International Journal of Control and Automation
Vol.9, No.3 (2016), pp.15-22
http://dx.doi.org/10.14257/ijca.2016.9.3.02

Estimated Energy Consumption for a Building Based on Weather and Time Conditions Using Neural Networks

Nicolás Marrugo[1], Jorge Ospina[2] and Darío Amaya[3]

[1]Research assistant, [2]Research assistant, [3]Main researcher of GAV research group of Nueva Granada Military University

{[1]tmp.nicolas.marrugo, [2]u1801423, [3]dario.amaya}@unimilitar.edu.co

Abstract

When looking for a way to develop a program with artificial intelligent, the most common related concept is the artificial neural network (ANN). Due to being a computational model of a biological neural network that can be used for simulating, processing or prediction. This article have as objective the mathematical implementation of a feedforward neural network in C# programming language. To achieve the previous, the behavior of a MATLAB ® neural network was analyzed, with the purpose of making the data adjustment that allows identify the energy consumption for a building from weather and time variables.

Keywords*: C#, data adjustment, feedforward neural network, MATLAB*

1. Introduction

Artificial neural networks (ANN) form part of the intelligent artificial field, due to being a calculation model characterized for having very efficient algorithms that operate massively in parallel. This structure is appropriate to do cognitive tasks such as patterns learning, classification or optimization [1-3].

Neural networks are commonly used nowadays, even though they date from the forties and fifties, when the first concepts and developments by Warren Mc Coulloc (neurophysiologist) and Walter Pitts (Mathematical) began being published [4]. Their concepts were not successful at the time, because the computational resources needed for obtaining an ANN with good results were very high [5-6].

Taking into account the actual level of processing and the usefulness of the ANN, is very common to find specialized toolbox to design, train and simulate an artificial neural network on different programming languages [7-9].

The artificial neural network consists of neurons which are equivalent to biological neurons[10]. The concept of artificial neuron is defined as a system which is able to do tasks of automation classification by learning through a number of tagged examples and determined the equation applicable to these examples [11-12].

The interconnection of artificial neurons can change depending of the architecture of the neural network. Nowadays, different architectures exist but for this particular case, the feedforward neural network is analyzed. Which is considered as the most studied at scientific level and it is the most useful in many application fields. One of them is the estimation, analysis and prediction of meteorological variables [13-14].

The study of meteorological variables, especially for urban zones allows studying and estimating the energy consumption of a building from present climate conditions. Amongst the most common variables analyzed are the humidity, temperature, wind and solar radiation, which can be related with the building consumption in illumination and air conditioning [15-16].

Taking the previous into account, this paper proposes the mathematical implementation of a feedforward neural network in C#, with which data adjustment

ISSN: 2005-4297 IJCA

can be done for the measurements of humidity, temperature, wind and solar radiation in the area. With the purpose of giving the user an estimate of building energy consumption from these factors. Obtaining as a result, a support tool for prediction programs of meteorological variables or energy consumption without the use of specialized toolbox.

2. Methodology

In this particular case, the database used for the training of the feedforward neural network is based on the study done by the Oak Ridge National Laboratory, located in Oak Ridge, Tennessee [18]. This study analyzed the meteorological factors of dry bulb temperature, global horizontal irradiance, wind speed, dew point temperature, wind direction and relative humidity for the year 2010, and their relation with the energy consumption of a building. The average values are shown in the table 1.

Table 1. Average Values of Meteorological Factors from Oak Ridge for the Year 2010

VARIABLE	AVERAGE VALUE
Temperature (°C)	14,65
Humidity relative (%)	69,13
Global horizontal irradiance (W/m^2)	183,71
Wind speed (m/s)	1,48
Direct irradiance (W/m^2)	182,61
Diffuse irradiance (W/m^2)	72,7
Wind direction (°)	192,47

By using the METEONORM software, a database for the previously mentioned location, for the year 2010 was rebuild. Once the example database is obtained, it relates the meteorological and time factors as inputs, and the energy consumption as the output.

The next step is the construction, training and evaluation of the feedforward neural network for its later implementation in C#, following the process shown in figure 1.

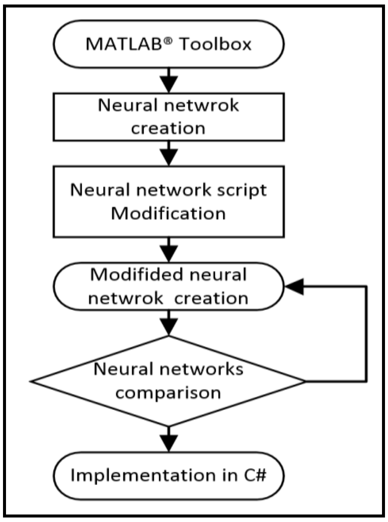

Figure 1. Methodology for the Implementation of the Neural Network in C#

The feedforward neural network was created using the neural network toolbox of MATLAB®, *nftool*, in which the database with the example data was uploaded. The previous was done by applying the architecture shown in figure 2.

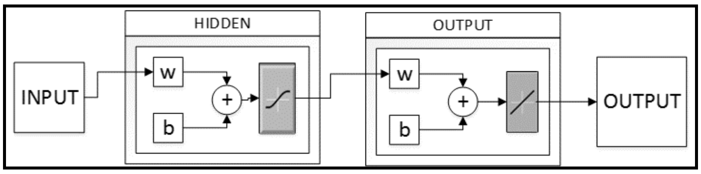

Figure 2. Neural Network Architecture Apply in MATLAB®

From the architecture, the hidden layer (HL) can be mathematically deduced, resulting in equation 1.

$$HL = tansig\,(Hw * input + Hb) \tag{1}$$

Where Hw represents the weights of the hidden layer and Hb represents the bias of the hidden layer. Subsequently, the output layer can also be deduced mathematically (OL). Its representation is shown in equation 2.

$$OL = Ow * HL + Ob \tag{2}$$

Where Ow represents the weights of the output layer and Ob represents the bias of the output layer. In order to do the data adjustment, each layer has their own activation function. In this case, the hidden layer has a hyperbolic tangential function and the output layer has a linear function.

The hyperbolic tangential (tansig) is the most used function in the neural networks. It was chosen as the activation function for the hidden layer, due to its flexibility and the range of the results it manages, being between 1 and -1 [19, 20]. In equation 3, the mathematical expression of tansig is shown.

$$tansig = \frac{2}{1 + exp^{(-2*input)}} - 1 \tag{3}$$

After the feedforward neural network in MATLAB ® is generated, the construction script of neural network is then made. This script must be modified due to MATLAB® performing functions of pre-processing and post-processing of the input and output signals before initializing the neural network training. These functions are remove using the following code lines: "*net.inputs{1}.processFcns={};*" for inputs and "*net.outputs{2}.processFcns={};*" for outputs [21].

By executing the previous lines, a modified neural network was created. This ANN will be implemented mathematically in C#. The result of equation 2 was compared with the ANN for the same input.

For the implementation of equation 2 in C#, it is necessary to import the matrices of the modified neural network from MATLAB® to C#. In this case, the IW{1} matrix that corresponds to the weights of the hidden layer, LW{2} that corresponds to the weights of the output layer, b{1} that corresponds to the bias of the hidden layer and b{2} that corresponds to the bias of the output layer were imported.

Similarly, the example database was used as well to construct, train and simulate a neural network using the most common toolboxes of C#. In this case, the AFORGE.net and ACCORD.net were used, with the purpose of comparing their results with the C# version of equation 2.

3. Results

Due to MATLAB® being used specifically for mathematical processing, it is the best option to construct, train and simulate a neural network. Therefore, following the previous methodology, the differences between the *nftool* neural network and modified neural network were identified and shown in the table 2.

Table 2. Comparison of Training between the Nftool ANN and the Modified ANN

ANN	*Nftool* ANN	Modified ANN
Time (S)	106	80
Epoch (#)	29	42
Data adjustment (%)	95,87	95,64
Neurons number (#)	100	100
Processing memory (Gb)	1.5	2.3
Number of trainings (#)	1 - 5	3 - 8

From table 2, it can be observe, that in order to get a data adjustment of 95%, a higher number of trainings was necessary for the modified ANN. Despite of having the same number of neurons, the modified ANN had a higher computational consumption but minor training time than the *nftool* network. Subsequently, the tests between the two neural networks were made through the evaluation of each database input. The results are shown in figure 3.

Due to the data adjustment, is normal that the networks presents a few peaks in error, as shown in figure 3. It proves that both neural networks have a similar behavior. Additionally is possible to observe that the modified neural network has an average error of 5%.

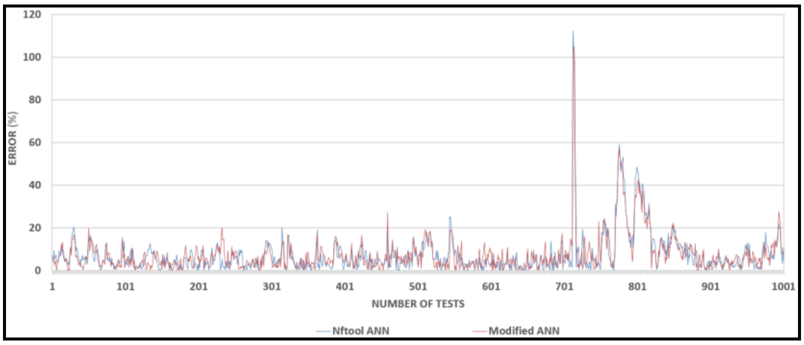

Figure 3. Error behavior for the Nftool ANN and modified ANN

Once validated the modified neural network, it was implemented in C# with the purpose of making the data adjustment of the example database. Which related the meteorological and time variables with the energy consumption. This means, that by having as input of environment factors and time for a specific zone of database, the ANN is able to estimate how much energy is consumed.

As the validation method, a test with 110 different inputs was made. The energy consumption obtained by the ANN was compared with the average energy consumption of the example database, achieving the error shown in figure 4.

The ANN in C# presents an average error of 3, 7%, in relation with the average energy consumption of the example database. Moreover, the comparison between the mathematical implementation of a feedforward neural network and the most common toolbox use in C# was made. The results obtained are shown in table 3.

Table 3. Comparison of Training between Toolbox of C# and Modified ANN

Parameters	AFORGE	ACCORD	Modified ANN
Time (S)	--	--	80
Epoch (#)	40000	40000	42
Data adjustment (%)	98	98	95,64
Neurons numbers (#)	100	100	100
Processing memory (Gb)	>3.5	>3.5	2.3
Number of trainings (#)	>3	>3	03-ago

In the C# toolbox, the error, number of epoch and the number of neurons of the neural network must be set by the user. Due to the toolbox not having specialized methods and the example database being of a considerable size, the process of training requires 3 times more of computational consumption than one used by MATLAB® for the same process. This can be seen in table 3, in the processing memory row.

Taking into account, the amount of decimals used in the data, a moderate capacity of processing was required in order to have a precision of four decimals. Due to only having a 4 GB of RAM memory for processing, the neural network can achieve an error of 7% or higher that corresponds to a data adjustment of 75% or minor. For this reason, it was not possible to use the toolbox in C# for the training of the neuronal network.

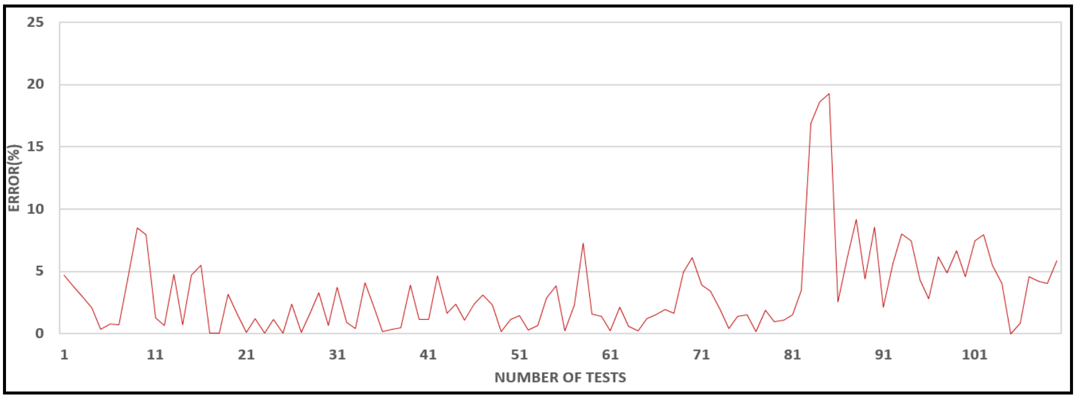

Figure 4. Feed forward Neural Network Error in C#

Conclusions

The use of a mathematical implementation of a feedforward neural network in a C# program, leads to save a 28, 5% in computational consumption compared with the specialized toolbox AFORGE and ACCORD.

Since the implemented ANN in C# had the same precision as the artificial neural network created in MATLAB®, the data adjustment done for the example database presented an average error of 4%, which is very low even for meteorological variables.

Finally, with this work, it was demonstrated that the trained ANN could be integrated in estimation or prediction systems. Due to the meteorological and time factors involved in this study, it was possible to ratify the direct relationship that exist between building energy consumption and factors such as the day, hour and warm or cool day. Proving that the use of this extra information generates more accurate data than the one from common studies of building energy consumption.

Acknowledgement

The authors would like to express their special gratitude to the Research Vice-chancellorship of Nueva Granada Military University for financing the research project IMP-ING 1576 titled: *"Abastecimiento energético autónomo mediante SFRC para áreas de laboratorio en el campus de Cajicá "*, 2015.

References

[1] R.F. López and J.M.F. Fernández, “Las Redes Neuronales Artificiales”, Netbiblo, **(2008)**.
[2] F. de M. Anegón, V.H. Solana and V.G. Bote, “La aplicación de Redes Neuronales Artificiales (RNA): a la recuperación de la información”, Bibliodoc: anuari de biblioteconomia, documentació i informació, **(1998)**, pp. 147-164.
[3] J. Párraga, “Implementación en C# de una Red Neuronal Artificial en el control locomotor de una hormiga robótica”, Revista Tecnológica - ESPOL, vol. 26, no. 2, December **(2013)**.
[4] S.N. Sivanandam and S.N. Deepa, “Introduction to Neural Networks Using Matlab 6.0”, Tata McGraw-Hill Education, **(2006)**.
[5] J. Trujillano, J. March and A. Sorribas, “Aproximación metodológica al uso de redes neuronales artificiales para la predicción de resultados en medicina”, Med Clin (Barc), vol. 122, no. s1, **(2004)**.

[6] J.G. Mora-Esperanza, "La inteligencia artificial aplicada a la valoración de inmuebles: un ejemplo para valorar Madrid", CT: Catastro, no. 50, **(2004)**, pp. 51-68.
[7] D. Ballabio, V. Consonni and R. Todeschini, "The Kohonen and CP-ANN toolbox: A collection of MATLAB modules for Self Organizing Maps and Counterpropagation Artificial Neural Networks", Chemometrics and Intelligent Laboratory Systems, vol. 98, no. 2, **(2009)**, pp. 115-122.
[8] Y. Huang, H. Yan, L. Feng and Q. Wu, "Vehicle Production Forecasting Based on BPNN and Interface Between C# .NET and Matlab [J]", Computer Technology and Development, vol. 11, **(2008)**, p. 011.
[9] X. Wang, S. Cao and H. Ding, "Application of Method of Calling Matlab Neural Network Toolbox in VB to Surveying Software Design", Bulletin of Surveying and Mapping, vol. 11, **(2011)**.
[10] M.C. Aristizábal, "Evaluación asimétrica de una red neuronal artificial: aplicación al caso de la inflación en Colombia", Lecturas de Economía, vol. 65, no. 65, **(2006)**, pp. 73-116.
[11] H. Tabares, J. Branch, and J. Valencia, "Generación dinámica de la topología de una red neuronal artificial del tipo perceptron multicapa", Revista Facultad de Ingeniería, vol. 38, **(2006)**, pp. 146-162.
[12] J.-Z. Wang, X.-J. Wang and H.-G. Ni, "An Algorithm of Neural Network and Application to Data Processing in Concrete Engineering", Informatica, Lith. Acad. Sci., vol. 14, no. 1, **(2003)**, pp. 95-110.
[13] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks", in 2004 IEEE International Joint Conference on Neural Networks, vol. 2, **(2004)**, pp. 985-990.
[14] G.-B. Huang, L. Chen and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes", IEEE Transactions on Neural Networks, vol. 17, no. 4, **(2006)**, pp. 879-892.
[15] S.L. Wong, K.K.W. Wan and T. N. T. Lam, "Artificial neural networks for energy analysis of office buildings with daylighting", Applied Energy, vol. 87, no. 2, **(2010)**, pp. 551-557.
[16] N. Kandil, R. Wamkeue, M. Saad and S. Georges, "An efficient approach for short term load forecasting using artificial neural networks", International Journal of Electrical Power & Energy Systems, vol. 28, no. 8, **(2006)**, pp. 525-530.
[17] A.K. Yadav and S.S. Chandel, "Solar energy potential assessment of western Himalayan Indian state of Himachal Pradesh using J48 algorithm of WEKA in ANN based prediction model", Renewable Energy, vol. 75, **(2015)**, pp. 675-693.
[18] M. Bhandari, S. Shrestha and J. New, "Evaluation of weather datasets for building energy simulation", Energy and Buildings, vol. 49, **(2012)**, pp. 109-118.
[19] I. Sahin and I. Koyuncu, "Design and Implementation of Neural Networks Neurons with RadBas, LogSig, and TanSig Activation Functions on FPGA", Electronics and Electrical Engineering, vol. 120, no. 4, April **(2012)**.
[20] D.E. Khodja, A. Kheldoun and L. Refoufi, "Sigmoid function approximation for ANN implementation in FPGA devices", in Procedure of the 9th WSEAS International Conference On Circuits, Systems, Electronics, Control, and Signal Processing, Stevens point, WI, **(2010)**.
[21] O.O. David and A.T. Chioma, "Artificial neural network for the modulus of rupture of concrete", Advances in Applied Sciences Research, vol. 4, no. 4, **(2013)**, pp. 214-223.

Authors

Nicolas Marrugo, He received his B.Sc. degree in mechatronics engineering from Piloto University, Colombia, in 2014. Currently, he is a research assistant in GAV research group at Nueva Granada Military University. His research interests include renewal energies, automation system.

Jorge Ospina, He Studied at UMNG (Nueva Granada Military University), Bogota, Colombia, where he received his B Sc. degree in Mechatronics Engineering in 2014. Currently, he is working as a Research Assistant at UMNG in the Research Group GAV in different Mechatronics fields.

Dario Amaya Hurtado, He was educated at UAN, Bogotá, Colombia receiving the B Sc. degree in Electronics Engineering in 1995 and the M.Sc. degree in Teleinformatic in 2007 by the Faculty of Engineering at the Francisco José de Caldas District University, UFJC in Bogotá, Colombia. He was awarded the Ph.D. degree in 2011 in Mechanical Engineering at Campinas State University, São Paulo, Brazil, working on hybrid control – He has worked as a professor and researcher at the Nueva Granada Military University, Colombia since 2007 been

