Current Journal of Applied Science and Technology

Volume 42, Issue 6, Page 10-18, 2023; Article no.CJAST.97654
ISSN: 2457-1024
(Past name: British Journal of Applied Science & Technology, Past ISSN: 2231-0843, NLM ID: 101664541)

A Hybrid Method of Backpropagation and Particle Swarm Optimization for Enhancing Accuracy Performance

I. Made Widiartha [a], Anak Agung Ngurah Gunawan [b*], E. R. Ngurah Agus Sanjaya [a] and Kartika Sari [c]

[a] Departement of Informatics, Faculty of Mathematics and Natural Sciences, Udayana University, Indonesia.
[b] Departement of Physics, Faculty of Mathematics and Natural Sciences, Udayana University, Indonesia.
[c] Departement of Mathematics, Faculty of Mathematics and Natural Sciences, Udayana University, Indonesia.

Authors' contributions

This work was carried out in collaboration among all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/CJAST/2023/v42i64072

Open Peer Review History:
This journal follows the Advanced Open Peer Review policy. Identity of the Reviewers, Editor(s) and additional Reviewers, peer review comments, different versions of the manuscript, comments of the editors, etc are available here:
https://www.sdiarticle5.com/review-history/97654

Original Research Article

Received: 09/01/2023
Accepted: 18/03/2023
Published: 22/03/2023

ABSTRACT

Aims: Backpropagation is an algorithm for adjusting the weight of neural networks in the training stage. The performance of backpropagation has proven superior in optimizing the weight of neural networks; however, this method needs improvement in the initiation stage, where the random process creates local optimal solution. Applying an algorithm based on the global search is an alternative to solve the drawback of backpropagation. One global search method with superior performance is particle swarm optimization. In this research, we apply the hybridization of

**Corresponding author: E-mail: a.a.ngurahgunawan@unud.ac.id;*

backpropagation and particle swarm optimization (BP-PSO) to overcome the problem of backpropagation.
Study Design: Research Papers and Short Notes.
Place and Duration of Study: Department of Informatics, Faculty of Mathematics and Natural Sciences, Udayana University, between June 2022 and November 2022.
Methodology: The dataset used in this study is a handwriting image dataset of the mathematical symbol. There are 240 symbols consisting of 180 images for training and 60 for testing. The robustness of the PSO method in obtaining the optimum global solution is expected to help backpropagation out of local optimal solutions. The application of PSO is carried out at the initial weight initialization stage of the artificial neural network. The tuning parameters of the artificial neural network are the number of neurons in the hidden layer and the value of the learning rate. There are three combinations in the number of neurons in the hidden layer, namely 10, 20, and 30. Meanwhile, the learning rate values are five different combinations, namely 0.1 to 0.9, the minimum error value is 0.01, and the maximum number of epochs is 1000. We carry out five repetitions in each test scenario.
Results: The performance results showed that PSO has succeeded in optimizing backpropagation, where the accuracy of the BP-PSO is higher than BP without optimization. The accuracy of BP-PSO is 97.2%, while the BP is 94.4%. The optimal learning rate value and the optimal number of hidden layers are 0.1 and 30 neurons, respectively.
Conclusion: The performance results showed that PSO has succeeded in optimizing backpropagation, where the accuracy of the BP-PSO is higher than BP without optimization. The optimization process of weighting the artificial neural network as the initial weight for later retraining shows a higher average accuracy, while decreasing the average number of epochs does not optimize initial weight.

Keywords: Neural network; backpropagation; particle swarm optimization; classification; swarm intelligent.

1. INTRODUCTION

One of the methods in artificial intelligence that is inspired by the biological workings of the human neural system is the artificial neural network (ANN) method. ANN is developed as a generalization of mathematical models from the human or neurological understanding, just like humans whose brains always learn from the environment. They do that to manage the environment adequately based on the gained experiences. ANN learning is performed iteratively as the network is presented with training examples, similar to how we learn from experience [1]. ANN has superior performance when compared to other methods [2]. This method requires a training process to be able to recognize class data. The training process is a stage of renewing the weight of the neural network through a series of learning iterations where the results of this training are the best weight values in recognizing patterns. Researchers have made various methods to optimize the weight value of this artificial neural network. One method that has superior performance is backpropagation [3]. The performance of backpropagation has been proven to have outstanding performance in optimizing the weight of the neural network. Although this algorithm is better than some others, it also has a weakness where the resulting results can be local optimal values [4]. This weakness is influenced by the initial weight, which is randomly selected [5]. In some cases, backpropagation will experience poor performance compared to other method schemes. To solve this problem, we can insert a global search-based algorithm into backpropagation to find optimal initiation values of backpropagation weight.

There are various methods based on the search for global solutions, including genetic algorithms and multiple methods in intelligent swarms. If we compare the performance of genetic algorithms and swarm algorithms such as ant colony, bee colony, memetic, shuffled frog leap, and particle swarm optimization (PSO), the results show that the PSO algorithm has the most superior performance compared to these other algorithms [6]. In integer programming problems, PSO has been compared with the Genetic algorithm. The comparison results show that the PSO algorithm is superior in terms of complexity, accuracy, iteration, and program simplicity in finding the optimal solution [7]. Seeing the excellent performance of this PSO, this research applies the PSO method to optimize the backpropagation method to get out of the local optimal solution. We evaluated the performance comparison between the backpropagation method with and without PSO at the testing stage. We also measured the effectiveness of the PSO in optimizing the weight of the training results.

2. METHODOLOGY

2.1 Backpropagation

Backpropagation is an algorithm for training artificial neural networks to obtain the optimal network weights. This set of weights determines the network's ability to recognize hidden patterns. Backpropagation uses an error output to readjust the weights in the backward direction. This study used a multi-layer backpropagation neural network scheme, as shown in Fig. 1 [8].

The followings are the steps of the Backpropagation algorithm.

1. Initialize the weights in the network with small random numbers.
2. A pattern of data training is fed into the network input layers.
3. Calculate all output unit in the hidden and output layers.

$$z\_net_j = v_{j0} + \sum_{i=1}^{n} x_i v_{ji} \quad (1)$$

$$z_j = f\left(z_{net_j}\right) = \frac{1}{1+e^{-z\_net_j}} \quad (2)$$

4. Calculate the unit error (δ) for every output layer and hidden layer. δ is the error unit that will be used in updating the weight. t_k is the target output, Δw_{kj} is the weight different, and α is the learning rate.

$$\delta_k = (t_k - y_k) y_k (1 - y_k) \quad (3)$$

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (4)$$

5. Modify all the weights.

$$w_{kj}(new) = w_{kj}(old) + \Delta w_{kj} \quad (5)$$

The subsequent pattern from the training set is chosen, and step number 2 will be carried out until all patterns are processed. The algorithm ends when the error is less than the minimum criteria.

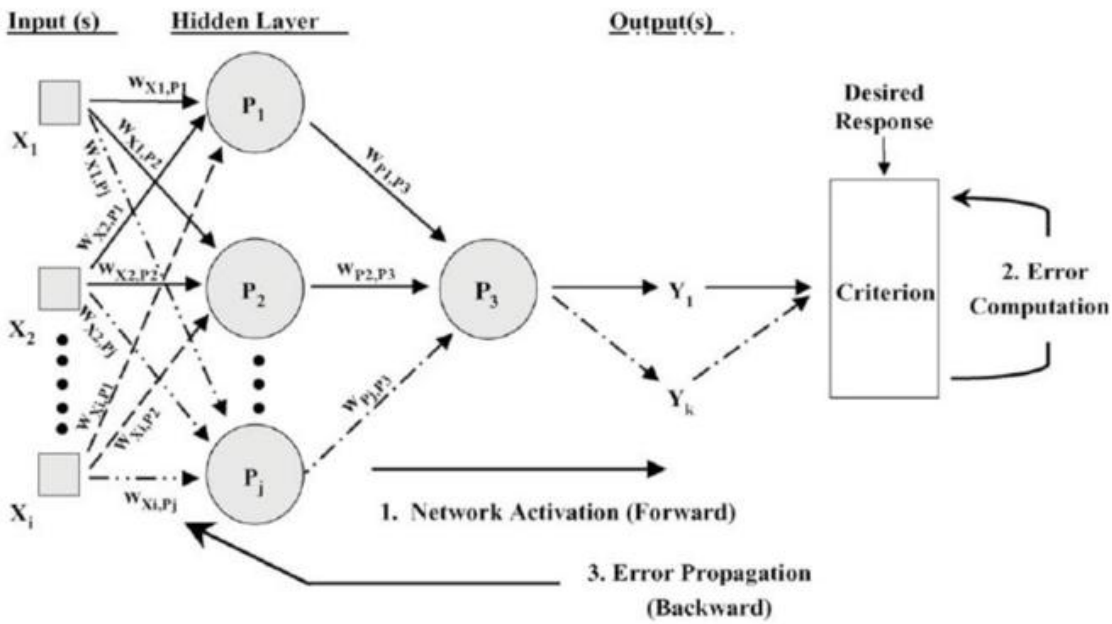

Fig. 1. Back-propagation network

2.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based stochastic optimization algorithm motivated by the intelligent collective behavior of animals such as flocks of birds or schools of fish [9]. PSO, as an optimization tool, provides a population search-based procedure. PSO performs searching via a swarm of particles that updates from iteration to iteration. Each particle moves in the direction of its previous best (pBest) position and the global best (gBest) position in the swarm [10] to obtain the optimal solution. The following are the stages of the particle swarm optimization algorithm:

1. Initialize particles with random speed and velocity.
2. Evaluate the fitness value of each particle. Then the comparison process between particles and Pbest is carried out. If the particle fitness value is better than the Pbest value, the Pbest value is converted into the particle fitness value.
3. Compare the Gbest value against the best value for all particles. If the fitness value obtained is better than the Gbest value, the Gbest value is converted into the particle fitness value.
4. Perform the process of changing the speed and rank of particles using formulas (6) and (7).
5. Repeat step 2 until it meets stopping criteria such as best fitness score or maximum iteration.

Some of the popular terms in PSO are as follows:

- Swarm: population/herd of an algorithm.
- Particles: individuals in a swarm.
- pBest (Personal best): the best rank of a particle.
- gBest (Global best): the best position of the particle on a swarm.
- Velocity (v): a vector that determines the movement of particles as they move.
- Inertia weight (w): parameters controlling the speed of a particle.
- Learning speed (c_1 and c_2): constants for particles (c_1) and swarm (c_2).

In making changes to velocity, PSO has three parts, namely the social part, cognitive part and momentum part. These three parts are used in the velocity tracking process. The following is the PSO equation with the inertia weight [11].

$$v_{ij}(t+1) = w * v_{ij}(\mathrm{t}) + c_1 * \mathrm{rand}() * \left(p_{ij}(t) - x_{ij}(t)\right) + c_2 * rand() * \left(g_{ij} - x_{ij}(t)\right) \tag{6}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t) \tag{7}$$

Description :

v	=	Particel Velocity
x	=	Particel Position
w	=	Inertia Weight
c1	=	pBest Constanta
c2	=	gBest Constanta
p	=	pBest
g	=	gBest
rand()	=	Random Number

2.3 Hybridization Scheme

The backpropagation learning algorithm's major drawback is local minimal problems [12]. The robustness of the PSO method in obtaining the optimum global solution is expected to help backpropagation out of the local optimal issues. The application of PSO is carried out at the initial weight initialization stage of the artificial neural network. The impact of random weight initialization can be seen in previous studies [13].

3. RESULTS AND DISCUSSION

3.1 Hybridization

The dataset used in this study to measure the accuracy of PSO in optimizing backpropagation is a mathematical symbol image. There are 20 math symbols written by hand and using the image equation in the Microsoft Word Application. For each symbol, we have 12 sample images. The total number of data is 240, and we divided it into two parts; 70% as training data and 30% as the test data. Before this dataset enters the pattern recognition stage with the backpropagation and PSO methods, we first preprocessed the data. The preprocessing steps are as follows:

1) Grayscaling: In this process, the RGB image is converted into a grayscale or gray image. The red, green, and blue component values will be taken for each image pixel. The method used for grayscaling is the weighted method. This method will produce grayscale values by adding up 30% from red, 59% from green, and 11% from the blue.
2) Binarization: We convert the grayscale image into a black-and-white or binary image using the thresholding method. The thresholding method uses a value to check

the possibility of whether the gray color of the image pixels is converted to black or white. The threshold value used is 127 because it is in the middle of 0 to 255.

3) Segmentation: This binarization process uses a gray image where the color value of each pixel will be taken and compared with the threshold value. If the color value is greater than the threshold value (> 127), then it will be changed to 255 (white) and 0 (black color) otherwise. The result of this binary process is a binary image that only consists of white and black.
4) Normalization: The image resulting from the segmentation process has different sizes, so it is necessary to normalize the image size. There are two stages in the normalization process, namely the padding and scaling stages. In the padding stage, values will be added to the sides of an image that is not square. At the scaling stage, the image size is changed to 64x64 pixels. First, we insert the image height and width value and the image to be normalized. We then seek the largest value between the height and width of the image. If the height value exceeds the width, the left and right values will be added according to the difference between the height and the width divided by 2. If the width value exceeds the height, the top and bottom values will be added according to the difference between width and height divided by 2. Each side of the image will be added according to the top, bottom, left, and right values so that the image will have the same size. This stage is called the padding stage. We convert the image size to 64x64 pixels in the scaling stage. The image produced from this normalization process is a 64x64 pixels image.

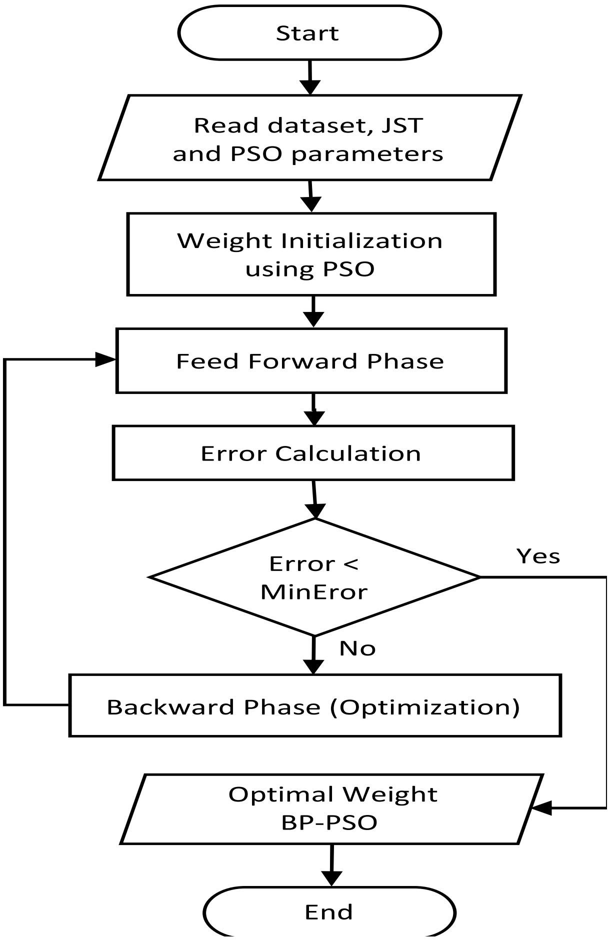

Fig. 2. Hybridization scheme

After completing the four stages, we carry on with the feature extraction stage. This is done to get the feature values for each dataset used in the pattern recognition stage. We extract features from the image obtained from the preprocessing step in the feature extraction stage. The method used to perform the feature extraction process is a gradient histogram. This feature extraction method is based on the slope or direction of the pixels in relation to the surrounding pixels. The result of this feature extraction is a vector which is a characteristic of the image.

3.2 Training and Testing Result

This study's first result of system testing is the best artificial neural network architecture for recognizing mathematical symbols. The parameters of the artificial neural network used are the number of neurons in the hidden layer and the learning rate value. There are three combinations in the number of neurons in the hidden layer, namely 10, 20, and 30. Meanwhile, the learning rate values are five different combinations, namely 0.1 to 0.9. The minimum error value is 0.01, and the maximum number of epochs is 1000. In each different test scenario, we implemented five tests.

The second system test result is the optimization effect of the initial weight value of the neural network using the particle swarm optimization algorithm. The results of this study will compare the accuracy value of the backpropagation method neural network with a random initial weight (ANN BP) and the backpropagation neural network with the initial weight generated using PSO.

We investigated the system's accuracy in recognizing mathematical symbols in the testing stage. This study will make several changes to the learning rate to determine the effect on the resulting accuracy value. The learning rate value will be used to find the program's accuracy in recognizing mathematical symbols. The value of the accuracy rate is calculated using equation (8).

$$P(N) = \frac{IN}{N} * 100\% \qquad (8)$$

Description:

P (N) = level of accuracy
IN = The number of successfully recognized data
N = Total number of data

Tables 1-3 are the test results with Hidden Neurons 10, 20, and 30. Figs. 3-5 present the results of the test scenario in graphical form. From the results of the scenario testing that has been carried out, the optimal results of the backpropagation weight parameter are in the learning rate value. Fig. 3 shows the changes in the performance of the backpropagation neural network with ten hidden neurons. The highest accuracy value of 92.8% is achieved when the learning rate is 0.9 for backpropagation with PSO, compared to 88.9% when the learning rate is 0.91 for backpropagation without PSO optimization. We can also see from the figure that in almost all tested learning rate values, BP PSO performance always has superior accuracy rates compared to the backpropagation algorithm without PSO optimization. This clearly shows the effectiveness of PSO in optimizing the backpropagation neural network.

Table 1. Test results with 10 hidden neurons

Learning Rate	**Epoch**		**Accuration**	
	BP	**BP PSO**	**BP**	**BP PSO**
0.1	1000	996	84.4	88.9
0.3	398.2	342.4	87.2	88.9
0.5	295.4	218.4	86.1	90.0
0.7	329.2	150.6	87.8	91.7
0.9	146	117.4	84.4	**92.8**
0.91	321.4	116.2	**88.9**	88.9
0.93	319.4	121.2	83.3	88.9
0.95	295.4	218.4	86.1	90.0
0.97	125.4	107.2	86.1	84.4
0.99	121.6	104	83.3	88.9

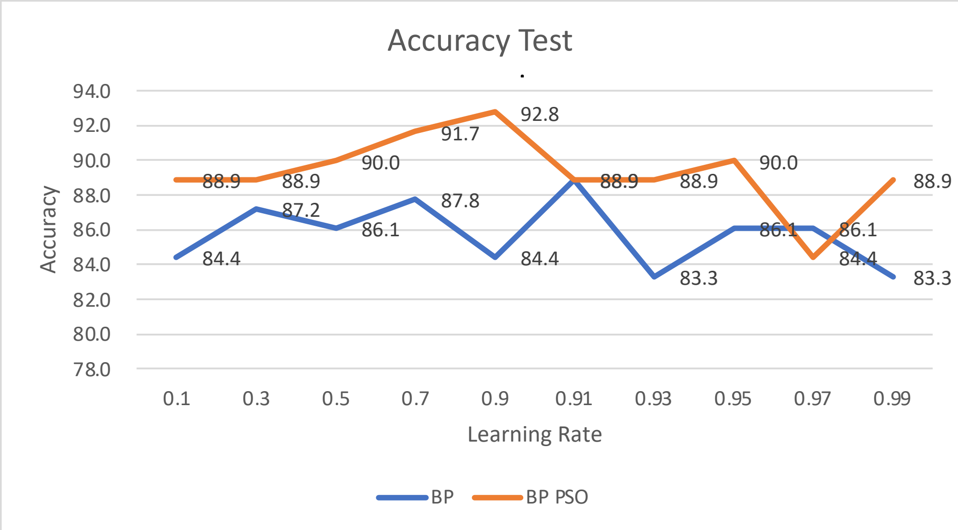

Fig. 3. PSO test results with 10 hidden neurons

Table 2. Test results with 20 hidden neurons

Learning Rate	Epoch		Accuration	
	BP	BP PSO	BP	BP PSO
0.1	564.4	560.8	93.3	95.0
0.3	202	187.2	93.3	95.6
0.5	159.6	116.2	91.7	93.9
0.7	86.2	82.8	91.1	95.6
0.9	70.2	64.6	94.4	96.1
0.91	70.8	64.2	89.4	95.0
0.93	77.6	63.4	90.6	95.6
0.95	188	112.4	84.4	90.6
0.97	64.2	61.6	91.1	94.4
0.99	259.6	59.2	92.2	94.4

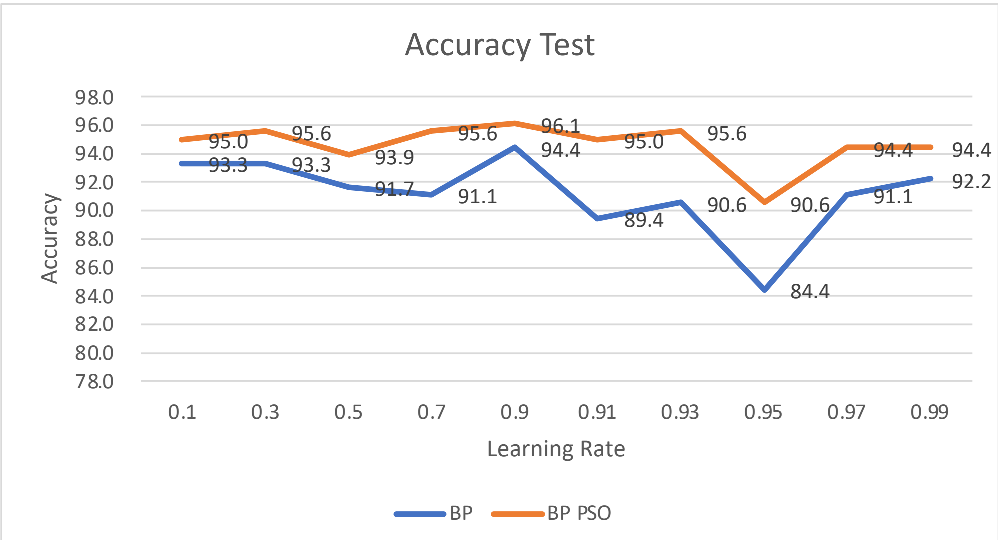

Fig. 4. PSO test results with 20 hidden neurons

Table 3. Test results with hidden neurons 30

Learning Rate	Epoch		Accuration	
	BP	BP PSO	BP	BP PSO
0.01	1000	1000	78.3	87.2
0.03	70.8	64.2	89.4	95.0
0.05	920.2	895.6	91.7	94.4
0.07	618.8	623.8	92.2	95.6
0.09	596.4	487.8	92.8	96.7
0.1	579.8	444	91.7	97.2
0.3	161.8	150.6	93.3	96.1
0.5	96.2	91.6	92.2	96.1
0.7	103.2	65.8	92.2	96.1
0.9	54.4	52.2	92.8	95.6

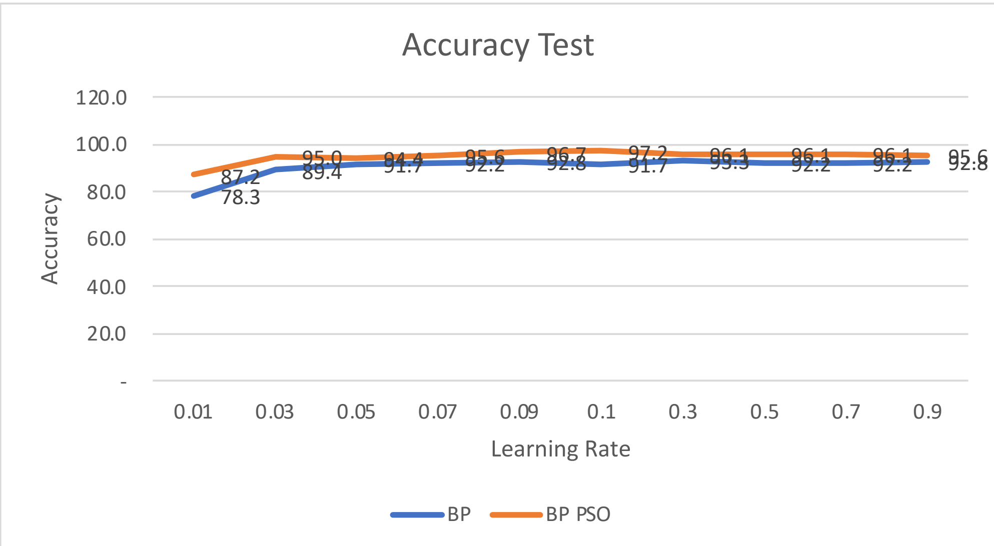

Fig. 5. PSO test results with 30 hidden neurons

4. CONCLUSION

From the results of trials and evaluations of the research that has been done, the following conclusions can be drawn.

1. We achieve the highest average accuracy of 94.4% with a 0.1 learning rate and 30 neurons in the hidden layer for neural network without optimization, and 97.2% otherwise.
2. Learning rate affects the neural network training process. A low learning rate of 0.1 will lead to a better average accuracy than a high learning rate of 0.9. However, a contrasting effect can be seen in the number of epochs. A more significant number of epochs will lead to a better average accuracy when we use the same number of neurons in the hidden layer.

ACKNOWLEDGEMENTS

We are extremely grateful to the Faculty of Mathematics and Natural Sciences, Udayana University for the funding of this study. This study was funded under the Study Program's Flagship research scheme. All the research activities were conducted at the Computing Laboratory at the Informatics Department, Faculty of Mathematics and Natural Sciences, Udayana University.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Basheera IA, Hajmeer M. Artificial neural networks: Fundamentals, computing, design, and application. Journal of Microbiological Methods. 2000;43(1):3-31.
2. Mahdiyah U, Irawan MI, Imah EM. Study comparison backpropagation, support vector machine, and extreme learning

machine for bioinformatics data. Journal of Computer Science and Information. 2015;8 (1):53-59.
3. Amalia OA, Putri AHM. Comparison of backpropagation artificial neural network and SARIMA in predicting the number of railway passengers. Journal of Physics: Conference Series. 2020;1663(2020):1-14.
4. Du W. Research on evaluation of enterprise performance based on bp neural network improved by levenberg-marquardt algorithm. Proceedings of the 2015 International Conference on Automation, Mechanical Control and Computational Engineering. 24-26 April 2015, Jinan, Shandong. 2015;169-173.
5. Lee A, Geem ZW, Suh KD. Determination of optimal initial weights of an artificial neural network by using harmony search algorithm. Applied Sciences. 2016;6(6): 1-17.
6. Garg A, Juneja D. A comparison and analysis of various extended techniques of query optimization. International Journal of Advancements in Technology. 2012;3(3): 184-194.
7. Wihartiko FD, Wijayanti H, Virgantari F. Performance comparison of genetic algorithms and particle swarm optimization for model integer programming bus timetabling problem. IOP Conference Series: Materials Science and Engineering. 12 October 2017, Tangerang, Indonesia. 2017;1-6.
8. Kaur J, Kaur P. A Review: Artificial neural network. International Journal of Current Engineering and Technology. 2018;8(4): 904-910.
9. Wang D, Tan D. Liu L. Particle swarm optimization algorithm: an overview. Springer-Verlag Berlin Heidelberg. 2018; 22(2):387-408.
10. Zhang Y, Balochian S, Agarwal P, Bhatnagar V, Housheya OJ. Artificial intelligence and its applications. Mathematical Problems in Engineering. 2016;2(6):1-12.
11. Kong F, Jiang J, Huang Y. An adaptive multi-swarm competition particle swarm optimizer for large-scale optimization. Mathematics. 2019;7(521):1-13.
12. Burse K, Manoria M, Kirar VPS. Improved back propagation algorithm to avoid local minima in multiplicative neuron model. International Journal of Electrical and Computer Engineering. 2010;4(12): 1776-1779.
13. Chang YT, Lin J, Shieh JS, Abbod MF. Optimization the initial weights of artificial neural networks via genetic algorithm applied to hip bone fracture prediction. Hindawi Publishing Corporation Advances in Fuzzy Systems. 2012; 1-9.

Peer-review history:
The peer review history for this paper can be accessed here:
https://www.sdiarticle5.com/review-history/97654