



PERGAMON

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Engineering Applications of Artificial Intelligence 15 (2002) 587–600

Engineering Applications of

ARTIFICIAL
INTELLIGENCE

www.elsevier.com/locate/engappai

Immunity-based hybrid learning methods for approximator structure and parameter adjustment[☆]

Yixin Diao^{a,*}, Kevin M. Passino^b

^a IBM Research, 19 Skyline Drive, IBM T. J. Watson Research Center, Hawthorne, NY 10532, USA

^b Department of Electrical Engineering, Ohio State University, Columbus, OH 43210, USA

Received 18 November 2002; received in revised form 27 January 2003; accepted 28 January 2003

Abstract

From the point of view of information processing the immune system is a highly parallel and distributed intelligent system which has learning, memory, and associative retrieval capabilities. In this paper we present two immunity-based hybrid learning approaches for function approximation (or regression) problems that involve adjusting the structure and parameters of spatially localized models (e.g., radial basis function networks). The number and centers of the receptive fields for local models are specified by immunity-based structure adaptation algorithms, while the parameters of the local models, which enter in a linear fashion, are tuned separately using a least-squares method. The effectiveness of the procedure is demonstrated through a nonlinear function approximation problem and a nonlinear dynamical system modeling problem.

© 2003 Published by Elsevier Science Ltd.

Keywords: Hybrid learning; Radial basis function neural networks; Artificial immune systems

1. Introduction

When a neural network is used to approximate a function by interpolating data from that function, the tuning flexibility of the approximator mapping is heavily dependent on the structure of the neural network (e.g., the number of nodes or layers). Generally, a larger approximator structure provides more tuning flexibility with the potential to achieve higher approximation accuracy (Barron, 1993). However, it may require a larger training data set, and may cause convergence, complexity, overfitting, and generalization problems. Due to these trade-offs, we generally want a simplest approximator structure that can achieve our desired level of approximation accuracy. Approximator structure tuning algorithms can be classified into two broad categories: incremental (a.k.a., growing or constructive) algorithms and decremental (a.k.a., pruning or destructive) algorithms. Incremental algorithms start with a

network that is too small to solve a problem, and add new nodes to the network until certain terminating performance criteria is met (Kwok and Yeung, 1997). A variety of incremental algorithms have been developed such as cascade correlation (Fahlman and Lebiere, 1990), projection pursuit regression (Friedman and Stutzle, 1981), dynamic node creation (Ash, 1989), Meiosis networks (Hanson, 1990), resource-allocating network (Platt, 1991), and selective node activation (Fabri and Kadiramanathan, 1996). They are different to each other in the structures of the growing networks, the methods to assign the weights of the newly added nodes, the strategies for freezing or retraining the existing network weights, and the algorithms for parameter training. In contrast, decremental algorithms start with a large network capable of good estimation, and trim it until the error becomes unacceptable. This can be achieved either by estimating the sensitivity of the error function to the removal of an element and then removing the elements with the least effect, or by adding terms to the objective function that reward choosing a small structure (Reed, 1993; Hassibi and Stork, 1993; Weigend et al., 1991; Sietsma and Dow, 1991; Kruschke, 1988).

[☆]This work was supported by the NASA Glenn Research Center under grant NAG3-2084.

*Corresponding author. Tel.: +1-9147847226; fax: +1-9147846040.

E-mail addresses: diao@us.ibm.com (Y. Diao), passino@ee.eng.ohio-state.edu (K.M. Passino).

In this paper we develop two approaches for approximator structure and parameter tuning using a hybrid learning approach inspired from the natural immune system. Here, “hybrid” refers to two essential characteristics of the learning procedure: (i) we use immunity-based concepts to specify the structure and some approximator parameters and a conventional least squares method to specify the remaining parameters, and (ii) we tune both parameters and structure of the approximators. To the best of our knowledge, this is the first time that the metaphor from the immune system is incorporated with radial basis function neural networks for structure and parameter adjustment.

The paper is organized as follows. In Section 2 the biological model of the immune system is described (from which the motivation for our approaches are derived). In Section 3 the spatially localized model in the form of radial basis function neural networks is discussed, which serves as the approximator structure for hybrid learning. In Section 4 we propose two immunity-based hybrid learning approaches capable of allocating appropriate receptive fields and pruning the redundant units for the spatially localized models. A nonlinear function approximation problem is presented to demonstrate the potential of the proposed methods. At the end of this section we will also comment on the relationships between the work here and the most relevant work in the applications of artificial immune systems. In this way we will further clarify the contributions of this paper. In Section 5 the immunity-based hybrid learning method is applied to model an aircraft jet engine (General Electric XTE46).

2. Inspiration from the immune systems

The natural immune system is a complex system which discriminates between own body cells (self) and foreign pathogens (non-self), and defenses against pathogens. From the point of view of information processing, the immune system is a highly parallel and distributed intelligent system with the capabilities of learning (to recognize relevant patterns, e.g., antigenic peptides), memory (to memorize patterns encountered previously), and associative retrieval (to construct pattern detectors to distinguish between self and non-self). Moreover, the problems of pattern recognition and classification are not solved by a single recognizing unit but rather by a network of local antigen–antibody interactions (*mutual recognitions*) at a system level. According to its importance and complexity, the immune system has absorbed many research efforts resulting in several theories to explain various aspects of immunological phenomena. Among them, the immune network model has gained more

attention which aims to explain the behavior of B lymphocytes (B cells).

As illustrated in Fig. 1, the B lymphocytes interact with the antigens to stimulate the immune response. There are many antibodies on the surface of the B-lymphocytes that act as antigen detectors. The specialized portion of the antibody used for identifying other molecules (of antigens or antibodies) is called *paratope*. The region on any molecule that can be recognized by the paratope is called *epitope*. If the epitope of a foreign antigen molecule matches the paratope of an antibody, the antibody can attach itself to the antigen so that the antigen can be neutralized (leading to its eventual demise). Since antibodies, like other molecules, have epitopes as well, they may also be recognized by other antibodies. Therefore, different antibody types together with antigens compose a complex reaction network, which is called the immune network by Jerne (1974). The immune network can be used to represent the relationship between antigens (being neutralized) and antibodies (being stimulated), and the relationship among antibodies (either being stimulated or being suppressed as they have both epitopes and paratopes). In the immune network, a node represents a type of antigens or antibodies, and the links between nodes represent the affinity between them. The affinity is determined by the matching degree between the paratope and the epitope, and the population of each type of antibodies (or antigens). An antibody type is thought to be stimulated when its paratope recognizes the epitope of antigens or other types of antibodies, so that the corresponding lymphocyte is stimulated to reproduce more lymphocytes, and the lymphocytes secrete more antibodies (with certain mutation rate). This process is called *clonal selection*. On the contrary, an antibody or antigen may be suppressed if its epitopes are recognized by others. Furthermore, the diversity of the immune system is maintained through the replacement of the dead lymphocytes (caused by aging or being suppressed by others) by new lymphocytes generated in the bone marrow (through the reshuffling of the host DNA that codes for the antibody genes).

Based on Jerne’s hypothesis that the immune system is a regulated network of molecules and cells (i.e., antibodies and antigens) that recognize one another, Perelson presented an immune network model (Farmer et al., 1986; Perelson, 1989) which has received a lot of attention among researchers (and we would also like to apply it for the function approximation problem). The affinity between the network nodes are defined in the form of logic operation of binary strings. Let i and j denote two nodes (two types of antibodies) in the immune network, which are represented by binary strings of length l . The matching index between them

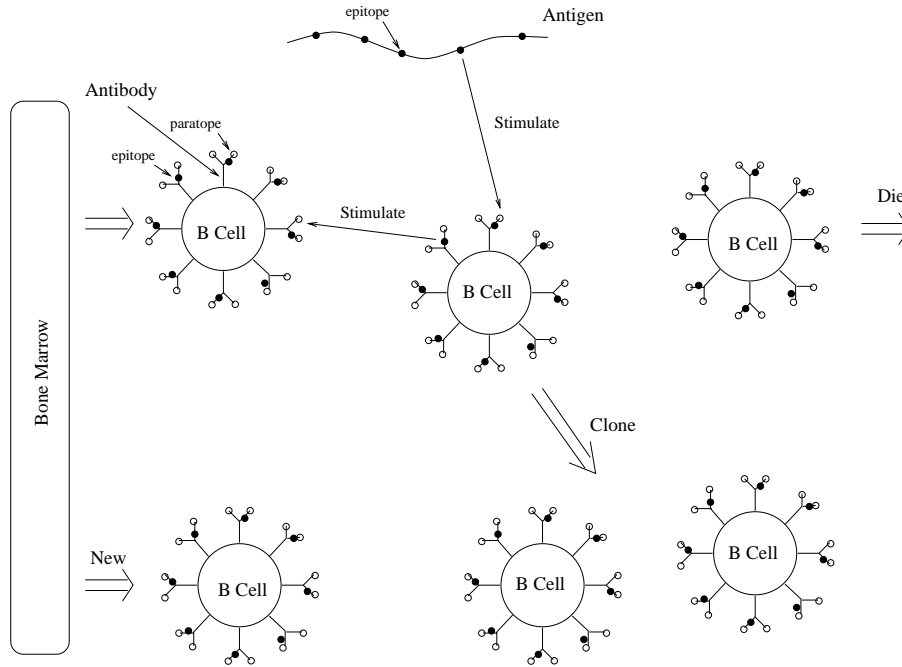


Fig. 1. A schematic illustration of the immune system.

is given by

$$m_{ij} = \sum_k G \left(\sum_n e_i(n+k) \wedge p_j(n) - s + 1 \right), \quad (1)$$

where n denotes the bit of a string, k denotes the shift alignment, $e_i(n+k)$ denotes the value of the $(n+k)$ th bit of the epitope string on the i th antibody type, and $p_j(n)$ denotes the value of the n th bit of the paratope string on the j th antibody type. The symbol \wedge denotes the *exclusive or* operation, s is a matching threshold below which two antibodies will not react at all, and $G(x) = x$ for $x > 0$ and $G(x) = 0$ otherwise.

In the immune network model, the connection strength between different types of antibodies is determined not only by the matching index, which is fixed given the types of antibodies, but also by the concentrations of the antibodies, whose dynamics are given by the following differential equations:

$$\dot{x}_i = c \left[\sum_{j=1}^N m_{ji} x_i x_j - k_1 \sum_{j=1}^N m_{ij} x_i x_j + \sum_{j=1}^n m_{ji} x_i y_j \right] - k_2 x_i, \quad (2)$$

where x_i , $i = 1, 2, \dots, N$, denote the concentrations of different antibody types and y_j , $j = 1, 2, \dots, n$, denote the concentrations of antigen types. The first term represents the stimulation from other antibody types, the second term represents the suppression from other antibody types, and the third term represents the stimulation from other antigen types. The form of these term is dictated by the fact that the probability of a collision between an antibody type i and an antibody type j (or an antigen type j) is proportional to $x_i x_j$

(or $x_i y_j$). The parameter c is a rate constant that depends on the number of collisions per unit time and the rate of antibody production stimulated by a collision. The constant k_1 represents a possible inequality between stimulation and suppression. The last term ($-k_2 x_i$) models the tendency of cells to die in the absence of any stimulation at a rate determined by k_2 .

Besides the above differential equations to describe the dynamics of the antibody concentrations, an essential aspect of the immune network model is that the list of antigen and antibody types is dynamic, changing as some types are added or removed, so that N and n change with time (but on a time scale slower than that occurs in x_i). In particular, a minimum threshold on concentrations is placed so that a variable (a given antibody type) and all of its reactions are eliminated when its concentration drops below the threshold, which simulates the death of unstimulated (useless) antibodies. Moreover, new types of variables may be added according to the reproduction of new antibodies.

Inspired by the immune system, many intelligent computational methods, known as artificial immune systems or immunity-based systems (Dasgupta and Forrest, 1999; Dasgupta, 1999), have been studied and successfully used in a variety of applications such as pattern recognition (Hunt et al., 1995, 1999; Gilbert and Routen, 1994), pattern classification (Bersini, 1999; Seront and Bersini, 1994; Bersini, 1992; Bersini and Varela, 1994; Pramanik et al., 2002; Wang and Jiao, 2001), function optimization (Bersini and Seront, 1992; Fukuda et al., 1999; de Castro et al., 2002; de Castro and Timmis, 2002), anomaly detection (Dasgupta, 1996;

Huang, 2002), decision making (Dasgupta, 1998), and computer security (Kephart, 1994; Kephart et al., 1995, 1999). However, note that with the introduction of the dynamics of antibody concentrations, the dynamical behavior of the immune network becomes very complicated: the connection parameters between the nodes are not independent but are determined by the dynamics of the antibody concentrations (and the concentrations of lymphocytes). This is different to the neural networks where the weights can be adjusted independently, and makes the immune system, in principle, more complex than the brain-nervous system. Thus, compared to other biological inspired techniques (such as neural networks), the application of artificial immune systems is still an emerging field and has received relatively little attention so far.

3. Spatially localized model architecture

Biological studies have undoubtedly provided a large amount of inspirations for the design of engineering systems. In particular, the applications such as nonlinear modeling have been strongly benefited from the analogy to the neural networks. Although immune networks and neural networks share many similarities at the level of system behavior, they are different at the respective building block level. Each node of the immune network represents a pattern, and the links between them represent the similarity (between the patterns) and the concentration (of each pattern). Some studies have been given to compare the artificial immune systems with neural networks (Hoffmann, 1986; Dasgupta, 1997) and other learning systems such as learning classifier systems (Farmer et al., 1986; Bersini and Varela, 1990) and autocatalytic chemical reaction networks (Farmer, 1990). In this paper we would like to incorporate the metaphor from the immune system with the spatially localized model architecture of the radial basis function neural networks.

In neurobiological studies, the concept of localized information processing in the form of receptive fields has been known and demonstrated by experimental evidence (e.g., locally tuned and overlapping receptive fields have been found in parts of the cerebral cortex, in the visual cortex, and in other parts of the brain). This suggests that such local learning offers alternative computational opportunities to learning with “global basis functions” such as the multilayer perceptron neural network with sigmoidal activation functions (Schaal and Atkeson, 1998). Inspired by these biological counterparts, the radial basis function neural network model has been presented, which can be defined by

$$y = \frac{\sum_{i=1}^M b_i R_i(x)}{\sum_{i=1}^M R_i(x)}, \quad (3)$$

where y is the output of the radial basis function network, $x = [x_1, x_2, \dots, x_n]^T$ holds the n inputs, and $i = 1, 2, \dots, M$ represent M receptive field units. The strength parameters b_i represent the “strengths” of the receptive field units, which are constants. The “radial basis functions” $R_i(x)$ (a.k.a., radial response functions or kernel functions) define the activation extents of the corresponding receptive fields with the characteristics that their responses decrease monotonically with distance from a central point. There are several possible choices for the receptive field functions $R_i(x)$. Typically, Gaussian-shaped functions are used for analytical convenience, that is

$$R_i(x) = \exp(-\frac{1}{2}(x - c_i)^T D_i(x - c_i)), \quad (4)$$

where $c_i = [c_1^i, c_2^i, \dots, c_n^i]^T$ parameterize the locations (centers) of the receptive fields in the input space, and $D_i = \text{diag}((1/\sigma_1^i)^2, (1/\sigma_2^i)^2, \dots, (1/\sigma_n^i)^2)$ determine the shapes (relative widths) of the receptive fields.

Actually, to improve modeling flexibility of the radial basis function networks, it is also possible to replace the strength parameters b_i with the strength parametric functions

$$b_i(x) = a_{i,0} + a_{i,1}x_1 + \dots + a_{i,n}x_n, \quad (5)$$

where $a_{i,j}$, $i = 1, 2, \dots, M$ and $j = 0, 1, 2, \dots, n$, are constant strength function parameters. Moreover, by using this new definition (5) of the “strength,” we can apply the inspiration from the immune system to structure and parameter adjustment of the radial basis function networks.

As illustrated in Fig. 2, we view the radial basis function neural network as an artificial immune system. The function to be approximated is defined (represented) by the training data set $G = \{(x(l), y(l)) : l = 1, 2, \dots, P\}$ including P input–output data pairs. These data pairs are thought to be the “antigens” that have been encountered by the artificial immune system. By learning the characteristics of these antigens, the artificial immune system is expected to approximate the function well for new antigens (testing data). The receptive field units in a radial basis function neural network are analogous to the antibodies in the immune system. Each strength parametric function $b_i(x)$ represents a “pattern,” which describes a linear mapping valid in a local region. The valid regions of the patterns are defined by the receptive field functions $R_i(x)$, where the parameters c_i specify the centers of the regions and D_i specify the sizes of the regions. Note that there could exist some overlapping between different patterns since $R_i(x)$ are defined as Gaussian functions. This implies that for a data point in the input space, it could be within more than one valid regions of the receptive field units and thus excite more than one strength functions. Therefore, the output of the radial basis function is a weighted average as defined in (3). This is similar to the

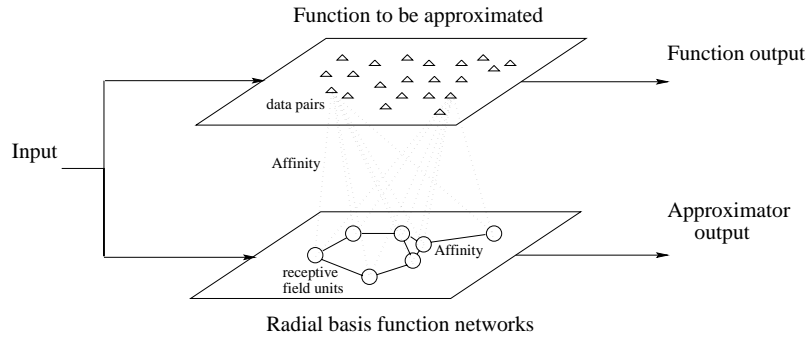


Fig. 2. Radial basis function neural network: an artificial immune systems.

immune system where an antigen could match more than one antibodies since the match, as defined in (1), is a partial match, and the overall results of these matches are to defend against pathogens.

It is also worthy to note that as we are developing an artificial immune system for the function approximation problem, the matching degree defined here is different to the logic operation of binary strings, which is generally used in the immune network model to calculate the matching index (1) (and may be applied to some other pattern recognition problems). There are two matching indices used in this function approximation artificial immune system: the *unit matching index* and the *function matching index*. The value of the radial basis function $R_i(x(l))$ is used as the unit matching index, which measures the closeness of a data pair $(x(l), y(l))$ to the pattern (stored in the i th receptive field unit) in the *input space*. The function matching index is a measure used in the *function space* to describe how well this data pair (antigen) can be approximated by the radial basis function network (the artificial immune system). This index is defined as

$$A_l = \left| y(l) - \frac{\sum_{i=1}^M b_i(x(l))R_i(x(l))}{\sum_{i=1}^M R_i(x(l))} \right|, \quad (6)$$

where $x(l)$ and $y(l)$ are the l th input–output data pair. The matching index between the receptive field units (antibodies) are naturally different than that between the radial basis function network and data pairs. It is intended to be used as a heuristic measure of redundancy of the receptive field units so that we define it as

$$f_{i,k} = m_{i,k} \delta_{i,k}, \quad (7)$$

$$m_{i,k} = \sqrt{\sum_{j=1}^n (c_j^i - c_j^k)^2}, \quad (8)$$

$$\delta_{i,k} = \sqrt{\sum_{j=1}^n (a_{i,j} - a_{k,j})^2}, \quad (9)$$

where i and k represent two different receptive field units, $m_{i,k}$ is the Euclidean distance between the centers of the i th unit and the k th unit, and $\delta_{i,k}$ is the Euclidean

distance between the strength function parameters of the i th unit and the k th unit.

Note that in the immune system the affinity between the antigens and antibodies is determined not only by the matching index, but also by the concentrations of the antibodies (whose dynamics are given by (2)). If a type of antibodies is stimulated by the antigens (or other antibodies), more antibodies of the same type will be generated. On the contrary, the antibodies may be suppressed if their epitopes are recognized by others. This phenomenon is resembled in the artificial immune system to the function of the relative width D_i but in a different way (and only the interconnection between the antibodies, the receptive field units, is considered). The basic idea is that if a part of the function mapping is complicated, then more receptive field units are required in this region (so that they are distributed closely) in order to have an accurate mapping (since each receptive field unit only stores a linear mapping (5)). Thus, their corresponding relative width should be small, which implies that the linear mapping is only valid in a small region. On the contrary, if the receptive field units are located sparsely, the relative width can be large. This kind of the “concentration” in the artificial immune system is described by the following equation:

$$D_i = D + E \tanh\left(\frac{h}{\sum_{k=1, k \neq i}^N 1/m_{i,k}}\right), \quad (10)$$

where i and k represent two different receptive field units, $m_{i,k}$ represent the Euclidean distance between the centers of these two units, h is a constant to determine the rate of changing the relative width according to the distance between the centers, and D and E are constant diagonal matrices to specify the possible adjustment range of the relative width.

The learning process of the artificial immune system to approximate the functions is the process to tune the parameters and the structure of the radial basis function neural network. Note that the tunable parameter vector can be composed of both the radial basis function parameters c_j^i and σ_j^i and the strength function parameters $a_{i,j}$. This is referred to as the *nonlinear in*

the parameter case because the parameters enter the function approximator in a nonlinear way. A nonlinear in the parameter spatially localized model can be tuned by a variety of gradient methods such as the steepest descent method and Levenberg–Marquardt method. Alternatively, we may decompose the parameter vector into a linear part (consisting of the strength function parameters) and a nonlinear part (composed of the radial basis function parameters). By having the tunable parameter vector θ be composed of strength function parameters $a_{i,j}$ only (and specifying the radial basis function parameters c_i and D_i in advance), we will have a linear in the parameter radial basis function network. Note that the linear in the parameter radial basis function networks are also universal approximators and have the capabilities of forming an arbitrarily accurate approximation to any continuous nonlinear function. Results from approximation theory indicate that if tuned properly the nonlinear in the parameter approximator can achieve good approximation accuracy with less parameters than does the linear in the parameter approximator. However, the problem is that we know how to tune the linear in the parameter approximators (e.g., using the least-squares method) but in certain cases they may not be able to reduce approximation error very well and we may need much more parameters (receptive fields). As for nonlinear in the parameter approximators, although they are desirable in theory, in practice we do not know as much about how to tune them properly and the gradient methods may often result in local minima. This motivates our work on understanding the radial basis function neural networks from the point of view of artificial immune systems, and facilitates structure and parameter adjustment of the spatially localized model by way of the hybrid learning approach. By this we mean that we can apply the immunity-based structure adaptation approach to specify the number and centers of the receptive field units, and use the conventional least-squares method to train the strength function parameters since they enter linearly.

4. Immunity-based hybrid learning algorithms

From the point of view of the biology, the function approximation problem may be thought of as a kind of artificial immune systems. The “antigens” represent the input–output data pairs and a collection of antigens defines the problem of function approximation. The “antibodies” represent the units of the approximator (e.g., the receptive field units of the radial basis function network), which are characterized by the tunable parameters and the number of units, and a collection of antibodies defines the function approximator. A good immune system usually consists of a compact collection

of antibodies capable of recognizing all antigens, which is analogous to finding the simplest approximator structure that can achieve the desired level of approximation accuracy. In the following, we present two immunity-based hybrid learning algorithms, namely, *learning from clonal selection* and *learning from internal affinity*. Actually, these two learning algorithms can be incorporated together for structure and parameter adjustment of radial basis function networks. We separate them to be two methods to facilitate a clear explanation.

4.1. Learning from clonal selection

This learning method is capable of allocating appropriate receptive field units according to the approximation errors (external criteria). The basic steps of this approach can be summarized as follows:

1. *Initialization*: The initial approximator may be composed of one (or perhaps a few) receptive field unit(s) using *a priori* knowledge of the function to be approximated. Usually, the centers of the receptive field units c_i are uniformly distributed on the input space of the problem, and the relative widths D_i are chosen according to Eq. (10).
2. *Recognition*: The function matching index between the data pairs $(x(l), y(l))$ and the radial basis function network is calculated according to Eq. (6).
3. *Reinforcement*: The “weakest” data pair $(x(w), y(w))$, i.e., the one whose value of the function matching index is the largest, is cloned. By “clone” we mean that we recruit a new receptive field unit, whose center is located at $x(w)$, into the existing radial basis function network. Its relative width is chosen according to Eq. (10). The step of reinforcement serves to compensate the poor approximation region by recruiting more receptive field units (which represent a local linear function) so that we could have a finer mapping in this area. This is analogous to the immune system where new antibodies are produced through clonal selection (with a certain mutation rate) or reshuffling of the host DNA. The difference is that in the artificial immune system we try to generate a clone based on the information from the antigen (we know the worst training data pair and we add a new antibody at that location) rather than from other antibodies (as in the natural immune system), which, we think, is more effective. As a result, the parameter vector of the radial basis function is enlarged and the tuning ability is improved.
4. *Parameter adaptation*: We use the training data set $G = \{(x(l), y(l)) : l = 1, 2, \dots, P\}$ to tune the “linear” parameters θ_1 consisting of the strength function parameters $a_{i,j}$. This may be implemented by least

squares method, that is

$$\theta_1 = (\Phi^T \Phi)^{-1} \Phi^T Y, \tag{11}$$

where $Y = [y(1), y(2), \dots, y(P)]^T$ and $\Phi = [\phi(x(1), \theta_2), \phi(x(2), \theta_2), \dots, \phi(x(P), \theta_2)]^T$. Note that here all the linear parameters are retrained in order to optimize the overall performance of the enlarged radial basis function network.

5. *Generalization:* The performance of the approximator is tested by evaluating approximation error defined as the maximum absolute error (MAE) between the output of the actual function and the output of the approximator. If the value of the fitness function is below a “satisfactory threshold,” we claim that a satisfactory approximator has been generated. Otherwise, go to Step 2 for structure and parameter adaptation again.

To study the performance of the proposed immunity-based hybrid learning approach, we seek to approximate a complicated nonlinear function

$$z = 0.1x + 0.05(x + y)^2 + 3(1 - x)^2 e^{-x^2 - (y+1)^2} - 10(\frac{x}{5} - x^3 - y^5) e^{-x^2 - y^2} - \frac{1}{3} e^{-(x+1)^2 - y^2}, \tag{12}$$

where $x \in [-4, 6]$ and $y \in [-4, 6]$, as shown in Fig. 3. Note that the nonlinearity represented in this function is not uniform. The “frequency” of the nonlinearity is high in some regions, while the function is relatively smooth in other areas. It is generally not easy to approximate such functions. (as we will see later by approximating it with neural networks) in that we try to approximate both

high frequency and low frequency components of the function.

We generate a training data set including 1000 data pairs. The “satisfactory threshold” is chosen to be 0.35. We start the approximation process with four receptive field units, whose centers, located at $(-2, -2)$, $(-2, 4)$, $(4, -2)$, and $(4, 4)$, are uniformly distributed in the problem space. The performance of the approximator becomes better along with the structure adaptation when more receptive field units are recruited into the radial basis function network, as shown in Fig. 4 (where the solid line represents the approximation errors with respect to growing numbers of receptive field units and

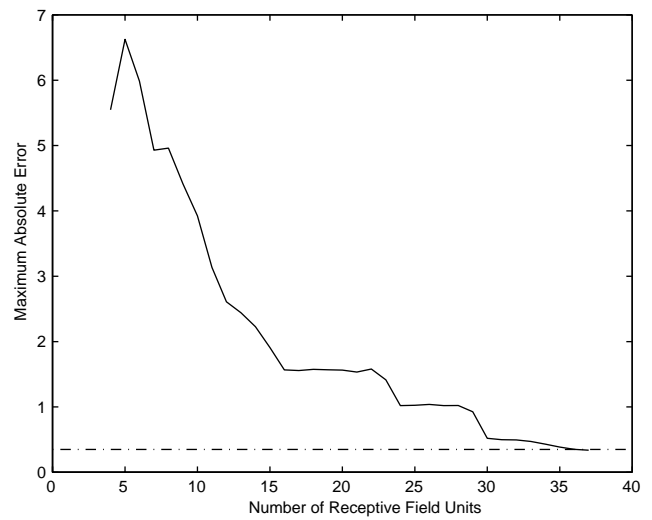


Fig. 4. Structure adaptation via recruitment of new receptive field units.

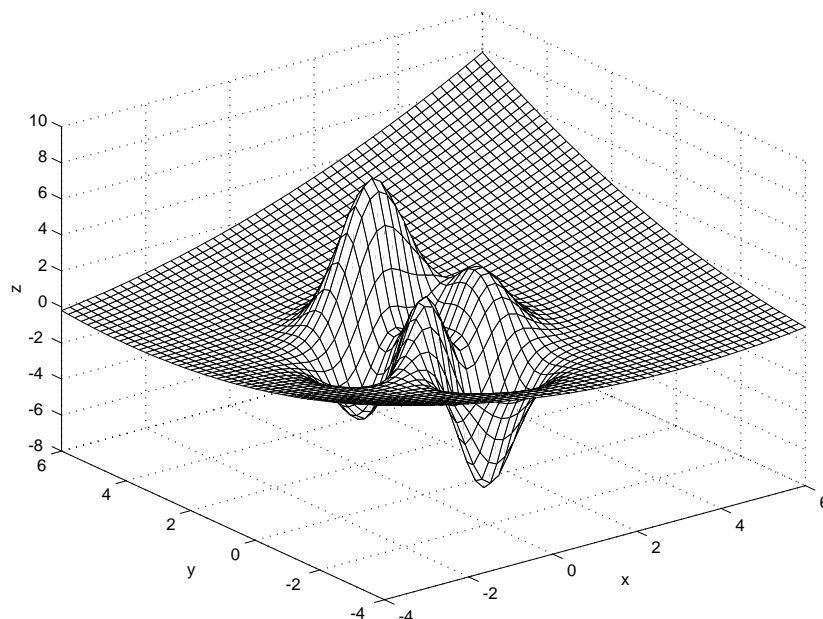


Fig. 3. Nonlinear function we seek to approximate.

the dash-dotted line represents the desired satisfactory threshold).

The immunity-based hybrid learning approach results in a radial basis function neural network with 37 receptive field units to satisfy the requirement given by the satisfactory threshold. As shown in Fig. 5, there are more units in the region with high nonlinearities, which is quite reasonable. The function surface implemented by this immunity-based hybrid learning system is shown in Fig. 6, which is very close to the function surface for which we seek to approximate. The maximum absolute error is 0.3580, which is slightly above 0.35, because we are using a different test data set consisting 2000 data pairs (rather than the training data set for structure and parameter tuning).

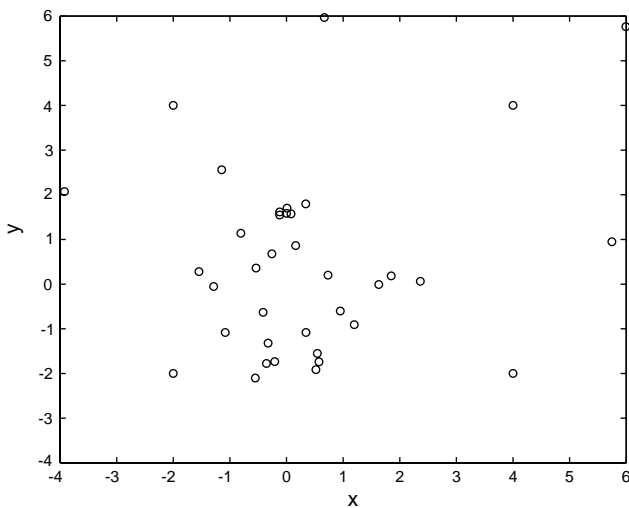


Fig. 5. Centers of the resulting receptive field units (using the learning from clonal selection approach).

The performance of a function approximator with both structure and parameter adaptation becomes remarkable when the function encountered is complicated enough. A function only with high frequency components may be easy to approximate (in the sense of structure adjustment), for instance, by increasing the grid of the receptive field units (given a large amount of well distributed training data). Here, we are trying to approximate a function which consists of both high-frequency and low-frequency components. Without a priori knowledge of the nonlinearity of the function, a very fine grid of receptive field units is required for radial basis function networks, which, actually, may result in a redundant approximation structure that wastes many computations. For example, in order to accurately approximate the function given in Eq. (12), a radial basis function network with 64 units (on an 8×8 grid) is required, whereas using 49 units (a 7×7 grid) is not appropriate due to the poor error surface. The proposed immunity-based hybrid learning system, however, is capable of approximating the function accurately enough with 36 well assigned receptive field units. Moreover, there is no generalization or overtraining problem encountered.

For the above function approximation problem one may argue that a nonlinear in the parameter approximator tuned by gradient methods may obtain good performance with relatively small number of parameters. However, it is difficult, in general, to find the optimal solution, and the parameters may get distracted by the multiple local minima, which we will illustrate later by an example of using multilayer perceptron neural networks tuned by the Levenberg–Marquardt method.

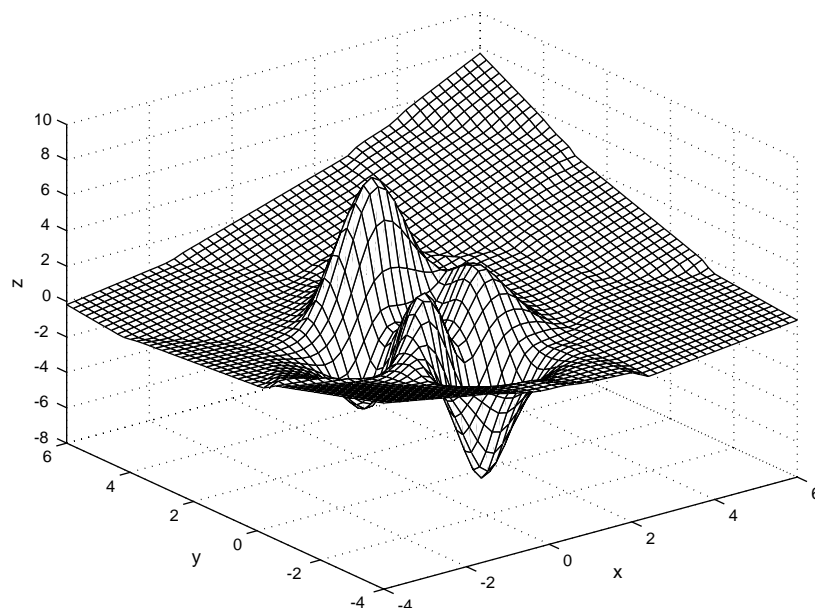


Fig. 6. Function surface of the immunity-based approximator (using the learning from clonal selection approach).

4.2. Learning from internal affinity

This learning approach performs structure adjustment according to the endogenous behavior of the radial basis function network and is capable of reducing the redundant receptive field units. The learning method is described as follows:

1. *Initialization*: The approach starts with a spatially localized model (radial basis function network) consisting of a large set of units, which is capable of good estimation. Sometimes, this can be obtained by a fine division of the input space and then by assigning a receptive field unit for each subregion, or by the learning from clonal selection method described in last section.
2. *Recognition*: We calculate the affinities between the receptive field units according to Eqs. (7)–(9).
3. *Refinement*: This process serves as structure adaptation for the approximator by eliminating the redundant units. We find the pair of receptive field units whose value of affinity measure is the smallest, and eliminate (any) one of them.
4. *Parameter adaptation*: We use a least-squares method to tune the linear parameters in order to maintain good approximation accuracy by allocating the computational power optimally among the remaining receptive field units, while the centers and relative widths of the receptive field units are kept frozen (to reduce computation complexity).
5. *Generalization*: The performance of the approximator is tested by calculating the fitness function of the hybrid learning system, which is defined as the MAE. If the value of the fitness function is above a “satisfactory threshold,” that is, the error becomes unacceptable, we stop the learning process and come up with the last satisfied approximator. Otherwise, go to Step 2 for structure and parameter adaptation again.

Again, to test the method we use the nonlinear function defined in Eq. (12). We start the approximation process from a radial basis function network with 100 receptive field units, whose centers are uniformly distributed in a 10×10 grid. The “satisfactory threshold” is chosen to be 0.7 because the centers of the this learning system are distributed more regularly and an acceptable function approximation surface corresponds to a larger value of satisfactory threshold (compared to 0.35 in the learning from clonal selection approach). The performance of the approximator degrades along with the structure adaptation when more units are eliminated from the learning system, and it finally results in a network with 32 receptive field units, whose centers are shown in Fig. 7. As shown in Fig. 8, the function surface implemented by this immunity-based learning system is

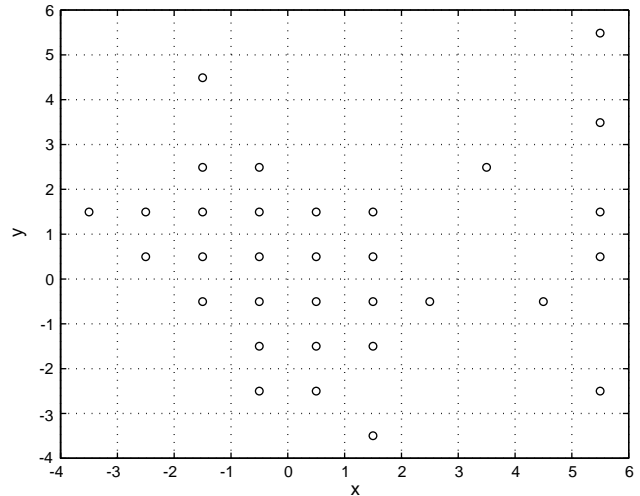


Fig. 7. Centers of the resulting receptive field units (using the learning from internal affinity approach).

also very close to the function surface for which we seek to approximate, and the maximum absolute error for the testing data set is 0.6774. Note that the MAE here is much larger than that in the learning from clonal selection approach but their function surfaces are very similar. This is because MAE is only one possible error measure that we are using to terminate the learning process and its effect may be different in different learning approaches. Thus, we view the function surface (as in Figs. 6 or 8) as a kind of general measure of the goodness of the approximator and we use it to determine the appropriate “satisfactory threshold.”

4.3. Remarks

Generally, the function approximation problem has not been explicitly studied in the applications of artificial immune systems. However, there do exist some connections between the function approximation problem and other applications, in particular, machine learning inspired by immune systems, pattern classification based on the endogenous double plasticity of the immune network, and function optimization with diversity and learning of the immune algorithm. In Seront and Bersini (1994), Bersini presented an immune recruitment mechanism for multilayer perceptron neural networks, which bared some resemblance to the approaches proposed in this paper. In Seront and Bersini (1994) a neuron is eliminated from the network if the average value of the weights connected to it is below a “useless threshold”. In addition, if these average values for all neurons in the network are above a certain “settling threshold”, which implies that the learning ability of the network is low because all neurons are almost saturated, a new neuron will be added to the network.

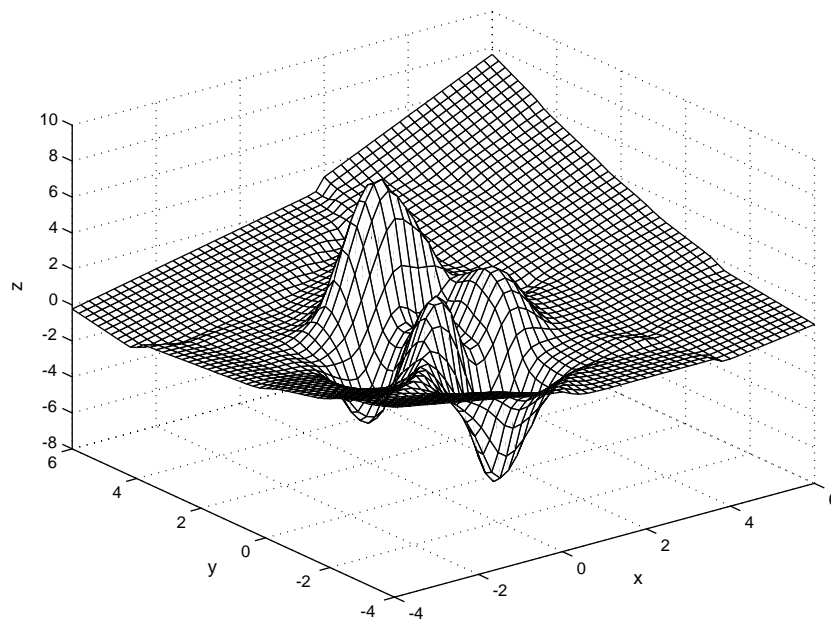


Fig. 8. Function surface of the immunity based approximator (using the learning from internal affinity approach).

In this paper we have studied immunity inspired learning approaches for radial basis function neural networks with a spatially localized model architecture (different to that of multilayer perceptron). Two new hybrid learning approaches are proposed. The learning from clonal selection approach is capable of adding more receptive field units and the structure adjustment is driven by the exogenous criteria (i.e., the error function). The learning from internal affinity approach can eliminate redundant elements from the structure according to the endogenous behavior of the spatially localized model (i.e., the similarities of the receptive field units measured by the Euclidean distance). We believe that constructing hybrid learning systems based on spatially localized models resemble the nature immune system better. The patterns (strength functions) are stored in a distributed manner in the receptive field units. Each receptive field unit is capable of local approximation, while a collection of these units (i.e., the radial basis function network) is capable of serving as a universal approximator. Compared to multilayer perceptrons (whose capability is gained through the collaboration of all the neurons), the radial basis function networks resemble the nature immune systems more (where each antibody is effective to recognize certain antigens). Furthermore, using the spatially localized model architecture provides the advantages of design flexibility for structure and parameter adaptation, that is, we can apply the immunity-based structure adaptation approach to specify the architecture and the parameters (centers and relative widths) that enter nonlinearly, followed by using least squares to train the linear parameters (strength function parameters). This also

often implies less computational complexity compared to tuning all parameters with the gradient methods. In other words, it is a kind of combination of local tuning (where the new unit is added only to compensate for the worst part of the approximator) and global tuning (where all the parameters of strength functions are retrained to achieve global optimization afterwards). In addition, note that in this paper we use the learning from internal affinity approach to remove the redundant units, which can be viewed as a kind of decremental algorithms. The decremental algorithms inherently assume the availability of a complex approximator sufficient for accurate mapping. This assumption may be relatively easier, achieved by a spatially localized model with a fine grid of units tuned by the least squares method rather than by a multilayer perceptron neural network (which are often tuned by gradient methods and may meet the problem of local minima).

However, we do not consider our approach universal. For other applications, one of the other methods may provide a better solution. Indeed, none of the existing algorithms are proven to be optimal in structure adaptation for learning systems and we could not show that either. For different problems and different measures of “best”, different algorithms may give structures with a minimal number of units (neurons or rules).

4.4. Comparisons

Compared to the learning from clonal selection approach, the learning from internal affinity approach results in more regularly distributed receptive field unit

centers, but its approximation performance is slightly worse than that of the *learning from clonal selection* approach. This is because the arrangement of its centers is restricted due to the initialization, whereas the assignment of the centers for the learning from clonal selection approximator is more flexible so that it may capture more characteristics of the function. Moreover, note that as shown in Fig. 5 some receptive field units generated in the learning from clonal selection approach are very close to each other. This suggests that we could integrate the learning from clonal selection approach and the learning from internal affinity approach together. For instance, we could execute the learning from clonal selection approach first to add units at the positions that are necessary, and then execute the learning from internal affinity approach to remove the redundant ones. We may expect the resulting approximator to have a smaller structure while still maintaining sufficient accuracy. Actually, by using this strategy for the above problem we reduce the number of units to 31 and still have a function surface accurately enough.

To illustrate the performance of the immunity-based approximators, we also implement a multilayer perceptron neural network (with one hidden layer), which is nonlinear in the parameters, tuned by the Levenberg–Marquardt method. The activation function for the neurons in the hidden layer is chosen to be the hyperbolic tangent function. We use 30 hidden neurons in the network, so that we have $(2 + 1)30 + (30 + 1) = 121$ parameters, which is equivalent to $40(2 + 1) = 120$ parameters for 40 units in the radial basis function network we used before. The function surface implemented by the neural networks, as shown in Fig. 9, is much worse than the ones obtained by the immunity

based approximators, and the maximum absolute error for the testing 200 samples is 1.2631. In order to improve the performance, we increase the number of hidden neurons to be 40 (i.e., 161 parameters). As shown in Fig. 10, the function surface is better and the maximum absolute error for the testing 200 samples is 0.7206. However, the result is still worse than that of the immunity based approximators, which implies that a local minimum may be encountered.

5. Application to nonlinear dynamical system modeling

Function approximation is a generic problem that permeates many fields. In this section, a nonlinear dynamical system modeling problem, in particular, model development for an aircraft jet engine (General Electric XTE46), is provided to demonstrate the potential of the proposed immunity-based hybrid learning methods.

Model development for jet engines is complicated as the engine behavior is different for different operating regions and engine quality parameters (due to the presence of engine-to-engine manufacturing differences and engine deterioration during normal operation). The fundamental dynamic characteristics of the XTE46 engine can be represented by a single-input single-output system in the form

$$\dot{x} = f(x, u, c, p), \quad (13)$$

$$y = h(x, u, c, p), \quad (14)$$

where $x = [\text{XNL}, \text{XNH}]^T$ is the state vector including the fan rotor speed and core rotor speed, $u = \text{WF36}$ is

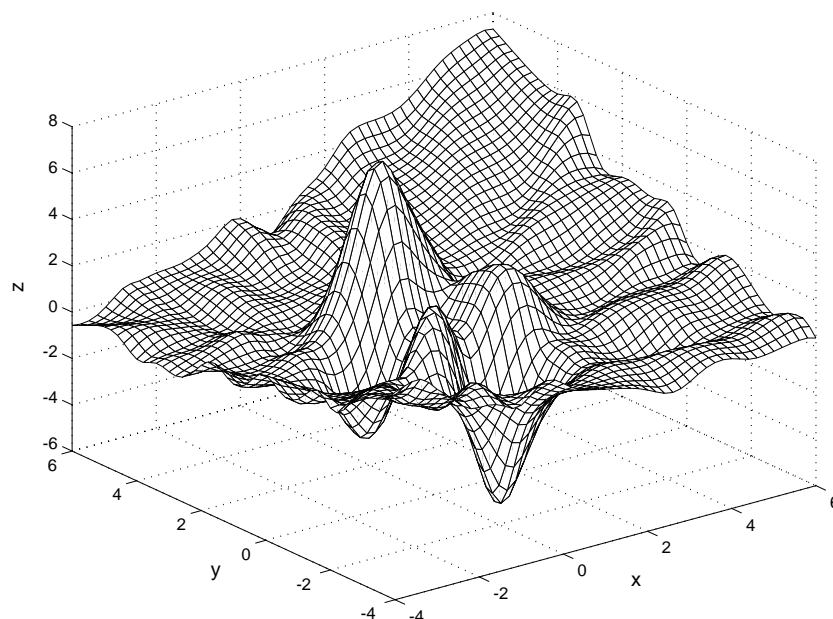


Fig. 9. Function surface of the multilayer perceptron neural networks with 30 hidden neurons.

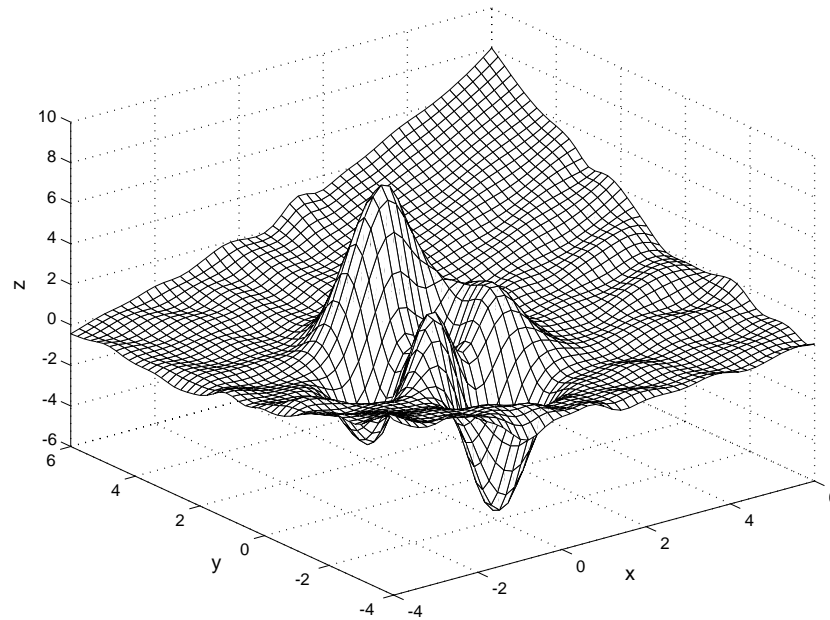


Fig. 10. Function surface of the multilayer perceptron neural networks with 40 hidden neurons.

the input variable (combustor fuel flow), and the output variable $y = \text{XN2}$ is the measurement of the state variable XNL . The vector $c = [\text{ALT}, \text{XM}, \text{PC}]^T$ represents the operating condition of the engine (the altitude, mach number, and throttle setting represented by power code). The vector $p = [\text{ZSW2}, \text{SEDM2}, \text{ZSW7D}, \text{SEDM7D}, \text{ZSW27}, \text{SEDM27}, \text{ZSW41}, \text{ZSE41}, \text{ZSW49}, \text{ZSE49}]^T$ represents the quality parameters of the engine. The function $f(\cdot)$ denotes the unknown nonlinear characteristics of the engine which is different at different operating conditions and for different quality parameters. Although it is theoretically possible to build one neural network to approximate the engine dynamics by interpolating all engine data collected from the whole space of operating conditions and quality parameters, it is not feasible in practice because of the huge amounts of data (millions of data pairs). Instead, we build a grid of neural network models to approximate local engine dynamics (specified by fixed values of operating conditions and quality parameters), and then interpolate these local models to generate the “global” model (actually, we build a “regional” model valid in the “climb” region). Note that it is possible to obtain the “regional” model by fuzzy interpolation on a grid of these local models (Diao and Passino, 2001). However, using a too coarse grid may only obtain limited model accuracy, while using a too fine grid may result in a redundant structure.

We use the learning from clonal selection approach to specify the structure of the “regional” nonlinear model. To illustrate the basic ideas we consider a simplified scenario where the engine dynamics are varying with respect to three key operating condition variables (ALT, XM, and PC) but the quality parameters

are fixed. The process of hybrid learning starts from generating a regional model consisting of only one local model located in the center (ALT = 15 000, XM = 0.7, and PC = 47.5) of the climb region (ALT \in [12 500, 17 500], XM \in [0.6, 0.8], and PC \in [45, 50]). Afterwards, more local models are recruited at the locations where poorest approximation performance is observed. The performance of the regional engine model becomes better along with the structure adaptation when more appropriate local models are recruited. Fig. 11 compares the performance of a regional model composed of 64 local models selected by the immunity-based approach with a regional model of the same number of local models but uniformly distributed on a $4 \times 4 \times 4$ grid. The operating condition in Fig. 11 is different to those where the local models are used. The solid line represents the system response of the XTE46 engine, the dashed line represents the response of the regional model constructed by the immunity-based approach, and the dotted line represents the response of the model generated from the grid. We can see that the immunity-based regional model is generally more accurate, especially when the XNL is low.

Note that for the engine modeling problem we use the learning from clonal selection method to allocate local models instead of receptive field units. This is an extension of the algorithm described in Section 4 due to the complexity of modeling a jet engine. Hence, we move the immunity-based structure selection problem from considering the units of a single neural network model to the units of a regional model composed of multiple local neural network models. This also demonstrates the versatile applicability of the immunity-based approach.

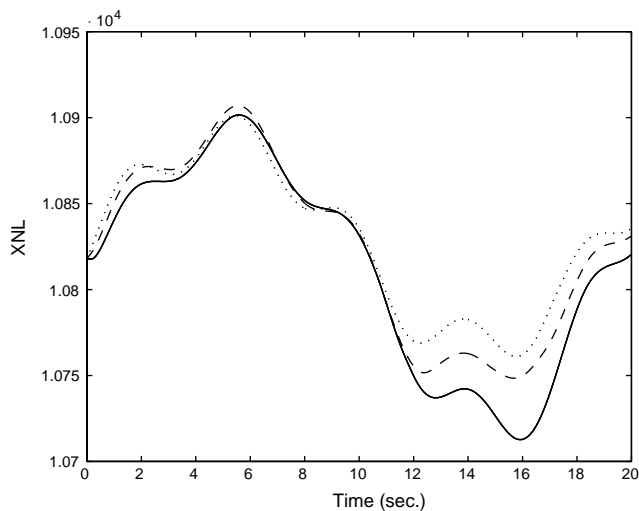


Fig. 11. Performance of the regional model.

6. Conclusion

From the point of view of information processing the immune system is a highly parallel and distributed intelligent system which has learning, memory, and associative retrieval capabilities. In this paper we present two immunity-based hybrid learning approaches for function approximation (or regression) problems that involve adjusting the structure and parameters of spatially localized models (e.g., radial basis function networks). The number and centers of the receptive fields for local models are specified by immunity-based structure adaptation algorithms, while the parameters of the local models, which enter in a linear fashion, are tuned separately using a least squares method. The effectiveness of the procedure is demonstrated through a nonlinear function approximation problem and a nonlinear dynamical system modeling problem.

References

- Ash, T., 1989. Dynamic node creation in backpropagation networks. *Connection Science* 1, 365–375.
- Barron, A.R., 1993. Universal approximation bounds for superpositions of a sigmoid function. *IEEE Transactions on Information Theory* 39 (3), 930–945.
- Bersini, H., 1992. Reinforcement and recruitment learning for adaptive process control. In: *Proceedings of the IFAC/IFIP/IMACS on Artificial Intelligence in Real Time Control*, Delft, The Netherlands, pp. 331–337.
- Bersini, H., 1999. The endogenous double plasticity of the immune network and the inspiration to be drawn for engineering artifacts. In: Dasgupta, D. (Ed.), *Artificial Immune Systems and Their Applications*. Springer, Berlin, Germany, pp. 22–44.
- Bersini, H., Seront, G., 1992. In search of a good evolution-optimization crossover. In: Manner, R., Manderick, B. (Eds.), *Parallel Problem Solving from Nature, II*, Elsevier, Amsterdam, pp. 479–488.
- Bersini, H., Varela, F., 1990. Hints for adaptive problem solving gleaned from immune networks. In: *Proceedings of the First Conference on Parallel Problem Solving from Nature*, Dertmund, Germany, pp. 343–354.
- Bersini, H., Varela, F., 1994. The immune learning mechanisms: reinforcement, recruitment and their applications. In: Patton, R. (Ed.), *Computing with Biological Metaphors*. Chapman & Hall, London, pp. 166–192.
- Dasgupta, D., 1996. Using immunological principles in anomaly detection. In: *Proceedings of the Artificial Neural Networks in Engineering (ANNIE'96)*, St. Louis, MO, pp. 443–448.
- Dasgupta, D., 1997. Artificial neural networks and artificial immune systems: similarities and differences. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Orlando, FL, pp. 873–878.
- Dasgupta, D., 1998. An artificial immune system as a multi-agent decision support system. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, San Diego, CA, pp. 3816–3820.
- Dasgupta, D., 1999. *Artificial Immune Systems and their Applications*. Springer, Berlin, Germany.
- Dasgupta, D., Forrest, S., 1999. Artificial immune systems: a new frontier in artificial intelligence. In: *International Conference on Intelligent Processing and Manufacturing of Materials*, Honolulu, HI, pp. 257–268.
- de Castro, L.N., Timmis, J., 2002. An artificial immune network for multimodal function optimization. In: *Proceedings of the 2002 Congress on Evolutionary Computation*, Honolulu, HI, pp. 699–704.
- de Castro, L.N., Von, Z.N., Fernando, J., 2002. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation* 6 (3), 239–251.
- Diao, Y., Passino, K.M., 2001. Stable fault tolerant adaptive fuzzy/neural control for a turbine engine. *IEEE Transactions on Control Systems Technology* 9, 494–509.
- Fabri, S., Kadiramanathan, V., 1996. Dynamic structure neural networks for stable adaptive control of nonlinear systems. *IEEE Transactions on Neural Networks* 7 (5), 1151–1167.
- Fahlman, S.E., Lebiere, C., 1990. The cascade-correlation learning architecture. In: Touretzky, D.S. (Ed.), *Advances in Neural Information Processing Systems 2*. Morgan Kaufmann, San Mateo, CA, pp. 524–532.
- Farmer, J.D., 1990. A rosetta stone for connectionism. *Physica D* 42, 153–187.
- Farmer, J.D., Packard, N.H., Perelson, A.S., 1986. The immune system, adaptation, and machine learning. *Physica D* 22, 187–204.
- Friedman, J.H., Stutzle, W., 1981. Projection pursuit regression. *Journal of the American Statistical Association, Theory and Models* 76, 817–823.
- Fukuda, T., Mori, K., Tsukiyama, M., 1999. Parallel search for multimodal function optimization with diversity and learning of immune algorithm. In: Dasgupta, D. (Ed.), *Artificial Immune Systems and Their Applications*. Springer, Berlin, Germany, pp. 210–220.
- Gilbert, C.J., Routen, T.W., 1994. Associate memory in an immune-based system. In: *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, Seattle, pp. 852–857.
- Hanson, S.J., 1990. Meiosis networks. In: Touretzky, D.S. (Ed.), *Advances in Neural Information Processing Systems 2*. Morgan Kaufmann, San Mateo, CA, pp. 533–541.
- Hassibi, B., Stork, D.G., 1993. Second order derivatives for network pruning: Optimal brain surgeon. In: Hanson, S.J., Cowan, J.D., Giles, C.L. (Eds.), *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann, San Mateo, CA.
- Hoffmann, G.W., 1986. A neural network model based on the analogy with the immune system. *Journal of Theoretical Biology* 122, 33–67.

- Huang, S.-J., 2002. Application of immune-based optimization method for fault-section estimation in a distribution system. *IEEE Transactions on Power Delivery* 17 (3), 779–784.
- Hunt, J.E., Cooke, D.E., Holstein, H., 1995. Case memory and retrieval based on the immune system. In: *First International Conference on Case Based Reasoning*, Sesimbra, Portugal, pp. 205–216.
- Hunt, J.E., Timmis, J., Cooke, D., Neal, M., King, C., 1999. Jisys: the development of an artificial immune system for real world applications. In: Dasgupta, D. (Ed.), *Artificial Immune Systems and Their Applications*. Springer, Berlin, Germany, pp. 157–186.
- Jerne, N.K., 1974. Towards a network theory of the immune system. *Annales Immunology* 125C, 373–389.
- Kephart, J.O., 1994. A biologically inspired immune system for computers. In: *Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems*, Cambridge, MA, pp. 130–139.
- Kephart, J.O., Sorkin, G.B., Arnold, W.C., Chess, D.M., 1995. Biologically inspired defenses against computer viruses. In: *International Joint Conference on Artificial Intelligence*, Montreal, Canada, pp. 985–996.
- Kephart, J.O., Sorkin, G.B., Swimmer, M., White, S.R., 1999. Blueprint for a computer immune system. In: Dasgupta, D. (Ed.), *Artificial Immune Systems and Their Applications*. Springer, Berlin, Germany, pp. 242–261.
- Kruschke, J.K., 1988. Creating local and distributed bottlenecks in hidden layers of back-propagation networks. In: Touretzky, D.S., Hinton, G., Sejnowski, T. (Eds.), *Proceedings of the 1988 Connectionist Models Summer School*. Morgan Kaufmann, San Mateo, CA, pp. 120–126.
- Kwok, T.Y., Yeung, D.Y., 1997. Constructive algorithms for structure learning in feedforward neural networks for regression problems. *IEEE Transactions on Neural Networks* 8 (3), 630–645.
- Perelson, A.S., 1989. Immune network theory. *Immunological Review* 110, 5–36.
- Platt, J., 1991. A resource-allocating network for function interpolation. *Neural Computation* 3, 213–225.
- Pramanik, S., Kozma, R., Dasgupta, D., 2002. Dynamical neuro-representation of an immune model and its application for data classification. In: *Proceedings of the International Joint Conference on Neural Networks*, Honolulu, HI, pp. 130–135.
- Reed, R., 1993. Pruning algorithms—a survey. *IEEE Transactions on Neural Networks* 4 (5), 740–747.
- Schaal, S., Atkeson, C.G., 1998. Constructive incremental learning from only local information. *Neural Computation* 10, 2047–2084.
- Seront, G., Bersini, H., 1994. An algorithm for self-structuring neural net classifiers. In: *Proceedings of the Second IEEE Conference on Neural Networks*, Orlando, FL, pp. 1307–1312.
- Sietsma, J., Dow, R.J.F., 1991. Creating artificial neural networks that generalize. *Neural Networks* 4 (1), 67–69.
- Wang, L., Jiao, L., 2001. An immune neural network used for classification. In: *Proceedings of the 2001 IEEE International Conference on Data Mining*, San Jose, CA, pp. 657–662.
- Weigend, A.S., Rumelhart, D.E., Huberman, B.A., 1991. Backpropagation, weight-elimination and time series prediction. In: Touretzky, D.S. (Ed.), *Connectionist Models: Proceedings of the 1990 Summer School*. Morgan Kaufmann, San Mateo, CA, pp. 105–116.