Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 14, No. 3, 2025

Optimizing BP Neural Network Image Restoration Based on an Improved Particle Swarm Optimization Algorithm

Ke Fang *, Jianhong Li

Sichuan Vocational and Technical College, Suining Sichuan, 629000, China

*** Corresponding author:** Ke Fang (Email: 799335999@qq.com)

Abstract: To address the shortcomings of BP neural networks, such as sensitivity to initial weights and thresholds and susceptibility to local optima, this paper proposes a particle swarm optimization (PSO) algorithm to optimize the BP neural network image restoration model. By searching for the initial weights and thresholds of the BP neural network, the dependence on these parameters is eliminated. The particle swarm optimization algorithm is modified, and the improved algorithm is used to search for the initial weights and thresholds of the BP neural network. The optimized network is then applied to restore the Cameraman image. Experimental results demonstrate that for the Cameraman image, the PSO-BP neural network achieves a peak signal-to-noise ratio (PSNR) improvement of 18.07% and 1.25%, respectively, compared to the standard BP neural network. Additionally, the structural similarity (SSIM) improves by 4.18% and 0.95%, respectively. These findings indicate that the optimized BP neural network delivers superior image restoration performance.

Keywords: Image Restoration; Intelligent Optimization Algorithm; BP Neural Network.

1. Introduction

Images serve as a vital source of information, yet their quality often degrades during generation and transmission due to factors such as blurring and distortion. High-quality images find extensive applications in fields like aerospace and industry, making image restoration technology of significant practical value.[1] Traditional restoration algorithms (e.g., Wiener filtering, L-R restoration) typically require known degradation models or point spread functions (PSFs), yet acquiring such prior knowledge proves challenging in real-world scenarios. Backpropagation (BP) neural networks possess adaptive and self-learning capabilities. They eliminate the need for matrix inversion, thereby avoiding the ringing artifacts inherent in traditional methods and yielding clearer restoration results. However, conventional BP networks are sensitive to random initial weights and thresholds, leading to high iteration counts, slow convergence, and susceptibility to local optima.[2]

In 2019, Xue HZ et al. proposed an image restoration method based on BP neural networks [2], establishing a BP-based image restoration model. Experimental results demonstrated improved restoration quality compared to traditional approaches. Nevertheless, this approach also exhibits certain limitations.[3] The BP neural network is highly sensitive to initial weights and threshold values, which are randomly assigned. This sensitivity can cause excessive iterations during training, making convergence difficult or even impossible. Furthermore, inconsistent training results increase error rates and may lead to getting stuck in local optima. Therefore, an intelligent optimization algorithm is employed to refine the initial weights and threshold values of the BP neural network. The optimized parameters are then assigned to the network for training, with the final trained network performing image restoration. [4] This approach resolves numerous challenges that traditional methods struggle with or cannot address, while also tackling issues such as slow convergence, susceptibility to local optima, and high error rates in BP neural networks. It offers new directions and perspectives for image restoration methods, facilitating the widespread application of image restoration across various fields.[5]

2. Particle Swarm Optimization Algorithm

The Particle Swarm Optimization (PSO) algorithm is an intelligent algorithm that mimics the behavior of a predator-prey population. First proposed by Kennedy and Eberhart in 1995, it originated from research on avian predation.[6] This algorithm operates as a collective population of numerous entities, exhibiting highly complex behavioral patterns. Each particle embodies all dimensions of the solution to the optimization function. Through defined position formulas, the particles' locations are continuously updated to determine optimal positions, thereby achieving a global optimum solution. The particle swarm algorithm offers advantages such as ease of implementation, minimal parameter configuration, and rapid convergence, yielding excellent results across numerous fields.[7]

2.1. Principle of the Particle Swarm Optimization Algorithm

Assume a swarm of m particles exists within a D-dimensional target search space. The position of the i-th particle is $x_{id} = \left(x_{i1}, x_{i2}, ..., x_{iD}\right)$, and its velocity is $v_{id} = \left(v_{i1}, v_{i2}, ..., v_{iD}\right)$ .Within the existing space D, the particle swarm will continuously update its positions to search for the local minima of all individuals. [8] The optimal position of a single particle is denoted as $pBest_{id} = \left(pBest_{i1}, pBest_{i2}, ..., pBest_{iD}\right)$ .Throughout the process, particle velocities are continuously updated based on learning factors, inertia weights, and individual optimal positions. The updated particles then identify the global optimum position through ongoing comparisons. Global Optimal Position $gBest=\left(gBest_1, gBest_2, ..., gBest_D\right)$,Among them $i=1,2,...,m$ .The speed and position update formula is as follows:

$$v_{id}\left(t+1\right)=v_{id}\left(t\right)+c_1r_1\left(pBest_{id}-x_{id}\left(t\right)\right)+c_2r_2\left(gBest-x_{id}\left(t\right)\right)$$

$$x_{id}\left(t+1\right)=x_{id}\left(t\right)+v_{id}\left(t+1\right)$$

In the above formula, t represents the number of iterations, which c_1、c_2 is a constant, and denotes the learning factor, $c_1=c_2=2$, r_1、r_2 is a random number between [0,1], $pBest_{id}$ representing the current individual's optimal position, and $gBest_{id}$ represents the overall optimal position, i.e., the global optimum solution. The search area must not exceed the designated spatial boundaries $v_{id}\in\left[-v_{\max}, v_{\max}\right]$, $x_{id}\in\left[-x_{\max}, x_{\max}\right]$ .At that time $v_{id}>v_{\max}$, $v_{id}=v_{\max}$,At that time, $v_{id}<v_{\min}$, $v_{id}=v_{\min}$ .The maximum particle velocity is determined by the size of the search space.If the range is between $\left[-x_{\max}, x_{\max}\right]$,So $v_{\max}=k\times x_{\max}$, $k\in\left[0.1,0.2\right]$ .When the optimization result meets the predefined requirements or reaches the maximum number of iterations, the algorithm search terminates.[9]

To further enhance the search capability of particle swarm optimization, two primary improvement methods are currently employed.

(1) Particle Swarm Optimization with Inertia Factor

The improved particle swarm algorithm velocity update formula is as follows:

$$v_{id}\left(t+1\right)=wv_{id}\left(t\right)+c_1r_1\left(pBest_{id}-x_{id}\left(t\right)\right)+c_2r_2\left(gBest-x_{id}\left(t\right)\right)$$

$$x_{id}\left(t+1\right)=x_{id}\left(t\right)+v_{id}\left(t+1\right)$$

In the equation, w represents the inertia factor. Its function is to influence the current velocity by adjusting the rate of change from the previous phase. During the global search phase, the corresponding inertia factor is relatively large; during the local search phase, the inertia factor is relatively small. This demonstrates that the inertia factor can regulate the algorithm's search capability. During the initial search phase, a larger inertia factor is typically set to enhance overall search capability. As the number of search iterations decreases, the inertia factor diminishes, thereby strengthening local search capability. To improve particle search speed and convergence accuracy, the inertia weight formula is defined as:

$$w=w_{\max}-\frac{w_{\max}-w_{\min}}{T_{\max}}*t$$

(2) Particle Swarm Optimization with Increased Contraction Factor

Clerc proposed incorporating a scaling factor into traditional algorithms. This approach accelerates convergence by imposing constraints through the scaling factor. The speed formula is as follows:

$$v_{id}\left(t+1\right)=\chi\left(v_{id}\left(t\right)+c_1r_1\left(pBest_{id}-x_{id}\left(t\right)\right)+c_2r_2\left(gBest-x_{id}\left(t\right)\right)\right)$$

$$\chi=\frac{2}{\left|2\text{-}\varphi\text{-}\sqrt{\varphi^2-4\phi}\right|}$$

2.2. Basic Process of Particle Swarm Optimization for BP Neural Networks

(1) First, set the parameters of the neural network and optimize the particle dimensions. For the BP neural network, the number of weights and thresholds to be optimized is: $H=\left(n+1\right)\times m+\left(m+1\right)\times s$, where n denotes the number of input layer nodes, s denotes the number of output layer nodes, and m denotes the number of hidden layer nodes.

(2) Set the learning factor c_1、c_2 value, maximum iteration count $T_{\max}$, dimension D, and population size M, $pBest_{id}$、$gBest_{id}$ Randomly initialize particle velocity and position within the specified range.

(3) Calculate the fitness function. This is computed based on both the global optimum $gBest_{id}$ and local optima $pBest_{id}$.

(4) Update the particle's velocity and position according to the velocity and position formula.

(5) Does the output signal result meet the preset value? If it meets the requirement, terminate the update and proceed to Step 6. If it does not meet the requirement, proceed to Step 3.

(6) The algorithm terminates, outputting the optimal weights and thresholds, which are then fed into the backpropagation neural network.

The PSO-BP flowchart is shown in the figure.

3. PSO-BP Image Restoration Experimental Results and Analysis

Similarly, we first apply a 9×9 Gaussian blur with variance 1 to the image. Since we are using a BP neural network with a 9:20:1 structure, the dimension D is 221, meaning there are 221 weight and threshold parameters. The maximum iteration count T for the particle swarm is set to 200.Learning Factor c1=2, c2=1.8, vmax=0.5, wmax=0.9, wmin=0.3, Group M=40. Simulation results are shown in the figure

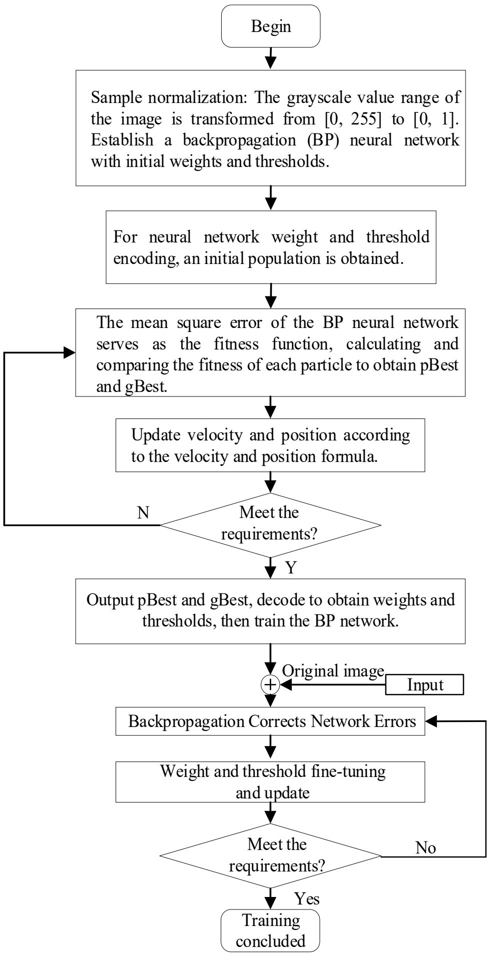
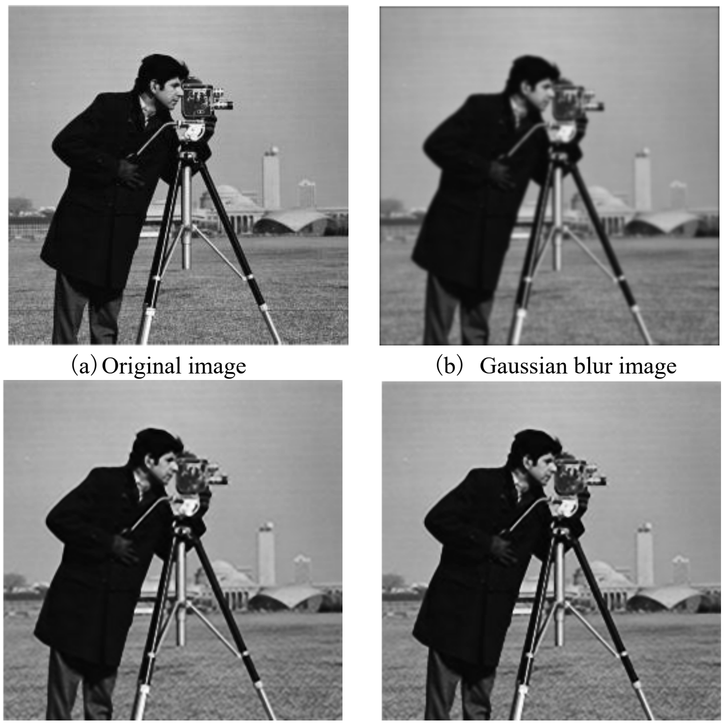


Fig 1. PSO-BP flow chart



(a) Original image (b) Gaussian blur image

(c) BP Recovery (d) PSO-BP Restoration

Fig 2. Cameraman reconstruction of PSO-BP neural network

Table 1. PSNR and SSIM of Cameraman restored by PSO-BP

Restoration Method	PSNR (Db)	SSIM
Wiener filter	20.0993	0.7747
BP Neural Network	30.4985	0.9429
PSO-BP	31.5182	0.9532

As shown in the figure, the PSO-BP image restoration method outperforms the other two approaches, producing clearer reconstructed images. Table 3.4 demonstrates that PSO-BP image restoration achieves a 56.81% and 3.34% improvement in Peak Signal-to-Noise Ratio (PSNR) and a 23.04% and 1.09% improvement in Structural Similarity (SSIM), respectively, compared to the other two methods. This confirms that the PSO-BP neural network delivers superior image restoration performance, producing clearer reconstructed images.

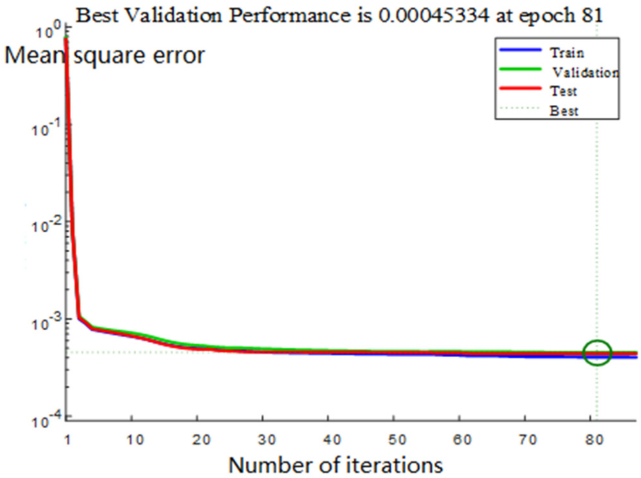


Fig 3. PSO-BP training convergence graph

As shown in the figure above, the particle swarm optimization-enhanced backpropagation neural network (PSO-BP) converged after 81 iterations, whereas the standard backpropagation neural network required 137 iterations to converge. The optimized BP network reduced the number of iterations needed for image restoration, thereby improving efficiency.

4. Conclusion

To address the drawbacks of BP neural networks, such as slow convergence and susceptibility to local optima, this chapter provides a detailed introduction to the particle swarm optimization (PSO) algorithm. By employing an improved PSO algorithm to optimize the initial weights and thresholds of the BP neural network, we resolve issues including sensitivity to initial parameters, slow convergence, and the tendency to get stuck in local optima. First, the particle swarm algorithm is enhanced by incorporating a contraction factor and an inertia factor. These modifications accelerate convergence speed and improve convergence accuracy. Finally, the improved particle swarm algorithm is applied to optimize the weights and thresholds of the BP neural network for image restoration. Experimental results demonstrate that the ICS-BP image restoration achieves higher Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) than both the BP neural network and the PSO-BP neural network. This indicates that ICS-BP delivers superior restoration performance, producing restored images that more closely resemble the original images while converging more rapidly.

References

[1] Huo Y K, Wei G, Zhang Y D, et al. An adaptive threshold for the Canny operator of edge detection[C] proceedings of the 2010 International Conference on Image. Analysis and Signal Processing Piscataway: IEEE, 2010:371-374.

[2] Liu C J, Ding W F, et al. Prediction of highspeed grinding temperature of titanium matrix composites using BP neural network based on PSO algorithm[J]. International Journal of Advanced Manufacturing Technology, 2017, 89(5-8): 2277-2285.

[3] Jiang Yi. Distributed Power Source Planning Based on Adaptive Mutation Particle Swarm Algorithm [J]. Transactions of the Chinese Society for Electric Power Engineering, 2012, 27(3):194-198.

[4] Meng Yazhou, Ma Yu, Bai Bing, et al. Otsu's Lung Tissue Segmentation Algorithm Based on Particle Swarm Optimization. Liquid Crystal and Display, 2015(6):1000-1007.

[5] Wang Yuanping. Research on Energy Consumption Balancing in Linear Wireless Sensor Networks [D]. Jiangsu: China University of Mining and Technology, 2016.

[6] Li, D. P., Chen, G. X., & Li, K. L. A usability-based particle swarm task scheduling algorithm for heterogeneous systems. Computer Engineering and Science, 2012, 34(6): 44-49.

[7] P Barthelemy, J Bertolotti. A Lvy flight for light. Nature, 2008, 453(7194): 495-498.

[8] M F Shlesing. Mathematical physics: Search research Nature, 2006, 443(7109): 281-282.

[9] Huang Y Q, Zhang J, Li X, et al. Thermal error modeling by integrating GA and BP algorithms for the highspeed spindle[J]. The International Journal of dvanced Manufacturing Technology, 2014, 71(9-12): 1669-1675.