Afrika Statistika

Afrika Statistika
Vol. 15 (3), 2020, pages 2413 - 2429.
DOI: http://dx.doi.org/10.16929/as/2020.2413.166

ISSN 2316-090X

Empirical determination of optimal configuration for characteristics of a multilayer perceptron neural network in nonlinear regression

Castro Gbêmêmali Hounmenou [1,2,*], **Roméo Jésukpégo Tohoun** [2], **Kossi Essona Gneyou** [3] **and Romain Glèlè Kakaï** [2]

[1]Institut de Mathématiques et de Sciences Physiques, Porto-Novo, Benin
[2]Laboratoire de Biomathématiques et d'Estimations Forestières, University of Abomey-Calavi, Benin
[3]Laboratoire de Modélisations Mathématiques et Applications, University of Lome, Togo

Received on April 5, 2020; Accepted on November 1, 2020.

Abstract. In this paper, we determine an optimal configuration for characteristics of a multilayer perceptron neural network (MPL) in nonlinear regression for predicting crop yield. Monte Carlo simulation approach has been used to train several databases generated by varying the internal structure of 3-MLP from simple to complex for 5 different algorithms most commonly used. Results showed that the optimal configuration is obtained with the Levenberg Marquard algorithm, 75% of the number of input variables as number of hidden nodes, learning rate 40%, minimum sample size 150, tangent hyperbolic and exponential functions in the hidden and output layers respectively. This configuration has been illustrated with real life data.

Key words: artificial neural network; machine learning; sample-size effect; nonlinear models; prediction
AMS 2010 Mathematics Subject Classification Objects : 62-07 ; 62J02 ; 65C05

* Castro Gbêmêmali Hounmenou [1,2,*] castro.hounmenou@imsp-uac.org
Roméo Jesukpégo Tohoun : romeotohoun@gmail.com
Kossi Essona Gneyou: kgneyou@gmail.com
Romain Glèlè Kakaï : romain.glelekakai@fsa.uac.bj

Résumé. (Abstract in French) Dans cet article, nous déterminons une configuration optimale pour les caractéristiques d'un réseau de neurones de type perceptron multicouche (PMC) en régression non linéaire pour prédire le rendement des cultures. L'approche de simulation Monte Carlo a été utilisée pour entraîner plusieurs bases de données générées en variant la structure interne du modèle 3-MLP des cas simples aux complexes pour 5 différents algorithmes les plus couramment utilisés. Les résultats ont montré que la configuration optimale est obtenue avec l'algorithme d'apprentissage Levenberg Marquard, 75% du nombre de variables d'entrée comme nombre de nœuds cachés, taux d'apprentissage 40%, taille minimale de l'échantillon 150, fonctions d'activation tangente hyperbolique dans la couche cachée et exponentielle dans la couche de sortie. Cette configuration a été testée par des données réelles et a donné de meilleurs résultats.

The authors.

Castro Gbêmêmali Hounmenou, M.Sc., is a preparing his Ph.D. degree at Institut de Mathématiques et de Sciences Physiques, University of Abomey-Calavi, Benin, under the supervision of the third and the fourth authors.

Roméo Jesukpégo Tohoun, M.Sc., is a preparing his Ph.D. degree at Faculty of Agronomic Sciences, University of Abomey-Calavi, Benin, under the supervision of the fourth author.

Kossi Essona Gneyou. Ph.D., is a Full Professor at the Department of Mathematics, Faculty of Sciences, University of Lome, Togo.

Romain Glèlè Kakaï, Ph.D., is a Full Professor at the Faculty of Agronomic Sciences, University of Abomey-Calavi, Benin.

1. Introduction

Let $y = \phi(x, \theta) + \epsilon$ be a regression model, where $x = (x_1, \ldots, x_p)^\top \in \mathbb{R}^p$ is the vector of input variables, $y \in \mathbb{R}$ is the ouput variable, $\phi : \mathbb{R}^p \to \mathbb{R}$ is the unknown function of the regression model which establishes the relationship between x and y; θ is a vector of the model parameters and $\epsilon \in \mathbb{R}$ is residual variable. The linear regression model $y = \theta_0 + \sum_{i=1}^{p} \theta_i x_i + \epsilon$ is the most used in practice. Its main assumptions are: normality and homogeneous of residuals, absence or lower multicollinearity between input variables, etc. (Uyanik and Güler, 2013, Wang *et al.*, 2014). In real-world situations, these assumptions do not always hold. Thus, a widely used alternative is the general regression model, $y = \theta_0 + \sum_{i=1}^{p} \theta_i f_i(x_i) + \epsilon$ where $f_i(x)$ is a basic expansion. For instance $f_i(x) = x^i$ or $f_i(x) = cos(ix)$ correspond to polynomial and trigonometric regression respectively.

Current models are usually more complex and often nonlinear (Wang *et al.*, 2014, Badran and Thiria, 2002, Lindsey, 2001, Bates and Watts, 1988). Among these new tools are Multilayer perceptrons neural networks (MLP). They belong to a very rich family of continuous functions whose main characteristic is to allow a great modeling flexibility. MLP have demonstrated their effectiveness in predicting empirical data than traditional methods and are applied in various fields (Cuauhtémoc, 2015, Cottrell *et al.*, 2012, Perai *et al.*, 2010). For example, in agriculture and climatology, relationship between y and x can be mapped by ϕ where $y = \phi(x)$: (a) Forecasting yields of maize, sorghum, rice, beans, cotton, etc. where y = crop yield; x = agronomic, edaphic, climatic and economic parameters ; (b) Prediction of rainfall, with y = rainfall and x = cloud and atmospheric parameters ; (c) Calculation of long-wave atmospheric radiation where y = radiation flux and x = temperature, humidity, O_3, CO_2, cloud parameter profiles, surface flux, etc.

Attractiveness for MLP models is due to their great capacity of generalization and their consideration of non-linearity, noise of data, multicollinearity between input variables and often the dynamic nature of data (Mário *et al.*, 2017, Cottrell *et al.*, 2012, Kordos and Duch, 2008, Graupe, 2003, Bishop, 1995, Chen *et al.*, 1990). However, the accuracy of the approximation of ϕ by a MPL can be affected by some of its characteristics. These are the learning algorithm and the hyper-parameters: number of hidden nodes, set of transfer functions and learning rate, etc. (Pentors and Pieczarka, 2017, Nagori and Trivedi (2014), Chi-Chung *et al.*, 2012, Gaudart *et al.*, 2004, Utgoff and Stracuzzi, 2002) but also the sample size (Pasin, 2015, Amari *et al.*, 1997). When one or more of these factors are not well specified, the explanatory quality of the model and its predictive performance are negatively affected. They may lead to local minima, over-learning or non-convergence problems, or sometimes to convergence after a large number of iterations. Therefore, some questions are directly linked to the use of such models: How can we select the best structure? What is the adequate learning algorithm to update the model parameters (weights)? What should be the minimum size of data to train MLP? In this paper, we focus on the identification of the optimal combination of characteristics of a multilayer perceptron to obtain its best performance in a crop yield forecasting context. Specifically, we aim (i) to analyze the

influence of structure of a multilayer perceptron on the learning algorithm; (ii) to identify minimum sample size to train a multilayer perceptron and (iii) to compare the performance of Backpropagation algorithm and its extensions for crop yield prediction.

2. Multilayer perceptron neural network (MLP) and factors affecting its predictive perfomance

2.1. Specification of model and learning process

Let $x = (x_1, \ldots, x_p)^\top \in \mathbb{R}^p$ be the vector of inputs, $w_i = (w_{i0}, \ldots, w_{ip})^\top \in \mathbb{R}^{p+1}$ be a parameter vector for the hidden unit i ($1 \leq i \leq m$), $m \in \mathbb{N}$ and $\beta = (\beta_0, \ldots, \beta_m)^\top \in \mathbb{R}^{m+1}$ be a parameter vector for the only output unit. Multilayer perceptron (MLP) function with m hidden units and one output unit can be written as follows:

$$F(\theta, x) = g\left(\sum_{i=1}^{m} \beta_i f\left(\sum_{j=1}^{p} w_{ij}x_j + w_{i0}\right) + \beta_0\right) \tag{1}$$

where $\theta = (w_{10}, \ldots, w_{m0}; w_{11}, \ldots, w_{1p}; \ldots; w_{m1}, \ldots, w_{mp}; \beta_0; \beta_1, \ldots, \beta_m)$; g and f (real value functions) are output and hidden-unit activation functions respectively.

Let $\Theta_m \subset \mathbb{R}^{m(p+2)+1}$ be a compact (i.e. closed and bounded) subset of possible parameters of the regression model family $\mathcal{S} = \{F_\theta(x), \theta \in \Theta_m, x \in \mathbb{R}^p\}$ with $Y = F_\theta(X) + \epsilon$.

Let $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ with n a strictly positive integer be the observed data coming from a true model $(X_t, Y_t)_{t\in\mathbb{N}}$ for which the true regression function is F_{θ^0}, for an θ^0 in the interior of Θ_m.

Learning consists to estimate the true parameter θ^0 from the observations D. This can be done by minimizing the mean square error function:

$$E(\theta) = \sum_{t=1}^{n} \frac{1}{2}(y_t - F(\theta, x_t))^2 \tag{2}$$

with respect to parameter vector $\theta \in \Theta_m$. Different algorithms are used and based on gradient descent procedure.

2.2. Back-propagation algorithm

The basic idea consists to compute the partial derivatives $\frac{\partial E(\theta)}{\partial w_i}$ and $\frac{\partial E(\theta)}{\partial \beta_i}$ by using the Chain rule. There are two steps: First is propagation learning, which allows to compute the error and partial derivatives and the second is back propagation learning, which allows to compute the resulting weight update. From one algorithm to another, only the second step change. We present, the most used and some drawbacks.

2.3. Standard Back-Propagation with Gradient Descent (SBP)

The weights are updated with SBP algorithm by using of the following equation:

$$\begin{aligned}\theta(k+1) &= \theta(k) + \Delta\theta(k) \\ &= \theta(k) - \eta \times \nabla E(\theta)(k) \\ &= \theta(k) - \eta \times \frac{\partial E(\theta)}{\partial \theta}(k)\end{aligned} \tag{3}$$

where $\Delta\theta(k)$ is known as gradient descent, η is learning rate parameter, k is number of iterations and $\nabla E(\theta)$ is the gradient of error.
Parameter η plays a major role in convergence of the algorithm. Small values take large number of iterations to converge to the desired solution. The large values lead to a faster convergence but in some cases, it may overshoot the optimal solution (Gori and Tesi, 1992). In some situations, there is no guarantee to find a global minimum of the error-function. Another problem with gradient descent is the influence of the partial derivate on the size of the weight-step. To make leaning more stable, a momentum term was added to compute the resulting weight update.

2.4. Back-Propagation with Momentum (MBP)

The use of momentum facilitates oscillation attenuation and renders fast convergence (Samanta *et al.*, 2006).

$$\theta(k+1) = \theta(k) - \eta \times \frac{\partial E(\theta)}{\partial \theta}(k) + \mu\Delta\theta(k-1) \tag{4}$$

where $\mu =$ momentum coefficient ($0 \leq \mu \leq 1$). The parameter μ scales the influence of the previous weight-step on the current one. This method works well on many learning tasks but it is not a general technique for gaining stability or speeding up convergence. Usually, when using gradient descent with momentum, the learning rate decreases to avoid unstable learning.

2.5. Quickprop Back-Propagation Algorithm (Quickprop)

It is a variation of momentum algorithm, which was developed to improve the convergence of BP (Fahlman, 1988).

$$\theta(k+1) = \theta(k) - \eta \times \frac{\partial E(\theta)}{\partial \theta}(k) + \max\left(\tau, \frac{\frac{\partial E(\theta)}{\partial \theta}(k)}{\frac{\partial E(\theta)}{\partial \theta}(k-1) - \frac{\partial E(\theta)}{\partial \theta}(k)}\Delta\theta(k-1)\right) \tag{5}$$

where $\Delta\theta(k)$ is the change of the weights in a MLP at the k^{th} iteration and τ is a parameter called the maximum growth factor, which limits the step size (the default value for τ is 1.75). The value of τ can significantly affect the performance of Quickprop. If this value is too small, the algorithm can converge to the overall minimum. Else if it is too large, the network may become unstable and fail to converge.

2.6. Resilient Back-Propagation Algorithm (Rprop)

Rprop stands for 'Resilient backpropagation' and is a local adaptive learning scheme (Riedmiller and Braun, 1993).

$$\theta(k+1) = \theta(k) + \Delta\theta(k) \tag{6}$$

$$\Delta\theta(k) = \begin{cases} \eta^+ \times \Delta(k-1) & \text{if } \frac{\partial E(\theta)}{\partial \theta}(k-1) \times \frac{\partial E(\theta)}{\partial \theta}(k) > 0 \\ \eta^- \times \Delta(k-1) & \text{if } \frac{\partial E(\theta)}{\partial \theta}(k-1) \times \frac{\partial E(\theta)}{\partial \theta}(k) < 0 \\ \Delta\theta(k-1) & \text{else} \end{cases} \tag{7}$$

where $0 < \eta^- < 1 < \eta^+$. Increase and decrease factors are fixed to $\eta^+ = 1.2$ and $\eta^- = 0.5$ based on both theoretical considerations and empirical evaluations. This reduces the number of free parameters to two, namely Δ_0 and Δ_{max}. It is a slight expensive in computation compared with ordinary back-propagation.

2.7. Levenberg–Marquardt algorithm

It is a modification of Newton's method for nonlinear optimization (Marquardt, 1963).

$$\theta(k+1) = \theta(k) - (H - \eta I)^{-1} \nabla E(\theta)(k) \tag{8}$$

where H is a Hessian matrix.
It is based on the concept of quadratic approximation of error function in a local region and finds the minimum solution in a single iteration. If the quadratic approximation is not appropriate, the algorithm may diverge. Searching of an optimal solution using this method requires calculation of the inverse of the Hessian matrix, which should be positive definite.

2.8. Internal structure of a MPL and sample size

There is no precise rule for determining the optimal number of hidden layers and hidden units (El Badaoui *et al.*, 2017, Larochelle *et al.*, 2009, Kenyon and Paugam-Moisy, 1998, Hornik *et al.*, 1991) and therefore it remains one of the unsolved tasks in this research area (Egriogglu *et al.*, 2008). High value of these parameters increases the number of possible computations and for small value, the learning ability of MLP can be affected.
The choice of transfer functions may strongly influence complexity and performance of neural networks (Duch and Jankowski, 1999). Although sigmoïd transfer functions are the most common, there is no *a priori* reason why models based on such functions should always provide optimal decision borders (Duch and Jankowski, 2000). A large number of alternative transfer functions has been described in the literature and there is no precise rule for choosing optimal combination in a MLP.
With regards to the sample size, there is no clear information about a minimum sample that may be consistent enough with the true field of the variable to regress (Pasin, 2015, Amari *et al.*, 1997, Hush and Horne, 1993). The inappropriate choice of the optimum training sample size leads to an under or over-fitting.

3. The dataset

Dataset considered is related to an assessment study of effect of environment factors, pests and number of treatments on cotton yield. The input variables are $x = \{x_1 = \text{Rainfall}, x_2 = \text{Temperature}, x_3 = \text{Average moisture}, x_4 = \text{Insolation}, x_5 = \text{Evapotranspiration}, x_6 = \text{Average density of Helicoverpa to the hectare}, x_7 = \text{Average density of pests to the hectare and } x_8 = \text{number of treatment}\}$ and the output variable is $y = \{\text{cotton yield}\}$. They come from bio-climatic experimental sites (Alafiarou, Angaradebou and Gogounou) of Agricultural Research Center Cotton and Fiber of Benin Republic (CRA-CF, 2008). This dataset is chosen because it presents a strong multiple collinearity between input variables and a non-linear relationship among x and y such as:

$$\begin{aligned}\hat{\phi}(x) &= -24550 + 0.6339x_1 - 839\log(x_2) - 298x_3 + 218.3x_4 + \\ &+ 743.4x_5 + 16450x_6^2 - 0.2115\log(x_7) - 163.5x_8^2.\end{aligned} \tag{9}$$

4. Simulation plan

• *Identification of probable PDF of X*

Ten usual distributions (Normal, Lognormal, Exponential, Gamma, Uniform continuous, Poisson, Negative binomial, Logistic, Geometric and Weibull) were tested on each component of x with "*fitdistrplus*" package (Delignette-Muller and Dutang, 2015) from software R 3.3.6 (R Core Team, 2019). The best distribution is retained when the value of Akaike's Information Criterion (AIC) is the lowest. Results are presented in Table 1.

Table 1. Distribution of explanatory variables

Variables	Distribution	Parameters
$X_1 = Rainfall$	Log-normal	$Log-\mathcal{N}(\mu = 6.86,\ \sigma^2 = 0.16)$
$X_2 = Mean\ temperature$	Normal	$\mathcal{N}(\mu = 28.17,\ \sigma^2 = 0.35)$
$X_3 = Mean\ humidity$	Log-normal	$Log-\mathcal{N}(\mu = 3.94,\ \sigma^2 = 0.04)$
$X_4 = Insolation$	Log-normal	$Log-\mathcal{N}(\mu = 2.08,\ \sigma^2 = 0.06)$
$X_5 = Evaporation$	Log-normal	$Log-\mathcal{N}(\mu = 3.97,\ \sigma^2 = 0.05)$
$X_6 = Number\ of\ Helicoverpa$	Negative binomial	$\mathcal{BN}(\nu = 307.43\ ;\ p = 0.29)$
$X_7 = Total\ density\ of\ insects$	Negative binomial	$\mathcal{BN}(\nu = 552.18\ ;\ p = 0.36)$
$X_8 = Number\ of\ treatment$	Poisson	$\mathrm{P}(\lambda = 0.8)$

• *Generation of population and sample size*

A population of size $N = 10000$ was considered. The output variable Y was generated using equation (9) added to the error ϵ linked to (9). The error was generated following $\mathcal{N}(\mu = 0, \sigma^2 = 1)$. The input variables ($X_1$ to X_8) related to Y were defined using their respective distributions in table 1. Samples with different sizes n_i ($n_i = 25$, 50, 75, 100, 150, 200, 300 and 400) were extracted from the population using the bootstrap technique. Seventy five percent of each sample is used to train the

neural network and 25% to test train the network concerning its the generalization capacity. Before performing the training and testing, the samples were normalized using min-max normalization technique (Priddy and Kelle, 2005):

$$new_v = \frac{v - \min_z}{\max_z - \min_z}(new \max_z - new \min_z) + new \min_z \tag{10}$$

where v is an observation of vector z and new_v is a normalized observation.

• *Prediction with* $3-$*MLP in R software*

The function "*mlp*" of *RSNNS* package (Bergmeir and Benítez, 2012) was used for the prediction. A $3-$MLP model (see equation 1) was used by varying learning algorithms and hyper parameters for each sample size. Five learning algorithms were considered: 1. Standard backpropagation, 2. Backpropagation with momentum, 3. Quickprop algorithm, 4. Resilient backpropagation and 5. Levenberg Maraquardt algorithm. With respect to hyper parameters, 4 combinations of activation functions, AF (f and g, see equation 1) were used: (i) Logistic-Linear (LL); (ii) Logistic-Exponential (LE); (iii) TanH-Linear (TL) and (iv) TanH-Exponentiel (TE). The expression of activation functions considered are: Linear, $h(x) = x$; Logistic, $h(x) = \frac{1}{1+e^{-x}}$; Exponential, $h(x) = e^x$ and Tangent hyperbolic, $h(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. In additional, 19 numbers of nodes in the hidden layer were considered: $2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19$ and 20. In addition, 4 learning rates, LR were considered: $20\%, 40\%, 60\%$ and 80% and as well as the 8 sample sizes.
A total of 500 replications was performed on each sample size to the analyze performance of the method. Initial weights were generated randomly according to the uniform law in the range -3 and 3. The stopping criteria used are the combination of a fixed number of epochs, NE$=$ 1000 and a sufficiently small training error less than or equal to 10^{-6}.

• *Performance criteria and statistical method comparison*

The performance criteria used are: (i) Coefficient of correlation, r; (ii) Coefficient of determination R^2 and (iii) Mean absolute error, MAE (Kazem and Yousif, 2017, Elarabi and Taha (2014)). In the formula below, y and F_θ respectively denote observed output and predicted values output of y, $\bar{y}$ and $\bar{F}_\theta$, their mean and n the test data size.

$$r = \frac{\sum(y_t - \bar{y}_t)(F_\theta(x_t) - \bar{F}_\theta(x_t))}{\sqrt{\sum(y_t - \bar{y}_t)^2} \times \sqrt{\sum(F_\theta(x_t) - \bar{F}_\theta(x_t))^2}} \tag{11}$$

$$R^2 = \frac{\sum(y_t - F_\theta(x_t)) \times (\sum y_t \times \sum F_\theta(x_t))}{\sqrt{(\sum y_t^2 - (\sum y_t)^2)(\sum F_\theta(x_t)^2 - (\sum F_\theta(x_t))^2)}} \tag{12}$$

$$MAE(\%) = \left| \frac{1}{n} \sum_{t=1}^{n} \left(\frac{F_\theta(x_t) - y_t}{y_t} \right) \right| \times 100 \tag{13}$$

The model giving the optimal configuration of its characteristics for a given learning algorithm and with an optimal sample size is the model for which a strong correlation between predicted and observed data ($|r| \geq 0.8$) (Kazem and Yousif, 2017), is observed with R^2 closed to "1" (Shahin *et al.*, 2008) and with low value of MAE (Kazem and Yousif, 2017).
To assess factors which affect performance of the MLP model, *ANOVA* procedure was run on R^2, r and MAE for each learning algorithm.
Interaction plot was considered for significant interactions between hyper parameters of MPL and the learning algorithm.
$Mean$, $minimum$, $maximum$ and $coefficient\ of\ variation$ of the criteria considered (R^2, $|r|$ and MAE) were used to compare learning algorithms performances.

5. Main results

5.1. Analysis of hyper-parameters' effect on the $3-$MPL performance according to learning algorithm

Hyper-parameters' effect on the $3-$MPL performance criteria (R^2 and r) shows same trends. Third and 4th order-interaction effects between number of nodes (N), activation functions (F), learning rate (LR) and sample size (S) on R^2 and MAE are not significant at 5% level for all algorithms except Quickprop. These factors taken individually or their 2nd order interaction effect significantly influence the performance of the $3-$MLP for each algorithm and from one algorithm to another at the 0.1% threshold (Table 2). In addition, interactions N:F and N:L do not affect the performance of the neural model for all the algorithms considered (Table 2).

Table 2. Effect of hyper-parameters on $3-$MLP performance: p-values from ANOVA

Variables	df	SBP		BPM		Quickprop		RBP		LM	
		R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE
Nodes (N)	18	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
LR (L)	3	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.276	0.007
AF (F)	3	0.001	0.001	0.001	0.001	0.001	0.001	0.746	0.001	0.001	0.001
Size (S)	7	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.165	0.001	0.001
N : L	54	0.799	0.001	0.981	0.001	0.080	0.765	0.001	0.001	0.701	0.263
N : F	54	0.001	0.001	0.053	0.001	0.015	0.001	0.955	0.001	0.176	0.001
L : F	9	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.921	0.001
N : S	126	0.001	0.001	0.134	0.098	0.002	0.143	0.161	0.391	0.071	0.394
L : S	21	0.001	0.001	0.001	0.081	0.001	0.001	0.001	0.935	0.001	0.421
F : S	21	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.340

Grey color indicates no significant difference at the 5% threshold.

Interaction plot related to nodes and LR on R^2 and MAE reveals increase in the performance of $3-$MLP with numbers of nodes in the hidden layer and with learning rate for learning algorithms: SBP and BPM (see Figure 1). The Quickprop algorithm

is more sensitive to variation in the number of hidden units and the learning rate (R^2 and MAE) compared with other algorithms. Performance of the LM algorithm varies very slightly according to the number of nodes in the hidden layer, oscillating between ($R^2 = 95$) and ($R^2 = 99.99$) with lower value of MEA. Moreover, beyond 6 nodes and 40% of learning rate, the error increases.

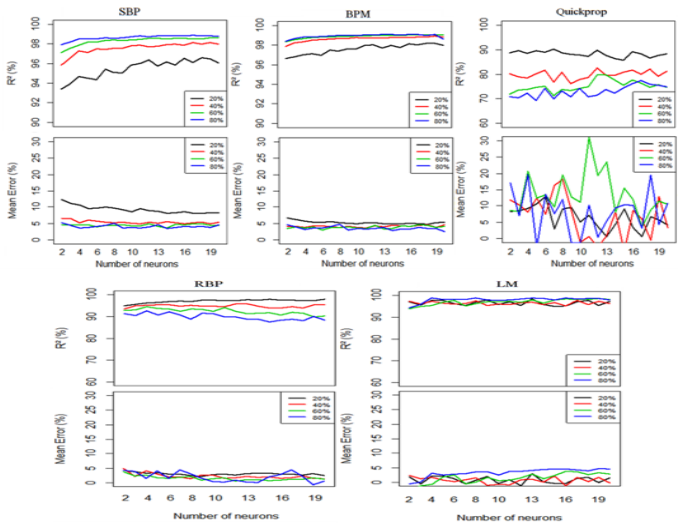

Fig. 1. Interaction plot of nodes and LR on R^2 and MAE

Figure 2 shows that the combination of TanH-linear (TL) as activation functions in the neuronal model gives the best performance for SBP, BPM and Quickprop algorithms. In addition, LM algorithm records the best performance with the combination of TanH-Exponential function (TE) from a LR= 40%.

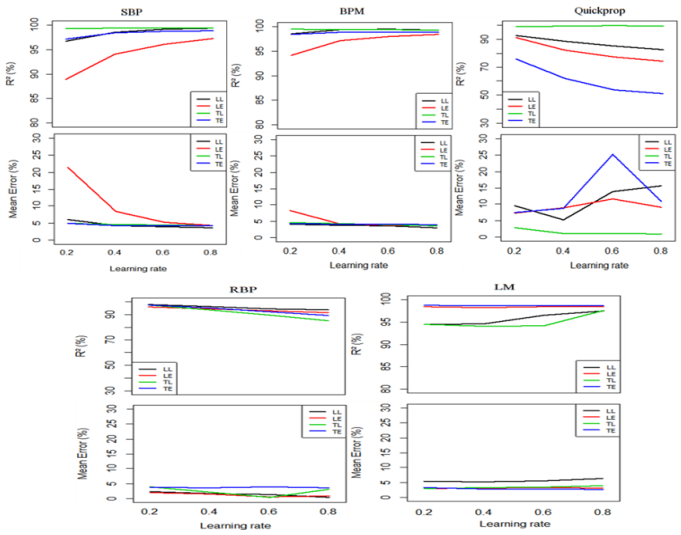

Fig. 2. Interaction plot of activation function and learning rate for R^2 and MAE

5.2. Analysis of sample size effect on $3-$MLP performance according to learning algorithms

Effect of sample size on the performance of $3-$MLP depends on the learning rate and learning algorithm considered (Figure 3). Values of R^2 and correlation coefficient of all algorithms except Quickprop, increase generally with sample size regardless of the learning rate. Low values of MAE are obtained for $n_i = 150$ for any algorithms and learning rates.

5.3. Relative performance of learning algorithms considered

Table 3 presents the performance of the learning algorithms considered for 40% learning rate, 6 nodes with $n_i = 150$. Based on the $8-6-1$ structure, Quickprop and RBP give the lowest R^2 (respectively 95.98 and 96.58) and the highest MAE (respectively 12.03 and 5.71). LM presents the highest R^2 (99.05) and the lowest MAE (1.08) followed by BPM and SBP algorithms respectively.

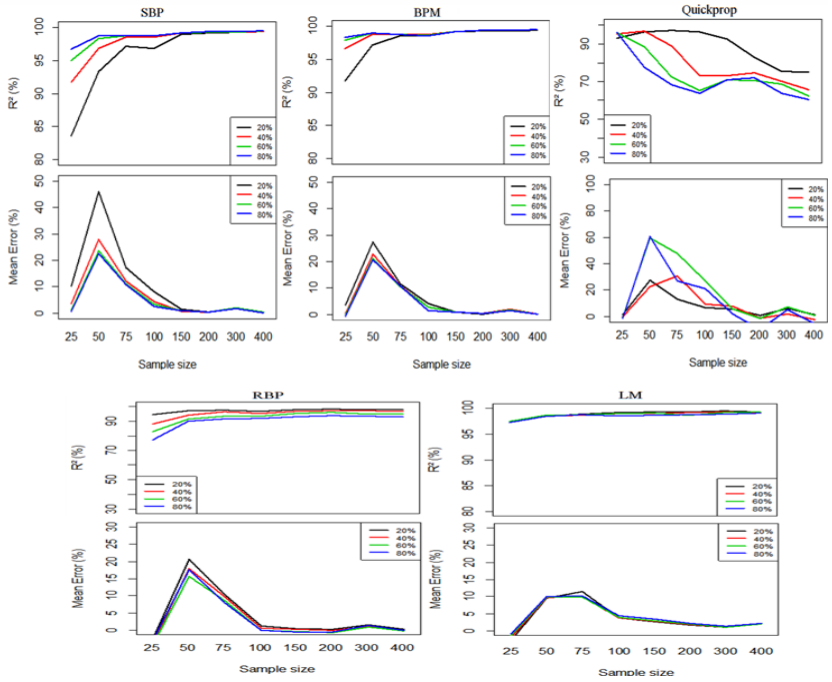

Fig. 3. Interaction plot of the sample size and LR on R^2 and MAE

Table 3. Comparison of learning algorithm performances

Algorithm	AF	Structure	R^2		r		MAE	
			m	$cv(\%)$	m	$cv(\%)$	m	$cv(\%)$
SBP	TL	8-6-1	98.31	1.02	99.10	0.50	3.05	5.15
BPM	TL	8-6-1	98.87	1.01	99.09	0.48	2.32	4.69
Quickprop	TL	8-6-1	95.98	3.68	98.50	4.59	12.03	17.89
Rprop	LL	8-6-1	96.58	2.77	96.76	3.44	5.71	13.57
LM	TE	8-6-1	99.05	0.12	99.06	0.03	1.08	0.35

m: mean; cv: coefficient of variation; R^2: coefficient of determination; r: coefficient of correlation; MAE: mean absolute error.

5.4. Application on real dataset

Results from the simulation study applied to two different databases linked to Alafiarou and Gogounon sites are presented in Table 4. Whatever the neural model

considered with their best characteristics, it outperformed the classical nonlinear model (see Equation 9) . Among the learning algorithms considered, LM was identified as the best ($R^2 = 98.82$) regardless of the database (Table 4). It is followed by BPM ($R^2 = 97.98$) and SBP ($R^2 = 96.95$). In addition, RBP and Quickprop recorded the lowest performance ($R^2 = 96.12$ and $R^2 = 96.51$ respectively).

Table 4. Application on real dataset

Algorithm	AF	Structure	Data of Alafiarou			Data of Gogounou		
			R^2	r	MAE	R^2	r	MAE
SBP	TL	8-6-1	96.95	98.40	3.89	96.52	97.90	3.18
BPM	TL	8-6-1	97.98	98.65	2.27	98.21	99.08	2.04
Quickprop	TL	8-6-1	96.51	97.06	8.59	96.04	97.03	7.81
Rprop	LL	8-6-1	96.12	98.22	4.17	96.03	97.72	4.75
LM	TE	8-6-1	98.82	98.96	0.85	98.94	99.06	0.73
			R^2	r	AIC	R^2	r	AIC
Eq.09			55.59	64.01	1142.26	56.33	65.97	1071.76

R^2: coefficient of determination; r: coefficient of correlation; MAE: mean absolute error

6. Discussion

With the use of SBP and BPM algorithms for learning a $3-$MLP neural model, increasing the number of nodes in the hidden layer improves the performance of the network. This observed trend is contrary to the use of the Quicprop algorithm. These results lead to the conclusion that the choice of the number of nodes in the hidden layer depends on the learning algorithm. The optimal number of hidden neurons obtained is equal to 75% of the number of neurons in the input layer and is in line with conclusion from Salchenberger *et al.*(1992). But some authors like Elarabi and Taha (2014), Masters (1993) have reported that there is no direct and precise accurate method to determine the best number of nodes in a hidden layer. The best performance of a 3-MLP with respect to the optimal choice of learning rate is 40%. This is not far from the 35% obtained by Nagori and Trivedi (2014). On the other hand, the 40% found are in line with those of Uma Rao (2011) who noted possible oscillatory response in case of large learning rate value because of the larger changes in the synaptic weight. These may lead network to be unstable. But our results are similar to those of Rajasekaran and Vijayalakshmi (2012) who stated that the best learning rate is 60%.
Several studies reported that the most commonly transfer function used for 3-MLP is a sigmoidal function in the hidden layer and a linear transfer function in

output layer (Dahikar *et al.*, 2015, Kaan and Arslan, 2007). Our results reveal that the choice of a good activation funct()ions depends on the learning algorithm to be used. Thus, we observed that the SBP, BPM and Quickprop algorithms give their best performance with the combination of tangent hyperbolic in hidden layer and linear in output layer.
A minimal sample size of 150 observations allowed the algorithms to perform better. Whereas, in the literature, the minimum sample size for using multilayer perceptron should be based on the number of connections in the network (Cottrell *et al.*, 2012, Amari *et al.*, 1997). The number of connections in the network depends on three parameters: input number, number of nodes in the hidden layer and output number. Since our study has only crunched the number of nodes in the hidden layer, we could not confirm or deny hypotheses which stated that the sample size is correlated with the number of connections in the network.
Levenberg-Marquardt algorithm (LM) gives a better predictive performance than the other four algorithms considered using the same hyper-parameters. This result is identical to those obtained by Hicham *et al.*(2013); Kaan and Arslan, 2007. However, HüskenIgel and Igel (2002) report in their in works that the Rprop algorithm is one of the best performing learning algorithms for neural networks with an arbitrary topology, but without taking into account the influence of hyper-parameters.

7. Conclusion

This study focused on the problem of optimal choice of the number of hidden neurons, appropriate set of transfer functions, learning algorithm, learning rate and required sample size to identify non-linear systems by multilayer perceptron neural networks (MLP), especially for crop yield prediction. Overall, Levemberg Marquard algorithm is the best algorithm which gives good predictive performance with minimum sample size equal to 150 associated to 40% as learning rate and a number of 6 nodes in hidden layer (75% of the number of input variables). The activation functions used are tangent hyperbolic in hidden layer and exponential in output layer. Further studies are required to control the interval of initial weight, the data normalization and application of results obtained to other fields. Moreover, the same work can be applied in classification using neural network.

Acknowledgment

This work was financially supported by the African Centre of Excellence in Mathematics and Applications (CEA-SMA). The authors are also grateful to the anonymous referee and to the associated editor for their helpful suggestions that lead to an improved paper.

References

Amari, S. I., Murata, N., Muller, K. R., Finke, M., and Yang, H.H. (1997). Asympotic statistical theory of overtraining and cross-validation. *IEEE Transactions on Neural Networks*, 8 (5), 985-996.

Asogwa, O. C., and Oladugba, A.V. (2015). On The Comparison of Artificial Neural Network (ANN) and Multinomial Logistic Regression (MLR). *West African Journal of Industrial and Academic Research.* 13(1), 1-9.

Badran, F., and Thiria, S. (2002). Les perceptrons multicouches : de la régression non-linéaire aux problèmes inverses. *J. Phys.* IV France 12 (1), 157–188.

Bates, D.M., and Watts, D.G. (1988). *Nonlinear regression analysis and its applications*. New York: John Wiley & Sons.

Bergmeir, C., and Benítez, M.J. (2012). Neural Networks in R Using the Stuttgart Neural Network Simulator: RSNNS *Journal of statistical software*, 46 (7), 26. https://DOI:10.18637/jss.v046.i07

Bishop, C.M. (1995). *Neural Networks for pattern recognition*. Oxford: Oxford University Press.

Chen, S., Billings, S., and Grant, P.M. (1990). Nonlinear system identification using neural networks. *Int. J. Control.* 51 (1), 1191–1214.

Chi-Chung, C., Sin-Chun, N., and Lui, K.A. (2012). Improving the Quickprop algorithm. *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 10-15. https://DOI:10.1109/IJCNN.2012.6252546

Cottrell, M., Olteanu, M., Rossi, F., Rynkiewicz, J., and Villa-Vialaneix, N. (2012). Neural Networks for Complex Data. *Künstliche Intelligenz* 26:1-8 .https://DOI:10.1007/s13218-012-0207-2

CRA-CF (2008). Centre de recherches agricoles coton et fibres (CRA-CF) de l'Institut national des recherches agricoles du Bénin (INRAB). *Rapport de l'étude de la modélisation des rendements du Coton sur les sites de Alafiarou, Angaradebou and Gogounou*, pages 56.

Cuauhtémoc, L-M (2015). Predictive accuracy comparison between neural networks and statistical regression for development effort of software projects. *Journal Applied Soft Computing archive*, Volume 27 Issue C, 434-449. https://doi>10.1016/j.asoc.2014.10.033

Dahikar, S.S., Rode, V.S., and Deshmukh P. (2015). An Artificial Neural Network Approach for Agricultural Crop Yield Prediction Based on Various Parameters, *International Journal of Advanced Research in Electronics and Communication Engineering*, 4(1): 94-98.

Delignette-Muller, M.L., and Dutang, C. (2015). fitdistrplus: An R package for fitting distributions. *Journal of statiscal stofware* 64 (4), 1-34. https://DOI:10.18637/jss.v064.i04

Duch, W., and Jankowski, N. (1999). Survey of neural transfer functions. *Neural Computing Surveys*, 2: 163–213

Duch, W., and Jankowski, N. (2000) Taxonomy of neural transfer functions. *Proceedings of the IEEE-INNS-ENNS International*, pages 6. https://DOI:10.1109/IJCNN.2000.861353

Egriogglu, E., Hakam Aladag C., and Gunay, S. (2008). A new model selection strategy in artificial neural networks. *Applied Mathematics and Computation.*195 (2), 591-597.https://doi.org/10.1016/j.amc.2007.05.005

El Badaoui, H., Abdallaoui, A., and Chabaa, S. (2017). Optimization numerical the neural architectures by performance indicator with LM learning algorithms. *J. Mater. Environ. Sci.* 8 (1), 169-179.

Elarabi, H., and Taha N.F. (2014). Effect of Different Factors of Neural Network on Soil Profile of Khartoum State. *American Journal of Earth Sciences*. 1(3), 62-66.

Fahlman S.E. (1988). Fast learning variations on back-propagation: An empirical study. *Proceedings of the 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski, eds., pp. 38 – 51, Morgan Kaufmann, San Mateo, California.

Fausett, L. (1994). *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA. ISBN:0-13-334186-0

Gaudart, J., Giusiano, B., and Huiart, L. (2004). Comparison of the performance of multilayer perceptron and linear regression for epidemiological data. *Computational Statistics & Data Analysis*, 44(4); 547-570.DOI:10.1016/S0167-9473(02)00257-8

Gori, M., and Tesi, A. (1992). On the problem of local minima in back-propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1) 76–86

Graupe, D. (2013). *Principles of Artificial Neural Networks*: 3rd Edition. World Scientific Publishing Company, 500p.

Hüsken, M., and Igel, C. (2002). Balancing learning and evolution. *GECCO'02 Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, Pages 391-398.

Hicham, El.B., and Abdelaziz, A., and Samira, C. (2013). Etude des effets des algorithmes d'apprentissage et des fonctions de transfert sur la performance des modèles statistiques neuronaux : Application dans le domaine météorologique. *International Journal of Engineering Research and Development* 9(6) 15-26.

Hornik K., Stinchcombe, M., and White, H., (1991). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Journal Neural Networks*. 3 (5), 551-560. https://DOI:10.1016/0893-6080(90)90005-6

Hush, D.R., and Horne, B.G. (1993). Progress in supervised neural networks. *IEEE Signal Processing*, 10 (1), 8-39. https://DOI:10.1109/79.180705

Kaan, Y., and Arslan, S. (2007). Stochastic modeling approaches based on neural network and linear–nonlinear regression techniques for the determination of single droplet collection efficiency of counter current spray towers. *Environ Model Assess*, 12 (1), 13–26. https://doi:10.1007/s10666-006-9048-4

Kazem, A.H., and Yousif H.J., 2017. Comparison of prediction methods of photovoltaic power system production using a measured dataset. *Energy Conversion and Management*, 148:1070-1081. http://dx.doi.org/10.1016/j.enconman.2017.06.058

Kenyo, C., and Paugam-Moisy, H. (1998). Multilayer neural networks and polyhedral dichotomies. *Annals of Mathematics and Artificial Intelligence*, 24 (1-4), 115–128.

Kordos, M., and Duch, W. (2008). Variable Step Search Algorithm for Feedforward Networks. *Neurocomputing*. Vol. 71, Issue 13-15.

Larochelle H., Bengio Y., Louradour, J. and Lamblin, P. (2009). Exploring strategies for training deep neural networks. *The Journal of Machine Learning Research*, 10:1–40.

Lindsey, J.K. (2001). *Nonlinear models for medical statistics*. 2nd ed. Oxford Stat. Sci. Ser. 24. Oxford Univ. Press, Oxford, UK.

Mário, W.L.M, Joel, J.P.C.R, Neeraj, K.J., and Al-Muhtadi, V.K. (2017). Evolutionary radial basis function network for gestational diabetes data analytics. *Journal of Computational Science*. http://dx.doi.org/10.1016/j.jocs.2017.07.015

Marquardt, W.D. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2), 431–441. https://doi.org/10.1137/0111030

Masters, T. (1993). *Practical neural network recipes in C++*, Academic Press, SanDiego, California.

Nagori, V., Trivedi, B. (2014). Fundamentals of ANN, Back propagation algorithm and its parameters. *International journal of science, technology and management*, 4(1), 69-76.

Pasin, A. (2015). Artificial neural networks for small dataset analysis. *J Thorac Dis*, 7(5), 953-960. https://doi:10.3978/j.issn.2072-1439.2015.04.61

Pentors, K., Pieczarka, K. (2017). Applying an artificial neural network approach to the analysis of tractive properties in changing soil conditions. *Soil and Tillage Research*, 165(C) 113–120. https://DOI10.1016/j.still.2016.08.005

Perai, A.H., Nassiri-Moghaddam, H., Asadpour S., Bahrampour, J., and Mansoori, G. (2010). A comparison of artificial neural networks with other statistical approaches for the prediction of true metabolizable energy of meat and bone meal. *Poultry science*, 89(7), 1562-1568. doi:10.3382/ps.2010-00639

Priddy, K.L., and Kelle, E.P. (2005). Artificial neural networks - an introduction. *SPIE Tutorial Texts in Optical Engineering*, Vol. TT68, pages 180.

R Core Team (2019). R 3.3.6: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austrianhttps://www.R-project.org

Rajasekaran, S., and Vijayalakshmi, G.A.P. (2012). *Neural networks, fuzzy logic and genetic algorithms: synthesis and applications*, New Delhi: PHI Learning.

Riedmiller, M., and Braun, H. (1993). A direct adaptive method for faster back propagation learning: The RPROP algorithm. In H. Ruspini, editor, *Proceedings of the IEEE International Conference on Neural Networks (ICNN)*, pages 586-591, San Franciscio,California.

Salchenberger, L.M., Cinar, E.M., and Lash, N.A. (1992). Neural networks: A new tool for predicting thrift failures. *Decision Science*, 23 (4), 899-916.

Samanta, B., Bandopadhyay, S., and Ganguli, R. (2006). Comparative Evaluation of Neural Network Learning Algorithms for Ore Grade Estimation. *Mathematical Geology*, Vol. 38, No. 2, 175-197. https://DOI:10.1007/s11004-005-9010-z

Shahin, M.A., Jaksa, M.B., and Maier, H.R. (2008). State of the art of artificial neural networks in geotechnical engineering. *Electronic Journal of Geotechnical Engineering*, 8 (1), 1-26.

Sibi P., Allwyn, Jones, S., and Siddarth, P. (2013). Analysis of different activation functions using back propagation neural network. *Journal of theoretical and applied information technology*, 47(3), 1264-1268.

Uma Rao, K. (2011). *Artificial Intelligence and Neural Networks*, Pearson Education, pages 468.

Utgoff, P.E., and Stracuzzi, D.J. (2002). Many-layered learning. *Neural Computation*, 14 (10): 2497–2529. https://doi/abs/10.1162/08997660260293319

Uyanik, G.K., and Güler, N. (2013). A Study on Multiple Linear Regression Analysis. *Procedia - Social and Behavioral Sciences*, Vol 106, 10, 3234-240.https://doi.org/10.1016/j.sbspro.2013.12.027

Wang, S.L., Zheng, L. and Dai, J.T. (2014). Empirical Likelihood Diagnosis of Modal Linear Regression Models. *Journal of Applied Mathematics and Physics*, 2, 948-952. http://dx.doi.org/10.4236/jamp.2014.210107