network paradigms on the one hand, and the representation list paradigm on the other.

We begin this paper with an introduction to emergence and illustrate properties of an emergent model with some examples of existing emergent systems. Following that, we summarized the approach of combining rough set theory and artificial neural networks and described this hybrid system. In the next part of this paper, we illustrated the emergent properties that exist in rough neural networks, and showed how to exploit emergence to extend problem-solving capabilities by combining rough set theory and artificial neural networks. Two new operators were added: duplicate and remove, and new functions were defined to assign fitness values for each neuron in the network. By using these new modifications, rough neural networks can perform their roles and be used in an emergent way. In the last part of the paper, we described experiments with real-life data from California State University and some medical datasets. We compared the results of conventional neural networks, rough neural networks, and our new model of rough neural networks.

References

[1] J. Z. Dong, N. Zhong, and S. Ohsuga, "Rule Discovery by Probabilistic Rough Induction," *Japanese Society of Artificial Intelligence*, (2000) 274–286.

[2] A. Garcia and D. Shasha, "Using Rough Sets to Order Questions Leading to Database Queries," in *Proceedings of the International Conference on*

Information Systems Analysis and Synthesis (ISAS'96), edited by Nagib C. Callaos, July 22–26, Orlando, USA.

[3] S. Tsumoto and H. Tanaka, "Discovery of Approximate Medical Knowledge Based on Rough Set Model, Principles of Data Mining and Knowledge Discovery," in *Second European Symposium*, PKDD'98, 1998.

[4] N. Gotts, "Emergent Phenomena in Large Sparse Random Arrays of Conway's Game of Life," *International Journal of System Science*, **31**(7) (2000) 873–894.

[5] R. Olson and R. Sequeira, "Review: Emergent Computation and the Modeling and Management of Ecological Systems," *Computers and Electronics in Agriculture*, **12** (1995) 183–209.

[6] Y. Hassan and E. Tazaki, "Combination Method of Rough Set and Genetic Programming," *The International Journal of System and Cybernetics "Kybernetes,"* **32**(9/10) 2003.

[7] Y. Hassan and E. Tazaki, "Emergence Computation using New Model of Cellular Automata," to appear in *Applied Artificial Intelligence Journal*, 2002.

[8] Y. Hassan, E. Tazaki, S. Egawa, and K. Suyama, "Decision Making using Hybrid Rough Sets and Neural Networks," to appear in *International Neural System Journal*, 2002.

[9] L. Rocha and C. Joslyn, "Simulations of Embodied Evolving Semiosis: Emergent Semantics in Artificial Environments," *Proceedings of the Conference on Virtual Worlds and Simulation 1998*, pp. 233–238.

[10] Z. Pawlak, "Rough Sets," *International Journal of Computer and Information Science*, **11**(5) (1982) 341–356.

[11] Z. Pawlak, "Rough Classification," *International Journal of Man-Machine Studies*, **20** (1984) 469–483.

[12] H. Nakayama, Y. Hattori, and R. Ishii, "Rule Extraction Based on Rough Set Theory and its Application to Medical Data Analysis," *IEEE SMC'99 Conference Proceedings*, 1999.

[13] L. Polkowski and A. Skowron, *Rough Sets in Knowledge Discovery* (Physica Verlag, 1998).

[14] Y. Hassan and E. Tazaki, "Emergence in Combined System Structure of Rough Set Theory and Genetic Programming," *Proceedings of KES'2001 International Conference*, 2001.

[15] Y. Hassan and E. Tazaki, "Emergent Phenomena in Cellular Automata Modeling," *The International Journal of System and Cybernetics "Kybernetes,"* **31**(9/10) 2002.

[16] L. Polkowski, S. Tsumoto, and Y. Lin, *Rough Set Methods and Applications* (Physica Verlag, 2000).

[17] P. Lingras, "Rough Neural Networks," in *Proceedings of the Sixth International Conference of Information Processing and Management of Uncertainty in Knowledge-based Systems* (IPMU'96), 1996.

[18] W. Ziarko, (editor) *Rough Sets, Fuzzy Sets, and Knowledge Discovery* (Springer-Verlag, Berlin, 1994).

[19] T. Lee, *Structure Level Adaptation for Artificial Neural Network* (Kluwer Academic Publishers, 1991).

[20] S. Egawa, K. Suyama, A. Yoichi, K. Matsumoto, C. Tsukayama, S. Kuwao, and S. Baba, "A Study of Pretreatment Nomograms to Predict Pathological Stage and Biochemical Recurrence After Radical Prostatectomy for Clinically Resectable Prostate Cancer in Japanese Men," *Japanese Journal of Clinical Oncology*, **31**(2) (2001) 74–81.

[21] S. Egawa, K. Suyama, K. Matsumoto, *et al.*, "Improved Predictability of Extracapsular Extension and Seminal Vesicle Involvement Based on Clinical and Biopsy Findings in Prostate Cancer in Japanese Men," *Urology*, **52**(3) (1998) 433–440.