Optimized neural networks in industrial data analysis

Liesle Caballero^{1,2} · Mario Joja¹ · Winston S. Percybrooks¹

Received: 1 August 2019 / Accepted: 15 January 2020 / Published online: 30 January 2020
© Springer Nature Switzerland AG 2020

Abstract

The research work presented here tackles the problem of finding the optimum value for hyper-parameters, such as number of layers and number of neurons per layer, for a fully-connected Artificial Neural Network (ANN), particularly in regression problems. A proposed optimization strategy is tested on different datasets related to diverse industrial applications: (1) prediction of the performance of exploration algorithms for mobile robots, (2) prediction of the compressive strength of concrete, (3) prediction of energy output from a power plant; and (4) prediction of wine quality. Different evaluation metrics, such as Pearson correlation coefficient (R), Square Correlation Coefficient (R^2), Absolute Relative Error (RAE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE), are used to determine the best performing prediction model when the hyper-parameter optimization algorithms are used. That result is compared to the performance of the best model previously reported in the literature on the same dataset in order to determine the gains achieved by using the hyper-parameter optimization strategy under test.

Keywords Machine learning · Artificial neural network · Optimization algorithms · Hyper-parameters

1 Introduction

When working on regression problems with feed-forward, fully-connected Neural Networks (FFNN) models, hyper-parameters, such as number of layers, number of neurons per layer, learning rate, activation function, among others, are set by the designer usually through hand tuning or trial and error method [1]. Unfortunately, the performance of these networks is very sensitive to the value of hyper-parameters and a manual selection procedure is time-consuming and does not guarantee an optimal solution, since typically an exhaustive search is not practical due to the large number of possibilities involved. Several recent works in machine learning are focused on improving hyper-parameter optimization methods [2] to facilitate the automatic determination of optimal ANN

hyper-parameters. In the literature we can find Bayesian optimization [3, 4] random search [5], tree-structure parzen estimators [5], among others.

The general objective in this research work is to apply a hyper-parameter optimization method to determine two specific hyper-parameters, the number of layers and number of neurons per layer in a FFNN. In this paper we propose the Hill Climbing with random restart and Tabu List optimization methods as a strategy to optimize the hyper-parameters named above for a regression model based on a FFNN in applications of small data.

We test the performance of the optimization method using different datasets publicly available on the web, all with related existing literature, finding that the values of the evaluation metrics obtained when using our proposed method are better than the values reported in the

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s42452-020-2060-5>) contains supplementary material, which is available to authorized users.

✉ Liesle Caballero, lieslec@uninorte.edu.co; Mario Joja, jojam@uninorte.edu.co; Winston S. Percybrooks, wpercyb@uninorte.edu.co |
¹Universidad del Norte, Barranquilla, Colombia. ²Institución Universitaria ITSA, Barranquilla, Colombia.

SN Applied Sciences (2020) 2:300 | <https://doi.org/10.1007/s42452-020-2060-5>

references [6–9]. The regression models predicted the following target variables: Performance of the Random Walk (RW) exploration algorithm, performance of the Random Walk Without Step Back (RW_WSB) exploration algorithm, performance of the Q_Learning exploration algorithm in autonomous robots, compressive strength of concrete, power consumed by a full-load plant and quality of red wine.

In order to compare the performance obtained when using the proposed hyper-parameter optimization method against a hand tuning design, the same evaluation metrics are used. The metrics involved are: Absolute Relative Error (%RAE), Pearson Correlation Coefficient (R), Root Mean Square Error (RMSE), Square Correlation Coefficient (R^2) and Mean Absolute Error (MAE).

The machine learning technique used to design the predictor is a fully-connected, multilayer perceptron, artificial neural network (ANN) with one or two hidden layers, suitable to represent a non-linear model and connected by feed-forward links. Small network structures, i.e., few layers (one or two) and few of neurons per layer (< 20), are sufficient to solve regression problems applied to small data (400–10,000 samples) with a low number of features (4–11).

This document is organized as follows: Sect. 2 describes the datasets used in the experiments; Sect. 3 presents the optimization methods based on Hill Climbing with random restart and Tabu List; in Sect. 4, experimental results are presented and compared; and Sect. 5 contains the conclusions.

2 Small datasets

Many recent applications of Machine Learning focus on the use of big datasets for training, however, there are still many interesting problems where the available datasets are small. It can be argued that the same type of optimization methods are not appropriate for both types of databases (big and small), therefore in this paper we propose a low-complexity method with excellent results for small databases. Table 1 summarizes the main characteristics of the six datasets used in this work. Three of the datasets were obtained from our own simulations of the RW, RW_WSB and Q_Learning algorithms for a specific robotic exploration scenario. The variable to be predicted in this application is defined as the maximum number of steps in which the robot reaches the target object for a given set of conditions. The experimental setup is described in [6] and

Table 1 Characteristics of the datasets used for testing

<i>RW, RW_WSB and Q_learning dataset</i>			
5 inputs	Inputs: Number of rows in the grid, Number of columns, Number of obstacles, Target object X coordinate, Target object Y coordinate.	Target variable	Maximum number of steps
		Type of data	Integer
		Type of variables	Quantitative
		No. instances	475
		Prediction model evaluation (metrics)	RAE and R
<i>Concrete compressive strength data set</i>			
8 inputs	Inputs: Cement, Blast furnace slag, Fly ash, Water, Superplasticizer, Coarse aggregate, Fine aggregate, Age.	Target variable	Concrete compressive strength
		Type of data	Real
		Type of variables	Quantitative
		No. instances	1030
		Prediction model evaluation (metrics)	RMSE and R^2
<i>Combined cycle power plant data set</i>			
4 inputs	Inputs: Temperature, Ambient pressure, Relative humidity, Exhaust vacuum.	Target variable	Energy output
		Type of data	Real
		Type of variables	Quantitative
		No. instances	9568
		Prediction model evaluation (metrics)	MAE and RMSE
<i>Wine quality data set</i>			
11 inputs	Inputs: Fixed acidity, Volatile acidity, Citric acid, Residual sugar, Chlorides, Free sulfur dioxide, Total sulfur dioxide, Density, pH, Sulphates, Alcohol.	Target variable	Quality [0–10]
		Type of data	Real
		Type of variables	Quantitative
		No. instances	1599
		Prediction model evaluation (metrics)	MAE

[10]. The other three datasets were downloaded from the public UCI repository [11]. One of them is used to predict the compressive strength of concrete from the quantities of each ingredient used in production [7]. Another dataset is used to predict the full-load electrical power output required for a power plant based on several features collected during a period of 6 years (2006–2011) [8]. The last dataset is used to predict the quality of red wine from some physiochemical properties [9]. The target variable is a valuation between 0 (very bad wine) and 10 (very excellent wine), made by wine experts.

3 Hyper-parameter optimization

Often, the hyper-parameters are determined by a random search inside a limited set of possibilities, picking the values that achieve the best performance. However, there is a optimization method proposed in [6] which calculates the number of layers and neurons per layer that maximizes the performance of the neural network for one or two layers and therefore, gets the best descriptive metrics of system behavior. This previous work detailed in [6] shows an improvement in predictor performance when Hill Climbing with Random Restart algorithm is applied to find the optimal ANN architecture comparing the results with a simple Hill Climbing algorithm, demonstrating that the results with the introduction of Random Restart are better. Hill Climbing is restarted several times using randomly chosen starting points, i.e. Random Restart method [12].

The prediction process include an optimization subprocess as shown in Fig. 1. This process consists in the application of an optimization algorithm to calculate the number of layers and number of neurons per layer in a neural network model. The Hill Climbing- Random Restart and Tabu List algorithms originally presented in [12] are proposed to search for the ANN topology that minimizes error metrics and maximizes, at the same time, the correlation between the predicted data $\hat{y}_i$ and the actual data y_i in a number of experimental conditions N .

The Fig. 2 shows a grid with the all possible structures of the neural networks to be trained, validated and tested. The number of neurons has been limited to an integer number in the range [1,20] and the number of layers has been limited to the integer range [1,2]. To ensure that optimization algorithms include single and two layer neural network structures, we vary the number of neurons in layer two between zero and twenty [0–20]. If a local optimum is found in the zero column of the search grid, it means that the structure of the neural network is single layer. Variations with respect to the rows of the zero column between one and twenty [1–20] determine the number of neurons in layer 1 when these

Fig. 1 Flowchart of the prediction model

Fig. 2 Search grid of the optimal neural network structure

structures are presented. For example, for the coordinate (20,0) shown in Fig. 2, a structure of 20 neurons in layer 1 and zero neurons in layer 2 (structure that does not have layer 2) is tested, obtaining a prediction model which is evaluated according to the metrics (% RAE, MAE, RMSE, R, R^2) defined. Now, for the coordinate (20,10) of the same figure a neural network structure of 20 neurons in layer 1 and 10 neurons in layer 2 is tested.

In an optimization problem, it is possible to find good solutions (local optimum), or the best overall solution (global optimum) [13]. For the work presented here, Hill Climbing and Tabu List optimization are proposed to search for the number of layers and neurons per layer that maximizes R , R^2 and minimizes %RAE, RMSE, MAE according to the case.

The Fig. 3 shows that Hill Climbing Random Restart and Tabu List algorithms converge and find the same minimum RAE, RMSE or MAE error on the search grid that satisfies the requirements of the problem. However, when comparing the two algorithms we note that Tabu List converges with

less iterations than those spent by Hill Climbing Random Restart.

For Fig. 3a it shows in the blue line curves that Tabu List spends approximately 50% less iterations than those spent by Hill Climbing Random Restart shown with red lines. Figure 3b, c and d show that each optimization algorithm finds in each application area the same minimum errors (RMSE and MAE) independently. A minimum RMSE of 1891 and 3,385 is found for the prediction model of the compressive strength of concrete and of the energy output from a power plant respectively. A minimum MAE of 0.2395 is found for the prediction model of wine quality.

In Fig. 4, the Y axis represents the number of neurons in layer 1, the X axis represents the number of neurons in layer 2, the Z axis is the value of the metric. The highlighted point in each figure are the maximum or minimum found by the optimization algorithms. The color scale shows in blue tones magnitudes close to zero and in red tones large magnitudes indicated by the scale next to the color map. In the case of the %RAE, RMSE and MAE metrics, where a value close to zero is desired, the blue areas provide good architectures, on the contrary, for the R and R^2 metrics, where a large value (close to 1) is desired, red areas provide good architectures.

In the case of RW exploration algorithm, the color maps in Fig. 4a and b indicate that an optimal structure is found when there are 19 neurons in layer 1 and 19 neurons in layer 2. At this point, [19:19], the %RAE is a minimum of 2.550%. The R is a maximum of 0.9957 when the structure is [16:10], that is, 16 neurons in layer 1 and 10 neurons in layer 2. However, the structure [19:19] is selected because for this structure, %RAE is higher than the other one and the difference between both R 's is close to zero.

In the case of the RW_WSB exploration algorithm, Fig. 5a and b indicate that an optimal structure is found when there are 18 neurons in layer 1 and 20 neurons in layer 2, with a minimum %RAE of 3.307%. The R is a maximum of 0.9957 when the structure is [14:10]. The structure [18:20] is selected by the same reasons described above.

For the Q_Learning algorithm, the optimal structure is [19:18] in Fig. 6a, 19 neurons in layer 1 and 18 neurons in layer 2. The minimum %RAE presented at this point is 5.412%. With regard to R in Fig. 6b, the maximum was found for the [14:19] structure, however the former is selected because the difference between both R is negligible but the difference between both %RAE is significant.

In the concrete compressive strength prediction, a minimum RMSE of 1.891 MPa and a maximum R^2 of 0.9868 was found at [17:18], therefore, the optimal structure has 17 neurons in layer 1 and 18 neurons in layer 2. See Fig. 7a and b.

For the power plant prediction, a minimum RMSE of 3.39 MWh were located at point [20:20]. See Fig. 8.

Fig. 3 Convergence curves to compare Hill Climbing Random Restart and Tabu List

Fig. 4 Location of the minimum %RAE (a) and maximum R (b) that presents the optimal structure ANN for the performance predictor of the RW exploration algorithm

Fig. 5 Location of the minimum %RAE (a) and maximum R (b) that presents the optimal structure ANN for the performance predictor of the RW_WSB exploration algorithm

Fig. 6 Location of the minimum %RAE (a) and maximum R (b) that presents the optimal structure ANN for the performance predictor of the Q_Learning exploration algorithm

Finally, for the case of wine quality prediction a minimum MAE of 0.239 was located at [20:20]. See Fig. 9.

4 Results and comparisons

In this section the results obtained when hyper-parameter optimization is used are compared against the best

Fig. 7 Location of the minimum RMSE **(a)** and maximum R^2 **(b)** that presents the optimal structure ANN for the concrete compressive strength predictor

Fig. 8 Location of the minimum RMSE that presents the optimal structure ANN for the predictor of power consumed by a full-load plant

Fig. 9 Location of the minimum MAE that presents the optimal structure ANN for the predictor of wine quality

previously reported results for each dataset, beginning with the RW, RW_WSB and Q_Learning exploration algorithms. The model used in the state of the art is an ANN whose number of neurons per layer is found using hand tuning while in the proposed model a Hill Climbing with Random restart and Tabu List algorithms were used. Table 2 presents the corresponding %RAE and

R reported in [6] and the %RAE and R reported in this work.

R is close to 1 when the proposed method is used. With respect to %RAE for RW, RW_WSB and Q_Learning, it was reduced by 92.47%, 90.41% and 87.38% respectively.

For the following application, prediction of concrete compressive strength, the reference method used an ANN too, but the searching of hyper-parameters was made using hand tuning. Table 3 presents the corresponding RMSE and R^2 reported in [7] versus the RMSE and R^2 reported in this work.

The structure obtained with the reference method was simpler than with our proposed method, but the metric RMSE was reduced in a 56.25% and R^2 was increased in a 5.85%.

With respect to power prediction both methods used ANNs and cross validation. Table 4 presents the corresponding RMSE reported in [8] versus the RMSE reported in this work. The proposed hyper-parameter optimization achieved a decrease of 37.3% in the RMSE.

Finally, Table 5 presents the corresponding MAE reported in [9] versus the MAE reported in this work. For the prediction model of red wine quality a reduction of 53.03% in MAE was achieved when the proposed optimization algorithm is used.

In general, the results show an improvement in the performance of the prediction model when optimization methods that calculates the hyper-parameters of the neural network (number of layers and number of neurons per layer) are included.

5 Conclusions

This research work extends previous work by the authors, [6, 11], demonstrating that an ANN can predict better when an optimization algorithm to determine the

Table 2 Comparative table of results in testing for prediction of the RW, RW_WSB and Q_Learning performance

Area of application	Results	Referenced method	Proposed method
Prediction of performance for algorithm RW	%RAE	33.89	2.55
	R	0.9194	0.9898
	Structure ANN	8	[19:19]
Prediction of performance for algorithm RW_WSB	%RAE	34.54	3.31
	R:	0.9203	0.9881
	Structure ANN	12	[18:20]
Prediction of performance Q_Learning	%RAE	42.89	5.41
	R:	0.8830	0.9717
	Structure ANN	9	[18:19]

Table 3 Comparative table of results in testing for prediction of the compressive strength of concrete

Area of application	Results	Referenced method	Proposed method
Prediction of the compressive strength of concrete	RMSE	4.32 MPa	1.89 MPa
	R²	0.929	0.9868
	Structure ANN	4	[17:18]

Table 4 Comparative table of results in testing for prediction of the power consumed by a full-load plant

Area of application	Results	Referenced method	Proposed method
Prediction of the power consumed by a full-load plant	RMSE	5.399 M Wh	3.385 M Wh
	Structure ANN	It was not specified	[20:20]

Table 5 Comparative table of results in testing for prediction of the quality of red wine

Area of application	Results	Referenced method	Proposed method
Prediction of the quality of red wine	MAE	0.51	0.2395
	Structure ANN	It was not specified	[20:20]

hyper-parameter (number of layers and number of neurons per layer) is applied. Three different optimization strategies were compared: hand tuning versus Hill Climbing-Random Restart and Tabu List. Although Hill Climbing-Random Restart and Tabu List reach the same target, we observed in the convergence curves that Tabu List converges faster than Hill Climbing-Random Restart.

In general, the comparison of the resulting evaluation metrics shows evidence that the proposed optimization methods have a significant and positive impact on model performance.

Reference results presented in the state of the art were obtained using hand tuning. With the proposed optimization algorithms, improvements were achieved on the performance metrics: Pearson correlation coefficient is close to one and MAE, RMSE and %RAE are reduced by 80% with respect to previous results. This

work focuses on optimizing the topology of a neural network, that is, number of layers and neurons per layer in order to improve the initial results reported in different references. It is necessary to compare the performance of the proposed method with others found in the literature, taking into account that such state-of-the-art methods are typically designed for big datasets and their performance may be hindered when there is less data available.

Acknowledgements Gratefulness to the CEIBA Foundation for its support in this research.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Bengio Y (2000) Gradient-based optimization of hyperparameters. *Neural Comput* 12(8):1889–1900
2. Feurer M, Springenberg JT, Hutter F (2015) Initializing bayesian hyperparameter optimization via meta-learning. In: Twenty-ninth AAAI conference on artificial intelligence
3. Probst P, Bischl B, Boulesteix A-L (2018) Tunability: importance of hyperparameters of machine learning algorithms. [arXiv:1802.09596v3](https://arxiv.org/abs/1802.09596v3)
4. Shahriari B, Swersky K, Wang Z, Adams RP, de Freitas N (2016) Taking the human out of the loop: a review of Bayesian optimization. *Proc IEEE* 104(1):148–175
5. Bergstra JS, Bardenet R, Bengio Y, Kégl B (2011) Algorithms for hyper-parameter optimization. In: *Proceedings of the 24th international conference on neural information processing*, pp 2546–2554
6. Caballero L, Jojoa M, Percybrooks W (2018) Optimized artificial neural network system to select an exploration algorithm for robots on bi-dimensional grids. Springer, Cham, pp 16–27
7. Yeh I-C (2006) Analysis of strength of concrete using design of experiments and neural networks. *J Mater Civ Eng* 18:597–604
8. Tüfekci P (2014) Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *Int J Electr Power Energy Syst* 60:126–140
9. Cortez P, Cerdeira A, Almeida F, Matos T, Reis J (2009) Modeling wine preferences by data mining from physicochemical properties. *Decis Support Syst* 47(4):547–553
10. Caballero L, Benavides C, Percybrooks W (2018) Machine learning-based system to estimate the performance of exploration algorithms for robots in a bi-dimensional grid. In *2017 IEEE 3rd colombian conference on automatic control (CCAC)*
11. UCI machine learning repository: data sets. <https://archive.ics.uci.edu/ml/datasets.html>. Accessed 26 Dec 2018
12. Koller D, Friedman N (2018) *Probabilistic graphical models: principles and techniques*. MIT Press, Cambridge
13. Kanaan S, Masip Rodó D, Escudero Bakx G, Benítez Iglésias R (2013) *Inteligencia artificial avanzada*. Editorial UOC—Editorial de la Universitat Oberta de Catalunya, Barcelona

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.