Computational intelligence: From mathematical point of view

QIAN Minping[1] and GONG Guanglu[2]

1. *Department of Probability and Statistics, Peking University, Beijing* 100871, *China*;
2. *Department of Applied Mathematics, Tsinghua University, Beijing* 100084, *China*

Abstract A simple but illustrative survey is given on various approaches of computational intelligence with their features, applications and the mathematical tools involved, among which the simulated annealing, neural networks, genetic and evolutionary programming, self-organizing learning and adapting algorithms, hidden Markov models are recommended intensively. The common mathematical features of various computational intelligence algorithms are exploited. Finally, two common principles of concessive strategies implicated in many computational intelligence algorithms are discussed.

Keywords: computational intelligence, simulated annealing, neural network, genetic algorithm, evolutionary programming, self-organizing learning, self-adapting algorithm, hidden Markov model, concession strategy.

WHAT is computational intelligence? In fact, there is no exact definition. People may have their own understanding. Along with the rapid development of large scale high quality computers so far, a number of new computational approaches, e.g. neural network, simulated annealing, genetic algorithm, evolutionary programming, hidden Markov model (HMM), adaptive algorithm, etc. have been proposed, which have been shown very powerful in dealing with the combinatorial explosion appearing in the large scale complex systems. These approaches usually not only share the advantage of generality, robustness, simplicity and convenience for parallel processing, but also hopefully become the bridge among the numerical calculation and the advanced intelligence behavior, such as semantic expression and thinking in images, etc. Therefore, "they are naturally considered to be the major significant influential key skills for the computational technique in the next decade"[1].

Recently, the above-mentioned approaches have attracted more and more attention owing to their successful application in various fields. From the mathematical point of view, the different intelligent computational approaches enjoy the following common features:

(i) In most of these methods, random factor is introduced; therefore, they are indeterminate, and sometimes they even support contradict solving ways. Lots of computational processing is in fact doing simulation of the stochastic processes on computers.

(ii) Most of them are the dynamical systems or random dynamical systems with adaptive mechanism, and sometimes the structures of them are even adjusted successively during the calculation.

(iii) These algorithms are designed for general objects. In contrast to those designed for special problems, heuristic methods for concrete problems with special treatments are not considered.

(iv) Many algorithms of them seem stupid in calculation for the low dimension case or in the simple cases, since in low dimension or in simple cases, extremely direct algorithms can be found through heuristic observation. While for the high dimension and complicated cases, these simple heuristic methods often become practically inexcusable, since the time spent in computing grows up exponentially with the increase of the scale.

The above-mentioned computational approaches prevailed mostly in the end of the 1980s or at the beginning of the 1990s. For instance, the artificial neural network, genetic algorithm and evolutionary algorithm, etc. are initiated by analogue and simplification of the process of intelligent activities of the brain of the human being and the competition of surviving, inheritance and mutation of living bodies, where numerical computation is more emphasized, and objects are described by data and distributions instead of by axioms and formulas. At that time, scholars of these groups are often excluded by those of the classical artificial intelligence based on logic and symbolic operations. So the nomenclature of computational intelligence has the flavor resistence. In fact, to understand the human intelligence and to think it purely as the logical thinking or mainly as the logical thinking have its one-sidedness. Actually, the

thinking in images and the non-logical inference are vital for the human intelligence, even say, the more important side. For example, in the pure logical inference, the statement "A implies B" simply means that the appearance of A includes that of B. However, the intelligence inference of the human being often goes in a converse way, e.g. during a medical treatment, the diagnosis of a doctor always goes from the statement "the disease A may have the symptom B" to infer "how big of the possibility of this patient having this disease". This is a typical non-logical inference, where the indeterminate factors are involved and so every decision there has to take the risk of false inference. Nevertheless, this kind of inference is the most possible cases appearing in human intelligent activities. Seeing from the development in the recent decades, the authors agree with Li[1, 2] that it seems more reasonable to consider the computational intelligence as a part of the artificial intelligence and to make our effort to combine the logic and the symbolic operations with the numerical calculation and to make the complement of them each other.

In recent years, in various application fields, one has made satisfactory achievement in trying to use various computational intelligent approaches to solve problems of the large scale complicate systems, e.g. at the beginning of the 1980s, Sejnowski and Rosenberg constructed the system of 'NetTalk'[3], which is exactly a small machine based on the artificial neural network. The amazing performance of this machine shows that it possesses the ability of learning and correctly reading English texts (even for the texts it has never learnt before) based on its accumulated experience from learning previously. This work started a hot surge of the artificial neural network. In the 1960s, Fogel *et al.*[4] proposed the Evolutionary Programming, and in the 1970s, Holland[5] initiated the genetic algorithm. These two algorithms are in fact very similar. They are to simulate the optimal process of survival for the fitness in the competition of biological world. Till the 1980s they were applied successfully to the fields of econometric prediction and others and became hot topics. Simulated annealing algorithm was proposed by Kirkpatrick *et al.* by borrowing the idea of physical annealing——increasing the temperature then cooling[6] as the means of escaping from the traps of local extreme in the global optimal problems. This algorithm has been shown effective for the multi-peak object functions, especially for those in the complicate high dimensional cases. It has become an optimal algorithm of broad use and study in recent years. HMM started to prevail in the speech signal analysis (in fact, it is the most effective algorithm for the speech recognition till now). Due to its flexibility in specifying objects of the HMM, like the forward multi-layer artificial neural network, it has been proven to be a very nice nonlinear statistical parameter model. In these years, much attention has been paid to the area of pattern recognition, communication systems and mathematical statistics, etc. Self-organizing learning (clustering) approach is first suggested by Kohonen as a neural network. Actually, the basic idea of self-organizing learning is quite different from that of artificial multi-layer neural networks, but it is rather similar to the stochastic approximation and the adopted algorithm discussed by Kushner, Metivier, etc. Since the results of learning by this approach are very good, this method also attracts much attention to its application and theoretical studies.

Although the computational intelligence approach has been widely applied to the real world problems with great complex, like the optimal selection (e.g. optimal searching problems of medicine molecular design, second and third level structure analysis of protein, etc.), symbolic regression, programming generating, game strategy discovering, etc. and it works quite well, from the theoretical point of view, except the simulated annealing, fairly complete discussion lacks. There are lots of interesting and deep theoretical problems for mathematicians.

Especially, the relation between various computational intelligence and parallel processing is an important theoretical and practical topic, which needs to be studied sophisticatedly. Some scholars studying traditional artificial intelligence, e.g. Simon, Feigenbaum, think that the parallel processing is not an important topic in artificial intelligence. They emphasize that thinking is essentially cascading. It is pointed out in ref. [1] that: Li Guojie "through deeply studying the interaction and connection between the parallel processing and the elicitation method of searching we have found the higher the quality of the function of the elicitation method the more of the parallel property. And so, there is no proper position for the parallel processing in the field of the artificial intelligence focusing on the symbolic logic inference".

Further, he said "but for the evolutionary calculating it is quite different. The evolutionary algorithm rather suits for large scale parallel processing. While the neural network itself is exactly in large scale parallel. It is not necessary to emphasize too much that the most efficient parallel way for the calculating of the genetic algorithm and the genetic programming is to take hundreds, or even thousands, of computers working independently for the genetic calculating individually or with disjoint sets of individuals. It goes such a way that after computing (without any communication in the processing), communicating and comparing at the end of the operation, it chooses the best result." "For the application to real world problems of intelligent computing, e.g. the real time continuous speech recognition, machine translating, general natural language understanding, global complicate computing auxiliary decision and so on, it can only be brought out on the large scale parallel-computers. In the practical application, it is necessary to solve the scalability of the artificial intelligence. It is a great challenge for the artificial intelligence scientists in nineties.", "We need to pay great attention to the study of the computational intelligence and the parallel calculating and surge the study of the artificial intelligence and the computational intelligence to a new high level." In the study of parallel processing, an important phenomenon, analogue to the critical phenomenon appearing in the interacting particle systems, is worthy paying attention, i.e. parallel operation can raise the calculating speed, however, too large scale parallel operation may even decrease the calculating speed. There is an optimal critical parallel scale[7].

The aim of the present paper is to give a simple survey on various approaches of computational intelligence with their features and applications, and, within the knowledge of the authors, to give a concise discussion of related known mathematical results and problems, in order to make the interested people coming up with valuable opinions and attract the attention of mathematicians to join the study of the computational intelligence, also in order to offer a few typical references for those who expect to know this new field. The listed references in this paper neither are complete, nor reflect their academic historical position, and are only to help those who want to know more about computational intelligence. Obviously, there are a number of excellent papers and books not listed.

This review is organized as follows. In sections 1—5, simulated annealing, neural network, genetic and evolutionary algorithm, self-organizing learning , adapting algorithm and hidden Markov model are discussed introductorily. In section 6, two common implicit principles——concession strategies, in various computational intelligence algorithms are illustrated from the mathematical point of view.

1 Simulated annealing

Simulated annealing is an approach of global optimization. Roughly speaking, the artificial noise is introduced to make the possibility of escaping from the trap of local optimum. After letting the noise decreasing to zero, the algorithm eventually converges to the set of the sites achieving the global optimum. In fact, early in 1965, Khas'minskii proposed this idea[8], but unfortunately it did not attract sufficient attention in computer science and optimal application fields. Till 1983, Kirkpatrick suggested a simulated annealing algorithm[6], which attracted great attention from the optimal application field and once became a hot point. In the following, we first give a simple example to explain the basic idea of the simulated annealing[9], which is a well written book on the simulated annealing and its general application.

Let the object function of the optimization problem be $f(i)$, with the argument taking discrete values. Denote

$$f_0 = \min_i f(i).$$

To obtain the global minimum of $f(\ )$, an homogeneous irreducible non-periodic positive recurrent Markov chain with the invariant measure

$$\mu_\beta(i) = \exp\{-\beta \cdot f(i)\} \cdot Z_\beta^{-1} \qquad \left(Z_\beta = \sum_i \exp\{-\beta \cdot f(i)\}\right) \tag{1.1}$$

is constructed. The distribution in (1.1) is very similar to the Boltzman distribution and the parameter

$$\beta = 1/T$$

is the inverse temperature, where T stands for the absolute temperature. If we consider the deterministic

algorithm without noise as the case of $T=0$, then the case of $T>0$ means that searching is done under positive temperature. Noticing that as $\beta\to+\infty$, ($T\to 0$) in (1.1), one sees that the invariant measure tends to a measure concentrated on the set of global minimum points of $f(\ )$. The real computing procedure is: first taking a positive temperature $T>0$ to run the Markov chain described above till it "reaches" the stationary; then starting at the present state of this Markov chain to run again the Markov chain at a lower temperature. Repeating this procedure with temperature decreasing successively, we will eventually have the Markov chain tending to the global minimum points of $f(\ )$ in probability. This procedure of temperature increasing to zero is just like the physical processing of cooling schedule called annealing and therefore is called the simulated annealing.

To accelerate the calculation, we introduce the inhomogeneous Markov chains. This means to change the transition matrix of the Markov chain at each step T. It can be proven that as the time goes to infinite, and T goes to zero at an appropriate slow speed, this inhomogeneous Markov chain tends to the set of the global minimum set of $f(\ )$ in probability. More precisely, there exists a positive constant a determined by $f(\ )$, and if we take

$$\frac{1}{T}=\beta_n\approx\gamma\cdot\log n\qquad(\gamma<a),\tag{1.2}$$

as $n\to\infty$, then T meets the demand of "going to zero at an appropriate slow speed". Of course, the closer γ to a the better.

In many cases, the underlying system may be very complicated with extremely high dimension and the object function possesses no explicit expression. Sometimes, although there is an explicit expression, it does not work practically, since the calculation is too complex. For this case, the locally changed neighborhood and the corresponding matrix are then introduced (the neighborhood of i is denoted by $R(i)$):

$$G=(g_{i,j}),\ g_{i,j}=\begin{cases}|R(i)|^{-1}, & j\in R(i);\\ 0, & j\notin R(i).\end{cases}$$

It determines that if the system is at the state i, then it can only change into a state in $R(i)$, and the n-step transition matrix is taken by

$$P(\beta_n)=(p_{i,j}(\beta_n)),$$

$$p_{i,j}(\beta)=\begin{cases}g_{i,j}\cdot\min\left\{1,\dfrac{\mu_\beta(j)}{\mu_\beta(i)}\right\}, & \forall j\neq i;\\ 1-\sum_{k\neq i}g_{ik}\cdot\min\left\{1,\dfrac{\mu_\beta(k)}{\mu_\beta(i)}\right\}, & j=i.\end{cases}$$

Thus, as temperature T decreases in an appropriate way, when we operate the inhomogeneous Markov chain defined by this family of transition matrices, it will converge in the sense of probability to the global minimal set of $f(\ )$.

A number of scientists have done a series of studies on the mathematical theory of the simulated annealing, among which the Taipei school has made outstanding contribution. How to control the noise level to make the algorithm converging to the global optimum, what is the upper bound of the above-mentioned constant a (i.e. the superior of a satisfying (1.2) and hence ensuring the success of the algorithm) and the improvement of the simulated annealing, etc. were studied systematically and rigorously, where the main mathematical tools involved are the large deviation theory of Markov processes, the spectral gap and the logarithm Sobolev inequality of the generator of Markov processes, etc.[10—15].

Actually, for a deterministic or stochastic algorithm possibly not reaching the global optimum by trapping at the local optimum, we can always introduce an artificial noise to do the simulated annealing for making the chance of escaping the local traps and to achieve the global optimum. In ref. [15] an algorithmic model is given, how to control the noise level in order to achieve the global optimum is pointed out, and the rigorous mathematical proof is shown.

Although theoretically the convergence in probability of the usual simulated annealing is proven, the expand time for the convergence is increasing exponentially with the reciprocal of the constant a. It

means that if a is rather small, then even $\beta = \gamma \log n$ ($\gamma < a$) is satisfied, and the algorithm will still not meet the demand of convergence in a practical allowable time period. And so people try constantly to improve the algorithm of the simulated annealing. Szu and Hartley proposed some algorithms to accelerate the convergence[16, 17]. However, some of the ideas may not be successful, for example, the Cauchy noise is taken instead of the Brownian motion (white noise) to accelerate the convergence speed of the simulated annealing (the idea is that the Cauchy process has its jumps, it may give better chance of escaping the local traps). It can be proven that for rather general cases, the Cauchy annealing does not converge[18]. This shows that the rigorous mathematical study is necessary. For complicated problems, in which the small probability event may happen again and again, without the basis of the rigorous theory, the heuristic may make mistakes. To adopt the color noises as the driver to escape the local extreme is also an approach of accelerating the convergence speed. Among them the TINA (Time Invariant Noise Annealing) approach is worthy to be paid more special attention. The basic idea is to take the noise bigger at the sites with bigger object function values, while to take the noise smaller as the object function gets smaller. Especially, in the case of known minimum (e.g. in the case of solving simultaneous nonlinear equations, the object function is taken to be the sum of the square of functions appearing in each equation and then the minimum of the object function is zero, which makes the zero points of the objective function the solutions of the equations), the variance of the noise may be taken as the square of the difference between the function and its minimum (usually, for bounding the variance, a threshold is taken for limiting the difference)[19—21].

For the optimization in continuous state space instead of the Markov chain above, we take the following stochastic difference equation:

$$x(t) = x(t-1) + (gradf)(x(t-1))) + (f(x(t-1)) - f_0) \cdot w(t), \qquad (1.3)$$

where $w(t)$ is an i.i.d. sequence of random variables (discrete white noise), f_0 is the minimum of $f(x)$. Thus, (1.3) gives the iterative formula for TINA with colored noise. In the case of minimal (maximal) value of object function unknown, the noise term can be changed into white noise multiplied by

$$g(f(x(t-1)) - c)^+) + (\gamma \cdot \log(t + t_0))^{-1},$$

where g is a bounded increasing function, c is a constant, the closer to the minimum of f the better. In ref. [21] the proof is given for the convergence of the continuous version of this algorithm. In the practical calculation, one can adjust the parameter c into smaller one by the increasing knowledge of f during the procedure of calculation to accelerate computing.

Besides, there are a number of theoretical problems, e.g. how to judge whether the computation is close enough to convergence, how to accelerate the computing further, and how to use the experience obtained from the previous computing to accelerate the calculation, etc. These problems may relate to the mathematical theory of large deviation theory of inhomogeneous Markov chains and Markov processes, their limiting behavior, the stochastic approximations, etc.

2 Neural networks

There are two kinds of neural networks: biological neural networks and artificial neural networks. For the former, the aim of this study is to understand the mechanism and working principle of brains and other neural systems: giving the mathematical model, and explaining why they have these functions and what are the key parameters, ··· especially to know how the advanced intelligent activities are going. While the artificial neural networks are basically analogizing biological neural networks, or following some of the principles of biological neural networks to construct software and hardware with high efficiency and functions, for instance, with some of the intelligence of human beings. The NETTALK mentioned in the introduction is a famous example for this. The research of artificial neutral networks focuses on the functions of a network rather than finds if there is some real biological systems having such a neural network. The most popular kind of neural networks is the multi-layer forward neural network. This network consists of multi-layer of neurons. There are 3 kinds of layers: input layers which consists of neurons input data

not accepting the interactions from the other neurons, hidden layers for neurons processing data, and output layers being able to be read from the outside network. The hidden layer may split into several layers. In spite of the fact that they are equivalent to one layer by the theory ok Kolmogorov[22], it is more convenient to adopt multi-layer of hidden neurons in practice. In the feed-forward network, the information can only be transferred in the order of layers. Each hidden layer or output layer consists of many perceptrons, a neuron accepting the interaction and the input from the other neurons of the network. In the feed-forward neural network the interactions are only allowed from the neurons of the preceding layer. The usually adopted model of the interaction is the following threshold controlled linear action:

$$h(x) = g(\sum_{y} V(x, y)\eta(y) - \theta(x)), \tag{2.1}$$

where $V(x, y)$, $\eta(y)$ and $\theta(x)$ are interaction potential from the neuron y to the neuron x, the state of the neuron y and the active threshold of the neuron x, where the neurons considered only have active and inhibit states; $V(x, y)h(y)$ is the force of the neuron y acting on x, and the argument of $g(\ )$ in the right hand side of (2.1) is the part of the interaction to x exceeding the threshold. The sum in the right hand side of (2.1) is taken over all y which has interaction to x, and the function g is called the threshold function, which usually is the Heaviside function (step function jumping at zero) or its smooth modification. The meaning of taking this is: in the deterministic case ($T = 0$), when the total interaction to x exceeds the threshold, x becomes active at the next sampling time; otherwise, it behaves inhibit. While in the stochastic neural network systems (equivalent to $T > 0$), at the next sampling time, the probability p of x taking the active state depends on the magnitude of the total interaction exceeding the threshold, i.e.

$$p = \exp(\beta \cdot h(x))/Z \quad Z = \sum_{x} \exp(\beta \cdot h(x)).$$

It is easy to see that the bigger $h(x)$ is, the bigger p is. The operating of the neural network in the stochastic case is a Markov chain similar to that in section 1. As for the case of neural elements taking more than two states, the preceding probability can be defined similarly.

The application of the multi-layer feed-forward network is divided into two phases: learning phase and operating phase. The learning phase aims to obtain parameters $\{V(x, y)\}$ and $\{\theta(x)\}$ through processing the sample such that under these parameters, the neural network can relatively optimally execute what we want. While the operating phase is that the state of neurons of the neural network moves according to the above rule to complete the desired task.

There are two kinds of learning of the neural network: supervised and unsupervised. For multi-layer feed-forward networks, the supervised learning (there is a teacher) is adopted more frequently, where the error back-propagation approach is a popular algorithm. In fact, the supervised learning usually means that a set of output samples with given input samples can be obtained and in the learning the given output is used to determine the quality of the network parameters. The goal of learning is to find the unknown parameters to make the output of the network of these parameters as close as possible to the output sample when we input the corresponding input sample. Therefore, it is reduced to a global optimal problem.

The main steps of the approach of the error back-propagation are: (ⅰ) set the initial values of the parameters; (ⅱ) by random order or one by one input the samples into the network with the initial parameters to obtain the output and compare it with the corresponding output sample; (ⅲ) adjust the network parameters of the previous layer (e.g. the (n-1)-th layer) to obtain the new parameters interacting to the n-th layer to make the output most close to the output sample and then find the output of the (n-1)-th layer to minimize the error of the n-th layer; (ⅳ) repeat (ⅲ) for the (n-1)-th layer instead of the n-layer and to adjust the parameters of the (n-1) layer, then find the output of the (n-2)-th layer to minimize the error of the (n-1)-th layer; (ⅴ) repeat (ⅳ) recursively to each previous layer, till the new parameters of all layers are found; (ⅵ) repeat steps (ⅱ) to (ⅴ), till the convergence is achieved.

The drawback of this algorithm is quite similar to that of the EM algorithm with missing data. In the process of optimization, since the object functions are nonlinear with the huge number of arguments, it often makes the computing falling into the traps of local extrema. To overcome this, the optimization tech-

niques, such as the simulated annealing, etc. are often applied.

Another kind of neural networks is the recurrent network, which is actually a dynamic system with the configurations of states of all neurons as a whole one thing, and the system moves by a deterministic or stochastic rule. The former is a dynamic system and the latter is a Markov chain. The Hopfield network[23] is a typical recurrent network, of which each neuron takes two values: active or inhibit, and changes its states according to the interaction and the rule given by (3.1) deterministically or stochastically by the above-mentioned probability p. The recurrent neural network appears more close to the biological neural network. Especially, the associative memory (or the content-addressable memory) introduced by Hopfield has completely different mechanisms from the traditional memory model (e.g. notepad, tape, disk, laser disk etc.). Retrieving of memory in the Hopfield network does not need any content lists or directories on the stored memory. Thus, it is the network most close to the function of the memory of brains so far. Hence, it provides valuable consultation for the computational intelligence from the angle of bionics. Hopfield network is also applied to the pattern recognition and many NP-hard problems, e.g. TSP problem, maximal connected set of graphs, etc.[24—26].

Boltzmann machine may be thought as a combination and an extension of the feed-forward network and the Hopfield network——it neither confines to "no interaction to the input layer is allowed", nor restricts to "the interaction is only permitted forwardly". On the other hand, this network has input and output neurons, and it is also a recurrent network and possesses the advantage of both of these two kinds of networks. So it has strong capacity and may describe and simulate very complicated systems. It was applied by Azencott *et al.* to the vessel recognition of radar signals, and worked quite satisfactorily[27]. However, the computing scale of the Boltzmann machine is extremely large so that it seems very difficult and complicated for learning. So far, the EM algorithm is widely used for learning and has shown its effect, of which the idea is fairly similar to the hidden Markov model in the following section 6. Because the mechanism of Boltzmann machine is rather complicated and the learning approach needs fairly strong theoretic background to understand. To adopt Boltzmann machine to solve the real world problems is not as popular as the feed-forward multi-layer networks. But the former has far more potential than the latter. The study is far from completeness and a big room is still left for mathematicians.

As a popular model for information processing and computer science, the neural networks have been observed tremendously. A number of excellent writings have appeared, which are easy to read owing to clear illustration[26, 28]. Ref. [29] is a well-written book, where the spin-glasses and the attractors are taken to be the model to describe the associate memory, and the working model of brains is discussed. The clearness of the physical idea of this book makes it a quite inspiring tutorial material for students even for professionals to understand the biological neural networks.

Besides, the self-organizing networks (Kohonen network, ART, etc.) are also the networks with strong functioning and have attracted wide attention. Its mathematical structure is similar to the adaptive algorithm. We will discuss it in section 5.

Beginning at the end of the 1980s and becoming hot in the 1990s, the study of neural networks gradually comes into a highly developing stage. People are trying to know the function and mechanism of neural networks and what is the key point in it. The veil on the neural network has been gradually raised. The fact is that, from the point of view of statistics, multi-layer feed-forward network is nothing but a parameter model of large capacity and computing convenience (only linear combination and thresholds are involved). It may be proven that with enough number of hidden elements the feed-forward network can well approximate any non-linear transformation. The learning based on the samples is exactly a parameter estimation problem of the hidden variables. Therefore, the learning algorithm of the multi-layer feed-forward network is essentially similar to the hidden variable estimation method. Moreover, the learning of the neural network with supervising is also a parameter estimation problem with hidden variables——statistics for incomplete data. Thus, EM algorithm, Markov chain Monte Carlo (MCMC), simulated annealing approaches also play their roles. These are the hot points deserving studying now.

Another important problem in neural network is how to determine the size, that is, to determine the

number of layers and the number of neurons in each layer. There is a dilemma here: too small size of the system will cause large systematical bias, while too large size will appear over-fitting and the stochastic fluctuation will be considered as the trend of the system. In ref. [30], Geman illustrated the problem both from theoretical analysis and from real calculating examples, and pointed out that it is harmful to enlarge the size blindly without paying enough attention to the right choice of the size of the networks. It seems that the only correct way to solve this dilemma is to do statistical analysis (exploratory statistical analysis) for the samples carefully to get the global understanding, and to try our best to make use of the *a priori* knowledge of the system. The self-organizing learning approach discussed in the later section may help the exploratory statistical analysis. Nevertheless, this dilemma is far from being solved and worthy to be studied.

The computing method of the neural network is a parallel one. While for the parallel scale, it is also not the larger the better. In ref. [7], the authors use examples to illustrate that there exists a critical bound of the parallel scale; if this critical scale is exceeded, the parallel computing may not achieve the effect of the corresponding cascade computing. So theoretical studies on the scale and the structure of the neural network are really vital.

To study the mechanism of biological neural networks is very helpful to the artificial networks. It not only provides the copy of bionics, but also reveals the key factors, hard points, main parameters and the guidelines for solving these problems. For example, the study of the Linsker network, the key problem of the graph information processing in the brains of human beings, the approximation of mathematical expressions and the main parameters, etc, have greatly inspired the design of the artificial neural network[31, 32]. More examples given in ref. [33] considered the recurrent networks like Hopfield's, and showed how they move among the attractors, what is the role of noise in controlling this motion and how the noise helps uncover the relation strength between attractors. All these provide important information for the simulation of the bionic neural network and application of the artificial neural network.

3 Evolutionary programming and genetic algorithm

Evolutionary programming and genetic algorithm are proposed by borrowing the evolution process of the nature. Notice that through long time natural selection, amazing optimal level can be achieved. People naturally think to mimic this process to do the optimization. Holland first proposed genetic algorithms[5]. Various practical examples have shown that there is an outstanding advantage in evolutionary programming and genetic algorithm——good robustness.

Genetic algorithm is a group optimizing process. In order to achieve the maximal (or minimal) values of the object function, we start at a group of initial values instead of one initial value to do optimization. This group of initial values is the mimic of a group of living bodies, and the optimizing process is just like the process of reproduction, competition, inheritance and mutation. The main steps of the genetic algorithm are:

(ⅰ) Set the initial values.

(ⅱ) Competition: in this step, several individuals from the initial group (the group of initial values) are selected to reproduce the next generation. For instance, to reflect eugenic principle, one may determine the individual selecting probability according to the values (or its reciprocal) of the object function.

(ⅲ) Reproduction: consists of evolution, hybridization and mutation. It may simply change the selected individuals by the optimization steps as we discussed before, or take additional mutation or hybridization algorithm to selected individuals. For the optimization problems of discrete variables (usually they are in an ultra-high dimension space), a commonly used hybridization approach is to cut selected individuals (vectors) into two or several same dimensional parts respectively and interchange them so that we can obtain new individuals, called offerings. In practice, for different practical problems, we may choose other reproduction ways (i.e. competition, inheritance and hybridization) according to our understanding about the problems.

(ⅳ) Use offerings instead of parents (two individuals of producing the offerings), repeat (ⅱ) and

(iii) to produce the next generation offerings successively till the maximal (or the minimal) value of the object function among the whole group cannot be improved any more.

As a matter of fact, the above method does not guarantee the achievement of the global optimization. It may still lead to local traps. However, since the object function with multi-variables is introduced in the genetic algorithm, which is actually the maximum (or the minimum) value of the original object function of single argument at the components of the multi-variable, there is relatively small chance of being trapped in the local extreme values (in fact, the set of points with minimal value of the new object function is enlarged). And so the effect of optimization may be improved. Moreover, people suggested to introduce competition between the offerings and their parents , which means not simply using offerings instead of their parents but choosing the winner of the competition between them as the "parents" for the next generation. In the genetic algorithm, mutation may be introduced such that the optimization program may get away from the local optimum with a small probability. Locally, this evolution seems to make the object function inferior, but it is this evolution that provides the possibility of escaping from local traps. Moreover, one can control the probability of such mutations to be small, such that after many steps of computing, it is eventually close to the optimum with a probability big enough. Thus, one can see that the basic principle of this method is quite similar to the simulated annealing algorithm. The evolutionary programming proposed by Fogol is similar in principle to the genetic algorithm, while the genetic algorithm evolves with structure variables and it is a group optimization suitable for the parallel calculating. Moreover, from the view point of mathematical structure of algorithms, cellular automata and lattice gas automata, which are very popular in physical and mechanical communities are similar to various evolutionary algorithms, and the appearing problems, the interesting special phenomena , the convergence behavior are also very similar. Although there have been some researches of convergence on the evolutionary programming and the genetic algorithm, there are a number of open problems for study, for example, what is the advantage brought by the group optimization, what is the principle of selecting the scale of the group and the method of the hybridization and the mutation, how to characterize the merit and the efficiency of these selections, and how to estimate the convergence probability and the convergence speed, and so on. In the interacting particle systems, lots of behavior related to the stochastic processes similar to the above systems have been obtained, which can be used for references. On the other hand, these problems have many similar points to the simulated annealing approach. The evolutionary programming and the genetic algorithm are Markov chains or Markov processes, and the mathematical tools, such as the large deviation theory of Markov processes, the spectrum gap, the logarithm Sobolev inequality, etc. are hopefully very helpful.

4 Self-organizing neural network and stochastic approximation

The self-organizing neural networks are essentially a kind of learning algorithm, which is a method to determine the model completely by the samples given. Just as mentioned above, if the scale of a network is fixed, then the supervised learning of the neural network is nothing but a problem of parameter estimation with complete or incomplete data. However, to determine the scale is not only extremely important, but also a problem of greater difficulty. As for the dilemma mentioned above, powerful tools still lack. The self-organizing network provides a tool for understanding and identifying the statistical distribution of the data more or less, therefore providing us reference for selecting the scale of the network. In the late 1980s, Kohonen and Grossberg introduced the self-organizing learning networks or adaptive learning networks with competition mechanisms. These networks are used to find a group of representative points from a large size of samples (these representative points are not necessary in the set of samples). Each point represents a part of samples located in its neighborhood. Hence when the gross of the representative points is much smaller than the sample size, this group of representative points determined from the network may be taken as the reference points for clustering or the pattern recognition and also may be used for the vector quantization, feature extraction and exploratory statistical analysis of the population represented by samples.

The self-organizing topological mapping of Kohonen (KTM)[35] is considered as the networks with

very strong functions, but it usually spends very long computing time. And the problem of determining the scale of itself (the number of representative points) can be transferred to an algorithm of determining the scale of network automatically under certain conditions to provide a rather good tool for the "vector quantization" in computer science. The advantage of KTM lies in its topological property, meaning that the group of representative points obtained from the network keeps the original topological property of the sample points, i.e. the "closer" between the sample points the "closer" between their representative points.

The calculating steps of KTM are:

(i) Choose randomly or deterministically a group of initial values, the size n of which depends on the gross of the representative points: $\{X(1,0), X(2,0), \cdots, X(n,0)\}$ (usually, each of $X(i,0)$ is a vector with the same dimension as each sample).

(ii) Randomly (i.i.d.) or by a deterministic principle read samples to get $Y(t)$.

(iii) Update the representative point as follows:

$$X(i, t+1) = X(i,t) + a(t)\, r(d(X(i,t), X(j,t)))\,(Y(t) - X(i,t)),$$

where $r(\ )$ is a strictly monotonic positive function, $a(\ )$ is a convergence factor, $d(X, Y)$ stands for the distance between X and Y; and

$$d(X(j,t), Y(t)) \leqslant d(X(i,t), Y(t)) \ (\forall\, i = 1, 2, \cdots n).$$

(iv) Repeat steps (ii) and (iii) till converging, and denote the final result by $\{X(i)\}$.

Here the strict monotone of $r(\ )$ ensures the result nearly possessing the "topological property", i.e. if i and j are relatively close, so do $X(i)$ and $X(j)$. Kohonen pointed out that if

$$\sum_t a(t) = +\infty, \quad \sum_t a^2(t) < +\infty,$$

then the algorithm converges. For the one-dimensional sample and the representative points arranged one-dimensionally, it is proven in mathematical rigorous[36] that the algorithm converges and the topological property holds for KTM, but not for the case of high dimensional sample points or representative points not arranged in one-dimension order. In the later case, in fact, the "topological property" cannot be guaranteed strictly. And in this case, the nearly "topological property" of the Kohonen networks is only illustrated by graphs without transparent exploitation and definition.

If the topological property of the representative points is abandoned, we may set the function $r(\ )$ to be zero except at the origin. This principle means that "the winner gets all" and only one representative point is updated each time. When the gross of the representative points is large, this method is far more rapid than the standard KTM algorithm. For example, if the size of the representative points is 300, then the algorithm of updating one representative point each time according to the principle that "the winner gets all" is 300 times faster than the standard KTM algorithm. Furthermore, in order to have the topological property, we have to adjust the number of all representative points. And then the time spent will be far longer than that of only choosing the representative points. It is proven in ref. [37] that the configuration (the set of representative points of vectors) in the algorithm that "the winner gets all" converges to an approximate distribution of the sample points (usually not having the topological property); however, the author did not give any estimate of the time of convergence there. While from the application point of view, this estimate is even more important than the convergence.

In the self-organizing algorithm of Kohonen, the size of the representative points have to be determined *a priori*, but when a group of data with large size or of very high dimension is given, one cannot know how many points are appropriate. On the contrary, if a group of appropriate representative points is found, it will provide good basis for the global understanding of the data and for the processing of the exploratory analysis. Thus, we are facing naturally the problem of determining the size of the representative points from data automatically. Grossburg *et al.*[38, 39] proposed the adaptive resonance theory (abbreviate as ART), of which the idea is that when a new sample point is inputted into the leaning programming, it affords information for updating only when the representative point of the previous instant is sufficiently close to this sample point. This makes a self-adapting resonance state, from which the nomination ART comes. This algorithm of the ART usually goes like follows: at the beginning, choose few or even no points of the representative points for the input sample by a pre-designed method of measuring similarity

and the threshold of "sufficiently closeness" according to the competition principle above, determine whether a representative point needs updating, which is sufficiently close to the inputted point, and update this representation point by it . If there is no old representative point sufficiently close to the newly inputted sample point, then a new representative point is added by taking the value of that inputted sample point. The algorithm possesses very good robustness. In ref. [40], two kinds of similarity control method , i.e. the ϵ-neighborhood control method and the center control method were given, where a series of setting parameters ϵ were also specified. It is a self-organizing method of selecting the representative points of the sample points for the pattern recognition of many patterns (here it is not required to choose only one representative point for one pattern). The author of ref. [40] illustrated that the experiment of the pattern recognition of handwriting digits obtained by using this algorithm is quite satisfactory in selecting the representative points (vector quantization), which gives very good correctness. Since the scale of the networks and the control parameters are not pre-required, this algorithm may be employed as a tool of exploratory statistical analysis[41].

The idea of self-learning network is very similar to the neural network algorithm[42, 43] proposed by E. Oja to get the principle components of a group of data or the eigenvalues and the eigenvectors of stochastic matrices. It is also an adaptively iterative algorithm. The central idea of the adaptive algorithm is to use the information obtained from the history of evolution to orient searching and computing. Its general calculating scheme is

$$X(t+1) = f(X(t),\ Y(t),\ \xi(t)),$$

where $X(t)$ stands for the state of the system at the time t, $Y(t)$ is the sample and $\xi(t)$ is the artificial noise added for the convergence of the searching process. The reason for the introduction of the artificial noise is that for the system with many extreme values, even the sample is inputted randomly, it is still possibly trapped in a local extreme value and so an appropriate added artificial noise may enable the system to achieve the expected global object[15].

Referring to the proof of the convergence of the simulated annealing, in studying the convergence and estimates of the convergent speed of the Kohonen algorithm, ART and its modified algorithm we may meet the mathematical problems involving the large deviation theory, the estimation of the spectral gap of the solutions of the stochastic iterative equations with discontinuous coefficients. These open problems are also of mathematical interest and with difficulties.

For adaptive algorithm and its various possibilities of applications, ref. [44] gives a comprehensive illustration for the historical situation before 1990. In this book, some rigorous discussions are given for the algorithms. Ref. [45] studies systematically the stochastic approximation from the angle of system identification, filtering and the estimation of the system parameters.

5 Hidden Markov models

The abstract mathematical theory and conceptions are not only vitally valuable for inspiring the establishment of the model, but also providing a suitable language of expression. In modeling problems, the randomness plays an important role. The hidden Markov model is a widely applied mathematical model with rich contents, including the non-linear filtering model. It is a kind of statistical model with incomplete data. And thus it is very flexible and adaptable for applications. On the other hand, it makes us possible to use conveniently the given knowledge of structures and properties about the objects studied.

A hidden Markov chain refers to a collection of an unobservable Markov chain (called the state process) and an associated observable process (called the measuring process). When multi-dimensional "time" (multi-indices) problems are considered, hidden Markov fields will be taken instead of hidden Markov chains[46]. Sometimes, the cases of indexing on graphs or indexing on trees are considered, and the Markov chain may become a Markov model on graphs or trees. All of them are called Markov models abbreviated by HMM. The most simple case happens when the time parameter $n \in \{0, 1, 2, \cdots, M\}$, and the measuring process is a given function of the state with an added i.i.d. perturbing noise. Then the relation between the measuring process $\{O_n: n \geqslant 0\}$ and the state process (Markov chain) $\{S_n: n \geqslant 0\}$ is satisfied by the following conditional probabilities:

$$P(O_n = v_n | S_M = s_M, S_{M-1} = s_{M-1} \cdots, S_1 = s_1, S_0 = s_0; O_{n-1} = v_{n-1}, \cdots, O_0 = v_0)$$
$$= P(O_n = v_n | S_n = s_n) \triangleq b_{s_n v_n}. \quad (5.1)$$

If both the state process and the measuring process take values in a finite set $\{1, 2, \cdots, N\}$, then a hidden Markov model is determined by the distribution of the initial random variable S_0(the initial distribution π_0 of the Markov chain), the transition matrix $\boldsymbol{P} = (p_{ij})_{i,j \leqslant N}$(transition function) of the Markov chain (or the Markov model) $\{S_n\}$ and the matrix $\boldsymbol{B} = (b_{ij})_{i \leqslant N, j \leqslant M}$ corresponding to the distribution of the measuring process under the condition of giving the state of the Markov chain (or the Markov model) at some "time" instant. Thus, the statistical distribution of an hidden Markov chain is completely described by the triple $(\pi_0, \boldsymbol{P}, \boldsymbol{B})$.

There are many different varieties of the hidden Markov models, e.g. the measuring process may be continuous, the conditional distribution in (5.1) may be normal or mixed normal, the state process may even also be continuous. Besides, the hidden Markov model may be a Markov field with both state process and measuring process being random fields of multi-parameters. Moreover, the noise appearing in the measuring process may be not independent of the state process, for example, assumption (5.1) may be changed into the following equality:

$$P(O_n = v_n | S_M = s_M, S_{M-1} = s_{M-1}, \cdots, S_1 = s_1, S_0 = s_0; O_{n-1} = v_{n-1}, \cdots O_0 = v_0)$$
$$= P(O_n = v_n | S_n = s_n, S_{n+1} = s_{n+1}) \triangleq b_{s_n s_{n+1}}(v_n). \quad (5.2)$$

The reason for the hidden Markov models being widely adopted comes from the fact that these models well reflect both randomness and the implicated structures of the objects, which makes us convenient to use the heuristic *a priori* understanding about the objects.

The central role played by the hidden Markov models in the speech recognition is well known[47—49]. In the speech recognition, the first step is to establish a corresponding such that one voice or sound (or one word, or one sentence, $\cdots$) corresponds to one hidden Markov chain, where the state space of the hidden Markov chain takes the set of all possible phonemes (or the refinements, or their combinations) included in this sound. For a sample of the measuring chain of a sound, the corresponding sequence of states successively appearing (including the state "empty") is the corresponding trajectory of the Markov chain of this HMM for this sound. For example, the sound of the Chinese word "Gao" may have a trajectory consisting of successive states (phonemes) like

$$\text{'ggg(ge)(ge)(ga)(ga)(ga)aaa(au)(au)uuu'}, \quad (**)$$

where one letter stands for the Chinese "Pin Ying" (the Latinized phoneme of Chinese sound), the two letters in the parenthesis stand for the transfer from one to another. The corresponding trajectory of the measuring process is the amplitude of the acoustic signal of each 'letter'. The sequence in (**) is exactly what we want to know, while it is unobservable. In order to establish the above corresponding relation, we need to "learn" from a group of observation samples of this sound (a number of acoustic signals of various pronunciation of the Chinese word "Gao"), that is, to do the parameter estimation for $(\pi_0, \boldsymbol{P}, \boldsymbol{B}.)$ without the information of states of the corresponding Markov chain (the state process), or say as statisticians, to do the parameter estimation by using the incomplete data, which is the learning phase.

After learning and obtaining the parameters of the hidden Markov model of each sound, one may do the recognition, the idea of which is to find the hidden Markov model (the triple $(\pi_0, \boldsymbol{P}, \boldsymbol{B}.)$ corresponding to a sound), by which the sample of the measuring chain (the acoustic signal of a sound) appears with the greatest probability. Then the result of the recognition is just the sound corresponding to that model found.

Recently, the hidden Markov field model has been applied to the recognition of the off-line handwriting Chinese characters and a satisfactory high recognition rate has been achieved, indicating the application potential of the HMM[50].

The above two problems have two common features: they both have an implicate basic structure and some randomness. For instance, the basic stroke and their relative link for a Chinese character are deterministic but the length of a stroke, the spacing and the slope are various from time to time and person to person. Similarly, a sound has its own basic structure, but the pronunciation of this sound changes ran-

domly with the shape of the mouth, the location in the oral cavity, the strength and the length of pronouncing, etc. Even the pronunciations of the same sound by the same person are different from time to time. It is quite satisfactory to describe this randomness with nice structure by the hidden Markov model.

In genetics, the inheritance of the attribute is directed by the gene information included in the chromosomes of the cell nucleus. Gene consists of the sequence of DNA bases. If gene is considered as the fragments of a book, the clones of DNA sequence are the sentences, sections and chapters. There are four DNA bases: A, T, G, C, which appear just like the letters in sentences. A DNA sequence is then exactly a sequence of letters A, T, G, C. The gene of the human being consists of 23 chromosomes (DNA sequences) with the total amount of about three billions bases. While the minimal DNA sequence of the living things (e.g. the gene of the virus) has about one million bases. To read out the DNA sequences of the bases has extremely significance for genetic engineering, heredity disease, cancer diagnose and therapeutic. However, so far, devices for fast reading DNA bases can only deal with the sequence of several hundred to a thousand bases. The method of fast breaking the sequence is random. The location of the breaking point and the number of fragments cannot be controlled and the fragments obtained after breaking are mixed up with the order of arrangement not being able to be distinguished. Moreover, three kinds of errors may appear: misreading, missing reading and adding fake bases. The rough idea of solutions goes like the following: simultaneously breaking several copies of the same DNA sequence of bases in a random way and reading out the broke fragments, by using the fact that two of the breaking points are different with the probability one, we can connect the fragments to reach the global order by cross-validation of different fragments. Churchill is the first one who proposed to use the hidden Markov model to solve this assembling problem. The basic idea is to express the fragments broken up from a long DNA sequence by a 'random gun' as a Markov chain with five states: B (beginning), R (right reading), W (wrong reading), I or D (missing or extra insert), and E (end), and to take the readout fragments from the sequence of bases as the observation chain (samples of measuring process) corresponding to this Markov chain. In this way, the problem is put into the frame of the HMM[51].

In applying the HMM, the first thing is to construct the model, namely to set up the state set and its size N and to determine the corresponding observation process. In order to get those simple connection between the observation chain (or field) and the Markov chain (or field) as in (5.1) or (5.2), we often have to manipulate properly the state space of the observation chain coming from the real world. For example, in practice, an off-line handwriting Chinese character is a picture in black and white pixels (a matrix with entries 1 and 0 only), and the dependence of the pixels (real observations) on strokes (the states of the hidden Markov field) does not meet the demand of (5.1) or (5.2). We have to do some processing, i.e. the so-called "feature extracting", to simplify the relation between the measuring chain and the hidden Markov chain such that (5.1) and (5.2) hold.

There are two phases in HMM with fixed size: learning phase and running phase. The learning phase is to find (or say "estimating") parameters (π_0, P, B) of HMM from a given set of samples of the measuring chain (the observation process), to determine the model completely, while the running phase is for a given sample of the measuring chain to estimate the sample path of the hidden Markov process corresponding to this given sample path or to find the best HMM fit for this sample from the set of models corresponding to various patterns obtained from the learning phase. In the speech analysis and in the handwriting Chinese character recognition the final answer wanted is the best model, which represents a particular sound or character, for a given sample of speech or handwriting character. This is a process of identification. Usually we determine the best model by Bayes inference, where the best model means the one under which the sample of observation happens with the biggest probability, while the assembling of DNA sequences is different, which wants to know how the bases are arranged, i.e. the state of the sample of the hidden Markov chain corresponding to the given set of observations (fragments) instead of the 'best model'.

In the learning phase, as we discussed above, it is a problem of parameter estimation for incomplete data. And one can do estimation in the light of EM algorithm; namely, (ⅰ) setting a set of initial parameters; (ⅱ) estimating the corresponding sample path of the hidden Markov chain by Bayes inference;

(iii) using the chain obtained from (ii), comparing it with the given observation samples to estimate the parameters by the maximum likelihood estimation; (iv) repeating steps (ii) and (iii) alternatively till converging. Also, one may set an initial sample path of the hidden Markov chain (model), and alternatively repeat steps (ii) and (iii) till converging. This is often used in the speech analysis and the handwriting Chinese character recognition, since in these case, usually a voice or a character is corresponding to a hidden Markov model (π_0, $\boldsymbol{P}$, $\boldsymbol{B}$). A standard sample of this character (voice) would be a nice initial.

It is worth pointing out that the key step to get a practically workable computing design is to turn the alternative procedure above into an iterative calculation formula. Otherwise, the EM algorithm itself is a NP hard problem. Besides, in practice, the dimension of the parameter space is so large that the calculation will be impossible to fulfill in a usual EM procedure. Therefore, the idea of Gibbs field——to change only one component of the state or one parameter at each time, can be applied, and the Markov chain Monte Carlo is also applicable.

Even though the hidden Markov model has been used in various fields, but a number of theoretical problems remain to be solved, which are important for the efficient application of this model. Theoretical results on the statistical learning consistency have been achieved[52]. Moreover, it is important to know if the states of the hidden Markov model are reasonably selected, if they contain necessary information on the problems in consideration, and if the observation process implies enough information for understanding of the whole model. Thus, to tell generally (not only for a specific problem) how we can answer these questions based on the information in hand (such as samples we have) is a sensible and important statistical and mathematical problem. As is mentioned above, a simple recursive formula using *a priori* known knowledge is the key to fulfill HMM.

For the data processing problems in stochastic control and filtering, the algorithm and iterative formula are discussed in detail mathematically and rigorously in ref. [53]. The Girsanov formula for difference equations with discrete state space is introduced as a basic tool there.

6 Two concessive strategies

It is natural for people to ask how the computational intelligence methods can solve those complex problems with very large size in a practically reasonable time , while the problems are really of NP hard complexity. In fact, the algorithm implies two concessive strategies. Noticing the randomness in the algorithm, people easily see that the results of such a calculation are random. Here, that the result can be reached in a certain time period ought to be understood in the sense that one can get the results by this kind of calculation within a certain period with a probability big (close to 1) enough. This means that the success of the calculation can only happen with a probability big enough, while we must take a risk of failure with a small probability when using this kind of algorithm. In other words, we trade the failure with a small probability for the fast success in most cases. This is just a concession. And in terms of mathematics, the convergence is in fact "converging in probability".

In recent years, in the field of artificial intelligence, there is another popular concession[54], i.e. take the good enough solution as the aim of the optimization instead of the best solution. In many cases, the difference between the best solution and some good enough solutions may be even smaller than the cost for improving the best solution from good enough solutions. Then it will be nonsense if one insists on getting the best solution. That is the so-called ϵ-optimization solution. It is possible to change the NP-hard complex problems into P-hard complex problems by finding ϵ-optimization solutions instead of the best solutions. These two concessions reflect the strategies of the intelligence of human beings. As a matter of fact, such strategies are frequently used in everyday life. For instance, it is not avoidable to meet various traffic accidents in modern cities, but people do not stop going out. We just try to use some regulations to limit the probability of being injured by accidents as small as possible.

In many methods of computational intelligence, no matter being aware or unaware, people are often using these two concessive strategies. As a matter of fact, when various computational intelligence methods are used to solve P-hard problems, it does not mean that problems can always be solved in the poly-

nomial time of the size, but the probability of calculation time of bigger than the polynomial function of the size tends to 0 as the size goes to infinity. This is true for the case where the mean time of the calculation in various situations is the polynomial function of the size of the problem.

Acknowledgement Project supported by the "863" Project of High Technology and the National Natural Science Foundation of China (Grant No. 19571045).

References

1 Li, G. J., Computational intelligence: An important field of research, *Fundamental Study of Intelligent Computer* '94 (eds. Li, W., Huai, J. P., Bai, S.) (in Chinese), Beijing: Press of Tsinghua University, 1994, 9—12.
2 Dai, R. W., Semantics and grammar recognition based on artificial neural networks, *Fundamental Study of Intelligent Computer*'94 (Li, W., Huai, J. P., Bai, S.) (in Chinese), Beijing: Press of Tsinghua University, 1994, 1—5.
3 Sejnowski, T. J., Rosenberg, C. R., Parallel networks that learn pronounce English text, *Complex Systems*, 1987, (1): 145.
4 Fogel, L. J., Owens, A. J., Walsh, M. J., *Artificial Intelligence Through Simulated Evolution*, New York: John Wiley, 1966.
5 Holland, J. H., Genetic algorithm and the optimal allocations of trials, *SIAM J. of Computing*, 1973, 2(2): 88.
6 Kirkpatrick, S., Gelatt, C. D., Jr., Vecchi, M. P., Optimization by simulated annealing, *IBM Research Report*, 1982, Rc 9355.
7 Macready, W. G., Siapas, A. G., Kauffman, S. A., Criticality and parallelism in combinatorial optimization, *Science*, 1996, 271(5 Jan.): 56.
8 Khas'minskii, R. Z., Application of random noise to optimization and recognition problems, *Problems of Information Transmission*, 1965, 1(3): 89.
9 van Laavhoven, P. J. M., Aarts, E. H. Y. L., *Simulation Annealing: Theory and Applications*, Holland: D Reidel Publishing Company, 1987.
10 Holly, R., Stroock, D., Simulated annealing via Sobolev inequalities, *Commun. Math. Phy.*, 1988,115: 553.
11 Chiang, T. S., Chow, Y., On the convergence rate of annealing Process, *SAIM J. Control Optim.*, 1988, 26: 1455.
12 Chiang, T. S., Chow, Y., A limit theory for a class of inhomogeneous Markov processes, *Ann. of Probability*, 1989, 17: 1483.
13 Hwang, C. R., Sheu, S. J., Large-time behavior of perturbed diffusion Markov processes with application to the second eigenvalue problem for Fokker-Plank operators and simulated annealing, *Acta Applicae Math.*, 1990, 19: 253.
14 Chiang, T. S., Sheu, S. J., Diffusions for global optimization in R^d, *SIAM J. Control Optm.*, 1987, 25: 737.
15 Fang, H., Gong, G., Qian, M. P., Annealing of iterative stochastic schemes, *SIAM J. Control Optim.*, 1997, 35(8): 1886.
16 Szu, H. H., Hartley, H. L., Fast simulated annealing, *Phy. Lett. A*, 1987, 123(3-4): 157.
17 Szu, H. H., Hartley, H. L., Non-convex optimization by fast simulated annealing, *Proc. of IEEE*, 1987, 75(11): 1538.
18 Fang, H., Gong, G., Qian, M. P., Disconvergence of Cauchy annealing, *Science in China*, Ser. A, 1996, 39(9): 945.
19 Li, Y., Qian, M. P., Convergence of TINA algorithms, *Acta Sci. Natur. Univ. Pekin.*, 1996, 32(5): 557.
20 Lei, G., Qian, M. P., Generalized time invariant noise algorithm and related bifurcation problem, *Tech. Report* (in Chinese), Beijing: Peking University Press, 1997.
21 Fang, H., Gong, G., Qian, M. P., An improved annealing method and its large time behavior, *Stochastic Processes and Their Appl.*, 1997, 71(1): 55.
22 Kolmogorov, A. N., On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition, *Dokl. Akad. Nauk.* (in Russian), 1957, 144: 953.
23 Hopfield, J. J., Neural networks and physical systems with emergent collective computational ability, in *Proc. of Nat. Acad. Sci., USA*, 1982, 79: 2554.
24 Hopfield, J. J., Pattern recognition computation using action potential timing for stimulus representation, *Nature*, 1996, 6 (July).
25 Zu, Z. B., Hu, G. Q., Kwong, C. P., Asymmetric Hopfield-type networks: theory and applications, *Neural Networks*, 1996, 9(3): 483.
26 Hertz, J., Krogh, A., Palmer, G. R., Introduction to the theory of neural computation, LN Vol.1, Santa Fe Institute, *Studies in the Science of Complexity*, Redwood City, California: Addison-Wesley Pub. Co., 1991.
27 Azencott, R., Boltzmann machines: high-order interactions and synchronous learning, *Stochastic Models, Statistical Methods and Algorithms in Image Analysis. Lecture Notes in Statistics* (ed. Barone, P.), Vol. 74, Berlin: Springer, 1992.
28 Zheng, J. L., *Artificial Neural Network* (in Chinese), Beijing: Higher Education Publishing House, 1992.
29 Amit, D. J., *Modeling Brain Function, the World of Attractor Neural Networks*, Cambridge: Cambridge University Press, 1989.
30 Geman, S., Bienenstock, E., Doursat, R., Neural networks and the bias/variance dilemma, *Neural Computation*, 1992, 4: 1
31 Linsker, R., Self-organization in a perceptual network, *Conputer*, 1988, 21(3): 105.

32 Feng, J., Pan, H., Analysis of Linsker-Type Hebbian learning: rigorous results, 1993 *IEEE International Conference on Neural Networks*, San Francisco, California, 1993.

33 Albeverio, S., Feng, J., Qian, M. P., Role of noise in neural networks, *Physics Rev.* E, 1995, 52(6): 6593.

34 Holland, J. H., *Adaptation in Natural and Artificial Systems*, *Ann Arbor*, Chicargo: The Univ. of Michigan Press, 1975.

35 Kohonen, T., *Self-organization and Associative Memory*, 3rd ed., Berlin: Springer-Verlag, 1989.

36 Burton, R. M., Pages, G., Self-organization and a.s. convergence of the one-dimensional Kohonen algorithm with non-uniformly distributed Stimuli, *Stochastic Processes and Their Appl.*, 1993, 47(2): 249.

37 Burton, R. M., Faris, W. G., A self-organizing cluster process, *Ann. of Appl Prob.*, 1996, 6(4): 1232.

38 Grossberg, S., Competitive learning: from interactive activation to adaptive resonance cognitive, *Science*, 1987, 1: 23.

39 Carpenter, G. A., Grossberg, S., The ART of adaptive pattern recognition by a self- organizing neural network, *Trans. IEEE on Computer*, 1988, 37(3): 77.

40 Qian, M. P., Competition learning approach of artificial neural networks, *Fundamental Study of Intelligent Computer '94* (eds. Li, W., Huai, J. P., Bai, S.) (in Chinese), Beijing: Press of Tsinghua University, 1994, 9—12.

41 Qian, M. P., Wu, D., The statistics and discussion on various distance of images and applications to fuzzifying technique, in *Proc. of the Asian Conference on Statistical Computing*, *Beijing*, 1993, 181—184.

42 Oja, E., Neural betworks, principle components, and subspace, *International Journal of Neural Systems*, 1989, 1: 61.

43 Oja, E., Karrunen, J., On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix, *Journal of Mathematical Analysis and Applications*, 1985, 100: 69.

44 Benveniste, A., Metivier, M., Priouret, P., *Adaptive Algorithm and Stochastic Approximations*, Berlin: Springer, 1990.

45 Chen, H. F., Zhu, Y. M., *Stochastic Approximations* (in Chinese), Shanghai: Shanghai Science and Technology Press, 1996.

46 Kunsch, H., Geman, S., Kehagias, A., Hidden Markov random fields, *Ann. Appl. Probab.*, 1993, 3(3): 577.

47 Rabiner, L. R., A tutorial on hidden Markov models and selected applications in speech recognition, in *Proc. IEEE*, 1989, 77(2): 267.

48 Rabiner, L. R., Juang Biing-hwang, *Fundamentals of Speech Recognition*, Hong Kong: Prince Hall International Inc., 1993.

49 Huo, Q., Chan Chorkin, Contextual vector quantization for speech recognition with discrete hidden Markov model, *International Symposium on Speech Image Processing and Neural Networks*, 13—16, April, 1994, Hong Kong, 698—701.

50 Deng, M. H., Qian, M. P., Method of Recognition of handwriting Chinese characters and their realization based on hidden Markov fields, *Symposium of the 3rd Session Intelligent Intersection of Computer in China and Intelligence Application Conference* (eds. Wu, Q. Y., Qian, Y. L.), Beijing: Electronic Engineering Press, 1997, 204—208.

51 Churchill, G. A., Accurate restoration of DNA sequences, *Case Study in Bayesian Statistics*, Vol. Ⅱ, *Lecture Notes in Statistics* (eds. Gatsonis, C., Hodges, J. S., Kass, R. E. *et al.*), 105, Berlin: Springer-Verlag, 1995, 90—148.

52 Leroux, B. G., Maximum-likelihood estimation for hidden Markov modeling, *Stoc. Processes and their Appl.*, 1992, 40(1): 127.

53 Elliott, R. J., Aggoun, L., Moore, J. B., *Hidden Markov Models*, Berlin: Springer-Verlag, 1995.

54 Ho Yu-chi Larry, Soft Optimization of Hard Problem, in *Proc. of International Conference on Control and Information* (invited talk), Hong Kong, 1996.

(*Received June* 19, 1998)