International Journal of Electrical and Computer Engineering (IJECE)
Vol. 12, No. 1, February 2022, pp. 770~775
ISSN: 2088-8708, DOI: 10.11591/ijece.v12i1.pp770-775

Solving multiple linear regression problem using artificial neural network

Mohammad S. Khrisat, Ziad A. Alqadi
Computer Engineering Department, Faculty of Engineering Technology, Applied University, Amman, Jordan

Article Info

Article history:

Received Nov 14, 2020
Revised Jul 13, 2021
Accepted Jul 26, 2021

Keywords:

Artificial neural network
Convolutional artificial neural network
Feedforward artificial neural network
Multiple linear regression

ABSTRACT

Multiple linear regressions are an important tool used to find the relationship between a set of variables used in various scientific experiments. In this article we are going to introduce a simple method of solving a multiple rectilinear regressions (MLR) problem that uses an artificial neural network to find the accurate and expected output from MLR problem. Different artificial neural network (ANN) types with different architecture will be tested, the error between the target outputs and the calculated ANN outputs will be investigated. A recommendation of using a certain type of ANN based on the experimental results will be raised.

Corresponding Author:

Mohammad S. Khrisat
Computer Engineering Department, Faculty of Engineering Technology, Applied University
Amman 11134, P.O. Box 15008 Jordan
Email: mkhrisat@bau.edu.jo

1. INTRODUCTION

Multiple rectilinear regressions (MLR) [1] are the foremost common kind of linear regression analysis. As a prognostic analysis, it will not justify the link between one continuous dependent variable and 2 or a lot of freelance variables. The independent variables will be continuous or categorical (dummy coded as appropriate). MLR presented in (1) and Figure 1 can be solved using MATLAB Figure 2 [2]-[4].

$$y = a_0 + a_1x_1 + a_2x_2 + \cdots + a_nx_n \quad (1)$$

Where:
y = dependent variable
x = explanatory variables
a = constants

The coefficients will be as follows: a={5.0, 2.0, 3.5}, So MLR problem can be represented using (2).

$$y = 5 + 2x_1 + 3.5x_2 \quad (2)$$

Artificial neural network (ANN) [5]-[7] is a computational mathematical model, which contains a set of fully connected neurons, which are arranged in two or more layers. Each neuron as shown in Figure 3 implements two operations [8]-[10]: i) finding the sum of products of the inputs and the associated weights [11], [12], ii) according to the selected activation function calculate the neuron output.

X1	X2	Target output
2.0000	-3.0000	-1.5000
3.0000	-4.5000	-4.7500
-2.0000	-25.0000	-86.5000
4.0000	0	13.0000
-5.0000	9.5000	28.2500
0	10.0000	40.0000
9.0000	11.0000	61.5000
10.0000	2.0000	32.0000
15.0000	3.0000	45.5000
-11.0000	12.0000	25.0000
12.5000	-8.0000	2.0000
18.0000	-2.0000	34.0000
22.3000	0	49.6000
4.0000	14.0000	62.0000
0	5.0000	22.5000
-6.8000	11.0000	29.9000

Figure 1. MLR example

```
>> x1=[2 3 -2 4 -5 0 9 10 15 -11 12.5 18 22.3 4 0 -6.8];
x2=[-3 -4.5 -25 0 9.5 10 11 2 3 12 -8 -2 0 14 5 11]
yy=(5+2*x1+3.5*x2);
X = [ones(size(x1'))  x1'  x2'];
a = X\yy'
```

Figure 2. MATLAB code to solve MLR

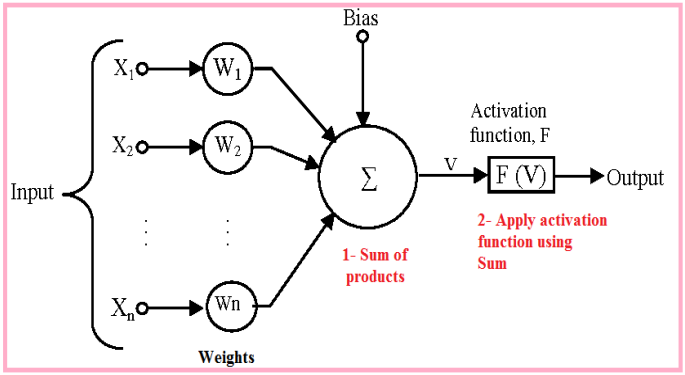

Figure 3. Neuron operations [13], [14]

Different types of ANN can be used to solve MLR problem, the first one is feed forward ANN with back propagation (FFANN), in this ANN as shown in Figure 4 each neuron in any layer are fully connected to the previous inputs [15]-[17]. The second type is cascade feed forward ANN (CFANN) [18], [19], in this type of ANN the inputs are used to feed the hidden layer as shown in Figure 5. The last type which will be investigated in this paper is Elman ANN (EANN), in this type as shown in Figure 6 a context inputs formed from the hidden layer are used as inputs [20], [21].

To use any type of ANN to solve MLR problem we have to follow the steps [22]-[24]: i) select the input data set, which is the values of explanatory variables; ii) select the target, which is the values of dependent variable; iii) create ANN, here we define the type of ANN, ANN architecture: How many layers are to be used, and how many neurons in each layer. In this step we have to define activation function for each layer, logsig or tansig for the input layer and the hidden layers, linear activation function for the output layer [25]; iv) initialize the weights; v) set the error parameter to zero; vi) select the number of training cycles; vii) train ANN; viii) save ANN, ix) run ANN to solve MLR problem.

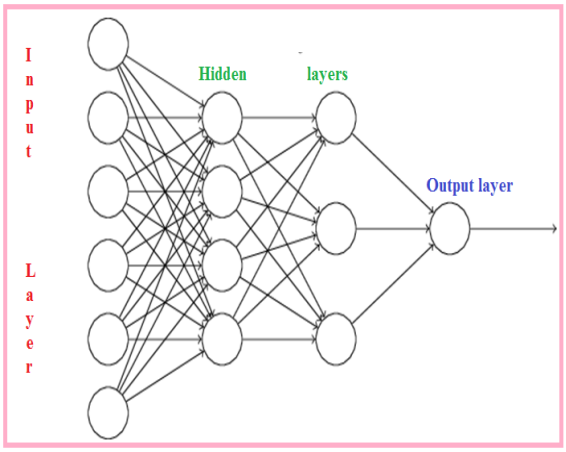

Figure 4. FFANN

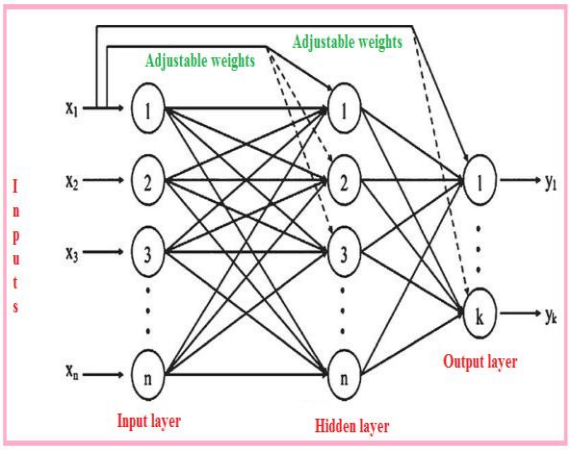

Figure 5. CFANN

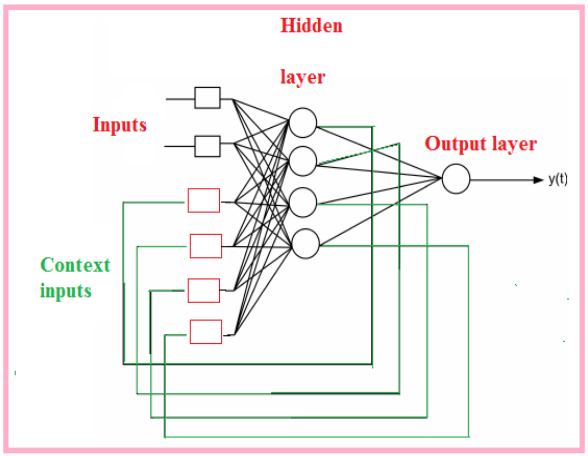

Figure 6. EANN

2. RESEARCH METHOD

Different types of ANN [11]-[13] with different architectures were used to solve MLR problem, some experiments were tested, and below we will investigate the results of these experiments:

Experiment 1:

Here we select a two layers ANN, input layer with two neurons and output layer with one neuron, Table 1 shows the obtained results of using various types of ANN: maximum error was calculated using (3).

$$MaxErr = \max\left(abs(y_1 - y)\right) \quad (3)$$

Figure 7 shows the obtained outputs for various ANN. Figures 8, 9, and 10 shows the error between the target and the calculated output of each ANN. From the obtained results here in this experiment we can see that CFANN has the best performance and we can recommend it to solve MLR problem.

Table 1. Experiment 1 results

ANN type	Number of layers	Neurons	Activation function	Max. Error	Training cycles
FFANN	2	2, 1	Logsig, linear	0.0079	10000
CFANN	**2**	**2, 1**	**Logsig, linear**	**0**	**9**
EANN	2	2, 1	Logsig, linear	31.1152	10000
FFANN	2	2, 1	Tansig, linear	0.0001059	10000
CFANN	**2**	**2, 1**	**Tansig, linear**	**0**	**10**
EANN	2	2, 1	Tansig, linear	24.7341	10000

Target	CF output	FF output	ELM output
-1.5000	-1.5000	-1.5000	-2.1458
-4.7500	-4.7500	-4.7500	-5.2117
-86.5000	-86.5000	-86.5000	-61.0822
13.0000	13.0000	13.0000	14.9681
28.2500	28.2500	28.2500	42.5609
40.0000	40.0000	40.0000	47.1758
61.5000	61.5000	61.5000	50.1927
32.0000	32.0000	32.0000	37.1704
45.5000	45.5000	45.5000	45.3666
25.0000	25.0000	25.0000	-5.3184
2.0000	2.0000	2.0000	-4.7442
34.0000	34.0000	34.0000	33.4190
49.6000	49.6000	49.6000	45.0598
62.0000	62.0000	62.0000	50.4441
22.5000	22.5000	22.5000	33.4453
29.9000	29.9000	29.9000	29.3088

Figure 7. Obtained outputs for each ANN

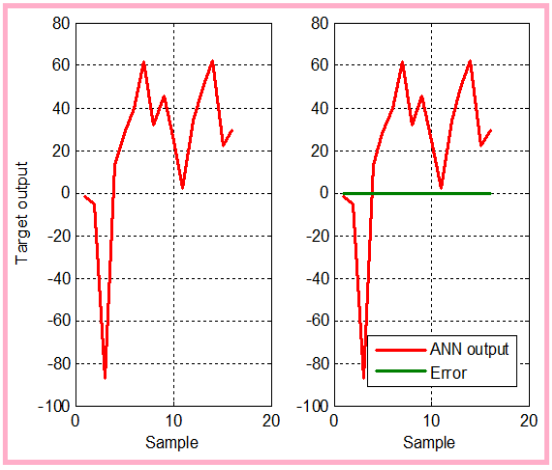

Figure 8. Calculated error for CFANN

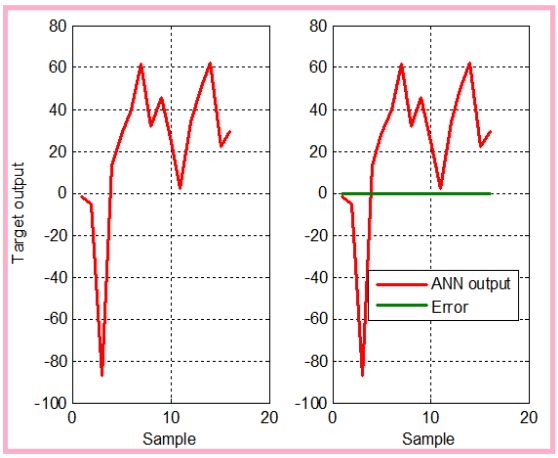

Figure 9. Calculated error for FFANN

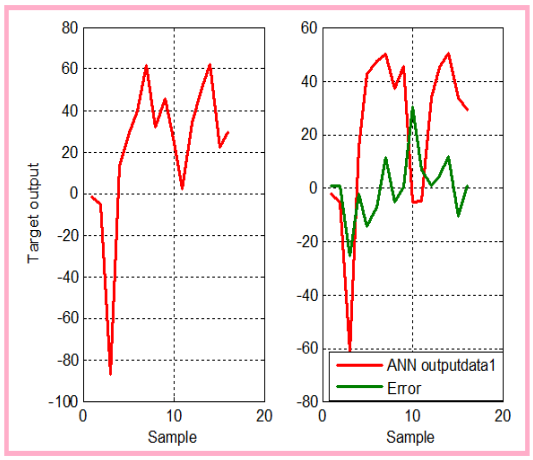

Figure 10. Calculated error for EANN

Experiment 2:

The architectures of ANNs here were changed to see how changing ANN architecture will affect the performance, Table 2 shows the experimental results. From Table 2 we can see that changing ANN architecture does not positively affect the performance; hence we can conclude that using CFANN with two layers is best solution of MLR problem, here the number of neurons in the input layer must equal to the number of explanatory variables. Figure 11 shows the calculated outputs of using CFANN to solve MLR problem with 4 explanatory variables (here the number of neurons in the input layer 4), while Figure 12 shows the error between the target and the calculated output.

Table 2. Experiment 2 results

ANN type	Number of layers	Neurons	Activation function	Max. Error	Training cycles
FFANN	3	2, 4, 1	Tansig, tansig, linear	0.00014588	10000
CFANN	**3**	**2, 4, 1**	**Tansig, tansig, linear**	**0**	**796**
EANN	3	2, 4, 1	Tansig, tansig, linear	55.5833	10000
FFANN	3	2, 4, 1	logsig, tansig, linear	0.00013311	10000
CFANN	**3**	**2, 4, 1**	**logsig, tansig, linear**	**0**	**1171**
EANN	3	2, 4, 1	logsig, tansig, linear	25.0369	10000
FFANN	3	2, 4, 1	logsig, tansig, linear	0.0048	10000
CFANN	**3**	**2, 4, 1**	**logsig, tansig, linear**	**0**	**114**
EANN	3	2, 4, 1	logsig, tansig, linear	58.0207	10000
FFANN	3	2, 4, 1	Tansig, logsig, linear	0.0034	10000
CFANN	**3**	**2, 4, 1**	**Tansig, logsig, linear**	**0**	**317**
EANN	3	2, 4, 1	Tansig, logsig, linear	29.515	10000

X1	X2	X3	X4	Target	CFANN output
2.0000	-3.0000	1.0000	-1.5000	4.3000	4.3000
3.0000	-4.5000	1.5000	-3.0000	4.8500	4.8500
-2.0000	-25.0000	-1.0000	-23.5000	-62.3000	-62.3000
4.0000	0	2.0000	1.5000	19.2000	19.2000
-5.0000	9.5000	-2.5000	11.0000	5.0500	5.0500
0	10.0000	0	11.5000	26.2000	26.2000
9.0000	11.0000	4.5000	12.5000	64.5000	64.5000
10.0000	2.0000	5.0000	3.5000	47.8000	47.8000
15.0000	3.0000	7.5000	4.5000	70.1000	70.1000
-11.0000	12.0000	-5.5000	13.5000	-13.2000	-13.2000
12.5000	-8.0000	6.2500	-6.5000	34.8000	34.8000
18.0000	-2.0000	9.0000	-0.5000	70.6000	70.6000
22.3000	0	11.1500	1.5000	92.4000	92.4000
4.0000	14.0000	2.0000	15.5000	51.4000	51.4000
0	5.0000	0	6.5000	14.7000	14.7000
-6.8000	11.0000	-3.4000	12.5000	1.3000	1.3000

Figure 11. Results of MLR with 4 variables

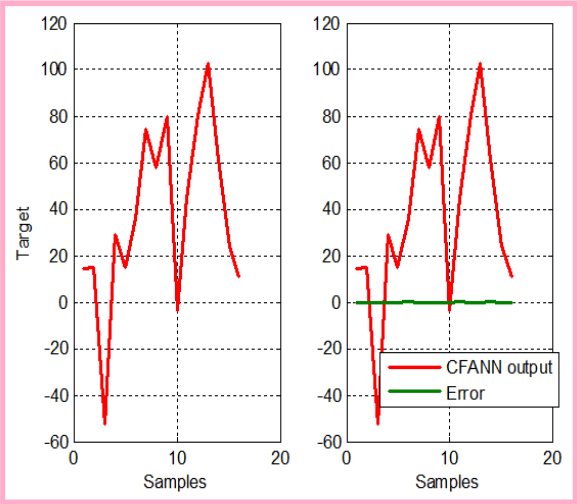

Figure 12. Calculated error for CFANN

3. CONCLUSION

Different types of ANNs with different architectures were used to solve MLR problem. Each ANN was treated with various input data set varying the activation function. It was seen that CFANN has the best performance, and here we can highly recommend CFANN to solve MLR problem with any number of explanatory variables by selecting two layers and adjusting the number of neurons in the input layer to match the number of explanatory variables in MLR problem. Provide a statement that what is expected, as stated in the "Introduction" chapter can ultimately result in "Results and Discussion" chapter, so there is compatibility. Moreover, it can also be added the prospect of the development of research results and application prospects of further studies into the next (based on result and discussion).

REFERENCES

[1] K. L. L. Khine and T. T. S. Nyunt, "Predictive geospatial analytics using principal component regression," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2651-2658, 2020, doi: 10.11591/ijece.v10i3.pp2651-2658.

[2] M. S. Rodrigo, "A system based on artificial neural networks for automatic classification of hydro-generator stator windings partial discharges," *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, vol. 16, no. 3, pp. 628-645, 2017, doi: 10.1590/2179-10742017v16i3854.

[3] M. Linveh and I. Ishai, "Using indicative properties to predict the density-moisture relationship of soil," *Transportation Research Record*, vol. 60P, pp. 22-28, 1978.

[4] N. Iksan, J. Sembiring, N. Hariyanto, and S. H. Supangkat, "Residential load event detection in NILM using robust cepstrum smoothing based method," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 2, pp. 742-752, 2019, doi: 10.11591/ijece.v9i2.pp742-752.
[5] F. Recknagela, M. Frenchb, and P. H. K. I. Yabunakad, "Artificial neural network approach for modelling and prediction of algal blooms," *Ecological Modelling*, vol. 96, no. 1-3, pp. 11-28, 1997, doi: 10.1016/S0304-3800(96)00049-X.
[6] Z. A. AlQadi, A. Y. Hindi, and O. D. Majed, "Procedures for speech recognition using LPC and ANN," *International Journal of Engineering Technology Research & Management*, vol. 4, no. 2, pp. 48-55, 2020.
[7] Z. Alqadi, M. S. Khrisat, Y. Eltous, S. A. Khawatreh, and M. O. Dwairi, "Building face recognition system (FRS)," *International Journal of Computer Science and Mobile Computing*, vol. 9, no. 6, pp. 15-24, 2020.
[8] J. Al-Azzeh N. Asad, Z. Alqadi, I. Shayeb, and Q. Jaber, "Simple procedures to create HSCS," *International Journal of Engineering Research and Management (IJERM)*, vol. 7, no. 5, pp. 6-10, 2020.
[9] DA. Hindi, M. O. Dwairi, and Z. Alqadi, "Analysis of procedures used to build an optimal fingerprint recognition system," *International Journal of Computer Science and Mobile Computing*, vol. 9, no. 2, pp. 21-37, 2020.
[10] H. E. Dytch and G. L. Wied, "Artificial neural networks and their use in quantitative pathology," *Analytical and Quantitative Cytology and Histology*, vol. 12, no. 6, pp. 379-393, 1990.
[11] M. A. Mohamad, H. Haron, and N. A. Ali, "Prediction of hardness in titanium aluminium nitride TiA1N coating process: A review," *2012 Fourth International Conference on Computational Intelligence, Modelling and Simulation*, pp. 111-116, 2012, doi: 10.1109/CIMSim.2012.63.
[12] U. Bai, Y. Li, Y. Liu, and Z. Ma, "Short-term prediction of distribution network faults based on support vector machine," *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2017, pp. 1421-1426, doi: 10.1109/ICIEA.2017.8283062.
[13] H. Yi, D. Ding, M. Lu, and Li. Li, "Fault location based on relevance vector machine and decision directed acyclic graph," *2013 25th Chinese Control and Decision Conference (CCDC)*, 2013, pp. 4679-4684, doi: 10.1109/CCDC.2013.6561780.
[14] Y. Temprado, F. J. Molinero, C. Garcia, and J. Gomez, "Knowledge discovery from trouble ticketing reports in a large telecommunication company," *2008 International Conference on Computational Intelligence for Modelling Control & Automation*, 2008, pp. 37-42, doi: 10.1109/CIMCA.2008.116.
[15] P. Kayal, S. Chanda, and C. K. Chanda, "An ANN based network reconfiguration approach for voltage stability improvement of distribution network," *2011 International Conference on Power and Energy Systems*, 2011, pp. 1-7, doi: 10.1109/ICPES.2011.6156643.
[16] F. J. Anscombe and J. W. Tukey, "The examination and analysis of residuals" *Technometrics*, vol. 5, no. 2, pp. 141-160, 1963, doi: 10.2307/1266059.
[17] R. A. Becker, J. M. Chambers, and A. R. Wilks, "The new S language a programming environment for data analysis and graphics," *Pacific Grove, CA: Wadsworth and Brooks/Cole*, 1988.
[18] J. Chang and D. J. Olive, "OLS for 1D regression models," *Communications in statistics: Theory and methods*, vol. 39, no. 10, 1869-1882, 2010, doi: 10.1080/03610920902923494.
[19] R. D. Cook, "Regression graphics: Ideas for studying regression through graphics," *New York, NY: Wiley*, 1998.
[20] S. Ghosh, "Note on a common error in regression diagnostics using residual plots," *The American Statistician*, vol. 41, pp. 338, 1987.
[21] H. L. Harter, "The method of least squares and some alternatives: Part IV," *International Statistical Review*, vol. 43, no. 2, pp. 125-190, 1975, doi: 10.2307/1402897.
[22] G. Ames, D. Witten, T. Hastie, and R. Tibshirani, "An introduction to statistical learning with applications," *R. New York, NY: Springer*, vol. 103, 2013.
[23] J. Lei and L. Wasserman, "Distribution free prediction bands for non-parametric regression," *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, vol. 76, no. 1, pp. 71-96, 2014.
[24] R. Bandi, J. Amudhavel, and R. Karthik, "Machine learning with pyspark-review," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 12, no. 1, pp. 102-106, 2018, doi: 10.11591/ijeecs.v12.i1.pp102-106.
[25] L. S. Mezher, "Design and implementation hamming neural network with VHDL," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 19, no. 3, pp. 1469-1479, 2020, doi: 10.11591/ijeecs.v19.i3.pp1469-1479.