ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 9(31), DOI: 10.17485/ijst/2016/v9i31/84278 , August 2016 ISSN (Online) : 0974-5645

Ridge Regression using Artificial Neural Network

Ali Muayad* **and A. N. Irtefaa**

University of Kufa, Iraq; alikhalefa2010@gmail.com, irtefaa.radhi@uokufa.edu.iq

Abstract

In this paper, a new suggested method using Ridge Neural Network (RNN) is presented to improve estimation based on using Ridge Regression method (RR). We compared between the proposed method and the existing back propagation algorithm of Artificial Neural Network (ANNs) by using Mean Square Error (MSE). The approach has been simulated using MATLAB. The results showed that the suggested method has the good performance in the sense that RNN method gives less error.

Keywords: Artificial Neural Network, Back Propagation, Estimation, MATLAB, Mean Square Error, Ridge Regression, Simulation

1. Introduction

A main branch of statistical inference is estimation. Statistical inference focus on doing some general statistical expectations about the population, usually described as one of the statistical distributions, and, in turn, rely on some features that distinguish this population to get some estimates for the model of population.

The strategy to calculate an estimate or approximation of the model is called the estimation.

One major problem in pattern recognition is how the mathematical model be learned from data[1]. To amend the estimation process, we may use computation algorithms to estimation the mathematical model from the data. One of the most important algorithms in the computational science is artificial neural network (back propagation algorithm of artificial neural network).

There are many popular techniques to estimate the nonparametric model. The alternative strategy for nonparametric models estimation is ANN[2]; given the universal approximation property[3].

The classical algorithm which have famous is ridge regression. Recently, it has been widely used in some papers such as Drucker[12,13].

In this study, we improve the strategy of the model estimation, by combining one of the important artificial neural network algorithms is back propagation algorithm. With ridge regression, this strategy is based on the estimation. The model which was used in the simulation is defined as:

$$f(x) = |x|^{\frac{2}{3}}$$

Random samples have been generated by MATLAB.

To design the human brain processes numerical, it can be used the Artificial Neural Networks for this purpose. The neural networks begin in early history, in the 1940's. That gives us the history for early programming of electronic computers[4].

Neural Networks have entered into lots of applications such as classification, pattern completion, optimization, feature detection, data compression, approximation, association, prediction, control, etc[5]. Neural networks are also in use in statistics for regression, estimation etc. The connection between the artificial and the real thing is also investigated and explained. Finally, the mathematical models involved in artificial neural network are presented and demonstrated. Now, every phase field has major fields, and the most important stages underlying accuracy of the results is the stage of estimating model, and have proposed several methods and statistical methods to estimate the unknown model such as ridge regression.

However, the development in the pattern of the models estimation will appear in the simulation part of this paper.

Author for correspondence

2. Artificial Neural Network (ANN)

To design the human brain processes as numerical models, it can be used the Artificial Neural Networks for this purpose. These models are used mainly for classification issues such as recognition pattern. Neural networks have entered into a lot of applications; some of the most popular applications are classification, pattern completion, optimization, feature detection, data compression, approximation, association, prediction, control, etc[5]. One of the important applications of neural networks is to approximate functions. The task of function approximation is to find an estimator $\hat{f}$ of f [6]. The most used networks in approximation is the feed-forward back-propagation network[7]. Let us consider the net architecture presented in Figure 1. Although the nodes are organized in two layers, we indicate by the subscripts i, h and o the input, hidden and output layers of the neural network in Figure 1. The input to any neuron is labeled I and its output is O, the inputs are normalized between 0 and 1. The input units distribute the values to the hidden-layer units. The net input to the hidden unit is in mathematical terms, we may describe a neuron k by writing the following pair of equations:

$$I_k = \sum_{i=1}^{N} W_{ik} X_i + \theta \tag{1}$$

$$h_k = \varphi(I_k) \tag{2}$$

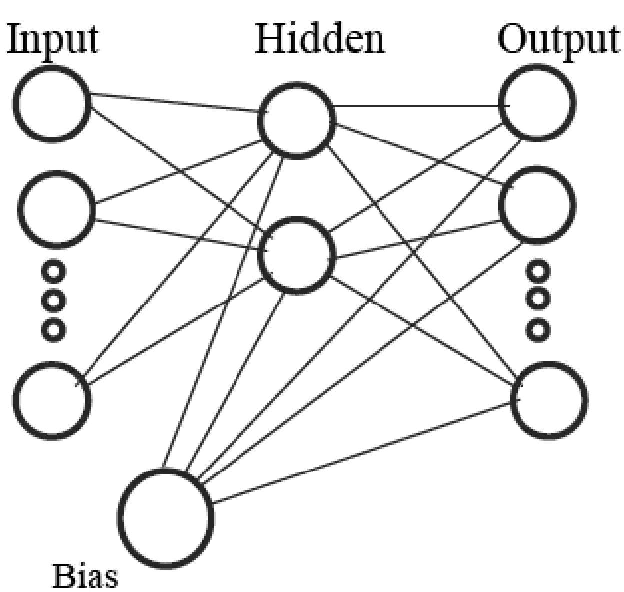


Figure 1. Multilayer neural network architecture.

where, the activation function $\phi(I)$ is called sigmoid (logistic). The sigmoid function is the most common form of activation function used in the construction of Artificial Neural Networks.

$$\phi(I) = \frac{1}{1+e^{-I}} \tag{3}$$

For each one of the input member W_i is called the weight, and θ is the bias.

The equations for the output nodes are:

$$O_k = \sum_{i=1}^{N} W_{ik} X_i + \theta \tag{4}$$

$$\phi(O) = \frac{1}{1+e^{-O}} \tag{5}$$

Where, the "O" is quantities for the output layer.

The error vector is $\delta_{pk} = (t_{pk} - O_{pk})$, where the desired target is t_{pk} and the actual output is O_{pk}, where the subscript "p" refers to the p^{th} training vector. The error to be minimized is the sum of the squares of the errors for all output units:

$$E_p = \frac{1}{2}\sum_{k=1}^{N} \delta_{pk}^{\ 2} \tag{6}$$

The factor of $\frac{1}{2}$ in Equation (6) is there for convenience in calculating derivatives later, since an arbitrary constant will appear in the final result. The presence of this factor does not invalidate the derivation.

To determine the direction in which to change the weights, we calculate the appropriate partial derivatives to determine how E varies with respect to each weight. The weights are then modified in proportion to $\frac{\partial E}{\partial W}$,

$$\Delta W = -\eta \frac{\partial E}{\partial W} \tag{7}$$

The factor η is called the learning-rate parameter; to updated weights for the output layer are:

$$W_k(r+1) = W_k(r) + \eta \delta_{pk} h_p \tag{8}$$

where,

$$\delta_{pk} = \delta_{pk} f'(I_{pk}) \tag{9}$$

For the updating the weights of hidden layer, we will replicate the same computations type for the same hidden layer as we did. We have,

$$\delta_{pj} = f'_{pj} \sum_k \delta_{pk} W_{kj} \quad (10)$$

So, the updated weights are:

$$W_{ji}(r+1) = W_{ji}(r) + \eta \delta_{pj} X_i \quad (11)$$

To make the first update for the processing, so we should repeat the previous steps to when we get less error[8].

3. Ridge Regression Method (RR)

The amendment of the Classical Least Squares method (OLS) that lets the regression coefficients for the biased estimators. To get a nearest values of estimated parameters, it must be used the unbiased estimator rather than busied with smaller MSE of regression coefficients.

Consider the standard model for multiple linear regression:

$$Y = X\beta + \square \quad (12)$$

The (n × 1) vector of the dependent variable values is represent as Y, while X is a matrix with size (n × p), P predictor variables values, and this matrix (1 × p), $\square$ is an (n × 1) vector of random variables, and β is a (p×1) vector of unknown regression coefficients.

Let $\hat{\beta}_{Rid}$ be the ridge estimator, then for any k ≥ 0

$$\hat{\beta}_{Rid} = \left(X'X + KI\right)^{-1}.X'Y \quad (13)$$

Where, I is identity matrix and the K value can be computed by:

$$K = \frac{ps_e^2}{\hat{\beta}'_{Ols}\,\hat{\beta}_{Ols}} \quad (14)$$

Where, $\hat{\beta}_{Ols}$ is the vector of estimators by (OLS)

$$\hat{\beta}_{Ols} = \left(X'X\right)^{-1}.X'Y \quad (15)$$

and

$$s_e^2 = \frac{\sum_i^n e_i^2}{n-p} \quad (16)$$

Where, e^2 is the mean squared error by (OLS)[9,10].

4. The Suggested Method (Ridge Neural Network)

The suggested method is based on ridge regression using artificial neural network; called Ridge Neural Network (RNN). In this method we combine between two methods: The ridge regression and back propagation algorithm of the Artificial Neural Network.

Firstly, the target in this context is:

$$t = f(x) \quad (17)$$

So, we want to calculate the output ($\hat{t} = \hat{f}(x)$), that is, to estimate the target.

After applying the ridge regression strategy to find ($\hat{B}$) as in Equation (19) below, we use back propagation algorithm of the neural network strategy to find output. The update of weights will be controlled as:

$$\delta = (t - \hat{t}) + \hat{B} * (1 - \hat{t}) * \hat{t} \quad (18)$$

Where,

$$\hat{B} = \left(X'X + KI\right)^{-1}.X'Y \quad (19)$$

5. Simulation Study

In this section, the new method is trained with ridge regression and the back propagation algorithm of Artificial Neural Network with different sample sizes (n = 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).

It was used to approximate the function:

$$y = |x|^{\frac{2}{3}} \quad (20)$$

This function was proposed in[11].

Firstly, we generate a random sample as an independent variable in the range [0,1].

For a comparison among the above three methods we use a neural network architecture with one of hidden layer using two neurons.

In addition, the Sigmund function is one of the types of activation functions, which is used in this study.

We can see both multilayer neural network architecture and a typical hidden or output layer neuron in Figure 1 and Figure 2, respectively.

The best method is chosen based on the minimum Mean Square Error (MSE):

$$MSE = \sum_{i=1}^{N} (t_i - \hat{t}_i)^2 \quad (21)$$

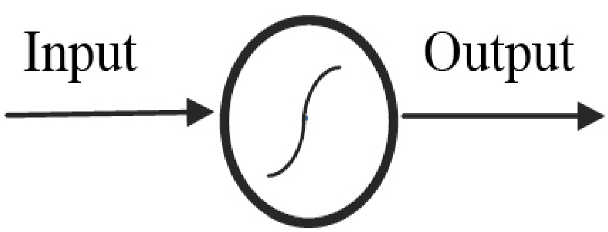
We can represent this simulation in the next flowchart to explain the strategy of steps of this simulation.

6. Results

We simulate back propagation algorithm of ANNs, RR and RNN for each of the above ten different sample sizes. The suggested method (RNN) is compared with others by MSE. The details of this comparison are shown in Table 1:

From above Table, we find that the suggested method RNN has less Mean Square Error (MSE) than the other methods for all types of sample sizes. Plotting of this error is done using MATLAB as displayed in the next figures. Figure 4 shows a comparison of error surface plots between the RNN and ANN respectively.

Since Ridge Regression (RR) does not involve weights, we use the bar chart error to compare between RR and RNN in performance as shown in Figure 5.



$$f(x) = \frac{1}{1 + e^{-x}}$$

Figure 2. A typical hidden or output layer neuron.

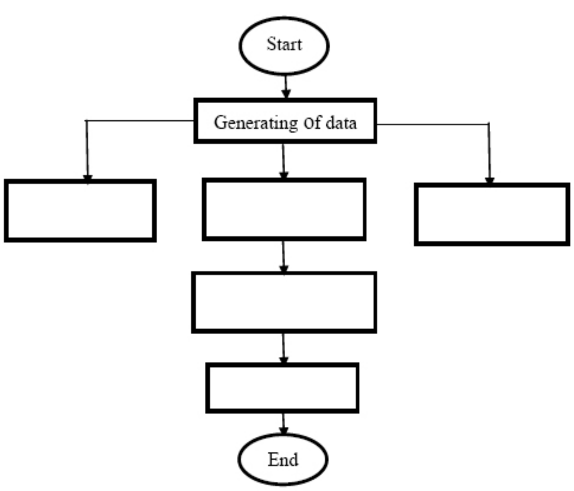


Figure 3. The flowchart of simulation strategy.

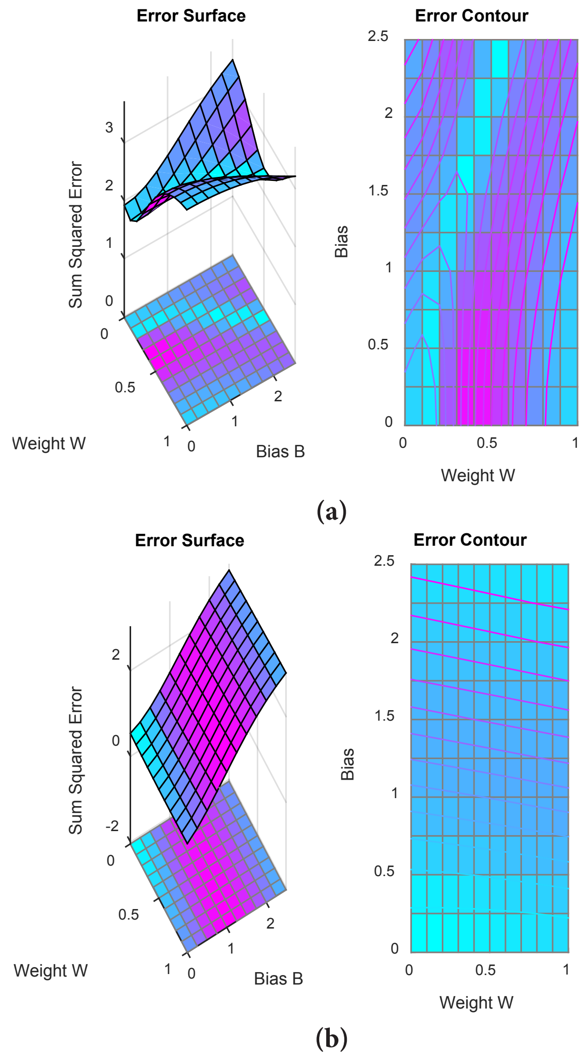
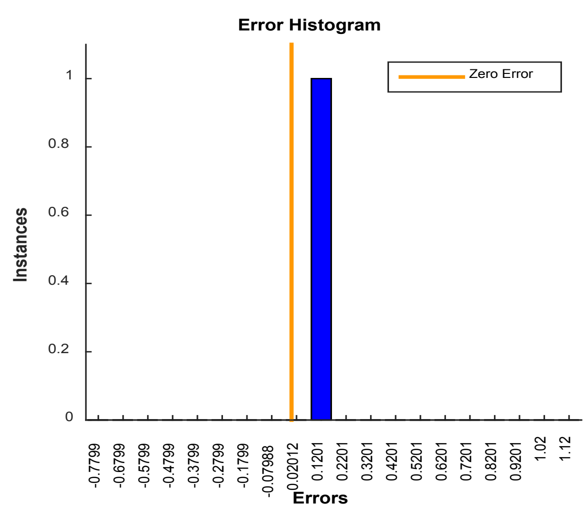
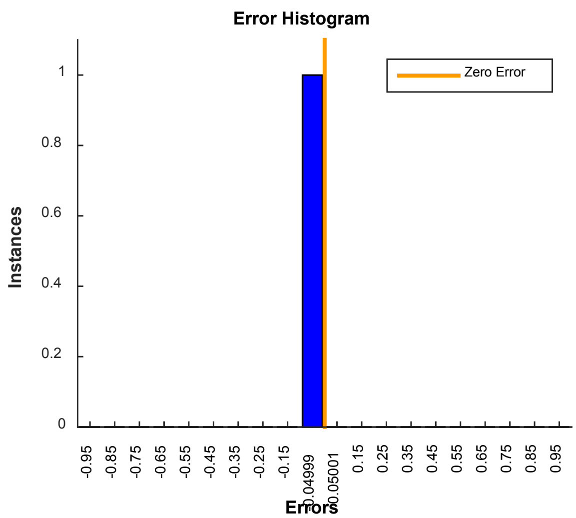


Figure 4. Error surface for. **(a)** RNN. **(b)** ANN.



(a)



(b)

Figure 5. Error histogram for. **(a)** RNN. **(b)** ANN.

Table 1. The MSE for ANN, RR and RNN comparison

	MSE				Sample size
	The new method (RNN)	**Ridge Regression Method (RR)**	**Artificial Neural Network Method (ANN)**		
	0.000008	0.0048	0.0003208	10%	
	0.000017	0.0031	0.0005396	20%	
	0.000026	0.0015	0.0013	30%	
	0.000034	0.0018	0.0017	40%	
0.000043	0.0020	0.0027	50%		
0.000052	0.0017	0.0037	60%		
0.000059	0.0012	0.0040	70%		
0.000068	0.0012	0.0048	80%		
0.000086	0.0010	0.0064	100%		

7. Conclusion

In this paper we proposed an estimation method based on ridge regression using Artificial Neural Networks (RNN). Performance of the proposed approach has been compared in the sense of getting the best estimate with two existing algorithms: Artificial Neural Network (ANN) and Ridge Regression (RR). Simulation results using MATLAB has shown that RNN outperforms other methods. The performance measure used in this comparison is the Mean Square Error (MSE), where RNN gave the least mean square error among the three methods.

8. References

1. Trentin E, Freno A. Unsupervised nonparametric density estimation. Proceedings of International Joint Conference on Neural Networks; USA. 2009.
2. Girosi F. An equivalence between sparce approximations and support vector machines. USA: Massachusetts Institute of Technology; 1997.
3. Kimeldorf G, Wahba G. Some results on tchebycheffian spline functions. Jounral of Mathematical Anylasis and Applications,USA. 1971; 33(1):82–95.
4. Kriesel D. A brief introduction to neural networks. Bonn, Germany; 2005. Available from: http://www.dkriesel.com/en/science/neural_networks
5. Fyfe C. Artificial Neural Network. USA: The University of Paisley; 1996.
6. Essai MTE, Mohammed H. From robust statistics to artificial intelligence: M-estimators for training feed-forward neural network. Al-Azhar Engineering 9th International Conference; Assiut University, Egypt. 2007.
7. Thomas M. Approximation of complex nonlinear functions by means of neural networks. Germany: Institute of Structural Mechanics, Bauhaus-University Weimar; 2005.
8. Freeman JA, Skapura DM. Neural networks,algorithms, applications,and programming techniques. USA: University of Houston; 1991.
9. Rashwan NI, El-Dereny M. Solving multicollinearity problem using ridge regression models. Int J Contemp Math Sciences. 2011; 6(12):585.
10. Hoerl E, Kennard RW. Ridge regression: Applications to nonorthogonal problems. Technometrics, American Statistical Association, JSOR, USA. 1970; 12(1):69–82.
11. Liano K. Robust error measure for supervised neural network learning with outliers. IEEE Trans Neural Networks,. 1996; 7:246–50.
12. Drucker H, Burges C, Kaufman L, Smola A, Vapnik VN. Support vector regression machines. Bell Labs and Monmouth University; 1996.
13. Saunders C, Gammerman A, Vovk V. Ridge regression learning algorithm in dual variables. Proceedings of the 15th International Conference on Machine Learning; USA. 1998.