

RISSET-36-2019-RPI-Warsito_
2019_J._Phys._3A_Conf._
Ser._1351_012089.pdf

by Budi Warsito

Submission date: 27-Jan-2020 04:01AM (UTC+0700)

Submission ID: 1246596424

File name: -36-2019-RPI-Warsito_2019_J._Phys._3A_Conf._Ser._1351_012089.pdf (1.03M)

Word count: 2622

Character count: 13856

PAPER • OPEN ACCESS

Hybrid of Particle Swarm and Levenberg Marquardt Optimization in Neural Network Model for Rainfall Prediction

To cite this article: Budi Warsito *et al* 2019 *J. Phys.: Conf. Ser.* **1351** 012089

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

Hybrid of Particle Swarm and Levenberg Marquardt Optimization in Neural Network Model for Rainfall Prediction

Budi Warsito^{1,2}, Rukun Santoso¹ and Hasbi Yasin¹

¹Department of Statistics, Faculty of Science and Mathematics, Diponegoro University Semarang Indonesia

²Magister Program of Information System, School of Postgraduate Studies, Diponegoro University Semarang Indonesia

budiwrst2@gmail.com

Abstract. Neural network model has undergone many modifications from the original model through the development of network architecture and optimization methods. By default, the gradient-based optimization method is a method used to find network weights. But some of the weaknesses and limitations of these methods inspired many researchers to try to use other methods. Non-gradient based heuristic method is a reasonable choice considering the learning algorithm in artificial neural networks is inspired by the characteristics of creatures so that optimization methods that also mimic life patterns in nature will be appropriate. One disadvantage of the heuristic method is the length of the iteration process. In this paper, a method which combines heuristic optimization methods and gradient-based methods is applied, namely particle swarm optimization (PSO) and Levenberg Marquardt. The weight obtained from the PSO method becomes the initial weight for the Levenberg Marquardt method. The proposed procedure is applied to the rainfall data in Cokrotulung Klaten. The results showed that this procedure succeeded in providing better predictions than the Levenberg Marquardt method.

1. Introduction

Neural network modeling in time series has experienced rapid development and has been applied in various fields. The interesting things for many researchers are the method to determine the optimal input, the number of units in the hidden layer, the activation function used in the hidden layer, and the optimization method used to get the optimal weight. Various modeling procedures have also been carried out to obtain optimal architecture both theoretical and applied studies. The optimization method to get optimal weight becomes one of the main focuses in neural network modeling. By default, standard optimization methods used to estimate network weights are gradient-based methods. Therefore, the activated activation function is a continuous and differentiated function. In the next stage, various heuristic optimization methods for optimizing a function also experienced many developments. This is also supported by advances in computing so that this facilitates the development of various new optimization methods that do not use gradients. Likewise, statistical and mathematical modeling has progressed so that various alternative models are formed to get better predictions. These models are increasingly complex, the consequence of which is the need for an appropriate optimization technique to obtain parameter estimates.

Development of heuristic optimization methods for optimizing a function is a new chapter in the field of statistical modeling. These methods are then used as a way to get parameter estimates from alternative models. Some of the optimization methods include genetic algorithm, ant colony, simulated annealing and particle swarm optimization. One disadvantage of the non-gradient method is that it requires a longer iteration time than the gradient-based method. Therefore, it is important to conduct experiments to combine the two methods so that the results are more efficient. This article discusses the use of particle swarm optimization as a method for determining the initial weight of a gradient-based optimization method, Levenberg Marquardt. The use of the Levenberg Marquardt method in the neural network model has been widely carried out. In fact, it becomes the default optimization method in the Matlab program for neural networks. Several previous studies relating to the use of particle swarm optimization for weight optimization also have been carried out [1-3]. In this paper, the use of Levenberg Marquardt for optimizing neural network models is done by using the PSO algorithm as a tool for determining initial weights. As an application, the proposed procedure is applied to the rainfall data at ZOM 136 Cokrotulung, Klaten. Rainfall prediction using neural networks has also been done by Lin and Wu [4].

2. Methods

In this study, network architecture has been determined. PSO optimization is used first, then the weight obtained is used as the initial value of the Levenberg Marquardt method. Explanations for each stage of modeling are explained in the following sections. The proposed procedure has

2.1. Neural Network

Neural network architecture consists of three layers, namely the input layer, the hidden layer and the output layer. In neural network modeling for time series data, the input layer consists of past data to lag p . In this study, determining the input using the best ARIMA method. The neural network architecture for time series data is presented in Figure 1.

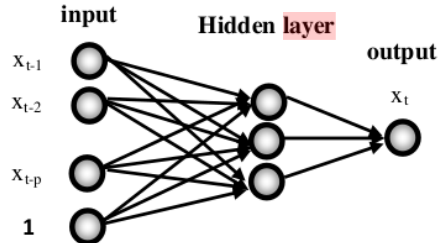


Figure 1. Neural network architecture for time series data

The architecture can be written in the form of a mathematical model as follows:

$$x_t = f^o(w^b + \sum_{j=1}^l w_j^o f^h(w_j^b + \sum_{i=1}^p w_{ji}^h x_{t-i})) \quad (1)$$

where f^h is an activation function in the hidden layer, the weight vector consists of the weight of the input layer to the hidden layer (w_{ij}), the weight of the bias to the hidden layer (w_{b_i}), the weight of the hidden layer to the output layer (w_j) and the weight of the bias to the output layer (w_b). In the standard backpropagation algorithm, the weights are optimized using certain gradient-based optimization methods. Initial weights are usually chosen randomly. In this study, PSO is used as a method for determining initial weights, then the gradient-based LM method is used to obtain the final weights.

2.2. Hybrid Particle Swarm Optimization-Levenberg Marquardt

PSO is inspired by the social environment of the life of a group of birds or fish that interact with each other. Without colliding, they move together towards a place by forming certain formations. This algorithm was proposed by Kennedy and Eberhart [5] for optimizing nonlinear continuous functions

and has been applied to machine learning [6]. In PSO optimization, the set of solutions is called a particle and as an initialization is generated randomly. The quality of each particle is evaluated by a fitness function. The particles will move and follow the optimum particles, i.e the particles with the best fitness. Each bird is described as a particle that represents a solution to an optimization problem. These particles have position and velocity. There are two important tasks in the PSO algorithm, namely velocity and position updates. The update process is an iteration form of the PSO algorithm. The particles used to update speed and position will continue to accelerate to the best position of the same particle in the previous iteration and the best particle as a whole. This process continues until the termination procedure is fulfilled. The standard procedure for stopping criteria is the maximum number of iterations or minimum errors. At the beginning of the process, PSO initializes the initial position and initial velocity then at each iteration, PSO will update the position and velocity using the following formula [7]:

$$v_i^{t+1} = \rho \cdot v_i^t + c_1 \cdot r_1 (pBest_i^t - x_i^t) + c_2 \cdot r_2 (gBest_g^t - x_i^t) \quad (2)$$

$$x_i^{t+1} = x_i^t + v_i^t \quad (3)$$

where

- ρ : inertia weight
- c_1 & c_2 : acceleration coefficients
- r_1 & r_2 : random value from continue uniform distribution (0,1)
- $pBest_i^t$: the best position of particle i at iteration t
- $gBest_g^t$: optimum global from particle g at iteration t
- x_i^t : position of particle i at iteration t

At each iteration to update position and velocity, the weight of inertia is also always updated. The updating of inertia weights is expressed by the following formula:

$$\rho = \rho_{max} - \frac{\rho_{max} - \rho_{min}}{t_{max}} \times t \quad (4)$$

where

- ρ_{max} : upper bound of ρ
- ρ_{min} : lower bound of ρ
- t_{max} : maximum of iteration
- t : current iteration

The initial position of the particles generated randomly and after going through an iteration process the final position of the particles is the weight of the optimization results. The particle speed in the initialization process is given a zero value because at the initial position the particle has not experienced movement. The steps of the PSO algorithm can be described as follows:

1. Determine the value of the initial position and the initial velocity of each particle randomly
2. Calculate the fitness value for each particle
3. Determine pBest and gBest
4. Calculate the weight of inertia
5. Update the position and speed
6. If the stopping criteria are met, return to 2
7. End

Determination of fitness in the initial population is very important because it is used to determine the best individual position (pBest) and the best global position (gBest). Fitness compared to the best population in the previous population. If the current results are better than the best global value of gBest, an update of the location and position of gBest is performed. The stages are repeated until the stopping criteria are met. After this stage, the optimal weights are obtained. These weights then

become the initial weights of the Levenberg Marquardt method. Levenberg Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. This method improves the solution to problems that are much harder to solve by only adjusting the learning rate repeatedly [8]. This represents a procedure for evaluating the inverse of the Hessian using single pass through the data set [9].

3. Results and Discussion

The data used in this research is ten daily rainfall data in Cokrotulung Klaten, Central Java from January 2010 to July 2018, that is 309 data. The data is divided into two parts, the first is 247 data (80%) as training and the remaining 62 data (20%) as testing. The activation function used in the hidden layer is the sigmoid logistic. The results of the investigation with the ARIMA model show that the input to the neural network model are the three variables in lags 1, 2 and 18 and the number of hidden units the same with the number of input. In the proposed procedure, four scenarios are built. In the first scenario, PSO optimization with a few generations is used, in this case it is determined to be equal to 10, the resulting weights are the initial weights of the Levenberg Marquardt method. In the second scenario, the generation used in PSO is 30, in the third scenario, the generation used is 50 while in the third scenario, the generation used is 100. In all four kinds of scenarios, optimization is continued by the Levenberg Marquardt method of 1000 epochs. The results obtained were compared with original Levenberg Marquardt of 1000 epochs and original PSO of 1000 generations. The results are presented in Table 1.

Table 1. Results of various scenarios.

	Scenario	MSE	
		in-sample ($\times 10^3$)	out-sample ($\times 10^3$)
PSO-LM	10 generations – 1000 epochs	4.3231	3.1671
	30 generations – 1000 epochs	4.2363	3.1529
	50 generations – 1000 epochs	4.4968	3.1632
	100 generations – 1000 epochs	4.2337	3.1775
PSO	1000 generations	4.2505	3.1564
LM	1000 epochs	4.3215	3.1547

The results of Table 1 show that the hybrid particle swarm method with 30 generations and Levenberg Marquardt with 1000 epochs gave better results than the only Levenberg Marquardt method. The time needed to complete the iteration process on a hybrid method with a few generations at the PSO stage is not much different from the Levenberg Marquardt method. Thus the time needed to complete the iteration with the hybrid method is not an obstacle. Therefore the basis for choosing the optimization method used is purely due to the errors achieved. In this case, the hybrid method is superior. An identical explanation can also be seen from its comparison with the only PSO algorithm. Although not much different, the hybrid method is slightly superior to PSO. The problem is the PSO method which requires much longer time to complete 1000 generations. Taking into account the effectiveness, the hybrid method is preferred for use. This method is able to improve the error rate of the Levenberg Marquardt method and is able to speed up the required iteration time. As an illustration, Fig. 2 displays the results of in-sample and out-sample predictions.

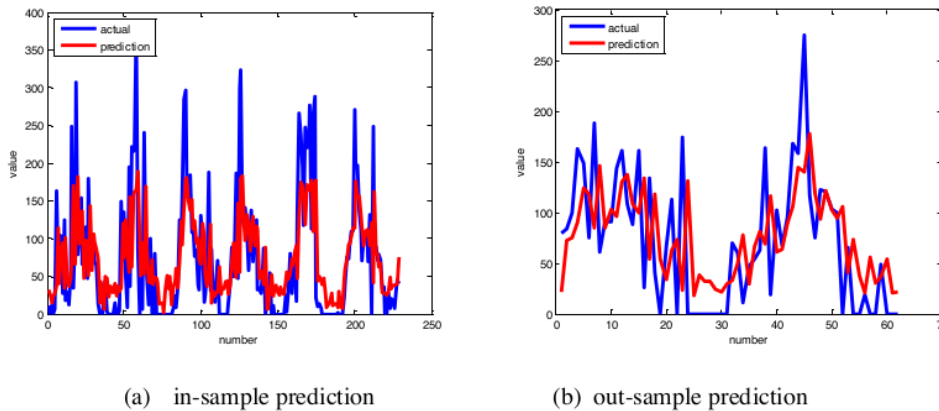


Figure 2. The in-sample and out-sample prediction of the hybrid PSO-LM method

Fig. 2 shows that the results of the in-sample prediction have successfully approached the actual. Seasonal patterns from the original data can be followed by predictions. Likewise with the results in the out-sample data, the predictions can also be close to the actual results. Thus the predicted results of the proposed procedure are able to produce predictions for both in-sample and out-sample predictions. In other words, in addition to being used to build the proposed procedure model, it is also good for forecasting.

4. Conclusion

The use of particle swarm to obtain the initial weights of Levenberg Marquardt in neural network model was developed. The proposed procedure has been used in various scenarios. A few generations of PSO was needed for obtaining the optimal weights prediction of the rainfall data by using Levenberg Marquardt. It better than use only Levenberg Marquardt and also faster than the only PSO. The developing of the hybrid procedure also been the interesting future works. It can be done by varying the gradient based optimization methods and make a comparison between them.

References

- [1] Lu W Z, Fan H Y, Leung A Y T and Wong J C K 2002 Analysis of pollutant levels in central Hong Kong applying neural network method with particle swarm optimization *Environmental monitoring and assessment*, **79(3)** pp 217-230.
- [2] Lu W Z and Xue Y 2014 Prediction of particulate matter at street level using artificial neural networks coupling with chaotic particle swarm optimization algorithm *Building and Environment* 78 pp 111-117
- [3] Ren C, An N, Wang J, Li L, Hu B and Shang D 2014 Optimal parameters selection for BP neural network based on particle swarm optimization: A case study of wind speed forecasting *Knowledge-Based Systems* 56 pp 226-239
- [4] Lin G F and Wu M C 2009 A hybrid neural network model for typhoon-rainfall forecasting *Journal of Hydrology* 375 pp 450-458
- [5] Kennedy J and Eberhart R 1995 Particle Swarm Optimization, Proceedings of the 1995 *IEEE International Conference on Neural Networks* Piscataway NJ USA pp 1942-1948
- [6] Kennedy J 2011 Particle swarm optimization *In Encyclopedia of machine learning* pp 760-766 Springer US
- [7] Shi Y and Eberhart R 1998 Parameter selection in particle swarm optimization *Proceedings of the International Conference on Evolutionary Programming* pp 591-601
- [8] Samarasinghe S 2007 *Neural Networks for Applied Sciences and Engineering* Aurbach Publications Taylor & Francis Group New York

- [9] Bishop C M *Neural Networks for Pattern Recognition* Oxford University Press

ORIGINALITY REPORT

6%

SIMILARITY INDEX

0%

INTERNET SOURCES

11%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

Yasin Hasbi, Warsito Budi, Santoso Rukun.
"Feed Forward Neural Network Modeling for
Rainfall Prediction", E3S Web of Conferences,
2018

Publication

6%

Exclude quotes On

Exclude matches < 3%

Exclude bibliography On