



Machine Learning Emulation in Nature-inspired Computation Systems

Jianhao Tan (Corresponding author)

Electrical and Information Engineering College, Hunan University

Changsha 410082, China

E-mail: tanjianhao96@sina.com.cn

Jing Zhang

Electrical and Information Engineering College, Hunan University

Changsha 410082, China

Fu Guo

Electrical and Information Engineering College, Hunan University

Changsha 410082, China

The research is financed by National Natural Science Foundation (Approval No. 60634020) and Hunan Province Natural Science Foundation (Approval No. 08JJ3132).

Abstract

The whole frame of nature_inspired computation systems is inquired into, the characteristics of machine learning in nature_inspired computation systems are researched, and a particular scheme on machine learning in nature_inspired computation systems is designed with environment being gathered present data; study unit adopting fuzzy optimization algorithm based on genetic algorithm; knowledge base adopting fuzzy optimization BP neural networks; executive unit being complicated industry process. The fuzzy optimization learning algorithm of fuzzy optimization BP neural networks is built, the flow chart of the algorithm is constructed, and the emulation test is made. At last, the design criteria of flash metal consumption are obtained, and the stability of the algorithm is verified through this example. The result shows that machine learning makes nature_inspired computation systems be able to gain know; edge automatically, their quality improved, their intelligent level advanced, and machine learning will greatly influence the memory mode, information input mode and system structures of nature_inspired computation systems.

Keywords: Nature_inspired computation, Machine learning, Emulation, Genetic algorithm, Fuzzy optimization, Neural networks, Flash metal consumption

1. Introduction

Based on the natural world, particularly on the functions, characteristics and mechanism of the organism, nature_inspired computation researches various information processing mechanisms contained in them, takes out the relevant calculation models, and designs the relevant algorithms which are applied in various fields (Cai Zixing, Xu Guangyou, 2004; Cheung L C Y, Holden T S I, 1991, pp499-508; Chien Steve, Decoste Dennis, Doyle Richard, et al, 1997, pp103-121; McCormark D M, Day R, 1993, pp 6; Ng E Y K, Fo K S C, Peh Y C, et al, 2002, pp152-157). Nature_inspired computation is usually a class of self-adaptive, self-organized, self-learning algorithms, including these researching fields such as evolutionary computation, neural computation, ecological terms, quantum computing and complex adaptive systems and so on which base the design of algorithms on the natural world mechanism, having the features of imitating the natural world and being able to solve many complex problems that the traditional method is hard to do. It has a very good use in such fields as optimizing designing, optimal controlling, computer network security, creative designing in large-scale complex systems.

According to the view that the nature_inspired computation is the internal mechanism in the natural world and the basis of intelligence, a general framework for nature_inspired computation is made (Chien Steve, Decoste Dennis, Doyle

Richard, et al, 1997, pp103-121). However, whether the natural_inspired computation is the internal mechanism of nature phenomenon needs waiting for future experiment and natural phenomena verification. Besides, the model which maps the computation mode existing in natural world broadly to the new nature_inspired computation mode has brought forward, and then the structure, principle and characteristic of the model have been analysed (Gong Tao, Cai Jingfeng, Cai Zixing, 2003, pp727-732). Nature_inspired computation can enhance many characteristics in general systems and give them new life. Because nature_inspired computation is an innovative one, the development of the calculation will promote the fundamental development of systematics including the generalization of system structure and the expansion of system domain. How to coordinate the relationship between the traditional computations and such emerging computations as nature_inspired computation and how to deal with the relationship among the branches of these emerging computations are pending further study.

On the basis of describing the whole framework of nature_inspired computation systems, the article studies emphatically the mechanism of action of machine learning in them. A particular scheme on machine learning in nature_inspired computation systems is designed with environment being gathered present data; study unit adopting fuzzy optimization algorithm based on genetic algorithm; knowledge base adopting fuzzy optimization BP neural networks; executive unit being complicated industry process. The principle, process and characteristics of the machine learning in nature_inspired computation systems are described and exemplified.

2. Whole Frame of Nature_inspired Computation Systems

Nature_inspired computation makes the high_level decision_making in the computer, the advanced mathematical modeling and synthetic approach in system theory, as well as linguistics methods dealing with inaccurate and incomplete information combined together, forming an agreed method to meet the project. It consists of such three levels as organization, coordination and implementation.

The organization level functions imitating human behavior, having highest-level intelligence in nature_inspired computation systems. It acts as reasoning, planning, decision-making and exchanging long-term memory information, as well as learning under the information from external environment and lower level feedback; Being an interface between the organization level and the implementation level, the coordination level functions coordinating these tasks in accordance with the instruction of the organization level; Being the lowest level, the implementation one completes various detailed tasks.

The intelligence of nature_inspired computation is mainly represented in high level. These problems encountered in the high level are often uncertain, and so adopting the knowledge_based organization level is just about right because the knowledge-based organization level can facilitate process information and use the instinct reasoning logic and experience of human being. The working process of the system can be described in two ways: from the horizontation, dividing a complex system into a number of interrelated sub_systems, each of which is equipped a controller in order to be easy to be directly controlled, as so to make the complex issue simplified in a large extent; from the vertication, breaking down the complex system from high to low according to the quantity of the knowledge or the level of intelligence needed for the system, which means that intelligence controller systems have two_fold meaning in the structure of the multi_level: on one hand, the controlled object has many levels, some of which have many uncertain and unknown factors. The more the levels are, the harder the according control is; On the other hand, there exists the control in many levels, which is related with the designing, development and running of nature_inspired computation systems. The whole framework of nature_inspired computation systems is shown in Figure.1 (a), and these parts instructed by the rough arrows are the machine learning sub_systems abstracted from nature_inspired computation systems, which constitute the main feedback loops in nature_inspired computation systems and whose diagram is shown in Figure.1(b).

3. Machine Learning in Nature_inspired Computation Systems

Knowledge, knowledge representation and reasoning algorithm using knowledge are the core of the nature_inspired computation systems, and the machine learning is the key among them. For centuries, philosophers and psychologists have believed: the basic mechanism on learning is trying translating successful performance action in one situation to another similar new situation. Learning is the process of acquiring knowledge, experiencing, improving performance, finding regularity and adapting environment. Figure.1 (b) gives a simple model on machine learning in nature_inspired computation systems. The model includes four basic sections of machine learning systems. The environment provides external information, similar to the role of the teacher, the study unit deals with the information provided by the environment, equivalent to a variety of learning algorithms, the knowledge base stores information by means of some knowledge representation mode, the executive unit completes some using the knowledge in the knowledge base.

A particular scheme on machine learning in nature_inspired computation system is given with environment being gathered present data, knowledge base adopting artificial neural network, study unit adopting fuzzy optimization algorithm based on genetic algorithm, and executive unit being complicated industry process. The diagram of the

system is shown in Figure.2.

3.1 Neural Network Model

The knowledge base of this article adopts fuzzy optimization BP neural networks (Horikawa S, 1992, 25-29). The input_output relationship of the neural network can be described as follows.

To Node i of the input layer, its input is

$$I_i = x_i \quad i = 1, 2, \dots, M \tag{1}$$

Its output is

$$O_i = I_i \tag{2}$$

To Node j of the hidden layer, its input is

$$I_j = \sum_{i=1}^M w_{ij} O_i \quad i = 1, 2, \dots, M; j = 1, 2, \dots, Q \tag{3}$$

Its output is

$$O_j = f(I_j) = \frac{1}{1 + (I_j^{-1} - 1)^2} = \frac{1}{1 + [\sum_{i=1}^M w_{ij} O_i - 1]^2} \tag{4}$$

The output layer has only a node, its input is

$$I = \sum_{j=1}^Q w_j O_j \tag{5}$$

Its output is

$$y = O = f(I) = \frac{1}{1 + (I^{-1} - 1)^2} = \frac{1}{1 + [\sum_{j=1}^Q w_j O_j - 1]^2} \tag{6}$$

The input-output relationship of the neural network can be written as $y = F(\mathbf{X}, \mathbf{w})$, where $\mathbf{X} = (x_1, x_2, \dots, x_M)$, $\mathbf{w} = (w_{ij}, w_j | i = 1, 2, \dots, M; j = 1, 2, \dots, Q)$. The process of the weight estimation of the neural network is: The N instances $(\mathbf{X}_k, Y_k^*), k = 1, 2, \dots, N$ which are known constitute a group of study samples, where the input x of the instance k can be represented as a vector $\mathbf{X}_k = (x_{1k}, x_{2k}, \dots, x_{Mk})$ of M variables, and the expected output of the instance K is the single output $Y_k^* = y_k^*, y_k = F(\mathbf{X}_k, \mathbf{w})$.

Then the problem of the weight estimation of the neural network is converted into the fuzzy optimization problem as discribed below (Jimenez F, Sanchez G, Cadenas J M, Gomez-Skameta A F. A, 2004, 23-26).

$$\max y_k = F(\mathbf{X}_k, \mathbf{w}), \quad k = 1, 2, \dots, N \tag{7}$$

Where $y_k, k = 1, 2, \dots, N$ is the fuzzy goal, whose expected value is y_k^* , whose symmetry tolerance is ε , and whose changing zone is $[y_k^* - \varepsilon, y_k^* + \varepsilon]$. In Domain A , we create fuzzy goal set $G_k, k = 1, 2, \dots, N$ corresponding to y_k , whose membership function $\mu_{G_k}(\mathbf{w})$ is defined as follows:

$$\mu_{G_k}(\mathbf{w}) = \begin{cases} 0 & y_k \leq y_k^* - \varepsilon \\ 1 - \frac{y_k^* - y_k}{\varepsilon} & y_k^* - \varepsilon \leq y_k \leq y_k^* \\ 1 - \frac{y_k - y_k^*}{\varepsilon} & y_k^* < y_k < y_k^* + \varepsilon \\ 0 & y_k \geq y_k^* + \varepsilon \end{cases} \tag{8}$$

If the fuzzy goal set $G_k, k = 1, 2, \dots, N$ is given in Domain A , the intersection $G = \bigcap_{k=1}^N G_k$ is called fuzzy superior set. The basic idea of the weight estimation of fuzzy optimization based on neural network is: in the decision space A , finding \mathbf{w}^* which makes the membership function $\mu_{G_k}(\mathbf{w})$ of the fuzzy superior set G get the maximal value. \mathbf{w}^* is

called the fuzzy optimum solution. $\mu_G(\mathbf{w})$ can be calculated by the following equation (Jimenez F, Sanchez G, Cadenas J M, Gomez-Skameta A F. A, 2004, 23-26).

$$\mu_G(\mathbf{w}) = \bigwedge_{k=1}^N \mu_{G_k}(\mathbf{w}) = \min(\mu_{G_k}(\mathbf{w}) | k=1, 2, \dots, N) \quad (9)$$

The mathematical model of fuzzy optimization can be described as follows: solving \mathbf{w}^* in order to make

$$\mu_G(\mathbf{w}^*) = \max(\mu_G(\mathbf{w})) = \max(\min(\mu_{G_k}(\mathbf{w}) | k=1, 2, \dots, N)) \quad (10)$$

3.2 Learning Algorithm

1) Weight Estimation of Neural Networks

The issue (10) is the unconstrained optimization problem, but its goal is not continuously derivative. This problem can not be solved by traditional methods but can use genetic algorithms to optimize, This article try to solve the problem by adopting the particular genetic algorithm using mutating along the weighted gradient direction, which was brought up by Wang Dingwei and Tang Jiafu (Tang Jiafu, Wang Dingwei, 2000). The algorithm regards the mutation as the main operator, then adoptes the arithmetic combination crossover operator in the later period, The basic idea of the algorithm is: firstly, randomly generating initial population containing *pop_size* individuals, which are chosen to generate descendants whose membership $\mu_G(\mathbf{w})$ on the fuzzy superior set \mathbf{G} gets increased. With the genetic algorithm executed, these individuals whose membership is less than α_0 (acceptable membership) get less chance of generating offsprings than others. With the genetic generation number increasing, these individuals whose membership is less than α_0 will eventually die, while others get survival. After a few of generations, the membership of all individuals are greater than α_0 , and most of individuals will be close to optimum solution.

To the individual \mathbf{w} , let $\mu_{\min}(\mathbf{w}) = \min(\mu_{G_k}(\mathbf{w}) | k=1, 2, \dots, N)$, if $\mu_{\min}(\mathbf{w}) \leq \mu_{G_k}(\mathbf{w}) < 1$, move along the gradient direction of $\mu_G(\mathbf{w})$ so as to improve the value of $\mu_G(\mathbf{w})$. The smaller $\mu_G(\mathbf{w})$ is, the more improvement it gets. Based on the above idea, the weighted gradient direction can be constructed as follows.

$$D(\mathbf{w}) = \sum_{k=1}^N \alpha_k \nabla \mu_{G_k}(\mathbf{w}) = \sum_{k=1}^N \alpha_k (\nabla \mu_{G_k}(w_j), \nabla \mu_{G_k}(w_{ij})) = (D(w_j), D(w_{ij})) \quad (11)$$

Where α_k is the weight along the gradient direction, which is defined as follows.

$$\alpha_k = \begin{cases} 0 & \mu_{G_k} = 1 \\ \frac{1}{\mu_{G_k} - \mu_{\min} + e} & \mu_{\min} \leq \mu_{G_k} < 1 \end{cases} \quad (12)$$

Where e is a small-enough positive number, $1/e$ is the maximal weight.

$$\nabla \mu_{G_k}(w_j) = \frac{1}{\varepsilon} \text{sgn}(y_k^* - y_k) \nabla y_j \quad k=1, 2, \dots, N \quad (13)$$

$$\nabla \mu_{G_k}(w_{ij}) = \frac{1}{\varepsilon} \text{sgn}(y_k^* - y_k) \nabla y_{ij} \quad k=1, 2, \dots, N \quad (14)$$

Now ∇y_j and ∇y_{ij} are derivated as follows:

$$\nabla y_j = \frac{\partial y}{\partial w_j} = \frac{\partial y}{\partial I} \frac{\partial I}{\partial w_j} = \frac{\partial O}{\partial I} \frac{\partial (\sum_{j=1}^Q w_j O_j)}{\partial w_j} = f'(I) O_j \quad (15)$$

$$\nabla y_{ij} = \frac{\partial O}{\partial O_j} \frac{\partial O_j}{\partial I_j} \frac{\partial (\sum_{i=1}^M w_{ij} O_i)}{\partial w_{ij}} = \frac{\partial O}{\partial O_j} f'(I_j) O_i \quad (16)$$

Whereas

$$\frac{\partial O}{\partial O_j} = \frac{\partial O}{\partial I} \frac{\partial I}{\partial O_j} = \frac{\partial O}{\partial I} \frac{\partial (\sum_{j=1}^Q w_j O_j)}{\partial O_j} = f'(I) w_j \quad (17)$$

So

$$\nabla y_{ij} = f'(I) f'(I_j) w_j O_i \quad (18)$$

Therefore

$$\nabla \mu_{G_k}(w_j) = \frac{1}{\varepsilon} \text{sgn}(y_k^* - y_k) f'(I) O_j \quad (19)$$

$$\nabla \mu_{G_k}(w_{ij}) = \frac{1}{\varepsilon} \text{sgn}(y_k^* - y_k) f'(I) f'(I_j) w_j O_i \quad (20)$$

$$D(w_j) = \frac{1}{\varepsilon} \sum_{k=1}^N \alpha_k \text{sgn}(y_k^* - y_k) f'(I) O_j \quad (21)$$

$$D(w_{ij}) = \frac{1}{\varepsilon} \sum_{k=1}^N \alpha_k \text{sgn}(y_k^* - y_k) f'(I) f'(I_j) w_j O_i \quad (22)$$

Where

$$f'(I) = \frac{2I^{-2}(I^{-1}-1)}{(1+(I^{-1}-1)^2)^2} \quad (23)$$

$$f'(I_j) = \frac{2I_j^{-2}(I_j^{-1}-1)}{(1+(I_j^{-1}-1)^2)^2} \quad (24)$$

According to Formula (10), to an individual w , its adaptive value is just the minimum membership in the fuzzy goal set. The aim of the optimization is to make the minimum membership increasing, so the priority should be given the goal with the minimum membership.

To w_j , the offspring $w_j^l(t+1)$ which is produced by the mutation of $w_j^l(t)$ along the weighted gradient direction, where l means the number of individuals and t means the generation number, can be described as follows:

$$w_j^l(t+1) = w_j^l(t) + \beta^l D(w_j^l(t)) \quad (25)$$

Where β^l is the random step length which has declining mean value with Erlang distribution. The Erlang distribution is generated by randomizer.

Similarly, to w_{ij} , there is :

$$w_{ij}^l(t+1) = w_{ij}^l(t) + \beta^l D(w_{ij}^l(t)) \quad (26)$$

From the weight and variation formulas (11) ~ (26), it is seen that the goals unable to be satisfied have the lest membership and can obtain the maximum weight $\frac{1}{\varepsilon}$, so the variation will guide the individual to the feasible region. As $\mu_G(w) > 0$, the goal with the minimum membership obtains the maximum weight. Therefore, the weighted gradient direction will improve the minimum membership value so as to make $\mu_G(w)$ improved. As a result, the weighted gradient direction will guide all individuals to approach the precise optimal resolution, and these individuals constitute one neighborhood including the precise optimal resolution.

In the genetic algorithm of the article, the variation along the weighted gradient direction is the main operator, the arithmetic combination crossover is only used in the later period, and the new population is selected in each generation by means of the proportional selection strategy.

2) Design of Fuzzy Optimization Algorithms

[Algorithm1: The Tans Algorithm]

Input: training samples

Output :neural network weight

According to the basic strategy discribed above, the algorithm is constructed as follows.

(1) Scanning the data base and reading the samples $(X_k, Y_k^*), k = 1, 2, \dots, N$.

(2)Constructing the objective function $y_k = F(X_k, w)$ whiose expected value is $y_k^* = Y_k^*$ and giving the symmetry tolerance ε .

(3)Giving the population scope pop_size and making $t=0$, then initializating the population $w_j^l(0)$, $w_{ij}^l(0), l=1, 2, \dots, pop_size$.

(4)To each individual $w_j^l(t)$ and $w_{ij}^l(t)$,

①Using Eq.'s(2)-(7), computing the value $y_k^l(t)$ of each objective function.

- ②Using Eq.(9), computing the membership $\mu_{G_k(t)}(w^l(t))$ of the fuzzy goal set.
- ③Using Eq.(10), determining the membership $\mu_{G^l(t)}(w^l(t))$ of the fuzzy superior set.
- ④Using Eq.(19), Eq.(20) and Eq.(12), respectively determining the gradient $\nabla\mu_{G_k(t)}(w_j^l(t))$ and $\nabla\mu_{G^l(t)}(w_{ij}^l(t))$, and the weight $\alpha_k(t)$ of each goal, and using Eq.(21) and Eq.(22), determining the weighted gradient direction $D(w_j^l(t))$ and $D(w_{ij}^l(t))$.
- (5)By the scale-selecting algorithm, choosing new parameter population $w_j^l(t+1)$ and $w_{ij}^l(t+1)$.
- (6)According to the arithmetic combination method, executing crossover operation.
- (7)Using Eq.(25) and Eq.(26), executing the mutation operation along the the weighted gradient direction.
- (8)Whether to meet the conditions for termination ?
 - ① If not, making $t=t+1$, then jumping to (4).
 - ② If so, exporting $w_j^l(t)$ and $w_{ij}^l(t)$ which makes $\mu_{G^l(t)}(w^l(t))$ the largest as BP fuzzy neural network weight.

4. Emulation Example

Machine learning systems can solve many engineering problems. This article only introduces the application example of the algorithm in determining flash metal consumption design criteria in a nature-inspired computation system.

4.1 Relationship among the Variables in Flash Metal Consumption Design Criteria

According to the former Soviet Union scholar Jiejielin's research result about the factors affecting flash metal consumption and the analysis of the field data collected from some factory, the relationship among the variables in flash metal consumption design criteria can be described as (Xiao Jingrong, Li Dequn trans, 1983)

$$\begin{cases} Q_{FY}/Q_Y * 100\% = f_1(Q_Y^{0.2}, S_Y, (D_0/D_Y)^2, (B_Y/H_Y), (D_0/D_Y)^2 * S_Y) \\ Q_{FZ}/Q_Z * 100\% = f_2(Q_Z^{0.2}, S, (D_Y/D_Z)^2, (B_Z/H_Z), (D_Y/D_Z)^2 * S) \end{cases} \quad (27)$$

Where Q_{FY} and Q_{FZ} are the flash metal consumption of pre-forging and finish_forging phases, Q_Y and Q_Z are the weight of pre-forging and finish_forging pieces, D_0 , D_Y and D_Z are the maximum diameter of rough, preforging piece and finish_forging piece, and S_Y and S are the shape complex coefficient of pre-forging and finish_forging phases.

4.2 Example

The samples tables responding to all sub-formula of Eq. 27 are respectively written in Table.1 and Table.2.

In the machine learning emulation system, the specific expression forms of all sub-formula of Eq.27 are minied using the Tans Algorithm 1. The neural network operating parameters are set as follows: input layer adopting 5 nodes, hidden layer adopting 15 node and output layer adopting 1 node. The GA parameters are set as follows: *MIND*(the number of individuals)=40, the number of excellent individuals being 5, *MAXGEN*(the maximum genetic number)=100, chromosome using real-code, *GGAP*(generation gap)=0.9, P_c (crossover probability)=0.7, P_m (variation probability)=0.05, and ε (symmetry tolerance)=0.1.

According to the running result, finally get

Weight matrix of Connecting the input layer with the hidden layer: to the first sub-formula of Eq.(27)

$$(w_{ij}^l) = \begin{bmatrix} 0.21 & 0.34 & 0.35 & 0.26 & 0.28 & 0.32 & 0.45 & 0.42 & 0.28 & 0.30 & 0.55 & 0.12 & 0.32 & 0.11 & 0.58 \\ 0.12 & 0.22 & 0.24 & 0.18 & 0.27 & 0.23 & 0.11 & 0.32 & 0.53 & 0.21 & 0.32 & 0.18 & 0.63 & 0.38 & 0.23 \\ 0.14 & 0.78 & 0.25 & 0.55 & 0.34 & 0.21 & 0.39 & 0.75 & 0.42 & 0.45 & 0.19 & 0.24 & 0.37 & 0.23 & 0.41 \\ 0.58 & 0.34 & 0.48 & 0.65 & 0.18 & 0.65 & 0.23 & 0.21 & 0.11 & 0.13 & 0.46 & 0.15 & 0.29 & 0.29 & 0.27 \\ 0.24 & 0.18 & 0.16 & 0.19 & 0.72 & 0.34 & 0.14 & 0.22 & 0.21 & 0.27 & 0.13 & 0.36 & 0.13 & 0.54 & 0.14 \end{bmatrix} \quad (28)$$

To the second sub-formula

$$(w_j) = \begin{bmatrix} 0.32 & 0.24 & 0.34 & 0.27 & 0.25 & 0.31 & 0.43 & 0.41 & 0.68 & 0.33 & 0.56 & 0.32 & 0.34 & 0.15 & 0.18 \\ 0.14 & 0.32 & 0.25 & 0.15 & 0.17 & 0.24 & 0.12 & 0.12 & 0.51 & 0.42 & 0.31 & 0.16 & 0.61 & 0.48 & 0.43 \\ 0.24 & 0.68 & 0.17 & 0.54 & 0.33 & 0.32 & 0.23 & 0.45 & 0.32 & 0.15 & 0.29 & 0.25 & 0.36 & 0.72 & 0.21 \\ 0.56 & 0.54 & 0.38 & 0.65 & 0.17 & 0.45 & 0.43 & 0.51 & 0.18 & 0.33 & 0.36 & 0.17 & 0.28 & 0.24 & 0.57 \\ 0.34 & 0.23 & 0.26 & 0.29 & 0.62 & 0.64 & 0.17 & 0.42 & 0.27 & 0.26 & 0.11 & 0.32 & 0.16 & 0.53 & 0.24 \end{bmatrix} \quad (29)$$

Weight vector connecting the hidden layer with the output layer: to the first sub-formula of Eq.(27)

$$(w_j) = [0.23 \ 0.42 \ 0.53 \ 0.11 \ 0.72 \ 0.25 \ 0.39 \ 0.43 \ 0.64 \ 0.13 \ 0.27 \ 0.13 \ 0.61 \ 0.29 \ 0.41] \quad (30)$$

To the second sub-formula

$$(w_j) = [0.13 \ 0.52 \ 0.23 \ 0.42 \ 0.35 \ 0.65 \ 0.18 \ 0.25 \ 0.72 \ 0.15 \ 0.38 \ 0.23 \ 0.11 \ 0.59 \ 0.38] \quad (31)$$

5. Conclusion

This article discusses the whole framework of nature_inspired computation systems, deeply researches the action mechanism of machine learning in nature_inspired computation systems, and gives a particular scheme on machine learning which is emulated. The results show that machine learning makes nature_inspired computation systems be able to automatically get knowledge, their performance improved, them given more intelligence. Machine learning will have a significant impact on the storage mode, information input mode and architecture of nature_inspired computation systems. In the scheme, because the initial weight of neural networks and the initial groups of genetic algorithm are randomly generated, the results of the algorithm will be more or less unstable. But ,if the algorithm model is reasonably choosed and the algorithm parameters are appropriatly setted , the results will be relatively stable.

References

- Cai Zixing & Xu Guangyou. (2004). *Artificial Intelligence and Application (Version III)*. Beijing: Tsinghua University Press.
- Cheung L C Y, Holden T S I. (1991). Survey of artificial intelligence impacts on information system & engineering. *Information and Software Technology*, 1991, 33(7): 499-508.
- Chien Steve, Decoste Dennis, Doyle Richard, et al. (1997). Making an impact artificial intelligence at the jet propulsion laboratory. *A.I.Magazine*, 1997, 18(1): 103-121.
- Gong, Tao, Cai, Jingfeng, Cai, Zixing. (2003). A Coding and Control Mechanism of Natural Computation [C] // Yen G Gary, L IU Derong. *Proceedings of the 2003 IEEE International Symposium on Houston, Piscataway*: IEEE Society Press, 2003, 727-732.
- Guimarse F G, Campelo F, Saldanla R R. (2006). A hybrid methodology for fuzzy optimization of electromagnetic devices. *IEEE Transaction on Magnetics*, 2006,15(4):15-19.
- Horikawa S. (1992). On Fuzzy modeling using fuzzy neural networks with BP Algorithm. *IEEE Trans on Neural Networks*, 1992, 39(5): 25-29.
- Jimenez F, Sanchez G, Cadenas J M, Gomez-Skameta A F. (2004). A multi_objective evolutionary approach for nonlinear constrained optimization with Fuzzy costs. *Systems, Man And Cybernetics*, 2004, 12(2):23-26.
- Jin Y W, Shen H, Li K Q, et al. (2005). Solution to multi-objective fuzzy optimization dynamic programming with uncertain information. *Parallel and distributed computing-Application and Technologies*, 2005, 32(2):53-57.
- Liu Y K. (2006). Convergent results about the use of fuzzy simulation in fuzzy optimization problems. *Fuzzy Systems*, 2006, 23(1):34-37.
- Mccormark D M, Day R. (1993). How artificial intelligence impacts E&P productivity. *World Oil*, 1993, 214 (10): 6.
- Ng E Y K, Fo K S C, Peh Y C, et al. (2002). Computerized detection of breast cancer with artificial intelligence and thermograms. *J of Medical Engineering and Technology*, 2002, 26 (4): 152-157.
- Shi, Zhongzhi. (2006). *Advanced artificial intelligence*. Beijing: Science Press.
- Tang, Jiafu, Wang, Dingwei. (2000). Fuzzy optimization theory and methods. *Control Theory and Applications*.
- Xiao, Jingrong, Li, Dequn trans. (1983). *Optimization and automation principles of hot volume die-forging process design*. Beijing: National Defense Industry Press.
- Yi, Jikai, Hou, Yuanbin. (2003). *Intelligent control technology*. Beijing: Beijing Technology University Press.

Table 1. Q_{FY}/Q_Y training samples of fuzzy optimization BP neural networks

$Q_Y^{-0.2}$	S_Y	$(D_0/D_Y)^2$	B_Y/H_Y	$S_Y * (D_0/D_Y)^2$	Q_{FY}/Q_Y
0.75390	1.62942	0.88054	2.93900	1.43477	0.17662
0.80757	2.69403	1.01835	3.00000	2.73980	0.13091
0.78038	3.36251	0.85031	4.26471	2.85919	0.15356
...					
0.79921	3.10378	0.78632	3.00000	2.44056	0.15168

Table 2. Q_{FZ}/Q_Z training samples of fuzzy optimization BP neural networks

$Q_Z^{-0.2}$	S	$(D_0/D_Z)^2$	B_Z/H_Z	$S * (D_0/D_Z)^2$	Q_{FZ}/Q_Z
0.76829	1.01682	0.99431	3.78000	1.01103	0.09915
0.93519	0.86370	0.79549	4.26670	0.68706	0.09338
1.00778	0.99236	0.95413	4.00000	0.94684	0.15615
...					
0.67096	1.16473	0.99598	3.00000	1.16005	0.16811

Table 3. Test results of flash metal consumption based on fuzzy optimization

NO.	Sample No	Predictive value Q_{FY}/Q_Y	Measured value Q_{FY}/Q_Y	Relative error (%)	Average Relative Error (%)	Computation time (s)	Genetic algorithm iteration
1	1	0.19186	0.17662	8.626	3.542	158	95
	2	0.13315	0.13091	1.712			
	3	0.15400	0.15356	0.287			
2	1	0.18442	0.17662	4.420	1.732	142	92
	2	0.13105	0.13091	0.105			
	3	0.14326	0.15356	0.671			
...							
5	1	0.16326	0.17662	7.562	4.484	161	88
	2	0.13375	0.13091	2.173			
	3	0.15927	0.15356	3.718			

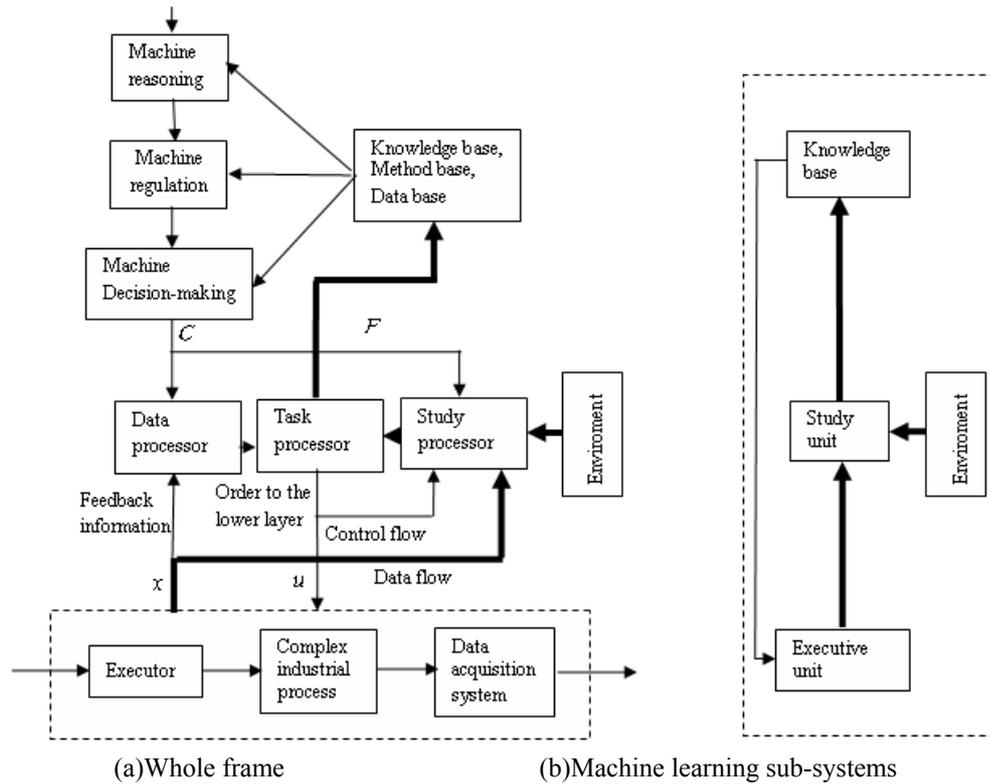


Figure 1. Nature_inspired computation systems

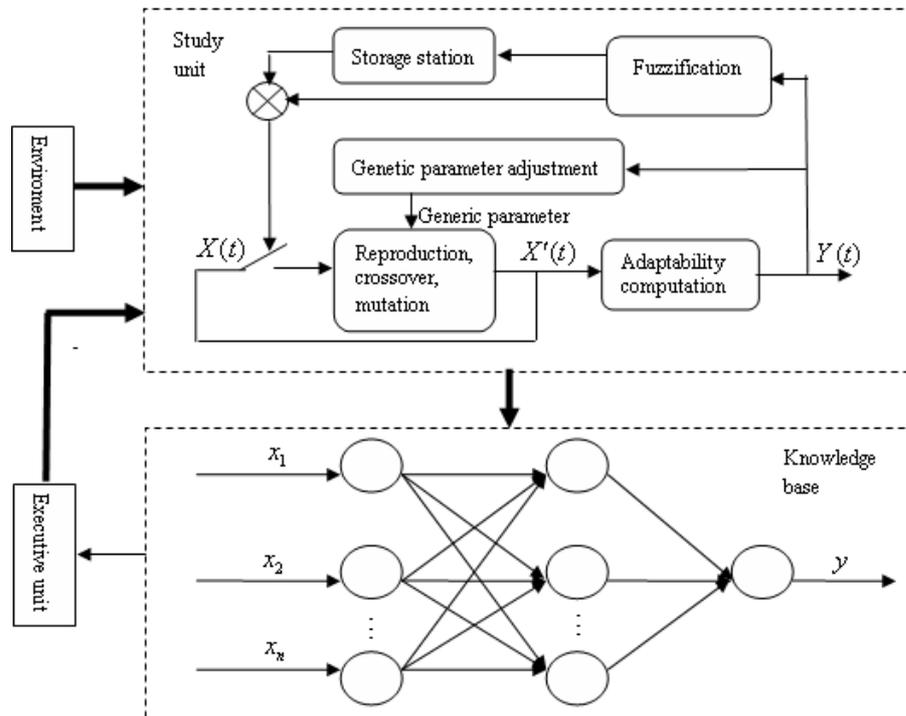


Figure 2. A particular scheme on machine learning in nature-inspired computation systems