ISSN: 0974-5823 Vol. 7 No. 2 February, 2022

International Journal of Mechanical Engineering

Iterated Local Search for Determining the optimal Configuration in Artificial Neural Networks

Joselito Medina-Marin

Universidad Autónoma del Estado de Hidalgo

Maria Guadalupe Serna-Diaz

Universidad Autónoma del Estado de Hidalgo

Alejandro Tellez-Jurado

Universidad Politécnica de Pachuca

Juan Carlos Seck-Tuoh-Mora

Universidad Autónoma del Estado de Hidalgo

Norberto Hernandez-Romero

Universidad Autónoma del Estado de Hidalgo

Samuel Medina-Garcia

Universidad Autónoma del Estado de Hidalgo

Abstract

One of the challenges in Artificial Neural Networks development is to find the correct network topology with the best performance. This research work proposes an Iterated Local Search algorithm to obtain an adequate configuration for the topology of Artificial Neural Networks. This proposal was tested with data gathered from a biotechnological experiment to obtain lignosulfonates in barley straw. In twenty iterations or less, the algorithm gets topologies for ANNs with r^2 near to one.

Keywords: Artificial neural networks, Iterated Local Search, Optimization.

1. Introduction

Artificial Neural Networks (ANNs) have many areas of application with outstanding results. There are interesting works related to management [1, 2], chemical engineering [3], energy systems [4, 5], industrial engineering [6, 7, 8], environment [9, 10], among others.

One of the most challenging issues in ANN development is how to design topologies with a coefficient of determination (r^2) near to 1. In [11], the authors used a Forest Type Mapping Data Set, and they found out that ANNs with three hidden layers provide the best performance.

A way to obtain the number of neurons in hidden layers is proposed in [12]. This method uses a quantitative criterion based on a signal-to-noise-ratio figure to detect overfitting. In [13] the authors apply a PSO to optimize the parameter settings of the ANN, and the topology with the best performance.

The authors of [14] propose a methodology to obtain the optimal number of hidden layers and their number of neurons by using

Tabu search and Gradient descent in combination with a learning algorithm. In another work, a grey wolf optimizer is used for the same aim. [15]

For this research paper, we are proposing the use of iterated local search (ILS) to obtain an ANN with good performance, by modifying the number of hidden layers and their corresponding number of hidden neurons.

2. Artificial Neural Networks

Artificial Neural Networks (ANNs), through parallel, distributed and adaptive computing, are able to learn from examples. These systems imitate the real neural systems of human beings trying to reproduce some of their abilities. [16]

An ANN is a directed graph and present the following properties:

1. Each node i is related with a state variable x_i.
2. Each connection (i, j) of nodes i and j has a weight $w_{ij} \in \mathfrak{R}$.

3. Each node i has a threshold θ_i.

4. For each node i an activation function $f_i(x_j, w_{ij}, \theta_i)$ is defined. It depends on the connection weights (w_{ij}), the threshold (θ_i), and the states of nodes j (x_j) connected to it. This function provides the new state of node i.

Figure 1 presents the structure of an ANN. Input layer is composed of neurons without input connections, the output layer has neurons without output connections, and the rest of the neurons belongs to the hidden layers.

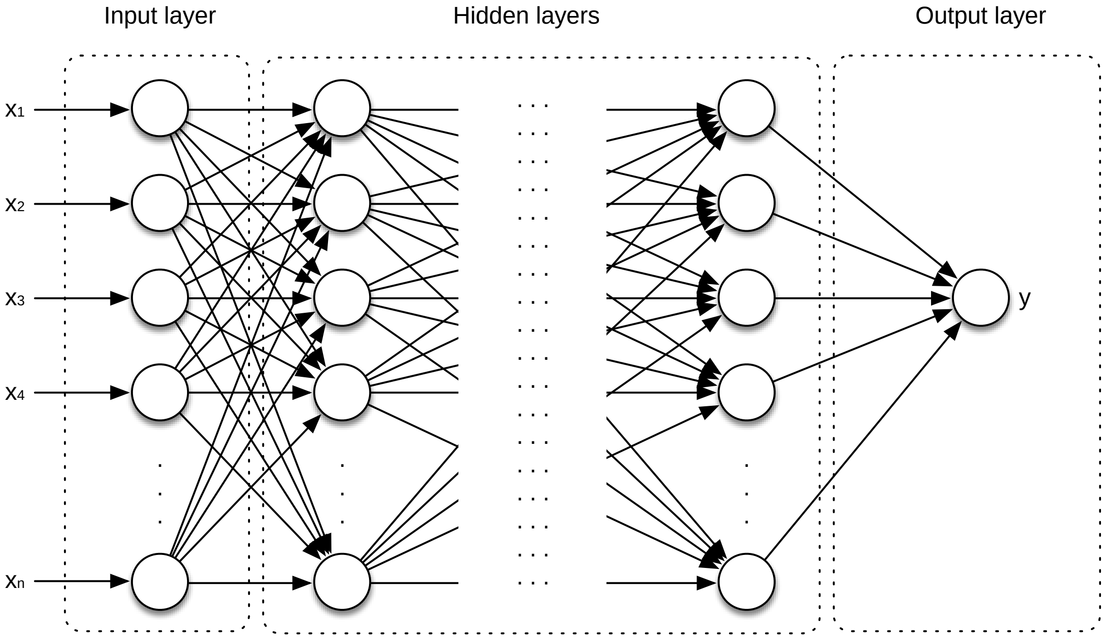


Figure 1. ANN structure with one input layer, hidden layers, and one output layer.

In this ANN structure, the information goes through each layer doing progressive processing. The operation for each neuron with n inputs and m outputs can be expressed as:

$$y_i(t) = f\left(\sum_{j=1}^{n} w_{ij}x_j - \theta_i\right), \forall i, 1 \leq i \leq m$$

Input layer neurons are not used to perform any computation, they only send the information to the first hidden layer neurons.

In the training phase, weights w_{ij} are updated to minimize the error, taking into account the output value (y_i) and the target used in the training of the ANN. An adequate ANN topology allows minimal errors in the prediction of ANN from input data.

3. Iterated Local Search

Iterated Local Search (ILS) is a strategy starting in an initial point, is obtained randomly, and it applies search locally to changes of the current point. If the next point is better than the current one, in the next iteration the new point is used to continue the local search; otherwise, the current point is kept for the next iteration with another perturbation. [17]

The algorithm for ILS is showed below [18]:

Step 1: s_0 =Initial_Solution

Step 2: s^* = Local_Search(s_0)

Step 3: while error condition is not satisfied

3.1 s' = Pertubation(s^*)

3.2 $s^{*'}$=Local_Search(s')

3.3 s^* = Acceptance_Criterion(s^*, $s^{*'}$)

Step 4: End

This algorithm can be applied in the design of the ANN topology, where the initial solution is the configuration of the ANN.

4. Dataset

Data used in the training of ANN was obtained from an experimental process of getting lignosulfonates from barley straw that was gathered in Hidalgo state, Mexico. The barley straw was ground and strained by using different meshes: size 8 (2mm), size 12 (1.68mm), and size 20 (0.84mm). After that, samples of 2g were blended with 50mL of three concentrations of sodium sulfite solution: 1%, 5%, and 10%. Next, the samples were heated at 3 atm and 137 ºC during 30, 60, and 90 minutes. The lignosulfonates were recovered from the cellulose pulp and solubilized material obtained in this process, by using a vacuum with pore filter paper of 1.6 □m.

The experimental design utilized a three-level three-factor (3^3). Three process variables were used (barley straw size, boiling time, and concentration of sulfite) and one response variable (performance of solubilized material in the recovering of lignosulfonates). 27 experiments with different combinations of variables values were carried out. These data are used in the training of ANNs and can be consulted in [19].

5. Application of ILS in ANN topology

In order to get an adequate ANN topology, the ILS is applied instead of searching for the best topology in all the possible combinations. We consider three hidden layers, taking into account the result presented in [11], and the initial solution is generated randomly. The number of neurons in each hidden layer varies from 1 to 10 neurons.

In the first step of the ILS, the initial solution is created.

$$s_0 = [rand(1{,}10) \quad rand(1{,}10) \quad rand(1{,}10)]$$

Next, we obtain an ANN with the configuration defined in s_0.

$$s^* = ANN(s_0)$$

A loop is initiated, while the condition is not satisfied we make a perturbation to the current ANN.

$$s' = perturbation(s^*)$$

The perturbation is applied in the following way: values of s_0, denoting the hidden layer neurons, are modified by increasing or decreasing by one the amount of neurons, in only one hidden layer. Thus, the position is selected by generating an integer random number between 1 and 6, which indicates the layer and the type of arithmetic operation (addition or subtraction). Figure 2 indicates the position of values to be perturbed. For instance, if the random number is 3, it means the quantity of neurons in the third hidden layer (z_i) decreases 1 neuron; but if the random number is 5, the neurons in the second hidden layer increases 1 neuron.

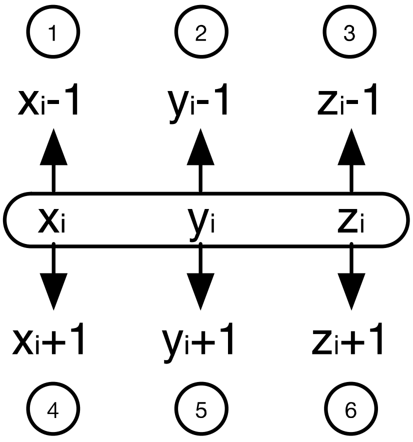


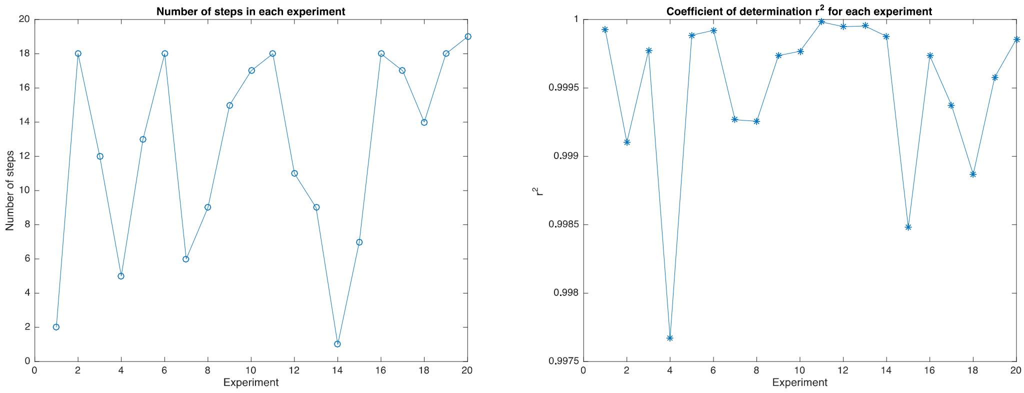
Figure 2. Number of neurons perturbation in hidden layers according to random selection.

After the perturbation, the new topology *s*' is used to train another ANN (*s**'). Both ANN are compared (*s** and *s**'), and we maintain the ANN with the best performance indicated by the coefficient of determination (r^2). These steps are repeated until an error condition related to r^2 is satisfied.

6. Results

The ILS algorithm is implemented as a Matlab® script, and the training of the ANN is carried out by using the toolbox of Neural networks included in this software.

In [11], 11110 ANNs were created and trained to search for the one with the best performance, and it was obtained an ANN with four hidden layers (9:5:9:3) and r^2=0.9825. However, in our approach, the ILS provides ANNs with good performance in a few steps, instead of searching in the state space of all the configurations. Figure 3 shows the execution of 20 instances of ILS, and how in 19 steps (Figure 3a) or less the algorithm achieves an r^2 very close to 1 (Figure 3b), in the worst case $r^2 = 0.9977$.



(a)

(b)

Figure 3. (a) Number of steps utilized to achieve an ANN with good performance. (b) Coefficient of determination r^2 obtained with our approach based on ILS.

7. Conclusions

The aim of this work is the proposal of a strategy to search for an ANN topology with good performance in fewer steps than looking for in all the possible combinations of neurons in hidden layers. Our proposal is based on ILS, and takes an initial topology to start the algorithm. The results show that in a few steps we can obtain ANNs with r^2 very close to 1, which can be used to predict values with a high level of certainty.

As further work, the ILS will be used to optimize the weights in the ANN, and our approach will be applied with another dataset to validate its feasibility.

8. Acknowledgements

This paper was financially supported by the Consejo Nacional de Ciencia y Tecnología in Project CONACYT CB-2017-2018-A1-S-43008.

9. References

[1] H. Hakimpoor, K.A.B. Arshad, H.H. Tat, N. Khani, and M. Rahmandoust, M. “Artificial neural networks’ applications in management.” *World Applied Sciences Journal*, 14(7), 1008-1019. 2011.

[2] Sharma, A., & Chopra, A. (2013). Artificial neural networks: Applications in management. Journal of Business and Management, 12(5), 32-40.

[3] Pirdashti, M., Curteanu, S., Kamangar, M. H., Hassim, M. H., & Khatami, M. A. (2013). Artificial neural networks: applications in chemical engineering. Reviews in Chemical Engineering, 29(4), 205-239.

[4] Kalogirou, S. A. (2000). Applications of artificial neural-networks for energy systems. Applied energy, 67(1-2), 17-35.

[5] Yalcintas, M., & Akkurt, S. (2005). Artificial neural networks applications in building energy predictions and a case study for tropical climates. International journal of energy research, 29(10), 891-901.

[6] Meireles, M. R., Almeida, P. E., & Simões, M. G. (2003). A comprehensive review for industrial applicability of artificial neural networks. IEEE transactions on industrial electronics, 50(3), 585-601.

[7] Jafari, R., Contreras, M. A., Yu, W., & Gegov, A. (2019, September). Applications of fuzzy logic, artificial neural network and neuro-fuzzy in industrial engineering. In Latin American Symposium on Industrial and Robotic Systems (pp. 9-14). Springer, Cham.

[8] Jafari, Raheleh, et al. "Applications of fuzzy logic, artificial neural network and neuro-fuzzy in industrial engineering." Latin American Symposium on Industrial and Robotic Systems. Springer, Cham, 2019.

[9] Al-Shawwa, M. O., Al-Absi, A. A. R., Hassanein, S. A., Baraka, K. A., & Abu-Naser, S. S. (2018). Predicting Temperature or Humidity in the Surrounding Environment Using Artificial Neural Network.

[10] Vlontzos, G., & Pardalos, P. M. (2017). Assess and prognosticate green house gas emissions from agricultural production of EU countries, by implementing, DEA Window analysis and artificial neural networks. Renewable and Sustainable Energy Reviews, 76, 155-162.

[11] Liu, P., Wang, J., Sangaiah, A. K., Xie, Y., & Yin, X. (2019). Analysis and prediction of water quality using LSTM deep neural networks in IoT environment. Sustainability, 11(7), 2058.

[11] Santoso, J., & Surendro, K. (2019, September). Determining the number of hidden layers in neural network by using principal component analysis. In Proceedings of SAI Intelligent Systems Conference (pp. 490-500). Springer, Cham.

[12] Liu, Y., Starzyk, J. A., & Zhu, Z. (2007). Optimizing number of hidden neurons in neural networks. EeC, 1(1), 6.

[13] Qolomany, B., Maabreh, M., Al-Fuqaha, A., Gupta, A., & Benhaddou, D. (2017, June). Parameters optimization of deep learning models using particle swarm optimization. In 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC) (pp. 1285-1290). IEEE.

[14] Gupta, T. K., & Raza, K. (2020). Optimizing deep feedforward neural network architecture: A tabu search based approach. Neural Processing Letters, 51(3), 2855-2870.

[15] Faris, H., Mirjalili, S., & Aljarah, I. (2019). Automatic selection of hidden neurons and weights in neural networks using grey wolf optimizer based on a hybrid encoding scheme. International Journal of Machine Learning and Cybernetics, 10(10), 2901-2920.

[16] Aggarwal, C. C. (2018). Neural networks and deep learning. Springer, 10, 978-3-319-94462-3, https://doi.org/10.1007/978-3-319-94463-0

[17] Khebbache, S., PRINS, C., & Yalaoui, A. (2008). Iterated local search algorithm for the constrained two-dimensional non-guillotine cutting problem.

[18] Lourenço, H. R., Martin, O. C., & Stützle, T. (2003). Iterated local search. In Handbook of metaheuristics (pp. 320-353). Springer, Boston, MA.

[19] Serna-Diaz, M. G., Arana-Cuenca, A., Medina-Marin, J., Seck-Tuoh-Mora, J. C., Mercado-Flores, Y., Jiménez-González, A., & Tellez-Jurado, A. (2016). Modeling of sulfite concentration, particle size, and reaction time in lignosulfonate production from barley straw using response surface methodology and artificial neural network. BioResources, 11(4), 9219-9230.