Spiking DNA Computing with Applications to BP Neural Networks Classification

Wenke Zang, Xiyu Liu and Ruifang Xia
1School of Management Science and Engineering, Shandong Normal University, Jinan, 250014, China
2Shandong University OuMa Software Co. Ltd, Jinan Shandong, 250014, China

Abstract: The study uses the idea of the extreme parallel to solve the BP neural network classification. Modification of the weights is not the traditional method which is to modify the connection weights between neurons repeatedly, but to find a group of weights in all possible weights combinations. The groups of weights are suitable for the relationship of the ideal input and the ideal output. Therefore, the model has some advantages compared with the traditional serial model in time miscellaneous. In the actual DNA computing, we also associate the coding problem with the model design. The coding problem is an important issue worthy to study in the DNA computing. There are many factors affecting the coding. The coding in this study is made when certain factors are overlooked.

Keywords: BP neural network, classification modal, DNA computing

INTRODUCTION

DNA computing is a newly emerging cross-research area, which is integrated with and penetrated by computer science and molecular biology. It is based on the coding DNA sequence as the operands. With the computing operations of molecular biology, its goal is to solve complex mathematical problems. DNA computing has such characteristics as the huge quantity, intensive information capacity and highly parallel processing operation. It reaches the right answer quickly through a powerful search strategy, which makes its computation much faster than that of the conventional computer. At present, many mathematicians and computing scientists are paying their attention to the design of the DNA computing, which has given birth to a new discipline, namely the DNA computing. It reveals to us that DNA can be used as the media of calculation to solve the mathematical problems, which is the true sense of biomathematics. Artificial neural network has a unique structure and a processing information method, which enables it to achieve significant results in many practical applications and solve some traditional complex mathematical problems. In recent years, in order to make the computer simulate the functions of human brain, advanced countries focus on the artificial neural network research, which has taken an important step to the computational intelligent (Chen and Yu-Zhen, 2003; Ravinderjit et al., 2002; Liu et al., 2000).


In this study, we use the idea of the extreme parallel to solve the BP neural network classification. The groups of weights are suitable for the relationship of the ideal input and the ideal output. In this way, the model has some advantages compared with the traditional serial model in time miscellaneous. Moreover, in the actual DNA computing, we also associate the coding problem with the model design and then we could not only optimize model design, but also continuously improve the encoding method.

BP NETWORK IN ARTIFICIAL NEURAL NETWORK

BP network is a hierarchical neural network with three or more layers. Neurons between the upper and lower layer are fully connected. That is to say, each neuron between the lower and upper layer is fully connected and it has no connection with other neurons within the same layer. Typical BP network is a three layers feed-forward network, which is consisted of the
input layer, one hidden layer and the output layer. The input number in each neuron of the hidden layer and the output layer is the Sigmoid function. The classification model is usually expressed as a binary (0/1) vector and so as the set of classes.

By training in \( p \) actual samples \((X', Y')\), \((X'', Y'')\), ..., \((X', Y')\) we want to achieve the connection weights between each neuron, that is, \( w_{ij}, v_{t} \) \((i=1, 2, ..., n, j=1, 2, ..., m, t=1, 2, ..., q)\). The final goal is to meet the correspondence between the ideal input and desired output of the samples. The training process is guided by the teacher’s instruction. When a sample mode pair is provided to the network, the neuron’s activation value is calculated. When a sample mode pair is output of the samples. The training process is guided by the teacher’s instruction. When a sample mode pair is obtained, it is compared with the actual output and corrects connection weights to the direction of reducing the error between the ideal output and actual output and corrects connection weights to the direction of reducing the error between the ideal output and actual output. Then it presses to the output layer and then each neuron in the output layer gets the input response of network. Then it presses to the direction of reducing the error between the ideal output and actual output and corrects connection weights from the output layer through the middle layer, layer by layer and finally back to the input layer. The connection weight, which is got after the training, tests other samples that do not belong to the sample sets. Its results can also achieve the correct correspondence.

If you input samples \( P_{s}, (X^{(p)}, Y^{(p)}) \), after certain ways of training, we get two groups of connection weights value, namely \( W(P_{s}) \) and \( V(P_{s}) \). \( W(P_{s}) \) includes all the connection weights between the input layer and the middle layer in network while \( V(P_{s}) \) includes all the connection weights between the middle layer and the output layer in network. Now, the solution to \( W(P_{s}) \) and \( V(P_{s}) \) is not single, but a scope in weighted space, or several scopes. As for all the sample pairs \( P_{s} = 1, 2, ..., P \) can satisfy: \( Y^{(p)} = F \left( X^{(p)}, W^{(p)}, V^{(p)} \right) \)

Each of the solutions is \( (W^{(p)}, V^{(p)}) \) \((W^{(p)}, V^{(p)}), (W^{(p)}, V^{(p)}) \)... \((W^{(p)}, V^{(p)}) \). By learning the sample set, we get the solution which satisfies the corresponding relationship of all the sample pairs, that is, \( W = \bigcap_{i} W^{(i)}, V = \bigcap_{i} V^{(i)} \).

**DNA computing model in BP network:**

**Main idea:** We know that, for the typical BP network which consists of \( n \) input neurons, \( m \) hidden neurons and \( q \) output neurons, all the neurons make up three neuron layers. However, after receiving the weighting of the input layer, the middle hidden layers deals with the weighting and gets the corresponding value by the Sigmoid function. At the same time, it takes the function value as the input information and sends it to the output layer. The output layer neurons deal with the input and get the output of the whole network. Therefore, the middle hidden layer may be viewed as output layer relative to the input layer. Meanwhile, relative to the output layer, it can be viewed as input layer. Thus, BP network is decomposed into two-layer network. That is to say, the middle hidden layer has double meanings, both as the input layer to the first network and as the output layer to the second network.

From the experience and the learning mechanism of the BP network, we also know that the size of the weights is usually in the interval \([-1, 1]\). Meanwhile, all neurons in the hidden layer and output layer have the same input and output function-Sigmoid function. So, in order to establish a suitable model for DNA computing, we will give the following special treatment to the weights between the input layer and hidden layer, the weights between the hidden layer and output layer and the input and output function of the neurons as well:

- We equally divide the range \([-1, 1]\) into 20 and the size of the weights \( w_{ij} \) and \( v_{t} \) take discrete values: -1, -0.9, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. They are all the probable connection weights values of the neurons, between the input layer and hidden layer, also, between the hidden layer and output layer. That is, each weight value \( w_{ij} | v_{t} \) has 21 kinds of possible values.

- Meanwhile, we divide neurons’ value of the Sigmoid function which is \((0, 1)\) into 10 files. That is, the value of the function takes discrete values: 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. Sigmoid function is a monotonically increasing function, so for the input of the neurons of each hidden layer, which is all the input values weighting of input neurons, we consider that the relationship of the neuron’s weighted inputs and the function value is shown in Table 1. However, as to the output layer neurons, when the function value is greater than 0.7, namely, neurons net input values \( \text{net}>1.1 \), we think the output value is 1. When the function value is lower than 0.3, namely, neurons net input values \( \text{net}<1.1 \), we think the output value is 0.

When were taken to set a value for each possible value, for this moment, all \((m+n+q)\) weights constitute a combination of a set of weights. That is, \( w_{11}, w_{12}, ..., w_{1m}; w_{21}, w_{22}, ..., w_{2m}; w_{31}, w_{32}, ..., w_{3m} \) and \( u_{11}, u_{12}, ..., u_{1q}; u_{21}, u_{22}, ..., u_{2q}; u_{31}, u_{32}, ..., u_{3q}; \) \( u_{m1}, u_{m2}, ..., u_{mq} \) together is a set of weights:

<table>
<thead>
<tr>
<th>Neurons net input values (weight and net)</th>
<th>Sigmoid function value</th>
<th>Neurons net input values (weight and net)</th>
<th>Sigmoid function value</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.9&lt;net&lt;-1.7</td>
<td>0.1</td>
<td>-0.2&lt;net&lt;0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>-1.7&lt;net&lt;-1.1</td>
<td>0.2</td>
<td>1.1&lt;net&lt;1.7</td>
<td>0.8</td>
</tr>
<tr>
<td>-1.1&lt;net&lt;-0.6</td>
<td>0.3</td>
<td>1.7&lt;net&lt;2.9</td>
<td>0.9</td>
</tr>
<tr>
<td>-0.6&lt;net&lt;-0.2</td>
<td>0.4</td>
<td>net&gt;2.9</td>
<td>1.0</td>
</tr>
<tr>
<td>-0.2&lt;net&lt;0.2</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1: The relationship between the input value and function value of the neuron**
Then, we make the ideal input vector \(X^{(1)}\) of the sample \((X^{(1)}, Y^{(1)})\) input to the classifier and get all the possible output value of every neuron of the hidden layer from all the possible weights \(w_i\) and the input weighting sum. Again, we take all the possible output values as input values and find a lot of groups’ weights value combination of all possible weights \(v_j\). Then, it makes every group of weights combination in the whole weights combination to get the ideal output \(Y^{(1)}\) of the sample \((X^{(1)}, Y^{(1)})\). All the weights combination recorded as \((W^{(1)}, V^{(1)})\) as an entirety. It is the same for sample \((X^{(2)}, Y^{(2)}), \ldots, (X^{(p)}, Y^{(p)})\), which can also find the corresponding relationship \((W^{(2)}, V^{(2)}), \ldots, (X^{(p)}, Y^{(p)})\) of the ideal input and output to meet the sample.

Then, we find a group of common combination of weights \((W, V)\) in,\((W^{(2)}, V^{(2)}), \ldots, (X^{(p)}, Y^{(p)})\), which can satisfy the corresponding relationship of sample pair of all the training set.

Then, we take use of this suitable weights combination \((W, V)\) again. We calculate the unknown class input vector and then, the outputs of the output neurons get the class of the input vector directly. As follows

\[
\begin{align*}
X^{(1)} & \xrightarrow{w^{(1)}, v^{(1)}} Y^{(1)} \\
X^{(2)} & \xrightarrow{w^{(2)}, v^{(2)}} Y^{(2)} \\
\vdots & \xrightarrow{w^{(p)}, v^{(p)}} Y^{(p)} \\
t & \mapsto \text{integer section of then } = (w,v)
\end{align*}
\]

Input vector of the unknown class \(X \xrightarrow{(w,v)} \text{output Y}\)

Therefore, in order to realize the classifier using DNA language, we will divide the whole function of the classifier into 4 modules:

- Discriminate classifier of the unknown vector

**ENCODING METHOD**

**The molecular libraries of the initial sample**: We take a functional segment to encode the input values of the input neurons. For example, the \(i^{th}\) input neuron’s input value is 0 and its chain form is \(5'\text{Int}(i)05'\) and when the input value is 1, its chain form is \(5'\text{Int}(i)15'\). All the input neurons’ input has only two states, namely \(0\) or \(1\), so the molecule of initial sample has \((2n)\).

**The molecular libraries of weights**: The molecule in the molecular libraries of weights includes two categories: the chain of the connection weights \(w_{ij}\) between the input neurons and the hidden neurons and the chain of the connection weights \(v_{jl}\) between the hidden neurons and the output neurons.

Every chain in the first weight \(w_{ij}\) chain is single, which consists of 8 functional segments. The coding form is shown in (Fig. 1).

The first functional segment shows that the input value of the the \(i^{th}\) input neuron that the weights received is \(0\) or \(1\). It’s form is \(3'\text{Int}(i)05', 3'\text{Int}(i)05', 5'\text{Int}(i)15'\). They are complementary with the molecule of the initial sample \(5'\text{Int}(i)05', 5'\text{Int}(i)15'\).

The second functional segment “\(R_i\)” is the restriction recognition site of the enzyme I.

The third functional segment “\(\text{Int}(i), \text{Mid}(j)\)” shows the marker of the connection weights between the \(i^{th}\) input neuron and the \(j^{th}\) middle hidden neuron which is represented by weight \(w_{ij}\).

The fourth functional segment encodes the size of the weight \(w_{ij}\). Because the weights \(w_{ij}\) and \(v_{ji}\) have 21 possible values: \(-1, -0.9, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\). It can be marked by “\(w\)” and “\(w\)” here has 21 possible values. Different values correspond to different encoding.

The fifth functional segment “\(\text{Int}(i)\)” is the restriction recognition site of the enzyme II.

The sixth functional segment is used to encode the resulting value of the product between weight \(w_{ij}\) and the input value \(0\) or \(1\) of the \(i^{th}\) input neuron. Different results...
However, when the input value of the \(i\)th input neuron is 0, the result of the product is 0. Thus, the corresponding functional segment of the output neuron is.

The eighth functional segment “Rin” is the restriction recognition site of the enzyme III.

The ninth functional segment “Rin” is the restriction recognition site of the enzyme II.

The tenth functional segment “(Mid (j+1), Out (t))”, shows the connection weight marker between the (i+1)th input neuron and the tth output neuron.

Therefore, in this coding mechanism, every weight chain \(v_{jt}\) contains the input information of the jth hidden neuron. It also contains both the connection weight between the jth hidden neuron and the tth output neuron.

hidden neuron output value determination molecular:

\[ 3'R_j(a, c)S' \quad (a = 0, 0.2, 0.8, 0.9, 1) \]

Since there are 10 kinds of values for a and m hidden layers, the number of hidden neuron output value determination molecular is 10m.

Two types of molecular in weights molecule library are placed in different test tubes according to the difference of hidden neurons and output neurons. Namely, \(W_{1j}, \ldots, W_{mj}\) is placed in a weight test tube, while \(V_{1j}, \ldots, V_{mj}\) in the other weight test tube. In this way, if there are \(m\) hidden neurons and \(q\) output neurons, what need are \(m+q\) weights test tubes. And each sample needs such a set of weights test tubes as \(U_{1i}, U_{2i}, \ldots, U_{mi}\) and \(V_{1i}, \ldots, V_{qi}\). In this way, there are \(P \times (m+q)\) test tubes for \(P\) pairs of samples. Meanwhile, we also need to set the weight test tube \(U_1, U_2, \ldots, U_m\) and \(V_1, V_2, \ldots, V_q\) which can be used in extracting the weight value chain that corresponds to the values of public weight combination (W, V).

**BIOLOGICAL OPERATIONS**

- **Input sample generator (the following actions for each sample):**
  - As for each sample \((X^{(k)}, Y^{(k)})\) of the training set P samples, the component of the vector corresponding coding of ideal input X(k) is fixed, respectively on P surfaces. One sample is fixed on each surface. The encoding of the input of each sample, which is single-stranded, is complementary with the initial
sample molecules. For example, if the first k sample is \(((1, 1, 0, 1, 0), (1, 0, 1))\) and the ideal input vector is \((1, 1, 0, 1, 1)\).

- Pour the solution of the initial sample molecules on the surface and then, the chain of initial sample molecules that is complementary with each component of the ideal input vector \(X^{(k)}\) of the kth sample will have fully complementary hybridization reaction with the single chain on the surface and form double strand. And then by controlling the temperature, some complementary double strand will break down. Then addexo nuclease on the surface and destroy the single chain. There maining double-stranded on the surface is formed when the chain of initial sample molecules that is complementary with each component of the ideal input vector \(X^{(k)}\) of the kth sample is fully complementary with k-chain and encoded on the surface of each component of k samples of the ideal input vector \(x^i\) single-stranded.

- Warm the kth surface and make double strands single, which are put into a test tube \(v_j\). All the molecules in this test tube are the single strands of each component of the ideal input vector \(X^{(k)}\) of the kth sample. In this way, we extract the ideal input \(X^{(k1)}\)in the kth sample from the initial sample of molecular libraries.

- When \(X^{(k)}\) is extracted, pour the solution of test tube \(v_j\) into each test tube of the kth set of weights tube \(U^{(k)}, U^{(k)}_{2..}, U^{(k)}_{m} v_j\) and prolong the reaction with the molecules in the test tube \(v_j\) as primers. Then addexo nuclease to the solution to destroy the single strands and make the solution in m test tubes go through electrophoresis, respectively. Remove those single strands that are much shorter than the weight chain and put the remaining strands into test tubes \(A^{(1)}, A^{(2)}, ..., A^{(j)}, ..., A^{(m)}\) according to different output neurons.

So, we have passed the ideal inputs of the initial samples to classifier through the above operations.

- **Find out \(W_1, W_2, ..., W_p\)** (respectively for each sample, do the following separately): We can find out weights combination \((W^{(1)}, Y^{(1)})\) that is fitted to \((W^{(1)}, Y^{(1)})\) by the ideal inputs and outputs of sample \((W^{(2)}, Y^{(2)})\). Similarly, by samples \((W^{(2)}, Y^{(2)})\), ..., \((W^{(m)}, Y^{(m)})\), we can find out weights combination \((W^{(2)}, Y^{(2)}), ..., (W^{(m)}, Y^{(m)})\) suited to themselves, respectively.

In the first module, we have passed all sample inputs to the classifier and transferred input information to the weight chain \(W_{\text{in}}\) between input layer neurons and hidden layer neurons. At the same time, there are product results of input value and its corresponding weights on the weight chain \(W_{\text{in}}\). Here, in order to calculate the input values of hidden neuron \(j\), that is the weighted sum of values of input layer neurons, we should connect weight chains corresponding to the same hidden neuron according to the order of input neurons, i.e., \(w_{ij}^1 \times x_i^1 + w_{ij}^2 \times x_i^2 + ... + w_{ij}^m \times x_i^m\), to constitute the chain of weighted sum of I-layer. Then, according to the length scope of each weighted sum, connect the input values of jth hidden neuron in its end, namely the input values of jth hidden neuron for the output neurons.

Then, pass output values of each hidden layer neuron \(j (j = 1, 2, ..., m)\) to the corresponding weight chain \(V_{ji}\) between hidden layer neurons and input layer neurons \(t (t = 1, 2, ..., q)\). After that, according to the order of hidden neurons, connect weight chains related to the same output neuron \(t\), to constitute the chain of weighted sum of the second layer. Then, select the chain of weighted sum suited to ideal outputs, according to the ideal outputs of output neuron \(t\). With the encoding of weight chain and the number of input neuron and hidden neuron, we can determine a threshold \(\phi_t\) of the weighted sum chain of the second layer. That is to say, if the length of a certain weighted sum chain of the second layer, which is generated before, is larger than the threshold \(\phi_t\), it demonstrates that the output value of output neuron of the weighted sum chain is 1; otherwise, if the length of a certain weighted sum chain of the second layer is smaller than threshold \(\phi_t\), it shows that the output value of output neuron of the weighted sum chain is 0. So, we reserve the weighted sum chain suited to the ideal output by the two length thresholds. And then, each reserved weighted sum chain contains a set of corresponding weights combination of input neuron and hidden neuron, hidden neuron and output neuron \(t\). So, we separately find out \((W^{(1)}, Y^{(1)}), (W^{(2)}, Y^{(2)}), ..., (W^{(m)}, Y^{(m)})\). The specific biological operation is described as following:

- Put enough restriction endonuclease I into the test tubes \(A^{(1)}, A^{(2)}, ..., A^{(j)}, ..., A^{(m)}\). Then use gel electrophoresis to the solution of each test tube and remove the short strands which have been cut down. And then, add enough ligase and get a series of weighted sum chains of the first layer.

- Heat test tubes of the previous step to melt strands and extract S'~3' single strands. Then respectively use electrophoresis to extract single stranded and in the different length, add output values with different function values to determine polymerase and ligase and perform extended reaction and then put the result chains into the test tubes \(B^{(1)}, B^{(2)}, ..., B^{(j)}, ..., B^{(m)}\) respectively, according to the differences of hidden neurons. For the same ideal input, the outputs of hidden neuron \(j\) vary in weights combination, so the length of DNA strands in the same test tube \(B^{(j)}\) is different.
Then heat the test tubes \( B_1, B_2, \ldots, B_m \) to melt the chain and after rapid cooling, extract \( S' = 3' \) single strands from each test tube separately. And then put enough \( S'R'R_3' \), by insertion and deletion system, delete the product result values of the first layer and output values of input neuron and put the result chains obtained from the reaction into the test tubes \( C_1, C_2, \ldots, C_i, \ldots, C_n \), respectively. Then a result chain in the test tube \( C_{i} \) contains a weights combination between all input neurons and the same hidden neuron \( j \) and the form of result chain is as Fig. 3:

- Heat the test tubes \( C_1, C_2, \ldots, C_i, \ldots, C_n \) to melt the chain and after rapid cooling, extract \( S' = 3' \) single strands from each test tube separately, then put polymerase and ligase and perform extended reaction and then put the result chains into the test tubes \( D_1, D_2, \ldots, D_j, \ldots, D_m \), respectively, according to the differences of output neurons.

- Then heat the test tubes \( E_1, E_2, \ldots, E_i, \ldots, E_m \) to melt the chain and after rapid cooling, extract \( S' = 3' \) single strands and then put enough \( S'R'mid(j) \), respectively, according to the differences of the input neurons.

- Put enough restriction endonuclease III into the test tubes, remove some short chains through electrophoresis and put ligase into the remaining solution, then the weight chains of all implied neurons and the same output neurons link together and form the second layer’s weighted summation chains. Then put the gotten chains into different test tubes \( F_1, F_2, \ldots, F_i, \ldots, F_m \), respectively according to the differences of the output neurons.

- Let the solution in test tube \( F_1, F_2, \ldots, F_i, \ldots, F_m \) go through electrophoresis and retain the chains of every test tube that meet the second layer’s weighted summation chains’ threshold according to each output neuron’s ideal output in the sample \( (x^{(1)}, y^{(1)}) \). That is to say, if the \( i \)th component of \( y^{(1)} \) is 1, then we retain the chains in test tube \( H^{(1)} \), whose length exceed threshold \( \phi_i \). If the \( i \)th component of \( y^{(1)} \) is 0, then we retain the chains in test tube \( H^{(0)} \), whose length is less than threshold \( \phi_i \). Then put the retained chains into different test tubes \( G_1, G_2, \ldots, G_i, \ldots, G_m \), respectively according to the differences of the output neurons.

- Heat test tubes to melt the chains \( G_1, G_2, \ldots, G_i, \ldots, G_m \), extract \( S' = 3' \) single strands after rapid cooling, then put enough \( S'R'mid(j) \) into the test tubes and delete the fragment \( L_{\gamma} = a^n \) of the result value, which is from the encoding weights of the second layer’s weighted summation chains multiplied by the input value through insertion and deletion system and put the gotten chains into different test tubes \( H^{(0)}, H^{(2)}, \ldots, H^{(m)} \) respectively according to the differences of output neurons. One result chain in test tube \( H_j^{(m)} \) includes weights of all input neurons and every implied neurons \( j \), a weight combination of all implied neurons and every output neuron \( t \) and at the same time, the combination can meet the ideal output of the \( k \)th pair sample.

- So, for the \( k \)th pair sample \( (x^{(k)}, y^{(k)}) \), we can find corresponding \((W(k), V(k))\) that fit ideal input and ideal output. In the same way, we do the above operations to each pair of samples, then we can get corresponding \((W(1), V(1)), (W(2), V(2)), \ldots, (W(P), V(P))\).

- **Figure out the intersection** \((W, V)\) of \((W^{(1)}, V^{(1)}), (W^{(2)}, V^{(2)}), \ldots, (W^{(P)}, V^{(P)})\): Figure out the intersection of \((W^{(1)}, V^{(1)})\) and \((W^{(2)}, V^{(2)})\) first, then figure out the intersection with \((W^{(3)}, V^{(3)})\) and so on until we figure out the intersection with \((W^{(P)}, V^{(P)})\). So the result we get is the intersection \( w = (W(1), V(1)), (W(2), V(2)), \ldots, (W(P), V(P)) \), namely the public weight value combination \((W, V)\) which makes all sample pairs get ideal output through ideal input. We judge whether two combinations are exactly equal or not on the basis that the two weight value combination chains are complementary or not. Each weight value combination chain includes weights between all input neurons and all implied neurons and the weight between every implied neuron and some output neurons. Here, we take how to obtain the intersection of \((W(k), V(k))\) and \((W(l), V(l))\) as an example. We gain \((W^{(k)}, V^{(k)})\) and \((W^{(l)}, V^{(l)})\), which are suitable for the \( k \)th pair sample and the \( l \)th pair sample, respectively in the last operation and they exist in form of double-strands in test tubes.

- Heat the test tubes \( H^{(1)} \) and \( H^{(0)} \), to unlock them respectively, then rapidly cool them. Then extract 3 \(-5'\) single strands from the test tube \( H^{(1)} \) and put them into the test tube \( F_1 \). Extract 3 \(-5'\) single strands from the test tube \( H^{(0)} \) and put them into the test tube \( F_0 \). Pour the solution in the test tube \( F_1 \) into the test tube \( F_0 \), then the chains in \( F_0 \) will take hybrid complementary reaction with the chains in \( F_1 \), by
controlling the temperature. If the chains in $I^0$, and the chains in $I^0$, take complementary reaction completely, it indicates that the weight value combination of the two chains is completely equal.

- Add exonuclease to the test tube $I^0$, in order to melt the single strands in the test tube and then the remaining chains in the test tube are the fully complementary double strands and partially complementary double strands. By the encoding rules of the weight value chain, we know that only the fragment encoding the size of the weights will not have complementary phenomena because corresponding weights are not equal. Then add fluorescent oligo nucleotide fragment $3' W 5'$ and $3' W 5'$ (21 kinds, respectively) into the test tube and the fluorescent oligo nucleotide fragment $3' W 5'$ and $3' W 5'$ will have complementary reaction with the fragments which are partially complementary double strands and are not occurring hybridization reaction in the test tube, then there will appear fluorescence on the partially complementary double strands.

- Get rid of the partially complementary double strands with fluorescence in the test tube $I^0$, and the remaining chains are fully complementary double strands, then put the remaining double strands into test tube $J^0$, so the combination of weights of the double strands in the test tube $J^0$, is the intersection of $I(W (1), W (2), ... , W (P))$ and $J(W (1), W (2), ... , W (P))$. Having the above operations, then, we place the final chain into the test tube $f^{(1)}, f^{(2)}, ..., f^{(P)}$. In this way, begin with the $((W (1), V (1)), ((W (2), V (2)), ..., (V (P)))$, combine with $W _p$ until $((W (1), V (1)), ..., (V (P)))$, we place the final chain into the test tubes $f^{(1)}, f^{(2)}, ..., f^{(P)}$.

- By adding sufficient number of restriction endonuclease II in the test tubes $J^0, J^0, ..., J^0$ and by having gel electrophoresis respectively, we remove the DNA strands whose length are less than or equal to 10 base pairs. Meanwhile, the remaining chains were respectively placed in test tubes $K1, K2, ..., Kq$.

- Warm test tubes $K_1, K_2, ..., K_q$ to melt the DNA chain. Extract the single chain $5' - 3'$ into test tube $L_1$ ($t = 1, 2, ..., q$) respectively. Pour the solution in the test tubes $L_1$ into the test tubes $U_j (j = 1, 2, m)$ and $V_1$ ($t = 1, 2, q$) which equipped with weight chains respectively. The DNA segment in test tube $L_t$ will react with weight chain in test tube $U_j$ and $V_1$. They will have hydridize complementary reaction. Control the temperature, then, make the third and the fourth functional segment in the weight chain be fully complementary with the DNA segment in test tube $L_t$ at the same time. But, you must be sure that only one functional complementary chain cannot form double chain. Then, add the exonuclease into the mixed solution to shred the single chain fully. Finally, the remaining chain is the “$2^W (i), W (j))” segment in the DNA segment in test tube $L_t$ that is completely equal to the “$2^W (w)$” segment; and the “$W (j), W (j))))” segment is completely equal to the “$2^W (v)$” segment. Then, place the remaining chain into the test tube $M_j (j = 1, 2, ..., m)$ and $N_t (t = 1, 2, ... , q)$ . Thus, the result chain in tube $M_j (j = 1, 2, ..., m)$ is a group of weight chain in $w$, which is the intersection of the $W (1), W (2), ... , W (P)$. The result chain in tube $N_t (t = 1, 2, ... , q)$ is a group of weight chain in $v$, which is the intersection of the $V (1), V (2), ... , V (P)$.

- **Discriminant classifier for the unknown vector:** In the last operation, we obtain $(w, v)$. Now, we react the first weight chain corresponding with a group of weight in $(w, v)$, with the unknown vector molecule. Then, we get the first layer weighting chain, based on the length of which we choose the output value of the hidden neurons. Then, we take the output value of the hidden neurons as the input value and make it respond to the second weight chain. In this way, we obtain the second layer weighting chain. Again, using the length of the weighting chain, we choose the output value of the output neurons, which is 0 or 1. Based on all the output of the output neurons, we can know the class of the unknown vector.

- Add the unknown vector into the test tube $M_j (j = 1, 2, ..., m)$. Take the unknown vector molecules as primer. Then, the unknown vector molecule can find the first functional segment of the weight chain in test tube $M_j (j = 1, 2, ..., m)$ hybridization automatically. Then, it forms double chains. The solution removes some of the chain which is much shorter than the weight value chain by gel electrophoresis. Put the remaining chain into test tube $O (j = 1, 2, ..., m)$. Add sufficient restriction endonuclease I in test tubes $O (j = 1, 2, ..., m)$. Make the solution of test tube $O (j = 1, 2, ..., m)$ go through gel electrophoresis and then remove the short chains. Based on the different output neurons, the remaining chains are put into test tube $P_j (j = 1, 2, ..., m)$, respectively. Then, by adding sufficient ligase, the remaining chains can form a series of weighting chain of the first layer.

- Warm every test tube in the last step to undo the chain. Extract the single chain $5' - 3'$. Then, we extract the single strands by electrophoresis and add in different range of length the output value that are accompanied by different function values to test molecule, polymerase and ligase into it. Then, the
prolonged reaction occurs. We put the result chain into the test tube \( Q_j \) (j = 1, 2, ..., m) respectively, based on different hidden neurons.

- Add restriction endo nuclease \( I \) into the test tube \( Q_j \) (j = 1, 2, ..., m). Cut down the stick head which contains the value of the hidden neuron. Then, put the stick head into the weight test tubes \( V_1, V_2, ..., V_t, ..., V_q \). Then, add polymerase and ligase and the prolonged reaction occurs. Then, add restriction endo nuclease \( m \) and make the solution of each test go through gel electrophoresis and remove the short chains. Then, add sufficient ligase into the solution, it can form a series of weighting chain of the second layer.

We define the length of all those weighting chain as threshold \( \phi_1 \) and define the length of the second layer weighting chain as threshold \( \phi_2 \). We use the size of the relationship to choose the output value as 1 or 0. That is to say, if the length of one weighting chain of the second layer is greater than the threshold \( \phi_1 \), we say the output value of the output neuron in the weighting chain is 1. Otherwise, if the length of one weighting chain of the second layer is lower than the threshold \( \phi_2 \), we say the output value is 0.

Thus, based on all the output of the output neurons, we can know the class of the unknown vector \( X \).

**CONCLUSION**

The passage uses the idea of the extreme parallel to solve the BP neural network classification. Modification of the weights is not the traditional constantly repeated modification of the connection weights between neurons, but to find a group of weights in all possible weights combination. The groups of weights are suitable for the relationship of the ideal input and the ideal output. Therefore, the model has some advantages compared with the traditional serial model in time miscellaneous. In the actual DNA computing, we also associate the coding problem with the model design. The coding problem is an important issue worthy to study in the DNA computing.

There are many factors affecting the coding. The coding in this study is made when certain factors are overlooked. We cannot completely eliminate the occurrence of various errors. Therefore, we should not only optimize model design, but also continuously improve the encoding method.

**ACKNOWLEDGMENT**

The research is supported by the National Science Foundation of China (No: 61170038, No. 60873058, No: 11171193), the Natural Science Foundation of Shandong Province (No. ZR2011FM001), the Shandong Soft Science Major Project (No. 2010RKMA2005).

**REFERENCES**


