Application of particle swarm optimization BP neural network algorithm in image compression

Hai-jun Wang*, Tao Jin
Department of Mathematics and Computer Engineering, Ordos Institute of Technology, Inner Mongolia Ordos, 017000, China
*Corresponding author’s e-mail: wanghajun11249@126.com

Abstract: In order to overcome the problems of weak global search ability, slow convergence speed and easy to fall into local minima in the process of image compression of BP neural network model, an image compression model based on particle swarm optimization algorithm and improved BP algorithm is proposed. In this model, a group of optimal approximate solutions of weights and thresholds of BP network are obtained through global search of particle swarm optimization according to objective function, and then the approximate solution is taken as the initial value of BP model, and the improved BP algorithm is used to conduct quadratic optimization training on these weights and thresholds to obtain the final image compression model. The experimental results show that with the same error accuracy, the quality of model compressed image reconstruction based on particle swarm BP neural network algorithm is significantly higher than that of BP and improved BP algorithm models.

1. Introduction
At present, image compression mainly adopts methods based on transformation technology, vector quantization technology and prediction method technology. These traditional methods are either computationally intensive or difficult to solve the compression function, so their practical application is limited [1]. Based on this, many scholars began to explore new methods, especially the neural network for image compression research is the most active. Among many neural network image compression algorithms, Back Propagation (BP) neural network is currently the most commonly used neural network model, but because BP neural network uses gradient descent algorithm, it has weak global search ability and convergence slow speed, easy to fall into local minima, etc. [2]. In order to overcome these problems, the improved LMBP algorithm is often used for modeling. However, since the improved algorithm is still based on the gradient descent algorithm, it cannot fundamentally solve the local minimum problem caused by the random selection of initial weights and thresholds of BP neural network. Particle swarm optimization (PSO) is a swarm intelligence algorithm proposed by Dr. Eberhart et al in 1995. The emergence of this algorithm provides an efficient solution to the optimization problem of neural network weights and thresholds [3,4]. Therefore, the PSO-BP hybrid image compression model combining particle swarm optimization algorithm and BP neural network algorithm is proposed. The experimental results show that the PSO-BP hybrid image compression model is more effective in image reconstruction than the BP algorithm and LMBP algorithm model.

2. Algorithm principle and calculation process
2.1 Principle of PSO Algorithm
PSO algorithm originated from the simulation of simple social system and was originally conceived to simulate a process of foraging of birds. This algorithm has a distinct biological and social background, and it considers the cognitive process of the whole and the individual at the same time. In the process of seeking consistent cognition, individuals not only remember their own experience, but also consider the experience of other partners: learn from their own experience, and learn from their peers. Through the research on the behavior of such similar biological groups, it is found that there exists a social information sharing mechanism in the biological groups, which is finally applied to the optimization problem, resulting in particle swarm optimization algorithm [5].

In PSO model, every solution of optimization problem is a "bird" in search space, which is called "particle". The PSO method first randomly initializes a group of particles in a feasible solution space, each of which is a feasible solution of the optimization problem, calculates the corresponding fitness value with an evaluation function to determine whether the optimization goal is reached, and then determines whether the optimization needs to continue iteratively. In each iteration, the particle updates itself by tracking two "extremums": one is the optimal solution found so far by the particle itself, called individual extremum point $p_{id}$; The other is the optimal solution found so far for the entire population, called global extremum $p_{gd}$. According to the principle of following the current optimal particle, the particle will change its speed and position according to equations (1) and (2) [6].

$$v_{id}^{(t+1)} = w v_{id}^{(t)} + c_1 r_1 (p_{id}^{(t)} - x_{id}^{(t)}) + c_2 r_2 (p_{gd}^{(t)} - x_{id}^{(t)})$$  \(1\)

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)}$$  \(2\)

In equations (1) and (2), $d$ and $i$ are cyclic traversals, $d = 1, 2, \cdots, n$, $i = 1, 2, \cdots, m$, $n$ is the dimensions of the search space, $m$ is the number of population particles, $t$ is the current evolutionary algebra, $w$ is the inertia weight, $r_1$ and $r_2$ are random Numbers distributed between [0,1], $c_1$ and $c_2$ are acceleration constants. In addition, in order to prevent the particle velocity from becoming too large, the upper limit of the velocity $V_{\text{max}}$ is set, that is, when $v_{id} > V_{\text{max}}$ in formula (2), $v_{id} = V_{\text{max}}$ is taken, and when $v_{id} < -V_{\text{max}}$, $v_{id} = -V_{\text{max}}$ is taken.

The pseudo-code for the execution of the PSO algorithm is shown below.

```pseudo
For each particle
  Initialize particle
End
Do
  For each particle
    Calculate fitness value
    If the fitness value is better than the best fitness value $p_{id}$ in history
      set current value as the new $p_{id}$
    End
  End
  Choose the particle with the best fitness value of all the particles as the $p_{gd}$
  For each particle
    Calculate particle velocity according equation (1)
    Update particle position according equation (2)
  End
While maximum iterations or minimum error criteria is not attained
```

2.2 The core idea of combining PSO algorithm and BP algorithm

The central idea of BP neural network optimized by PSO algorithm is to set the desired optimization variable (bird position) of PSO algorithm as the initial weights and thresholds of the neural network, and the position of each bird represents a set of weights and thresholds combinations of a neural network. The food concentration is the fitness function of BP neural network algorithm. The optimization process of the PSO algorithm is the process of adjusting the initial weights and thresholds of the BP neural
network. By using the excellent global optimization ability of the PSO algorithm, the initial weights and thresholds of the BP neural network can be adjusted to the global optimal value, which makes BP neural network no longer sets the initial weights and thresholds randomly, so as to avoid the network oscillation problem caused by randomly setting the initial weights and thresholds, making the model output more stable and more accurate. The core process of particle swarm optimization for BP neural network is shown in Fig.1.

![PSO-BP algorithm flow chart](image)

**Fig.1** PSO-BP algorithm flow chart.

### 2.3 The calculation process of PSO-BP algorithm

The calculation of the PSO-BP algorithm includes the following processes:

1. The parameter initialization of PSO-BP algorithm

   **Setp1**: Determine the size of the particle swarm—Randomly generate $m$ initial particle individuals, each of which is composed of a velocity matrix and a position matrix. For the dimension of the search space, suppose that the input layer of the BP model has $M$ nodes, the hidden layer has $N$ nodes, and the output layer has $P$ nodes, then the dimension of the search space is $n = (M + P) \times N + N + P$.

   **Setp2**: The setting of the inertia factor $w$—In order to keep the particles inertial, so that they have the tendency to expand the search space, and have the ability to explore new areas. This paper adopts the linear decreasing weight strategy [7] proposed by Shi and Eberhart in 1998. As shown in equation (3), it can make $w$ decrease linearly from $w_{\text{ini}}$ to $w_{t \text{end}}$ with iteration number:

   $$w(t) = (w_{\text{ini}} - w_{\text{end}}) \times \frac{T_{\text{max}} - t}{T_{\text{max}}} + w_{\text{end}}$$  \hspace{1cm} (3)

   In equations (3), $T_{\text{max}}$ is the maximum evolutionary algebra and $t$ is the current evolutionary algebra. $w_{\text{ini}}$ is the initial inertia weight, and $w_{\text{end}}$ is the inertia weight when iterating to the maximum algebra.

   **Setp3**: The setting of learning factors $c_1$ and $c_2$—The $c_1$ and $c_2$ represent the weights of the statistical acceleration terms that push each particle toward the $P_{\text{id}}$ and $P_{\text{gd}}$ positions. They are
parameters used to adjust the particle's own experience and the social group's experience in the entire optimization process. The $c_1$ and $c_2$ are usually fixed constants. Generally, $c_1$ and $c_2$ are defined to be equal and the value range is $[0,4]$.

(2) Determination of fitness function

Through the previous analysis, the particle swarm optimization algorithm has its own characteristics, so when determining the fitness function, it is slightly different from the genetic algorithm. The training mean square error accuracy $E$ is used as the fitness of the particles to promote the search of the population by the particles. The fitness function of the particles is shown in Equation (4).

$$\text{fitness} = E = \frac{1}{N} \sum_{i=1}^{N} (y_{i\text{(real)}} - y_{i\text{(ideal)}})^2$$ (4)

In equations (4), $N$ is the number of training samples, $y_{i\text{(real)}}$ is the ideal output value of the $i$-th sample, and $y_{i\text{(ideal)}}$ is the actual output value of the $i$-th sample. Therefore, the corresponding position of the particle with the lowest fitness value (least training error) when the algorithm iterates to stop is the optimal solution of the optimization problem.

(3) Update of particle velocity and position

**Step1:** Particle evaluation—According to the training error generated by each particle under the determined training samples of the BP network, the fitness is calculated by equation (4).

**Step2:** Extreme value update—compare the individual fitness value of the individual in the particle swarm with the individual fitness value before iteration. If the current value is better, make the current value replace the value before iteration and save the current position as its individual extreme value, otherwise its individual extreme value is the extreme value of the previous generation. For the global extreme value, if the current fitness value of a particle in the existing group is better than the global historical optimal fitness value, then the current fitness value of the particle is made the global optimal fitness value of the group, and save the current position of the particle as the global global extremum.

**Step3:** Speed update—Update the speed according to the $p_{id}$ and $gdp_{id}$ generated in Step2 iteration. In this paper, the speed update with additional terms [8] is used to perform the speed update. The formula is shown in (5), where $r_3$ is Random number between $[0,1]$.

$$v_{id}^{(t+1)} = w_{id}^{(t)} + c_1 r_1 (p_{id}^{(t)} - x_{id}^{(t)}) + c_2 r_2 (gdp_{id}^{(t)} - x_{id}^{(t)}) + r_3 (p_{id}^{(t)} - gdp_{id}^{(t)})$$ (5)

**Step4:** Update the solution—Update the solution with the speed generated by Step3 iteration, that is, adjust the weights and thresholds of BP neural network.

**Step5:** Judgment of the iteration stop—Evaluate the fitness of the new population generated by the iteration, and judge whether the algorithm reaches the expected error or the maximum number of iterations. If the conditions are met, stop the iteration, otherwise, return to Step2 to continue iteration.

(4) Generate the optimal solution

When the algorithm stops iterating, the value corresponding to $gdp$ is the optimal solution of the corresponding training problem, which is the initial optimal solution of the BP neural network model. The above optimal solution is substituted into the BP network model for secondary training and learning, and finally an image compression model is formed.

3. Sample structure and network structure determination

3.1 Sample structure

It is assumed that the image to be compressed is composed of $N \times N$ pixels, and the image to be compressed is divided into $M$ sub-image blocks, and each sub-image block is respectively composed of $p \times p$ sub-pixel blocks. During the training of BP network model, $M$ sub-image blocks were trained successively until all the training was completed. The method shown in formula (6) is a method of dividing a $128 \times 128$ pixel image into a $4 \times 4$ sub-pixel block image.
The sub-image block vector generated by the above method must also be normalized, where the mean distribution processing is adopted: set the gray scale range of the image to be processed as \([x_{\min}, x_{\max}]\), and the transformation field as \([y_{\min}, y_{\max}]\), and set the gray scale of the pixel to be processed as \(x_{\text{value}}\), then the calculation of the processed pixel point \(y_{\text{value}}\) is shown in formula (7). Figure 2 shows part of the sample data after normalization of part of the original image data of 128×128.

\[
y_{\text{value}} = \left( y_{\max} - y_{\min} \right) \left( x_{\text{value}} - x_{\min} \right) / \left( x_{\max} - x_{\min} \right) + x_{\min}
\]

\(\text{Fig.2 Part of the original image data normalized part of the sample data}\)

It can be seen from the figure above that after the conversion of the original data, the network input is reduced from 128 to 16, which greatly simplifies the network structure and reduces the network complexity, thus providing feasibility and convenience for the BP network to effectively compress the image.

### 3.2 Determination of BP Network Structure

In this process, the transformation of the input layer and the hidden layer can be regarded as the image compression coding process; and the transformation of the hidden layer and the output layer can be regarded as the process of decoding and reconstructing the image [9]. Therefore, the number of nodes in the input layer and the output layer of the BP image compression model is equal, and \(M = P = 16\). In image compression, the compression ratio \(K\) used by BP neural network for image coding is related to the number of input layer nodes \(M\) and the number of hidden layer nodes \(N\). The specific calculation formula is shown in (8) [10].

\[
K = M/N
\]

Therefore, for the BP neural network image compression model with different values of \(N\), different compression ratios of the same image can be achieved. After determining the number of layers, the number of layers and the weights and thresholds parameters of the compression model, it is necessary to determine the activation functions of each layer. In this study, according to the characteristics of the input and output data of the compression model, the tansig function and purein function are used in the hidden layer and the output layer respectively.

### 4. Experimental Analysis

In the experiment, the parameters of BP, LMBP and PSO-BP are set to be the same in the BP segment. Figure 3 shows the structure of the BP neural network model used when the compression ratio \(K = 2\) and \(N = 8\). In order to compare the effects of the reconstructed image after compression, in this study, the structural similarity (SSIM) function and the peak signal-to-noise ratio (PSNR) function were used to evaluate the reconstructed image. The larger the SSIM and PSNR values, the closer the reconstructed image output by the model is to the original image, the better the reconstruction effect. Table 1 shows the comparison of the reconstructed images using the three algorithms at a compression ratio of \(K = 2\). From the data in the table, it can be seen that although the image compression model based on the LMBP algorithm has a better image reconstruction effect, the model is optimized by the PSO algorithm. The reconstruction effect has been further improved. In order to further verify the effectiveness and
adaptability of the algorithm, the model with the compression ratio \( K = 4 \) is further verified. The reconstruction results given in Table 2 further prove that the PSO-BP algorithm is superior to the BP and LMBP algorithms in compressed image reconstruction.

![The BP model structure used in the compression model](image)

**Table.1** Image evaluation of compression model reconstruction with different algorithms when \( K = 2 \)

| Algorithm | Lena PSNR | Lena SSIM | Cameraman PSNR | Cameraman SSIM | Brain PSNR | Brain SSIM |
|-----------|-----------|-----------|----------------|----------------|------------|------------|
| BP        | 10.1538   | 0.0423    | 9.3761         | 0.0356         | 7.0835     | 0.0243     |
| LMBP      | 28.059    | 0.8635    | 25.0034        | 0.7565         | 22.7934    | 0.7541     |
| PSO-BP    | 28.4746   | 0.8751    | 26.5992        | 0.7888         | 24.4064    | 0.8281     |

**Table.2** Image evaluation of compression model reconstruction with different algorithms when \( K = 4 \)

| Algorithm | Lena PSNR | Lena SSIM | Cameraman PSNR | Cameraman SSIM | Brain PSNR | Brain SSIM |
|-----------|-----------|-----------|----------------|----------------|------------|------------|
| BP        | 9.8599    | 0.0373    | 9.1789         | 0.0261         | 7.1521     | 0.0208     |
| LMBP      | 27.2878   | 0.8627    | 24.5302        | 0.7685         | 22.4939    | 0.711      |
| PSO-BP    | 27.5424   | 0.8718    | 25.1512        | 0.8222         | 23.3526    | 0.7866     |

5. Conclusion

In this paper, a PSO-BP hybrid neural network algorithm combining particle swarm algorithm and BP algorithm is applied to image compression application research. The particle swarm algorithm is used to guide the selection of the initial weight threshold of BP neural network, and then the improved BP algorithm is used for secondary training. The experimental results show that compared with the image compression model based on the BP and LMBP algorithms, the model based on the PSO-BP algorithm has a more scientific weight threshold selection, which can effectively prevent the BP model from falling into the local minimum. Therefore, it is based on the PSO-BP The image compression model of the hybrid algorithm is significantly better in compression efficiency, and the reconstruction quality of the compressed decoded image is better. At the same time, the research in this paper can further broaden the application field of particle swarm neural network model and provide a new research idea for image compression.

Acknowledgments

The authors are grateful to the scientific research projects of Inner Mongolia higher education institutions support under grant No. NJZY19260 and College-level Project of Ordos Institute of Technology support under grant No. KYYB2018011.

Reference

[1] Hongmei Y., Dongmei W (2013). Application of Improved BP Neural Network in Hyper-spectral Image Compression. Journal of Graphics, vol.34,iss.5,pp:110-114.
[2] Li G M, He Z Q(2011). Embedded Image Compression Algorithm and FPGA Implementation Based on BP Neural Networks. Applied Mechanics & Materials, iss.65,pp:415-418.
[3] Y Tong-guang, G Wei-hua(2015). Stator resistance identification for induction motor based on particle swarm optimization neural network observer. Electric Machines & Control,
vol.19,iss.2,pp:89-95

[4] W Ganggang, L Qing, X Yurui(2016). Hologram Compression Using BP Neural Network by Modified Particle Swarm Optimization. Journal of Jilin University, vol.34,iss.1,pp:147-152

[5] Coello C A C, Pulido G T, Lechuga M S(2004). Handling multiple objectives with particle swarm optimization. IEEE Transactions on Evolutionary Computation, vol.8,iss.3,pp:256-279.

[6] Dong-Feng W, Li M, Wen-Jie Z(2016). Improved Bare Bones Particle Swarm Optimization with Adaptive Search Center. Chinese Journal of Computers, vol.39,iss.12,pp:2652-2667

[7] GAO Hai-bing, GAO Liang, ZHOU Chi(2004). Particle Swarm Optimization Based Algorithm for Neural Network Learning. Acta Electronica Sinica, vol.32,iss.9,pp:1572-1574.

[8] Hong-Zhen C, Liang H U, Wei-Wei G(2007). Modified particle swarm optimization based on random factor. Computer Engineering and Design, vol.28,iss.5,pp:2245-2247

[9] Shi-Fang W, Li-Guo L(2007). Application of a BP neural network based on genetic algorithm in image compression. Chinese Journal of Quantum Electronics, vol.24,iss.4,pp:425-428.

[10] Xiaoliang X, Guanglin Y, Xiaohui D(2008). Image Compression of BP Neural Network with SPIHT Algorithm. Journal of Electronic Measurement and Instrument, vol.22,iss.6,pp:7-11.