

A Novel Digital Recognition Method Based on Improved SVD-DHNN

Xuemei Yao¹, Jiajia Zhang², Juan Wang³, Jiaying Wei⁴

School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang, China^{1,2,3,4}
State Key Laboratory of Public Big Data, Guizhou University, Guiyang, China¹

Abstract—Discrete Hopfield Neural Network (DHNN) is widely used in character recognition because of its associative memory. It is a fully connected neural network. Its weight initialization is a random process. In order to give full play to the associative memory of DHNN and overcome the problems of pseudo-stable points and complex structure caused by random initialization, an improved SVD-DHNN model is proposed. Firstly, the weight of DHNN is optimized by the global search capability of PSO to help the model jump out of the pseudo stable point; secondly, the weight matrix of DHNN is readjusted by singular value decomposition (SVD). The contribution rate is used to trim the weights of DHNN, which can reduce the complexity of the network structure; finally, the validity and applicability of the new model are verified by means of digital recognition.

Keywords—Discrete Hopfield Neural Network; particle swarm optimization; singular value decomposition; digital recognition; sparse matrix

I. INTRODUCTION

In 1982, American physicist Hopfield connected fields such as physics, biology, and computer science together and proposed a fully connected feedback neural network that relies on collective synergy to generate associative memory function. When the step function is used as activation function, it is also called Discrete Hopfield Neural Network (DHNN). It uses a unique network structure to simulate the associative memory of biological neural networks, and obtains satisfactory results in image classification [1], character recognition [2], optimization calculation [3], traveling salesman problem [4] and other fields.

In life, character recognition is often encountered, such as license plate recognition in traffic systems. Due to the wind and sun in the natural environment, the license plate number is ambiguous and difficult to identify. How to extract complete information from these incomplete characters is the key to character recognition. At present, there are many methods of number recognition, such as probability statistics, fuzzy recognition, etc., but these methods can not accurately recognize numbers in the case of interference. Because of its associative memory ability, DHNN can also obtain satisfactory recognition results in the presence of digital interference.

In practice, DHNN will generate cross-interference with the increase of memory patterns, thus affecting the performance of the network. Many scholars had improved it. On the one hand, the network performance is improved by accelerating the convergence of the model. Zhang et al. [5] studied the global convergence problem of fractional Hopfield neural networks

and derived a simple test condition to ensure the global stability. Feng et al. [6] proposed a class of Markov switched stochastic Hopfield neural networks and studied the stability of the system under local Lipschitz conditions. Wang et al. [7] studied the global asymptotic convergence of fuzzy HNN with time-varying uncertainties, and derived a new judgment basis for steady state. Shi et al. [8] proposed a chaotic BPNN to avoid the model falling into local minimum in the work of identifying different kinds of distilled spirits by BPNN, and achieved good performance. Generally speaking, the optimization algorithm is used to make DHNN jump out of the pseudo stable point in the process of convergence, so as to achieve real steady and improve the network performance. Zhang et al. [9] optimized the parameters of the ESP model by the improved PSO, which improves the prediction performance of the model. Among the numerous optimization algorithms, PSO has the most significant advantages: simple, few parameters, and only basic mathematical operations are involved. Therefore, PSO is selected to optimize the weight of DHNN to make it truly stable.

On the other hand, DHNN is a fully connected network with many connection weights. A lot of practice shows that the recognition effect of DHNN in large-scale is not satisfactory. Many scholars have optimized DHNN by deleting weights. Keddous et al. [10] proposed a replacement the FC of CNN layers with DHNN. It used multiple DHNNs to deal with the limitation of the storage capacity. Zhao et al. [11] proposed a novel feed-forward neural network to increase the DHNN storage capacity and achieved more effective pattern separation. Sahoo et al. [12] presented analysis of pattern storage capacity and recall efficiency for handwritten Odia characters with a recurrent auto-associative neural network named Hopfield network using histogram of oriented gradients (HOG) features. Brot et al. [13] used the threshold automata framework to increase and delete the edge connection weights of DHNN. Luo et al. [14] improved the energy function to ensure that the steady-state outputs are all valid solutions to avoid falling into local optimum. Shi et al. [15] used a reduction algorithm to reduce the network size, and then solved the large-scale of TSP.

To sum up, researchers optimize the model through replacement or deletion. But the connection weight stores the information of target. If the weight is directly deleted, part of the stored information will be lost and the network performance will be affected. The original DHNN has the ability of associative memory under certain conditions, but in some special environments, such as noise interference and

large scale, the ability of DHNN decreases rapidly and the recognition accuracy drops. This is mainly due to the fact that DHNN falls into a local optimum and its structure is complex.

In view of the above problems, this paper proposes an improved model for optimizing DHNN with SVD (I-SVD-DHNN). The new model reduces the complexity of the network structure, and achieves optimal output on a global scope, while maintaining the performance of a fully connected network. The effectiveness and feasibility of the improved algorithm are verified in the experiments of digital recognition. Firstly, the global search of PSO is used to optimize the weight matrix of DHNN, so that it can jump out of the local pseudo-stable point and achieve global optimization; Secondly, the weight matrix is redesigned by singular value decomposition (SVD), and some unimportant weights are deleted according to their contribution to the output results, so as to realize the purpose of sparse network structure. The flow chart of I-SVD-DHNN model is shown in Fig. 1.

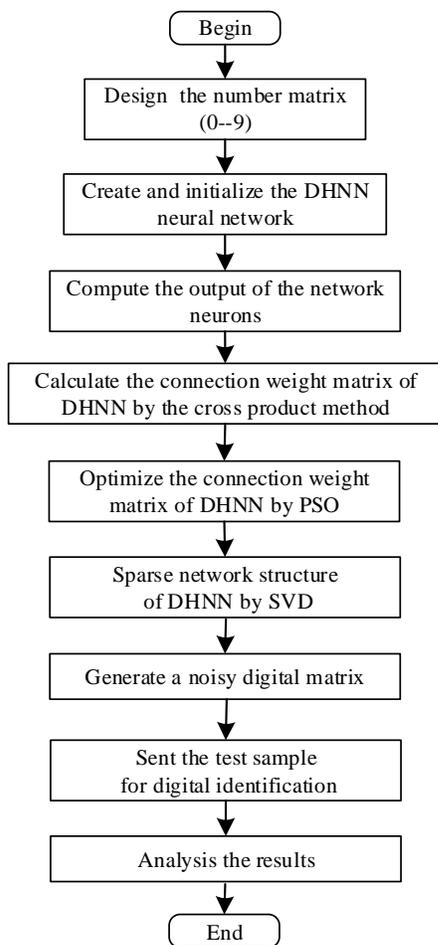


Fig. 1. Flow chart of I-SVD-DHNN.

II. METHODOLOGY

A. Introduction of DHNN

DHNN is a single-layer feedback network with binary output. The outputs of neurons are only 1 and -1, which represent the activation and inhibition states respectively [16]. Its network structure is shown in Fig. 2.

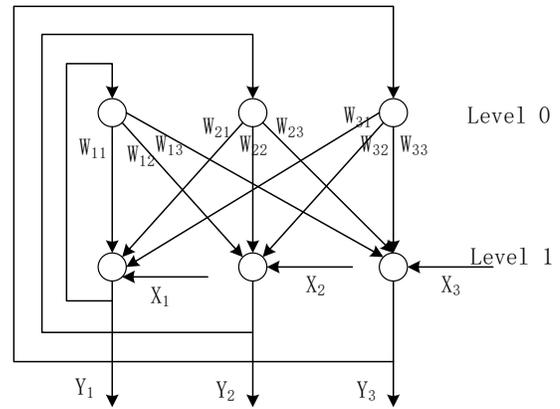


Fig. 2. Network structure of DHNN.

In Fig. 2, level 0 is used as input and there is no computation function. Level 1 is a neuron that performs the summation of the product of input and weight, and produces an output after processing by a nonlinear function (f). f is a simple threshold function, if the output of neuron is greater than threshold (θ), then the output is 1, otherwise it is -1. For binary neurons, its calculation formula is as follows:

$$u_j = \sum_i w_{ij} y_j + x_j \quad (1)$$

where x_j is the external input, w_{ij} is the weight matrix, and

$$\begin{cases} y_j = 1, & u_j \geq \theta_j \\ y_j = -1, & u_j < \theta_j \end{cases} \quad (2)$$

There are many design methods for w_{ij} . No matter which, it is necessary to ensure that the matrix is symmetric, which can make the network converge quickly and achieve stability. In this experiment, the outer product method is used to design w_{ij} . The outer product rule is as follows:

$$w = \sum_{k=1}^N [t^k (t^k)^T - I] \quad (3)$$

where, $\{t^1, t^2, t^3 \dots t^N\}$ is the sample vector to be memorized. Its value is 1 or -1. I is unit matrix.

For a network with n neurons in the output layer, the output at time t is an n -dimensional vector.

$$Y(t) = [y_1(t), y_2(t), y_3(t) \dots, \dots y_n(t)]^T \quad (4)$$

where $y_n(t)$ is the state of the n -th neuron at time t , with a value of 1 or -1. Accordingly, the state of node at the next moment ($t+1$) can be expressed as,

$$y_j(t+1) = f[u_j(t)] = \begin{cases} 1, & u_j(t) \geq 0 \\ -1, & u_j(t) < 0 \end{cases} \quad (5)$$

$$\text{where } u_j(t) = \sum_{i=1}^n w_{ij} y_j(t) + x_j - \theta_j \quad (6)$$

B. Optimize Network Weight

DHNN evolves in the direction of decreasing energy function. Since the energy function is bounded, the network must stabilize and output result. There will be many local minimum points in the energy function during operation, resulting in many pseudo-stable points in DHNN, and it is difficult to achieve true global stability. PSO is a better global

optimization algorithm. Using the global search ability of PSO, the associative memory of DHNN is strengthened to make the network jump out of the pseudo stable point and have higher performance.

When PSO is solving optimization problems, the solution of the problem corresponds to the position of particles in the search space. Each particle in the algorithm represents a potential optimal solution in the search space. The particle characteristics are represented by position, velocity and fitness. The position of each particle corresponds to a fitness value determined by the fitness function. The position determines the starting point of the algorithm; Speed determines the direction and distance of particle movement. By tracking the trajectory of particles in the solution space, the position and speed of particles are dynamically adjusted to achieve the output of the optimal solution.

The original weight of DHNN is taken as the global optimal particle. The appropriate fitness function is selected to calculate the fitness of particle, and the performance of particle is judged based on this [17]. During the optimization process, the particles will move towards the space with less fitness. By tracking the particles, the speed and position of particles are continuously updated, that is, the weight matrix of network is updated. The detailed optimization steps are as follows:

Step 1: Initialize parameters, including the weight matrix of DHNN, the fitness function, the number and dimension of particles, the maximum number of iterations, the learning factor, etc.

Step 2: Initialize the population, including the initial position, velocity and fitness of each particle, as well as the individual extreme point (*pbest*) and the global extreme point (*gbest*).

Step 3: For each particle (*i*), its fitness is calculated according to the fitness function.

```

If (fitness(xi) < fitness(pbesti))
{
fitness(pbesti) = fitness(xi);
pbesti = xi;
}
    
```

```

If (fitness(xi) < fitness(gbesti))
{
fitness(gbesti) = fitness(xi);
gbesti = xi;
}
    
```

Step 4: Determine whether the maximum number of iterations is met. If so, end the program. Otherwise, update the particle's velocity and position.

Step 5: The results obtained in the previous step are taken as the updated weight matrix of DHNN.

The Flow chart of algorithm is shown in Fig. 3:

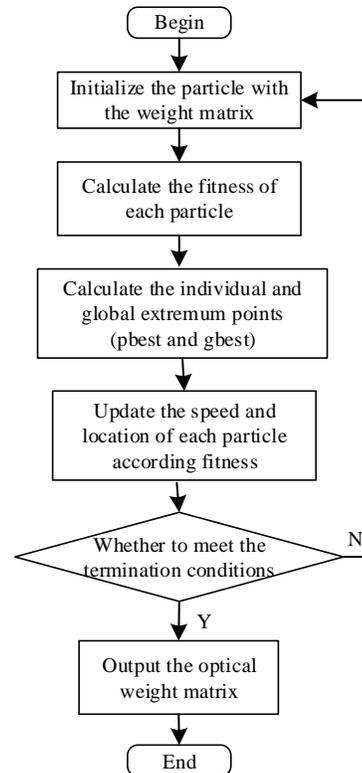


Fig. 3. Flow chart of PSO.

C. Sparse Network Structure

DHNN with *n* neurons has *n*² weights. Even optimizing by PSO, the number of weights remains unchanged. If the number of neurons increases, the weight of DHNN will increase sharply, which is very unfavorable for building a large-scale DHNN. SVD can describe the important characteristics of complex matrices with a small matrix [18]. Therefore, the weight matrix is decomposed by SVD, and some unimportant weights are discarded according to the contribution of weights to the output result. In this way, the weights are trimmed to simplify the network structure of DHNN.

Suppose that there are *n* target states (*t*₁, *t*₂, *t*₃ *t*_{*n*}), which are stored in the weight matrix. Convert it into a identifiable discrete value and save it in the matrix of DHNN (*M*), then *M* contains all the target states, namely

$$M = [t_1^T, t_2^T, t_3^T \dots \dots t_n^T] \quad (7)$$

The important information in *M* is extracted by SVD, that is

$$M = PVQ^T \quad (8)$$

where, *P* and *Q* are orthogonal matrices, and *V* is pseudo-diagonal. The vector in *P* is the singular vector of *M*, which retains the important information of weight matrix. Let's define a diagonal matrix.

$$D = diag[d_1, d_2, d_3 \dots \dots d_n] \quad (9)$$

where, d_i is the set eigenvalue. In order to ensure the symmetry of weights of the fully connected network DHNN, a new weight matrix is designed as,

$$W_{new} = PDP^T \quad (10)$$

W_{new} is obtained by SVD. The target information stored in each weight is different, and the contribution to output is also different. For each neuron, the weight with small contribution is deleted. C is defined as the contribution rate. The contribution rate of the i -th neuron is expressed as followed.

$$C_i = |W_{new-ij}| / \sum_{j=1}^n |W_{new-ij}| \quad (11)$$

The threshold of contribution rate is set to h , which is usually a constant, and $0 \leq h < 1$. If the contribution rate of the i -th neuron satisfies $C_i \leq h$, delete the corresponding weight. Otherwise, keep it. That is

$$W_{new-ij} = \begin{cases} 0 & (\text{delete}), C_i \leq h \\ W_{new-ij}(\text{retain}), C_i > h \end{cases} \quad (12)$$

The weight is trimmed by setting different h . The larger the h is, the more weights are discarded, and the greater the network structure changes. Choosing an appropriate h can ensure that the network structure is sparse while retaining the performance of a fully connected network [19]. The flow chart of sparse matrix is shown in Fig. 4.

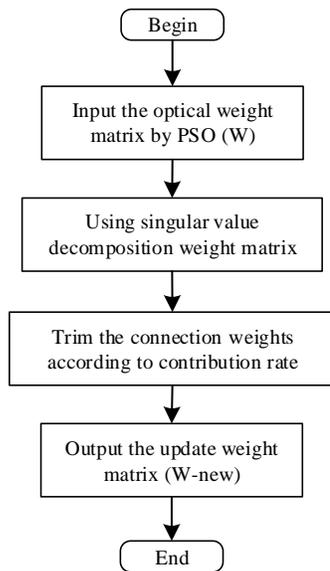


Fig. 4. Flow chart of sparse matrix.

III. EXPERIMENTAL DISCUSSION

A. Original Experimental Data Description

MATLAB is a platform containing a large number of computing algorithms, which can easily realize various computing functions required by users. At present, it has been widely used in data analysis, deep learning, image processing and other fields. According to the requirements, on the platform of MATLAB R2016a DHNN with associative memory is designed, which can correctly identify a total of 10

numbers from 0 to 9. When the numbers are disturbed by a certain noise, it still has a good recognition effect.

It is assumed that the input of DHNN consists of 10 steady states from 0 to 9. Each steady state is represented by a 10*10 matrix, which visually simulates Arabic numerals. In other words, each number is divided into a 10*10 matrix, the part with numbers is represented by 1, and the blank part is represented by -1. DHNN has the function of associative memory for the matrix of 10 digits. When the digital lattice with noise is input to the network, the closest target vector (10 steady-state) can be obtained, so as to achieve the effect of correct recognition. The matrix of numbers 1-2 is shown in Fig. 5, and so on for others.

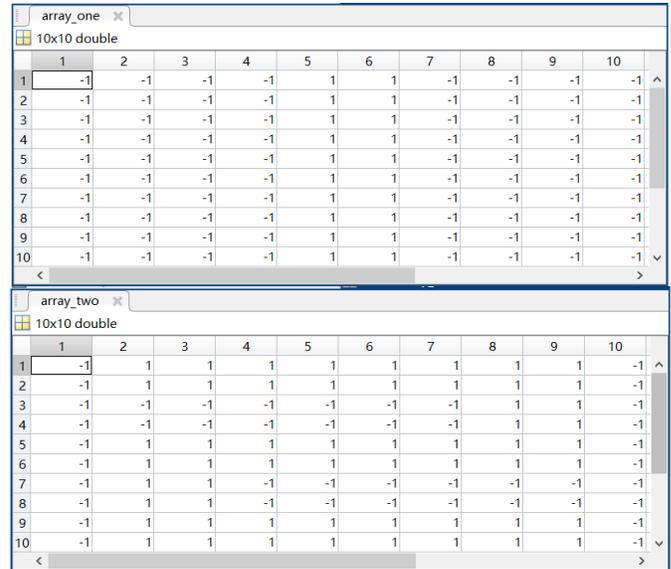


Fig. 5. Digital dot matrix of 1-2.

B. Noise Experimental Data Description

A noisy digital lattice is that some positions of lattice have changed. We employ random noise for this purpose. A random function is used to determine the position that needs to be modified, and then the digital lattice is noised. That is, according to the position determined by the random function, the original 1 becomes -1 and -1 becomes 1.

C. Digit Recognition of the Original DHNN

Suppose that DHNN memorizes a steady state of 10 numbers from 0 to 9. The steady state of standard number is used as training sample, and the digital lattice with adding noise is used as testing sample. The following two experiments are designed. In the first case, only 2 numbers are input, and the output of DHNN is observed as the noise intensity increases. In the second case, when the input increases to 4 numbers, how will the output change with the noise increases.

From the structure of DHNN, it is known that its stable state is the minimum of energy function during the training process. In the process of number recognition, the curve of energy function changing with the iteration is shown in Fig. 6.

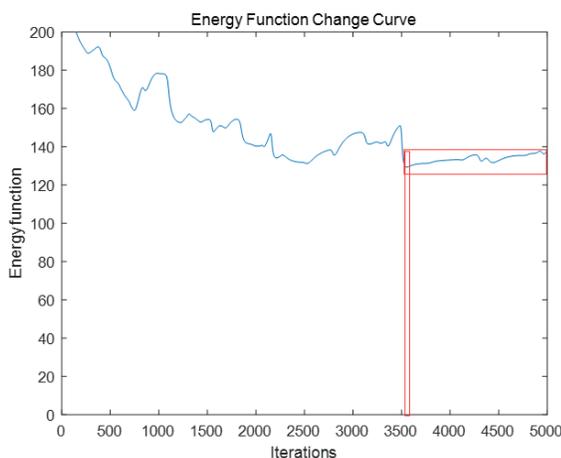


Fig. 6. Energy function change curve (DHNN).

Fig. 6 shows that the energy of DHNN decreases continuously with the iterative process. Until the iteration reaches 3500 times, the energy of DHNN changes very little (the red circled part in the figure). It indicates that the network tends to be in a stable state at this time.

In the first case, the lattices of numbers 1 and 2 are fed into DHNN as training samples. In the absence of noise (the noise is so small that it can be ignored), the network can accurately identify standard digital lattices. Subsequently, 10% (that is, 10% of the positions in 10*10 matrix have changed) and 30% of the noise intensity are added to the test sample and sent to DHNN again. The recognition result is shown in Fig. 7.

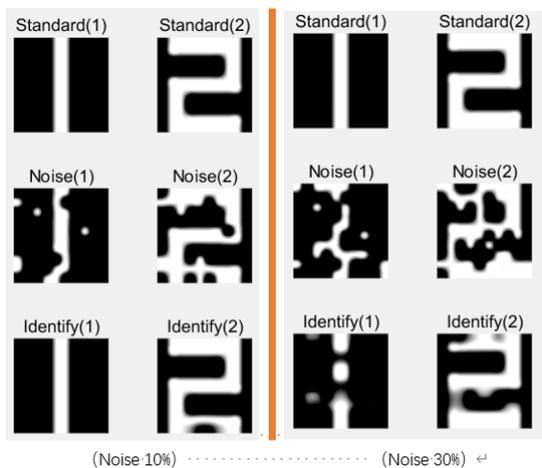


Fig. 7. Recognition effect with 2 numbers (DHNN).

Fig. 7 shows that the recognition clarity of DHNN is reduced after adding 10% noise, but it can be accurately identified, and the recognition effect of network is good. When the noise intensity increases to 30%, the recognition clarity of DHNN is further reduced compared to the previous one, but the numbers can still be distinguished, and the recognition effect of network is not very good. It shows that DHNN has the function of associative memory, which can correctly identify the digital lattice with certain noise.

In the second case, the number of input samples has been increased from 2 to 4. In the absence of noise, the recognition effect of DHNN is equivalent to having only 2 inputs. The accuracy reaches 100%. Subsequently, the digital lattice as input is replaced with a noise matrix with 10% and 30%. The recognition effect of DHNN is shown in Fig. 8 and Fig. 9.

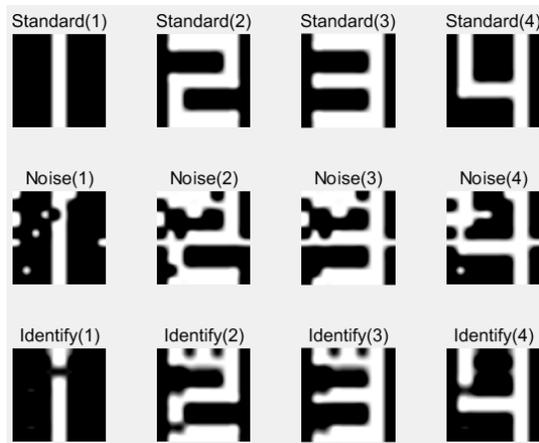


Fig. 8. Recognition effect adding noise 10% with 4 numbers (DHNN).

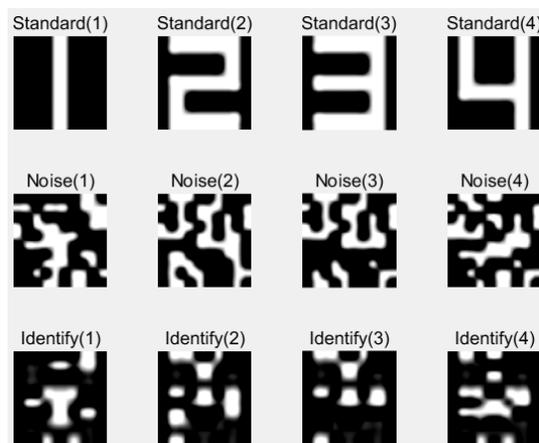


Fig. 9. Recognition effect adding noise 30% with 4 numbers (DHNN).

The output results show that when the input increase to 4 digits, the recognition performance of DHNN is significantly worse than 2 digits. When the noise intensity is 10%, Fig. 8 can clearly recognize the output of network, but compared to the Fig. 7, the recognition clarity of DHNN has dropped. When the noise intensity continues to increase, the associative memory ability of network further deteriorates, the error rate increases, and the recognition ability plummets. When the noise intensity reaches 30%, Fig. 9 shows that DHNN has difficulty in identifying 4 digits. The output of network is completely unrecognizable, and the recognition effect is extremely poor. In the case of 2 digits, although the recognition clarity is not good, the result can be roughly identified. Continue the experiment and found that when the input is 10 numbers from 0 to 9, even if the noise intensity is 10%, DHNN also fails.

From these two experiments, it can be seen that the associative memory ability of DHNN decreases with the increase of input scale. At the same time, the recognition

performance of network is inversely proportional to the noise intensity. The reason for this result is that there are many pseudo-stable points in DHNN, and it is difficult for the network to achieve a real steady state. Therefore, in order to maintain good performance of DHNN under high noise intensity, it is necessary to optimize the network structure.

D. Digit Recognition of the Improved DHNN

In order to jump out the pseudo-stable point, we optimize the structure of DHNN in two stages to make the network have better anti-interference. In the first stage, the global search ability of PSO is used to optimize the weight of DHNN to obtain weights with higher accuracy. In the second stage, SVD is used to redesign the weight matrix of DHNN to realize the sparseness of weights. Finally, the I-SVD-DHNN model is obtained.

Firstly, the weight of DHNN is optimized by PSO. In the experiment, the parameters of PSO are set as follows.

Inertia weight (W): it enables particles to maintain the inertia of motion and continue to expand the exploration space. In this experiment, $W=0.25$.

Acceleration coefficients C_1 and C_2 : it represents the weight of acceleration term each particle towards the pole position. In this experiment, $C_1=C_2=0.01$.

Number of Particles: In general, 20-40 can be selected. In this experiment, 20 particles are selected.

Secondly, the weight of DHNN is sparsed by SVD. For the weights optimized by PSO in the previous stage, SVD is used to extract their important features to form a new weight matrix. According to the contribution rate (C) of weights to the network output, it is decided whether to retain the corresponding weight. If $C>h$, keep the weight, otherwise delete it. In this way, a set of simple and sparse network weights can be obtained, which can ensure the network converging to a steady state. In this experiment, $h=0.15$.

In order to test the performance changes before and after the improvement, similar to Part C, the I-SVD-DHNN is first trained by using the standard digital lattice, and then different input and noise intensities are sent as test samples to observe the recognition effect of model. The contrast experiment is designed for two situations. Firstly, there are only 2 numbers. Secondly, the input is increased to 4 numbers.

Before starting the classification experiment, let's compare the stability of network. During the training process, the energy value of I-SVD-DHNN is also shrinking to the minimum. Its trend is shown in Fig. 10.

Fig. 10 shows that I-SVD-DHNN converges rapidly in the initial stage, and the energy value drops rapidly. After a period of small fluctuations, it tends to the minimum. It will reach a stable state after about 900 iterations (the red circled part in the figure). Compared with Fig. 6, the improved model I-SVD-DHNN has a rapid improvement in performance.

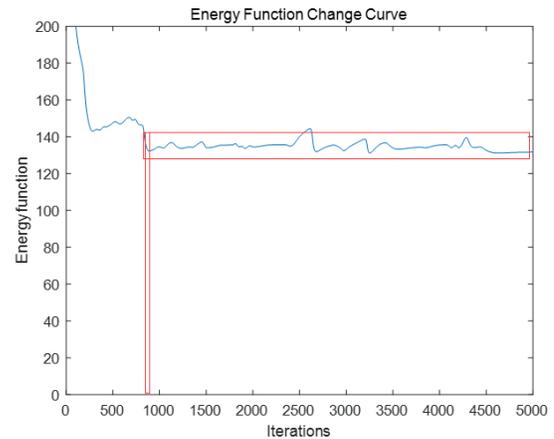


Fig. 10. Energy function change curve (I-SVD-DHNN).

1) The standard numbers 1 and 2 are sent into the I-SVD-DHNN. Subsequently, 10% and 30% of noise intensity are added for testing. The output of network is shown in Fig. 11.

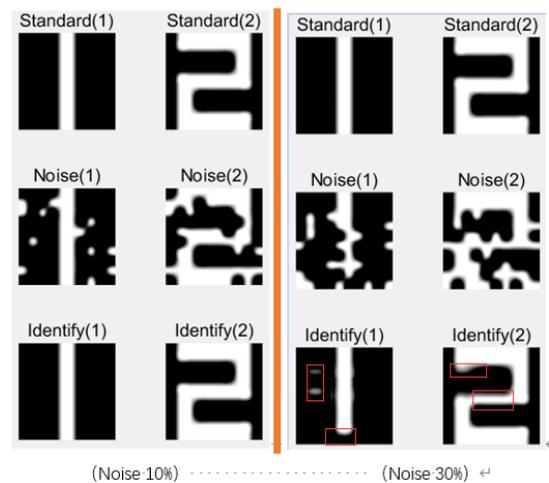


Fig. 11. Recognition effect with 2 numbers (I-SVD-DHNN).

Fig. 11 shows that after adding 10% of the noise, the network can accurately identify, and the recognition effect is close to the standard digital lattice, and the recognition rate reaches 100%. When the noise intensity is increased to 30%, the recognition performance of network decreases slightly. There are some borders with burrs (the red circled part in the figure), but the recognition effect is second only to the standard lattice. That is, the burrs have no effect on the output results. Compared with Fig. 7, the recognition performance of I-SVD-DHNN improves with the increase of noise intensity.

2) In I-SVD-DHNN, the number of training samples is increased to 4 digits, and the test sample are also added with 10% and 30% of noise intensity. The output of network is shown in Fig. 12 and 13.

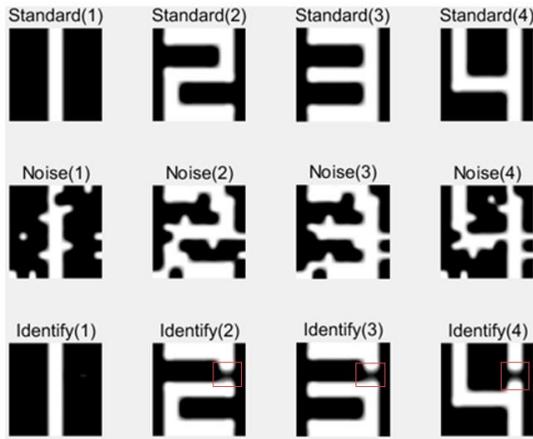


Fig. 12. Recognition effect adding noise 10% with 4 numbers (I-SVD-DHNN).

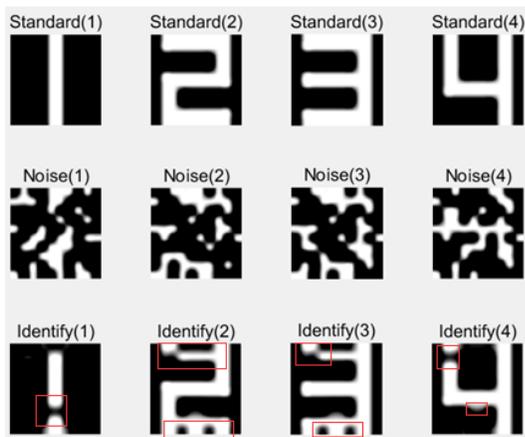


Fig. 13. Recognition effect adding noise 30% with 4 numbers (I-SVD-DHNN).

Fig. 12 and Fig. 13 show that the recognition performance of I-SVD-DHNN is significantly better than DHNN, when the input is 4 digits. When the noise intensity is 10%, the effect of Fig. 12 is slightly inferior to that of Fig. 11. Although it can be accurately identified, the number 2, 3, and 4 have a small missing (the red circled part in the figure), which does not achieve the effect of standard digital lattice. It shows that the recognition rate of I-SVD-DHNN is indeed related to the input scale of network. On the other hand, the effect of Fig. 12 is much better than that of Fig. 8 for the same input and noise intensity. When the noise intensity increases to 30%, Fig. 9 has completely failed while Fig. 13 still identified, but the accuracy of Fig. 13 has decreased. Compared with Fig. 12, some missing segments are added to the Fig. 13(the red circled part in the figure), which reduces the clarity of result, but it does not affect the output.

In order to further verify the effectiveness and practicability of I-SVD-DHNN, different thresholds are set to observe the difference of network recognition effects. Experiments are performed at 0.05 intervals. As t increases, more and more weights are removed. The recognition performance of network starts to decline. The deletion rate of weight and accuracy rate of result corresponding to different t are shown in Table I.

TABLE I. ACCURACY VS THRESHOLD

threshold	deletion rate of weight(%)	accuracy rate of result (10% noise)	accuracy rate of result (30% noise)
0	0	100	100
0.05	5	100	100
0.1	12	100	90
0.15	22	100	80
0.2	56	85	60
0.25	70	50	0
0.3	70	0	0

Table I shows that the recognition effect is good when $t=0$, which is equivalent to retaining all the optimized weights by PSO. When $0 \leq t \leq 0.15$, some of the network weights are removed, but it does not affect the recognition effect. When the noise intensity is added to 10%, the recognition rate reaches 100%. When the noise is increased to 30%, the recognition rate of 80% can also be achieved, indicating that some insignificant weights are deleted and the purpose of sparing network structure is achieved. When $t=0.2$, 56% of the weights are removed, but numbers with a noise intensity of 10% can still be identified. Relatively speaking, the recognition rate with a noise intensity of 30% drops further to 60%, and it is impossible to recognize all the test samples. When $t=0.25$, 70% of the weights are deleted, resulting in a large loss of information stored in the weights, only about half of the numbers can be recognized. If you continue to increase t , it is found that the deletion rate of weights remains the same. It means that the remaining weights cannot be deleted further. The network is in failure and the sample cannot be identified. The experimental results show that it is the best state to select 0.15 for the t , which can ensure the accuracy of digit recognition while achieving the sparse network structure.

To sum up, the recognition performance of I-SVD-DHNN is enhanced with the increase of input scale and noise intensity. The overall identification ability of network is improved and the anti-interference ability is strengthened. The conclusion is just consistent with the results described in Fig. 6 and 10. Both of them verify that the performance of I-SVD-DHNN is better than DHNN.

IV. CONCLUSION

As an important part of character recognition, digital recognition is widely used in transportation, post, business and other fields. Based on the original DHNN, this paper proposes an improved I-SVD-DHNN network model. The new model not only continues the associative memory ability of DHNN, but also improves its versatility and anti-interference. Firstly, it optimizes the weights of network by using the global search capability of PSO. The particle information with the minimum fitness is found as the optimized weight matrix through multiple iterations to avoid the network falling into local optimization. Secondly, it readjusts the weight matrix with the help of SVD and stores the information of target into it. Whether to retain or not is determined according to the contribution of each weight to output. By removing some insignificant weights, the purpose of sparse weight matrix is

achieved. Finally, the validity and applicability of new model are verified by means of digital recognition. Therefore, the following conclusions are drawn.

1) Optimize the weight matrix by PSO. It makes full use of the global optimization ability of PSO, so that the model tends to be stable in a short time, converges quickly, and obtains a weight matrix with higher accuracy to ensure the effectiveness of I-SVD-DHNN.

2) Sparse the weight matrix by SVD. It can effectively optimize the weight and greatly reduce the complexity of network structure, so that the optimized I-SVD-DHNN with sparse matrix has better recognition performance and it is more suitable for large-scale recognition scenarios.

3) The new method I-SVD-DHNN proposed in this paper has proved its feasibility and accuracy through experiments based on 0-9 digital samples. In the future, we will consider adding character samples, or even a mixture of the two, to further test the performance of the model.

ACKNOWLEDGMENT

This work is supported by the Foundation of State Key Laboratory of Public Big Data (No.PBD2022-22). Guizhou Minzu University Foundation Research Support Project (GZMUZK[2022]YB07 and GZMUJG202302). Natural Science Research Project of Department of Education of Guizhou Province (Grant No. QJJ2022047 and QJJ2022015).

REFERENCES

- [1] Jena K K , Bhoi S K , Nayak S R ,et al. Deep Convolutional Network Based Machine Intelligence Model for Satellite Cloud Image Classification[J]. Big Data Mining and Analytics, 2023, 6(1):12.
- [2] Belyaev M A , Velichko A A . Classification of handwritten digits using the Hopfield network[J]. IOP Conference Series Materials Science and Engineering, 2020, 862:052048.
- [3] Zhou X , Huang J , Lu F , et al. HNN-Based Generalized Predictive Control for Turbofan Engine Direct Performance Optimization[J]. Aerospace Science and Technology, 2021, 112(3):106602.
- [4] Lu J , Wu Z , Zhang X , et al. Quantitatively Evaluating the Effect of Read Noise in Memristive Hopfield Network on Solving Traveling Salesman Problem[J]. IEEE Electron Device Letters, 2020, 41(11):1688-1691.
- [5] Zhang L, Yang Y. Stability Analysis of Fractional Order Hopfield Neural Networks with Optimal Discontinuous Control[J]. Neural Processing Letters, 2019,50(1):581-593.
- [6] Feng L, Cao J, Liu L. Stability Analysis in a Class of Markov Switched Stochastic Hopfield Neural Networks[J]. Neural Processing Letters, 2019,50(1):413-430.
- [7] Wang J, Liu X, Bai J, et al. A New Stability Condition for Uncertain Fuzzy Hopfield Neural Networks with Time-varying Delays[J]. International Journal of Control, Automation and Systems, 2019,17(5):1322-1329.
- [8] Shi Z , Yu T , Zhao Q , et al. Comparison of algorithms for an electronic nose in identifying liquors. Journal of Bionic Engineering, 2008, 5(3):253-257.
- [9] Zhang Y , Li T , Na G , et al. Optimized Extreme Learning Machine for Power System Transient Stability Prediction Using Synchrophasors [J]. Mathematical Problems in Engineering, 2018, 23:1-8.
- [10] Keddous FE, Nakib A. Optimal CNN–Hopfield Network for Pattern Recognition Based on a Genetic Algorithm[J]. Algorithms. 2022; 15(11):1-18.
- [11] Zhao S , Chen B , Wang H , et al. A Feed-Forward Neural Network for Increasing the Hopfield-Network Storage Capacity[J]. International Journal of Neural Systems, 2022, 32(6): 2250027-2250043.
- [12] Sahoo R C , Pradhan S K . Pattern Storage and Recalling Analysis of Hopfield Network for Handwritten Odia Characters Using HOG[C]// ICMLCI. 2019.
- [13] Brot Hilla, Muchnik Lev, Goldenberg Jacob, et al. Feedback between node and network dynamics can produce real-world network properties[J]. Physica A, 2012,391(24):6645-6654.
- [14] Luo Y. Design and Improvement of Hopfield network for TSP[C]// International Conference on Artificial Intelligence. ACM, 2019.
- [15] Shi H , Rao Y. et al. A New Approach for Solving Medium Scale TSP Based on Hopfield Neural Network[J]. China Transportation Review, 2018,40(10):77-82.
- [16] Chen Y , Xue Y , Zhang X .A direct analysis method to state bounding for high-order quaternion Hopfield neural networks with mixed delays and bounded disturbances[J]. Mathematical Methods in the Applied Sciences, 2023,46(8):8646-8662.
- [17] Huang Y , Watanabe T , Bando M , et al. PSO algorithm parameter settings and optimal multi-rotor layout design[J]. International journal of sustainable aviation, 2022,8(2):116-135.
- [18] Liu X , Wu Y , Gao P , et al. Color image watermarking based on singular value decomposition and generalized regression neural network[J]. Multimedia Tools and Applications, 2022, 81(22):32073-32091.
- [19] Song F , Li L , Liu Y ,et al. Singular value decomposition based learning identification for linear time-varying systems: From recursion to iteration[J]. International Journal of Robust and Nonlinear Control, 2023, 33(12):6986-7003