Rice Yield Prediction – A Comparison between Enhanced Back Propagation Learning Algorithms

Puteh Saad[1,2], Nor Khairah Jamaludin[1], Nursalasawati Rusli[1], Aryati Bakri[2] and Siti Sakira Kamarudin[3]

[1]*Artificial Intelligence and Software Engineering Research Lab, School of Computer & Communication Engineering, Kolej Universiti Kejuruteraan Utara Malaysia(KUKUM), Blok A, Kompleks Pusat Pengajian, Jalan Kangar-Arau, 02600 Jejawi, Perlis, Malaysia.*
Telephone : 604 – 9798302/8310/8322/8144, Fax : 604 – 9798141
puteh@kukum.edu.my, nursalasawati@kukum.edu.my
[2]*Faculty of Computer Science and Information System, University Technology Malaysia, Skudai, Johor, Malaysia.*
puteh@fsksm.utm.my, aryati@fsksm.utm.my
[3] *School of Information Technology, Universiti Utara Malaysia, Sintok, Kedah, Malaysia.*
sakira@uum.edu.my

Abstract

Back Propagation algorithm(BP) has been popularly used to solve various problems, however it is shrouded with the problems of low convergence and instability. In recent years, improvements have been attempted to overcome the discrepancies aforementioned. In this study, we examine the performance of four enhanced BP algorithms to predict rice yield in MADA plantation area in Kedah, Malaysia . Amidst the four algorithms explored, Conjugate Gradient Descent exhibits the best performance.

Keywords:

Back-Propagation Algorithm, Quick Propagation, Conjugate Gradient Descent, Levenberg-Marquardt algorithm and Rice Yield Prediction.

1. Introduction

Back Propagation is by far the most widely used algorithm to train Multi-Layer Perceptron (MLP) for optimization, function approximation and pattern recognition. However it is afflicted with several deficiencies, the major ones are of low convergence and instability. The problem is classified as NP-complete [4]. The low convergence rate and instability are attributed to the following reasons:

i. the presence of local minima (*i.e.* isolated valleys) in addition to global minimum. Since Back-Propagation is basically a hill-climbing technique, it runs the risk of being trapped in a local minimum, where every small change in synaptic weights increases the error function. But somewhere else in the weight space there exist another set of synaptic weights for which the error function is smaller than local minimum in which the network is stuck. Clearly, it is undesirable to have the learning process terminating at a local minimum, especially if it is located far above a global minimum [1].

ii. the error surface is highly curved along a weight dimension, in which case the derivative of the error surface with respect to that weight is large in magnitude. In this second situation, the adjustment applied to the weight is large, which may cause the algorithm to overshoot the minimum of the error surface [9].

iii. The direction of the negative gradient vector may point away from the minimum of the error surface, hence the adjustments applied to the weights may induce the algorithm to move in the wrong direction. Consequently, the rate of convergence in BP training tends to be relatively slow [8].

The improvements made to overcome the above phenomena are performed using heuristic techniques and numerical optimization techniques. In this study three techniques are chosen; Quick propagation, Conjugate Gradient Descent and Levenberg-Marquardt. Quick propagation and Conjugate Gradient Descent are categorized as heuristic techniques and Levenberg-Marquardt as numerical optimization techniques.

Section 2 describes Quick propagation, Conjugate Gradient Descent and Levenberg-Marquardt enhancement techniques. Section 3 present the methodology adopted to perform the yield prediction. Section 4 discusses the results obtained and and the paper ends with a conclusion in Section 5.

2. The Enhanced Back Propagation Algorithms

Quick propagation computes the average gradient of the error surface across all cases before updating the weights once at the end of the epoch.

In the standard BP, the error function decreases most rapidly along the negative of the gradient however fastest convergence is not guaranteed. Conjugate gradient descent overcomes the discrepancy by constructing a series of line searches across the error surface. It first works out the direction of steepest descent, just as back propagation would do [2].

$$p_0 = -g_0$$

A line search is then performed to determine the optimal distance to move along the current search direction

$$x_{k+1} = x_k + \alpha_k p_k$$

where

x_k is the vector of current weight and bias
α_k is the learning rate
p_k is the gradient

The next search direction is determined so that it is conjugate to previous search directions. The general procedure for determining the new search direction is to combine the new steepest descent direction with the previous search direction:

$$p_k = -g_k + \beta_k p_{k-1}$$

The constant β_k is computed based on the Fletcher-Reeves update:

$$\beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}$$

The Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares, then the Hessian matrix can be approximated as:[6][3].

$$H = J^T J$$

and the gradient can be computed as

$$g = J^T e$$

where

J : Jacobian matrix contains first derivatives of the network errors with respect to the weights and biases.
e : a vector of network errors

The weights and biases are computed based on the following formula:

$$x_{k+1} = x_k - \left[J^T J + \mu I\right]^{-1} J^T e$$

where μ is a scalar value.

μ is decreased after each successful step and is increased only when a tentative step would increase the performance function. Hence, the performance function will always be reduced at each iteration of the algorithm.

3. The Methodology

In this study the domain chosen to evaluate the above algorithms is the rice yield prediction. The samples are obtained from Muda Agricultural Development Authority (MADA), Kedah, Malaysia from 4 areas consisting of 27 localities (A1 to G4). In this study, only three parameters are considered that affects rice yield namely; diseases, pests and weeds. Each parameter is further subdivided into various types. There are 12 types of diseases and pests and 11 types of weeds. The effect of each type of diseases is summed up to one value, similarly with pests and weeds, generating only 3 parameters. Hence the input data table consists of 3 columns representing factors affecting yield and the last column is the yield obtained. Rows represent localities. 5 seasons (1/95 to 1/97) for 27 localities, producing a total of 135 rows are utilized to train the Neural Network. One season (2/97) for 27 localities is used for prediction. Table 1 depicts an example of rice yield data used for training. Table 2 consists of data for the three parameters to predict the yield.

The Neural Network is trained using the above enhanced techniques, based on the data in Table 1. Since there are a total of three parameters affecting the yield, thus the number of nodes in the input layer is 3. The number of nodes in the output layer is one since there is one output. The number of nodes in the hidden layer is determined by training the Neural Network by varying it. Based on the smallest sum of square error values, the suitable number of nodes in the hidden layer for the 4 learning algorithms is depicted in Table 3.

Table 1:A Sample of Rice Yield Data to train the Neural Network

Locality	Pests	Diseases	Weeds	Yield
A1(1)	63.63	189.11	91.85	4223
B1(2)	538.46	283.51	236	4276
C1(3)	151.61	224.38	326.7	3652
D1(4)	189.84	809.26	341.31	4686
E1(5)	176.66	334.08	280.5	3948
A2(6)	936.05	439.96	540.5	4625
B2(7)	868.85	555.59	622.35	4712
C2(8)	560.16	600.63	2507.6	4704
D2(9)	562.05	476.2	394.3	5344
E2(10)	816.12	542.06	1958.59	4764
F2(11)	846.1	547.62	1759.2	4716
G2(12)	68.76	298.88	295.48	3998
H2(13)	297.45	455.05	434.95	4808
I2(14)	317.87	505.23	892.26	4005
A3(15)	9.37	668.71	126.61	3672
B3(16)	302.9	495.91	992.5	5196
C3(17)	142.15	580.36	95.23	5226
D3(18)	277.92	663.16	613.05	4406
E3(19)	239.5	516.06	76.8	3808
F3(20)	549.95	379.32	374.8	5292
A4(21)	205.44	528.04	154.53	4814
B4(22)	255.85	703.12	193.16	4621
C4(23)	378.56	545.12	514.1	4675
D4(24)	440.25	760.78	753.03	5529
E4(25)	470.14	707.12	497	4456
F4(26)	392.2	750.29	345.94	4883
G4(27)	216.35	478.69	650.9	4747

Table 2: Parameters Values that affect Rice Yield

Locality	Pests	Diseases	Weeds
A1(1)	74.7	21.8	161.34
B1(2)	90.74	22.95	136.94
C1(3)	88	26.54	124.97
D1(4)	56.81	53.21	305.4
E1(5)	138.03	204	323
A2(6)	459.95	31.5	476
B2(7)	299.03	17.64	107.4
C2(8)	524.95	132	1174.3
D2(9)	369.8	0.1	175.4
E2(10)	832.8	14.55	3136.62
F2(11)	478	0.1	278
G2(12)	125.15	4.2	102.6
H2(13)	185.05	2.6	573.05
I2(14)	102.2	8	269.25
A3(15)	18.26	0.6	81.94
B3(16)	152.52	51.25	789.9
C3(17)	361.21	0.1	221.32
D3(18)	225.79	0.5	707.45
E3(19)	31.02	0.48	37.76
F3(20)	114.19	4.89	141.05
A4(21)	211.49	157.1	182.3
B4(22)	87.1	492.7	273.5
C4(23)	827.01	380	689
D4(24)	799.15	128	497
E4(25)	473.95	52.5	517.5
F4(26)	398.1	26.5	472.5
G4(27)	457.1	51.02	698

4. Results and Discussion

The number of nodes in the input layer = 3
The number of hidden layer = 1
The number of nodes in the output layer = 1

Table 3 depicts the number of nodes in the hidden layer for each algorithm. Conjugate Gradient Descent and Levenberg-Marquardt use only two nodes in the hidden layer as compared to Back Propagation that uses double the value. Quick Propagation being a heuristic technique uses 3 nodes in the hidden layer.

Table 3: Number of Nodes in the Hidden Layer for the Learning Algorithms

Algorithms	Number of nodes in the hidden layer
Quick Propagation	3
Conjugate Gradient Descent	2
Levenberg-Marquardt	2
Back Propagation	4

Fewer nodes are required by the Conjugate Gradient algorithm is due to it's nature that perform a search for minimum value of error function in a straight line fashion as compared to Back Propagation algorithm that perform a search for a minimum value of error function proportional to the learning rate.

Levenberg-Marquardt algorithm compromises between the linear model and a gradient-descent approach, thus fewer nodes are used in the hidden layer. A move to a next step is allowed if the error value is less than that of the current value. The allowable downhill movement consists of a sufficiently small step.

As for the Quick Propagation, it enhances the Back Propagation algorithm by merely computing the average gradient before updating the weights. Thus, it still model the non-linear relationship between data, hence there is a slight improvement in the number of nodes in the hidden layer as compared to Back-propagation algorithm.

The Neural Network Model fitted with the above learning algorithm is then used to predict the yield based on the sample data in Table 2. A graph of average absolute error versus each of the above algorithms is plotted as depicted in Figure. 1. The results obtained tally with the number nodes used in the hidden layer. With the highest number of nodes in the hidden layer, Back Propagation algorithm shows the highest absolute error. The absolute error for Quick Propagation is slightly better that back-propagation, absolute error for Lavenberg-Marquart is better than Quick Propagation. Conjugate Gradient Descent displayed the lowest absolute error. The lowest error depicted by the Conjugate Gradient Descent algorithm is due to the search direction to obtain the minimum error value, assuring that the algorithm is not stuck at local minima.

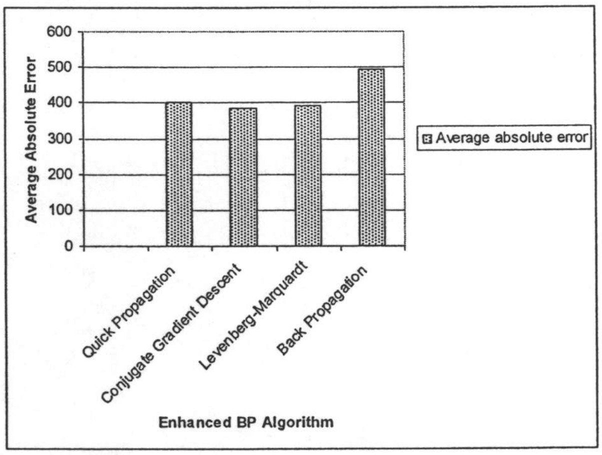

Figure 1: Average Absolute Error versus Different Enhanced Back Propagation Algorithms

In order to illustrate the performance of each learning algorithm, a graph of actual and predicted yield is plotted against locality. Figure 2 depicts the performance of Quick Propagation algorithm. Figure 3 depicts the performance of Conjugate Gradient Descent algorithm. Figure 4 depicts the performance of Levenberg-Marquardt algorithm and Figure 5 highlights the performance of Back Propagation algorithm.

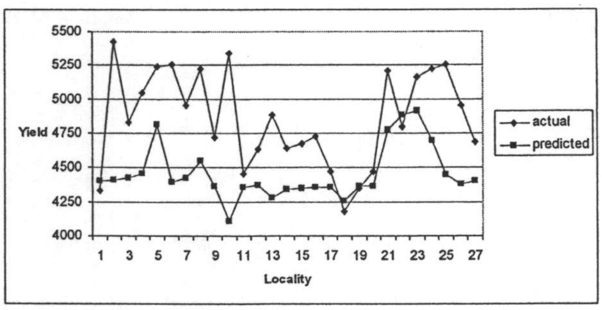

Figure 2: Yield versus Locality of Quick Propagation

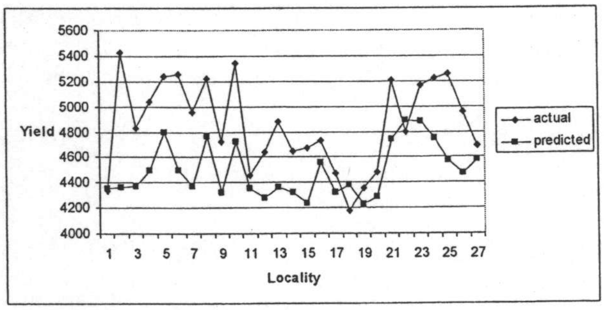

Figure 3: Yield versus Locality of Conjugate Gradient Descent

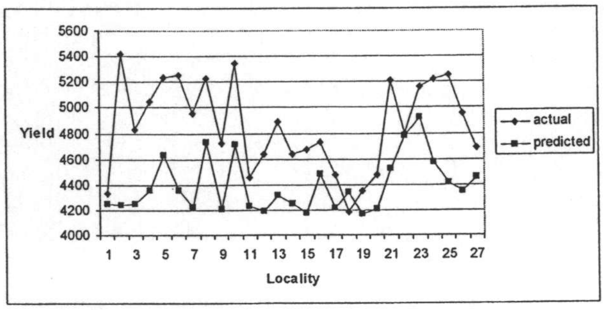

Figure.4: Yield versus Locality of Levenberg-Marquardt

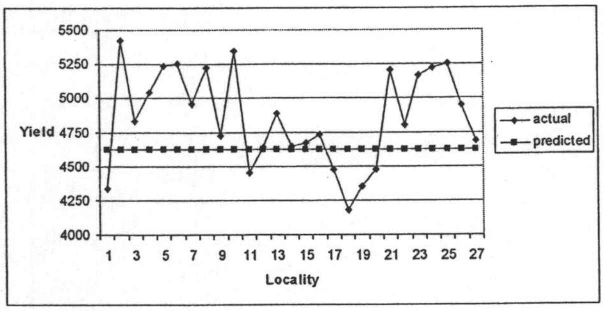

Figure 5: Yield versus Locality of Back Propagation

Based on Figure 2 to Figure 5, Conjugate Gradient Descent algorithm portrays the best performance as compared to the other. The next to it is Lavenberg-Marquart and then Quick Propagation. Back Propagation algorithm failed to produce the desired output due the major problem of being stuck at local minima The outstanding performance of the Conjugate Gradient Descent algorithm is due to the strategy that successive weight correction steps are orthogonal to the gradient. Thus attributing it to exhibit a quadratic convergence property that avoid the local minima phenomena. Lavenberg-Marquart is another alternative to choose in training the Neural Network for rice yield prediction. Its superiority as compared to Back Propagation algorithm is that the training is based on second-order derivative approach that avoid local minima problem and exhibit a faster convergence. However Lavenberg-Marquart algorithm has a major drawback that it requires the storage of some matrices that can be quite large for this kind of problems. Thus, when comparison is performed between Lavenberg-Marquart and Conjugate Gradient Descent, Conjugate Gradient Descent wins. From this finding, Conjugate Gradient Descent is adopted to train the Neural Network to predict the rice yield in this study.

5. Conclusion

In this study, 4 supervised learning algorithms are explored to predict rice yield based on weeds, diseases and pests in Muda Agricultural Development Authority (MADA), Kedah, Malaysia. It is found that Conjugate

Gradient Descent algorithm exhibits the outstanding performance as compared to Levenberg-Marquardt, Quick Propagation and Back-propagation algorithms. Although the Conjugate Gradient Descent is based on heuristics approaches performing the line searches of minimum error value, it is suitable for this kind of problem due to its quadratic convergence property. The next step of this study is to incorporate the Neural Network Model acts as intelligent component into the Intelligent Decision Support System to help paddy farmers to predict yield based on the factors affecting yield.

Acknowledgement

We are indebted to Ministry of Science Technology and Environment (MOSTE), Malaysia for providing funds to carry out this research project under the IRPA Grant No: 04-02-06-0066 EA001 with the title of "Development of Intelligent Decision Support System for Rice Yield Prediction in Precision Farming". We are also indebted to KUKUM for providing funds under the short term research grant with the title of "The development of Neural Network Learning Algorithm to forecast crop yield and recognition of 2-D images."

References

[1] Ahmed, W.A.M., Saad, E.S.M. and Aziz, E.S.A. 2001. Modified back propagation algorithm for learning artificial neural networks. In Proceedings of the Eighteenth National Radio Science Conference NRSC 2001. 345 – 352.

[2] Charalambous, C. 1992. Conjugate gradient algorithm for efficient training of artificial neural networks. In Proceedings of the IEEE. **139** (3). 301-310.

[3] Hagan, M.T. and Menhaj, M.B. 1994. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*. **5** (6). 989 – 993.

[4] Looney, C.G. 1997. *Pattern Recognition Using Neural Networks – Theory and Algorithms for Engineers and Scientists*. New York: Oxford University Press.

[5] Luh, P.B. and Li Zhang. 1999. A novel neural learning algorithm for multilayer perceptrons. In Proceedings of the Int. Joint Conf. on Neural Networks IJCNN '99. **3**. 1696 – 1701.

[6] Marquardt, D. 1963. An algorithm for least squares estimation of non-linear parameters. *J. Soc. Ind. Appl. Math*. 431-441.

[7] Ng, S.C. and Leung, S.H. 2001. On solving the local minima problem of adaptive learning by using deterministic weight evolution algorithm. In Proceedings. of the 2001 Congress on Evolutionary Computation. **1**. 251 – 255.

[8] Sidani, A. and Sidani, T. 1994. A comprehensive study of the backpropagation algorithm and modifications. Southcon/94. Conference Record. 80 – 84.

[9] Wen, J., Zhao, J.L., Luo, S.W. and Han, Z. 2000. The improvements of BP neural network learning algorithm. In Proceedings of the 5th Int. Conf. on Signal Processing WCCC-ICSP 2000. **3**. 1647 – 1649.