The Training and Application of RBF Neural Network Based on GWO
Xiao-qing ZHANG1,2,* and Zheng-feng MING1
1School of Mechano-Electronic Engineering, Xidian University, Xi’an 710000, China
2School of Physics and Electronic Engineering, Xianyang Normal University, Xianyang 712000, China
*Corresponding author

Keywords: Grey wolf optimizer, Neural network, Approximation, Chaotic synchronization.

Abstract. To optimize the hidden center matrix, Gaussian RMS width vector and the hidden-output weight matrix of RBF neural network, Grey Wolf Optimizer (GWO) and its several variants have been introduced. With the combination of the three parameters as the position vector of the grey wolf in GWO, selecting half of the average squared error as the optimizing object function, the RBF neural network based on GWO is named as RBF-GWO network. In the processing of training the parameters for RBF-GWO, the difference between the actual output of RBF-GWO network and the desired output value is considered as the guide of updating the position vector for the grey wolf, and in each iteration the optimal parameter values are stored in the position vector of Wolf α which will be returned to the RBF-GWO. The parameter training is completed until the end conditions of the iteration are satisfied. To verify the validity of RBF-GWO network, the continuous function approximation experiment and the chaotic synchronization anti-control experiment have been done in turn. Not only it is proved that the proposed RBF-GWO network is effective, but it is also found that the RBF-GWO network based on the WGWO has a relatively strong adaptive capacity in all experiments from the overall view.

Introduction
The Radial Basis Function (RBF) neural network was proposed by J. Moody and C. Darken. It is characterized by a shallow architecture, self-learning, self-organizing, adaptive, and avoiding local minima problem easily, and has been widely applied in all walks of life [1].

For RBF neural network, along with the economical and scientific development, it has become more difficult to set the weight values only depending on earlier experiences for these changing in real-time, non-linear, dynamic and complex problems. Thus, some better methods are needed to improve the adaptive ability of RBF neural network, or to enhance the fault tolerance ability. The least square method was applied to improve the RBF neural network in [2], the machine learning method was adopted to train the parameters of RBF in [3], and in [4] a variable projection method was used to perfect RBF neural network. Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) were also applied to optimize the parameters of RBF network respectively in [5] and [6], and their optimization effects were very well, which provide a good reference for the application of computational intelligence in RBF network training. Grey Wolf Optimizer (GWO) is a newer computational intelligence algorithm proposed in 2014 [7]. The training of RBF neural network based on GWO and several variant algorithms will be studied.

The GWO Algorithm
Grey wolf is in the top layer of the food chain, living and hunting in group. Usually there is a manager to lead a wolf pack hunting. The GWO is designed to imitate the behavior of the grey wolf pack in nature. In GWO, a grey wolf position represents a solution of the problem. The position of Wolf α shows the optimal solution, the position of Wolf β shows the sub-optimal solution, and that of Wolf δ shows the third best one. Wolf ω is a searching wolf, and it will do the searching behavior. Wolf α,
Wolf β and Wolf δ are thought as the leaders, and their number are usually set as one, while the number of Wolf ω is set to several dozens or hundreds. In each iterative process, the Wolf ω position is changed on one’s own initiative following as Eqs.(1)-(7), but the positions of Wolf α, Wolf β and Wolf δ are passively updated only when the Wolf ω solution is better than the one of Wolf α, Wolf β or Wolf δ. The solution of Wolf α is always optimal in each iteration.

\[ d_{alfa} = |C_1 \cdot X_{alfa} - X|, \]  
\[ d_{beta} = |C_2 \cdot X_{beta} - X|, \]  
\[ d_{delta} = |C_3 \cdot X_{delta} - X|, \]  
\[ X_1 = X_{alfa} - A_1 \cdot d_{alfa}, \]  
\[ X_2 = X_{beta} - A_2 \cdot d_{beta}, \]  
\[ X_3 = X_{delta} - A_3 \cdot d_{delta}, \]  
\[ X(t+1) = \frac{X_1 + X_2 + X_3}{3}. \]

where \( X_{alfa}, X_{beta}, \) and \( X_{delta} \) are the position vectors of Wolf α, Wolf β and Wolf δ respectively, \( X \) is the position vector of Wolf ω in the current iteration, \( X(t+1) \) the position vector in the next iteration, and Parameters \( A \) and \( C \) are calculated as following:

\[ A = 2a \cdot r_1 - a \]  
\[ C = 2 \cdot r_2 \]

where \( r_1 \) and \( r_2 \) are random vectors in [0,1], and Parameter \( a \) is a vector that decreases from 2 to 0 with the number of iterations increasing.

When the position value is out of the range, the following calculation could be done as:

\[ x^j_i(t) = \begin{cases} x_{max}, & \text{if } x^j_i(t) > x_{max} \\ x_{min}, & \text{if } x^j_i(t) < x_{min} \end{cases} \]

where \( x^j_i(t) \) is the \( j \)th position value of the \( i \)th Wolf ω in the current iteration, \( x_{max} \) is the maximum position value and \( x_{min} \) is the minimum one.

The stochastic search strategy has been adopted in GWO. Because of the characteristics of fast convergence rate, simple structure and easy programming, the GWO has been widely applied [8].

The exploiting ability and exploring ability of GWO are only balanced by a coefficient, and with the number of iterations increasing, its exploiting ability is increased but the exploring ability is greatly weakened, thus it would reduce the optimization speed, and easily fall into the local optimum. Several improved GWO algorithms have been proposed, such as binary GWO (bGWO1 and bGWO2) [9], the improved GWO based on evolutionary population dynamics (GWO-EPD) [10], the optimized GWO based on Mutation Operator and Eliminating-Reconstructing Mechanism (MR-GWO) [11], the modified GWO with hierarchical operators (WWGWO)[12], and the other improved GWO (WGWO) [13].

However, how effective is the GWO in RBF neural network? It remains unknown. In this paper, several GWO algorithms will be applied to RBF neural network for parameter training and testing.

38
RBF Network Based on GWO

To RBF neural network, it is important and difficult to determine the three vectors: the hidden center vector $\Phi$ whose dimension is $Nr \times Nh$, Gaussian RMS width vector $\Theta$ whose dimension is $Nh \times 1$ and the hidden-output weight vector $\rho$ whose dimension is $Nh \times Nn$. The “$Nn$” is the number of the output neuron. To get three optimized vectors, the GWO is applied in RBF network, and the network is noted as RBF-GWO. In RBF-GWO, the three vectors combination is treated as the position vector of the grey wolf. The optimal objective function is selected as:

$$
Object = \frac{1}{2n} \sum_{i=1}^{n} (y_{i,\text{expect}} - y_{i,\text{actual}})^2
$$

(11)

where $n$ is the number of the samples, $y_{i,\text{expect}}$ is the $i$th expected output, $Y_{\text{expect}} = [y_{1,\text{expect}}, \ldots, y_{n,\text{expect}}]^T$ and $y_{i,\text{actual}}$ is the $i$th actual output, $Y_{\text{actual}} = [y_{1,\text{actual}}, \ldots, y_{n,\text{actual}}]^T$.

The structure of RBF-GWO is shown in Figure 1. In RBF-GWO, first the samples are sent to the input ends ($U_1, \ldots, U_{Nr}$) for training, the output vector $Y_{\text{actual}}$ is got from the output layer of the RBF network. After comparing $Y_{\text{actual}}$ with the expected output value $Y_{\text{expect}}$, the compared results will be transferred to the GWO algorithm, in which the position of the grey wolf may be updated based on the compared results, and finally the Wolf $\alpha$ position vector will be sent back to the RBF network as its vector parameter $\Phi$, $\Theta$ and $\rho$, then one training is completed. One training ending, the next training starts until the training end condition is satisfied. Simple to understand, the training process of RBF network can be thought as the process to seek the minimum value of Eq. (11), and the training problem of RBF-GWO could be thought as a Minimization Problem.

![Figure 1. The structure of RBF-GWO.](image)

Continuous Function Approximation Experiment

To verify the effectiveness of RBF-GWO, three approximate experiments have been done. Three continuous functions are described in Table 1.

| Function Name            | Sigmoid      | Sine      | Cosine                        |
|--------------------------|--------------|-----------|-------------------------------|
| The range of training sample | [-3:0.1:3]  | [-2pi:0.1:2pi] | [1.25:0.05:2.75]        |
| The number of training sample | 61           | 126       | 31                            |
| The range of testing sample     | [-3:0.05:3]  | [-2pi:0.05:2pi] | [1.25:0.04:2.75]        |
| The number of testing sample    | 121          | 252       | 38                            |
| The network structure of RBF   | 1-20-1       | 1-15-1    | 1-15-1                        |
| The range of vector parameters in GWO | [-10,10]    | [-10,10]  | [-3,3]                        |
| Functional expression         | $y = 1/(1+\exp(-x))$ | $y = \sin(2x)$ | $y = (\cos(x \pi / 2))^7$   |

Although only the structure of RBF network based on the standard GWO has been introduced, the structures of RBF networks based on other GWO variants are similar to the one. Therefore, the other RBF-GWO structures will not need be explained in detail.

Table 1. The description of the three continuous functions.
The approximate results have been shown in Table 2, where AVE is the average error of ten repeated experiments between the output of RBF-GWO network and the expectations of the testing sample, STD is the average standard deviation of ten repeated experiments, and P is the average p value of Wilcoxon Test matching the network approximation output vector with the expected vector of the testing sample when the significance level was 0.05. The P value is more closing to 1, the difference between the network approximation sample and the testing sample is less obvious. In Table 2, the P value in Cosine experiment is bigger than that in Sine experiment, but smaller than that in Sigmoid experiment, meaning the approaching effect in Sigmoid experiment is the best, that in Cosine experiment is the next-best, and it is the most difficult to approximate the Sine function. Moreover, in the Sigmoid experiment, whether measured from P or AVE value, the approximation effect of RBF-GWO networks based on WGWO, GWO-EPD, MR-GWO and WWGWO are very well. In the Cosine experiment, the networks with P value above 0.9 included WGWO, GWO-EPD, MR-GWO and WWGWO, and the P value based on GWO was about 0.9. According to the results of Sine experiment, the P values based on WGWO, GWO and MR-GWO were greater than 0.8.

Table 2. The approximate results of RBF-GWO.

| Algorithm | AVE   | STD   | P     | Cosine | AVE   | STD   | P     | Sine   |
|-----------|-------|-------|-------|--------|-------|-------|-------|--------|
| WGWO      | 0.0008| 0.0001| 0.0012| 0.0111 | 0.0009|       |       |        |
| GWO       | 0.0005| 0.0000| 0.0009| 0.0008 | 0.0006|       |       |        |
| bGWO1     | 0.9893| 0.9911| 0.9854| 0.9947 | 0.9858|       |       |        |
| bGWO2     | 0.0152| 0.0111| 0.0119| 0.0105 | 0.0183|       |       |        |
| GA        | 0.0141| 0.0082| 0.0101| 0.0078 | 0.0221|       |       |        |
| GWO-EPD   | 0.9285| 0.8911| 0.9330| 0.9152 | 0.9620|       |       |        |
| MR-GWO    | 0.1033| 0.1117| 0.1432| 0.0550 | 0.1440|       |       |        |
| WWGWO     | 0.4131| 0.4270| 0.4263| 0.1579 | 0.4887|       |       |        |

To consider all algorithms more clearly, the total mean results are obtained as Table 3. The MAVE is the overall mean error of the three approximate experiments, the MSTD is the overall average standard deviation, and the M-p is the overall average p value for the RBF-GWO network based on each algorithm. From the results, it is obvious that the RBF-GWO network based on MR-GWO, WGWO and GWO is better than others when the approximation effect was measured by MAVE and M-p, and this result is in good agreement with the results in table 2. Generally, RBF-GWO networks can approach the continuous functions very well, especially based on WGWO and MR-GWO.

Table 3. The overall results of approximate experiments.

| Algorithm | MAVE | MSTD | M-rate |
|-----------|------|------|--------|
| WGWO      | 0.03979| 0.14257| 0.92736 |
| GWO       | 0.04323| 0.14579| 0.90348 |
| bGWO1     | 0.13714| 0.4112 | 0.49532 |
| bGWO2     | 0.15197| 0.36666| 0.51704 |
| GA        | 0.03447| 0.03749| 0.72235 |
| GWO-EPD   | 0.05206| 0.14575| 0.89671 |
| MR-GWO    | 0.02220| 0.05552| 0.93125 |
| WWGWO     | 0.05436| 0.17046| 0.87556 |

The Application of RBF-GWO in Chaotic Synchronization

Since 1975 chaos has become a hot topic in the kinetic theory field, especially in nonlinear systems. Although chaos has many definitions, no matter which definition, chaos have the same characteristics: the sensitivity to initial conditions, the aperiodic, fractal dimension and so on. Chaos not only needs controlled but also needs chaotification, anti-control of chaos. The anti-control of chaos is to take advantage of chaos. The chaotic synchronization anti-control means to make another chaotic system to synchronize the referential one by adding the synchronization controller. Here, the RBF-GWO network will be selected as the chaotic synchronization controller.
In this experiment, the PMSM chaotic systems are involved, as Eq. (12) and Eq. (13). The system of Eq. (12) will be as the reference model, and the one of Eq. (13) will be as the controlled system which will be controlled by RBF-GWO to make it synchronized with the system of Eq. (12).

\[
\begin{align*}
\dot{y}_{m1} &= -y_{m1} + y_{m2}y_{m3} + v_d \\
\dot{y}_{m2} &= -y_{m2} - y_{m1}y_{m3} + 17.5y_{m3} + v_q \\
\dot{y}_{m3} &= 5.46(y_{m2} - y_{m1}) - T_L
\end{align*}
\]  

(12)

\[
\begin{align*}
\dot{y}_1 &= -y_1 + y_2y_3 + v_d \\
\dot{y}_2 &= -y_2 - y_1y_3 + 28y_3 + v_q \\
\dot{y}_3 &= 3(y_2 - y_1) - T_L
\end{align*}
\]  

(13)

where \(y_1\) and \(y_{m1}\) are the normalized state variables of direct axis current, \(y_2\) and \(y_{m2}\) are the normalized state variables of quadrature axis current, \(y_3\) and \(y_{m3}\) are the normalized state variables of angular velocity, the \(v_d\) and \(v_q\) are the normalized variables of direct axis voltage and quadrature axis voltage respectively, and the \(T_L\) is the load torque.

Here, it is assigned that \(v_d = 0\), \(v_q = 0\) and \(T_L = 0\), and the synchronization results have been shown in Figs. 2-7. Figure 2, Figure 4 and Figure 6 are the results under controlling of RBF-GWO based on all algorithms including WGWO, GWO, bGWO1, bGWO2, GA, GWO-EPD, MR-GWO, and WWGWO. What can be seen obviously is that the results about bGWO1, bGWO2, and GA are not as good as others, since there are more failing tracking points than others. Therefore, Figure 3, Figure 5, and Figure 7 have been given without the algorithms of bGWO1, bGWO2, and GA. And in Figure 3, Figure 5, and Figure 7, it is shown obviously that the \(y_1\), \(y_2\), and \(y_3\) can effectively track the \(y_{m1}\), \(y_{m2}\), and \(y_{m3}\) respectively. To more clearly show the synchronization effect, 5001 points have been sampled with 0.01s for time interval in the range of \([0, 50s]\). The chaotic synchronization controlling errors of RBF-GWO based on all algorithms are shown in Table 4, where \(y1e\) is the average value of \(e_1 = y_{m1} - y_1\), \(y2e\) is the average value of \(e_2 = y_{m2} - y_2\), \(y3e\) is the average error of \(e_3 = y_{m3} - y_3\), and the AVE is the total average output errors about \(Y_m - Y\). From the AVE, the RBF-GWO network based on WGWO shows the best synchronous controlling effect, the one based on GWO-EPD takes the second place, and the one based on WWGWO takes the third place.
Summary and Discussion

Overall, the RBF-GWO network trainings based on all GWO algorithms are highly effective. Although they have the similar structure for all GWO algorithms, there still are many differences in RBF-GWO network training results. From the results of continuous function approximation experiments, it could be generally understood that the RBF-GWO networks based on MR-GWO and WGWO have the better training effects than others. And in Chaotic synchronous anti-control experiments, the best training effect should be belonged to the WGWO. Why is not an algorithm that performs the best in all experiments? That may provide the best explanation for No Free Lunch Theorem. However, it could still be seen that the WGWO performs well in all training experiments.
Table 4. The errors of the homogeneous synchronous.

| Algorithm | y1e  | y2e  | y3e  | AVE   | rank |
|-----------|------|------|------|-------|------|
| GWO       | 0.190| 0.312| 0.173| 0.22543| 5    |
| bGWO1     | 0.862| 0.571| 0.645| 0.693  | 6    |
| bGWO2     | 1.277| 0.800| 0.659| 0.91253| 7    |
| GWO-EPD   | 0.169| 0.192| 0.163| 0.17546| 2    |
| MR-GWO    | 0.242| 0.238| 0.165| 0.2156  | 4    |
| WGWO      | 0.139| 0.287| 0.096| 0.17446| 1    |
| WWGWO     | 0.183| 0.312| 0.133| 0.20983| 3    |
| GA        | 1.117| 0.675| 0.460| 0.7513  | 8    |

Acknowledgement

This research was supported by the Education and Teaching Reform Research Project of Xianyang Normal University (NO.2017Y004), and partly supported by the Local Service Researching project of Xian Yang Normal University (NO.XSYK18048).

References

[1] Q. Zhou, P. Shi, S. Xu, et al, Observer-based adaptive neural network control for nonlinear stochastic systems with time delay. IEEE Trans. Neural Netw. Learn. Syst. 24 (2013) 71-80.

[2] F. Ding, P.X. Liu, G. Liu, Multi-innovation least-squares identification for system modeling. IEEE Trans. Syst. Man, Cybern. B, Cybern. 40 (2010) 767-778.

[3] G.B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification. IEEE Trans. Syst. Man, Cybern. B, Cybern., 42 (2012) 513-529.

[4] M. Gan, H.X. Li, H. Peng, A variable projection approach for efficient estimation of RBF-ARX model. IEEE T. Cybernetics 45 (2015) 476-485.

[5] V. Fathi, G.A. Montazer, An improvement in RBF learning algorithm based on PSO for real time applications. Neurocomputing 111 (2013) 169–176.

[6] D.W. Chen Research on traffic flow prediction in the big data environment based on the improved RBF neural network. IEEE T. Ind. Inform. 13 (2017) online.

[7] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer. Adv. Eng. Softw. 69 (2014) 46-61.

[8] X.Q. Zhang, Z.F. Ming, Trajectory Planning and Optimization for a Par4 Parallel Robot Based on Energy Consumption. Appl. Sci-basel. 9 (2019) 2770.

[9] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection. Neurocomputing 172 (2016) 371-381.

[10] S. Saremi, S.Z. Mirjalili, S.M. Mirjalili, Evolutionary population dynamics and grey wolf optimizer, Neural. Comput. Applic. 26 (2015) 1257-1263.

[11] X.Q. Zhang, Z.F. Ming, An optimized GWO based on a mutation operator and eliminating-reconstructing mechanism and its application. Front. Inform. Tech. El. 18 (2017) 1705-1719.
[12] L. Rodríguez, O. Castillo, J. Soria, et al. A fuzzy hierarchical operator in the grey wolf optimizer algorithm. Appl. soft comput. 57 (2017) 315-328.

[13] L.G. Li, L.J. Sun, J. Guo, et al. Modified discrete grey wolf optimizer algorithm for multilevel image thresholding. Comput. Intel. Neurosc. 3 (2017) 285–296.