ANALYSIS OF ACTIVATION FUNCTIONS IN NEURAL NETWORKS FOR MINIMIZING THE LOSSES IN THE POWER FLOW SYSTEM

Rafael Rangel Szillat

Alessandra Freitas Picanco

rafael.r.szillat@gmail.com

alepicanco@ifba.edu.br

Instituto Federal de Educação, Ciência e Tecnologia da Bahia

Rua Emídio dos Santos, s/n - Barbalho, 40301-015, Salvador, Bahia, Brasil

Abstract. *Power Flow is a numeric instrument to determine permanent conditions of the electric system. It consists of nonlinear solution methods, which result in iterative algorithms methods to find the solution to this problem. In this context, several authors have studied techniques to numerical optimization of Load Flow from Newton-Raphson to reduce energy losses. The present paper shows an optimization Artificial Neural Network algorithm of the first order, using Levenberg-Marquardt and a Novel Levenberg-Marquardt, to reduce the losses of power system using the activation functions: Hyperbolic Tangent, Bi-hyperbolic Tangent, Gauss, Sigmoid, Elliot and Ackley. The results represent the losses of each function applicate in IEEE-6, IEEE-30, IEEE-118 and IEEE-300 bus.*

Keywords: *Power Flow, Levenberg-Marquardt, Losses, Neural Network*

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

1 INTRODUCTION

The electric power system has the objective to generate, transmit and distribute energy in an uninterrupted and quality way. In this context, study of power flow comes with a decision and support tool for planning the electrical system. Studies indicate that developing countries over 20% of the generated energy are lost, being about 13% of this loss occurs in distribution network (Ramesh et al., 2009) and (Abdelfatah and Brahim, 2011).

Power flow is very expensive and complicated, in this case, planning and operation the system becomes more difficult. For the solution of this problems are necessary mathematical methods that consist in the resolution of algebraic equations that determine the steady state of the system. Normally, that problems are solved with the Newton-Raphson (NR) and Gauss-Seidel Method, being the NR the most usual, for converging in a faster way with few iterations (Afolabi et al., 2015) and (Grainger and Stevenson, 1994).

This solution results in the values of magnitude and phase of the voltage of each bus, allowing to calculate how much of active and reactive power is transported in the transmission lines, and consequently, the losses of the system. However, due to the complexity of the operation and the continuous variations of generation and load consumption, the search for an optimal point for Power Flow (PF) solution requires a greater computational work.

Several authors search to optimize the reduction of active power losses. Whereas, for the analysis of these types of systems, there are a large number of data that must be analyzed, has a common solution, the use of optimization and analysis programs or learning algorithms such as Genetic Algorithms (GA) and Artificial Neural Network (ANN). According to Chopade and Bikdash (2011) one form to reduce losses in a system with nonlinear loads is reducing the lagging currents using capacitors at specific points, determined by the implementation of GA. Nakawiro and Erlich (2009) propose a mixture of offline ANN utilization, trained to solve PF problems and, subsequently, optimization this with GA.

Some studies show the application of ANN for power flow optimization. Poucar et al. (2005) suggest optimizing these problems with an alternative PF analyses model, diagonalizing the parameters of the admittance matrix of those systems, to improve the training conditions of the neural network, and later apply an ANN based on a second-order algorithm with the Levenberg-Marquardt (LM) method. Distribution feeder reconfiguration was studied by Olamaei et al. (2012), which, builds on a Honey Bee Mating algorithm, developed a code to reduce technical losses. Fathabadi (2016) present reconfiguration of the power distribution system using a new model of dynamic fuzzy c-means (dFCM) clustering based on ANN. Puppala and Chandrarao (2015) demonstrate that the voltage stability in PF can also be determinate using linear regression, with help from an ANN, which reduces de iteration number.

From that, in this research, based on the researched methods, optimize the values of magnitude and angle of voltage in the bus resulting from the PF, applying the Levenberg-Marquardt algorithms. Conform (Lagace et al., 2008) and (Pourbagher and Derakhshandeh, 2016) this technique could be used in ill-conditioned systems, optimizing the convergence of the NR method. This algorithm reduces the oscillation error of the nonlinear problems, efficiently finding the minimum local (Suratgar et al., 2005) and (More, 1997).

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering
P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

Based on this method, this paper shows how the power flow behaves for different activation functions trained with an optimization ANN algorithm, of the first order, using LM and a variation of this code: Novel Levenberg-Marquardt (NLM), which consists of updating the parameters of the activation function, as well such as the weights of the neural network. The use of this algorithm has the purpose of optimizing the system, determined by the interactive NR method reducing the technical losses of the PF system. This article is divided into: Section 2, which addresses the concepts of PF; Section 3 with the ANN concepts and the activation functions applied in this study; Section 4, modeling and Section 5, analysis of results.

2 POWER FLOW (STEVENSON AND GRAINGER, 1994)

The Power Flow study consists in simplifies the entire electrical system encompassing generations, transformations, transmissions and distributions of energy in the shape of a bus. These have the necessary information to construct the load flow of the electrical set. From this it is possible to read these data to provide operation actions, avoiding regimes of contingencies, shunts and the needing of expansion of new networks and substations of electrical power system.

Normally on the power flow's questions, there are four variables, which are associated with each bus. They are: Active Power (P), Reactive Power (Q), Phase associate to the voltage (Ɵ) and voltage magnitude (V). These data form equations, based on Kirchhoff laws, that cannot be solved by ordinary algebraic methods, being necessary an application of numerical methods to solute this kind of problem.

These buses can be subdivided into three types, being classified in according with the information they provided. This also classifies the information that the bus has and what can be calculated. Load Buses (PQ) are the buses that have installed the charges, where have only actives and reactive power. They can be exemplified by substations that feed cities and/or industries. In these buses, must be calculated the information referring to voltage and phase. Voltage-Controlled Buses or Generation Buses (PV), are where the energy generators are. They have this name due to voltage, that can be controlled by the variation of generation excitation, and the power injected into the system. In this bus, the reactive power of the system and the phase is calculated. Finally, there is the Reference Bus, known by Slack Bus. This bus is used to close circuit loop, in accord with the Kirchhoff laws. By this aspect, a magnitude and the voltage phase serves to reference to another bus. At this point, active and reactive powers are calculated.

Based in the buses information and the admittances existents in the power lines, equations are formed to calculate the lack of information in each bus, in other words, the voltage magnitude and phase at each point. These data are directly connected to the electrical losses of the system. For that, there is the NR method as an applied common way to solve PF with the network's parameters, as admittances and also active and reactive power generated and consumed. The Eq. (1) presents the element of the admittance matrix that associates the bus i with bus j.

$$Y_{ij} = G_{ij} + jB_{ij}. \tag{1}$$

Where, G_{ij} represents the conductance from bus i to bus j and B_{ij} represents the susceptance from bus i to bus j.

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

The flow of active power (P_i) and reactive (Q_i) in bus i are represented by Eq. (2) and Eq. (3).

$$P_i = V_i{}^2 G_{ii} + \sum_{j=1;j\neq i}^{N} V_i V_j [B_{ij} \cdot sen(\theta_i - \theta_j) + G_{ij} \cdot \cos(\theta_i - \theta_j)]. \quad (2)$$

$$Q_i = -V_i{}^2 B_{ii} + \sum_{j=1;j\neq i}^{N} V_i V_j [G_{ij} \cdot sen(\theta_i - \theta_j) - B_{ij} \cdot \cos(\theta_i - \theta_j)]. \quad (3)$$

Where V_i represents voltage magnitude in the bus i, V_j means to be tension on the bus j, θi is the angle of tension in bus i and θj is the tension in bus j.

The active and reactive losses of the flow system are defined in accord of the Eq. (4) and Eq. (5), respectively.

$$P_{loss} = \sum(P_{in} - P_{out}). \quad (4)$$

$$Q_{loss} = \sum(Q_{in} - Q_{out}). \quad (5)$$

The modeling of the PF presents restrictions on the voltage, Eq. (6) and the reactive power, Eq. (7). These restrictions are determined by the energy management company, to avoid under/over voltages, or excess of reactive power, to come to consumers and damage your equipment and/or make more spending. The values define the limit that the solution can have on the system.

The NR Method described on Eq. (8) is a deterministic method that is frequently used solving problems related to the PF. Where J_{NR} means to be the Jacobian Matrix of the NR Method given by the Eq. (9).

$$V_{min} \leq V_i \leq V_{max}. \quad (6)$$

$$Q_{min} \leq Q_i \leq Q_{max}. \quad (7)$$

$$J_{NR} \cdot \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} = \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}. \quad (8)$$

$$J_{NR} = \begin{bmatrix} \frac{\partial P}{\partial \theta} & \frac{\partial P}{\partial V} \\ \frac{\partial Q}{\partial \theta} & \frac{\partial Q}{\partial V} \end{bmatrix}. \quad (9)$$

3 NEURAL NETWORK (HAYKIN, 2001)

Artificial Neural Network is a numerical technique, inspired on the information processing by the human brain. On this way, ANN has characteristics of learning, that is, the acquired knowledge is stored on the synaptic weights, which are connected to neurons. Therefore, in a similar way to the human brain, ANN has the neurons, which are processing units where the mathematical functions, usually nonlinear, are calculated. The synapses are notices between the neurons from the pre-synaptic membranes (input data) to postsynaptic membranes (output data). This connection has an attraction force, known by "weights", they are distinguished for entrance and exit for each neuron. The bias has as function modify the input data to the activation function. The activation function is responsible for limiting and validating the amplitude of the neuron exit. The application of these nonlinear functions can be used inside an ANN to approximate any numerical value (Miguez, 2012) and (Mhaskar and Micchelli, 1992).

3.1 Activation Functions

The activation functions read on this paper were:

Hyperbolic Tangent – Presents growing practically linear on your center and asymptotic on your deflection points and your values fluctuate between -1 and 1. The hyperbolic tangent function is given by Eq. (10), and your respectability derivate is given by Eq. (11) (Miguez, 2012).

$$f(x) = a \cdot \tanh(b \cdot x). \tag{10}$$

$$f'(x) = a \cdot b \cdot (1 - tanh^2(b \cdot x)). \tag{11}$$

a) Bi-hyperbolic Tangent – It is an asymmetric function, that can behave with a lot of different ways in accord of your parameters, which allows your using in neural networks more complexes. This function is suggested by (Miguez, 2012), which demonstrated a function with fast convergence, that permits operation with a low number of neurons. The Eq. (12) presents the Bi-hyperbolic Tangent function and Eq. (13) the derivate.

$$f(x,\rho,a,b) = \sqrt{\rho^2\left(x+\frac{1}{4\rho}\right)^2 + a^2} - \sqrt{\rho^2\left(x-\frac{1}{4\rho}\right)^2 + b^2} + \frac{1}{2}. \tag{12}$$

$$f'(x,\rho,a,b) = \frac{\rho^2\left(x+\frac{1}{4\rho}\right)^2}{\sqrt{\rho^2\left(x+\frac{1}{4\rho}\right)^2 + a^2}} - \frac{\rho^2\left(x-\frac{1}{4\rho}\right)^2}{\sqrt{\rho^2\left(x-\frac{1}{4\rho}\right)^2 + b^2}}. \tag{13}$$

To the development of this network, was adopted a parameter ‘a’ equal to ‘b’, in a way that your mold gets closer to the others activation functions, with images varying between 0 and 1, offering a symmetrical property, according to what was presented by Xavier (2005).

b) Gauss – Usually requested to calculate probabilities, the Gaussian function allows, for the variation of parameters, find patterns to functions with high precision (Miguez, 2012) and (Berhold, 1973). The function is used as a comparison with other functions. Sibi, Jones and Siddarth (2013) using the Backpropagation (BP) method, and realized that the Gaussian function got the minor error on the rating problem of noxious mushrooms and eatables. This function can be described as Eq. (14), and the derivate is defined by Eq. (15).

$$f(x) = \frac{e^{-\frac{1}{2}\cdot\left(\frac{x-a}{b}\right)^2}}{a\sqrt{2\cdot\pi}}. \tag{14}$$

$$f(x) = \frac{(2\cdot a - 2\cdot x)\cdot e^{-\frac{1}{2}\cdot\left(\frac{x-a}{b}\right)^2}}{2\cdot a\cdot b^2\cdot\sqrt{2\cdot\pi}}. \tag{15}$$

c) Sigmoid – It’s the Function (16) with more applicability on neural networks, is purely growing and has relatives maximum/minimum values. This function has an asymptotical growing, which means to large values, the function operates in a saturation region.

This function has an advantage because it has a nonlinearity weak, your growth is practically linear, on the other hand, the function has a high cost, due to your

calculation being based on Taylor expansion (Miguez, 2012). The Eq. (16) expresses the Sigmoid function and Eq. (17) the derivative.

$$f(x) = \frac{1}{1+e^{-a(x-c)}}. \tag{16}$$

$$f'(x) = a\frac{1}{1+e^{-a(x-c)}} \cdot (1 - \frac{1}{1+e^{-a(x-c)}}). \tag{17}$$

Beaufays et al. (1994) uses this function on a neural network known by BP. It's applied to resolve complex problems of frequency control of the charge of nonlinear power systems.

d) Elliot – It's one more sigmoidal function – that has delimited values and an only growing bend. It's also a function, usually applied in neural network training, given by Eq. (18). Miguez (2012) and Sibi, Jones and Siddarth (2013) present this function as an alternative to activation function to the neural network, showing a good efficacy. The derivative of Elliot is express on Eq. (19).

$$f(x) = \frac{\frac{x}{1+|x|}+1}{2}. \tag{18}$$

$$f'(x) = \frac{1}{2(1+|x|)^2}. \tag{19}$$

e) Ackley – Has been used in optimization processes of genetic algorithms, tried to adapt it to the neural network, because it has a vast number of local minimums, according can be seen in Fig. 1 (Back, 1996). Lockett and Miikkulainen (2011) used the function in a genetic algorithm, known by Real-Space Evolutionary Annealing, verifying the convergence to the local minimums and the time for this intention. The function form is described according to Eq. (20) and his derivative in Eq. (21). Where a, b, c and d are non-negatives parameters.

$$f(x,y,z) = -a \cdot e^{-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}} - e^{-b\sqrt{\frac{1}{d}\sum_{i=1}^{d}\cos(2\cdot\pi\cdot x_i)}} + a + e. \tag{20}$$

$$f'(x,y,z) = \frac{a\cdot b\cdot x}{d\cdot e^{b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}}\cdot\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}} - \frac{b\cdot 2\cdot\pi\cdot\sin(2\cdot\pi)}{2\cdot d\cdot e^{b\sqrt{\frac{1}{d}\sum_{i=1}^{d}\cos(2\cdot\pi\cdot x_i)}}\cdot\sqrt{\frac{1}{d}\sum_{i=1}^{d}\cos(2\cdot\pi\cdot x_i)}}. \tag{21}$$

The described functions are exemplified in Fig. 2. Notice that in Fig. 2 it is possible to realize the difference, mainly on Gauss and Bi-hyperbolic Tangent functions, due to nonsigmoidal functions. Seen that Bi-hyperbolic tangent function has behaviors well different than the other functions. These parameters vary heavily in accord of the parameters selected on the Eq. (20).

3.2 Levenberg-Marquardt Training

The Levenberg-Marquardt (LM) algorithm has the aim to guarantee that the Hessian Matrix H(x) is invertible (Yu and Wilamowski, 2011). At this context, the approach uses the damped Gauss-Newton method, while h_{lm} is defined by the Eq. (22) (Madsen et al., 2004).

$$H(x).h_{lm} = (J(x)'.J(x) + \mu I).h_{lm} = -g. \tag{22}$$

Where μ is the damping parameter or combination coefficient that indicates the speed of the steepest descent direction and g is defined by Eq. (23), and e is the approximated error, the difference between the desired value (target) and the obtained, according to Eq. (24), x can assume the value of voltage or angle (V or θ). The value of x is always updated conforming Eq. (25), in $F(\chi)$ is the output of neural network function. The value of χ represents random values of starting that are associated with the new exit of function $F(\chi)$. These values are defined by the activation function.

$$g = J(x)'.e \tag{23}$$

$$e = x_t - x_{new}. \tag{24}$$

$$x_{new} = x + F(\chi). \tag{25}$$

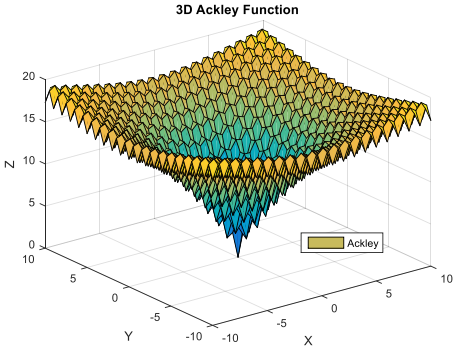

Figure 1. Ackley Function

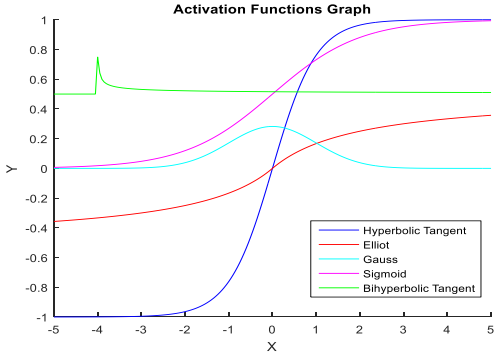

Figure 2. Activation Functions

The Jacobian matrix on the LM training is composed of the derivatives between $F(\chi)$ and the weights W, according Eq. (26) (Yu and Wilamowski, 2011).

$$J(W) = \begin{bmatrix} \frac{\partial F(\chi)}{\partial W_1} \\ \vdots \\ \frac{\partial F(\chi)}{\partial W_n} \end{bmatrix}. \tag{26}$$

The update of the values of μ is calculated by the rate of gain conform Eq. (27) where $L(0) - L(h_{lm})$ is defined in Eq. (28) (Madsen et al., 2004).

$$\varrho = \frac{E(x) - E(x + h_{lm})}{L(0) - L(h_{lm})}. \tag{27}$$

$$L(0) - L(h_{lm}) = \frac{1}{2}.h_{lm}'(\mu\, h_{lm} - g). \tag{28}$$

Where E is the medium quadratic error $E = \frac{1}{2}\sum e^2$.

4 MODELING

The electrical system modeling on ANN was based on a paper of Paucar et al. (2005). Paucar et al. work with the conductance normalization process and susceptance of the admittance matrix, while in this paper was used a diagonalization according to Eq. (29) and Eq.

(30) (Oliveira and Picanco, 2015). The aim is to minimize the input data of ANN, maintaining accuracy.

$$G_d = V_g^2 \cdot G_d^{\lambda} - P_d + Pg_{pv}. \tag{29}$$

$$B_d = -V_g^2 \cdot B_d^{\lambda} - Q_d. \tag{30}$$

Where Gd and Bd are, respectively, the conductance and susceptance, V_g is the tension in generation bus, extracted from the matrix after diagonalization by auto values; P_d and Q_d are the active and reactive powers in bus; Pg_{pv} is the active power in bus PV. It's worth pointing that the admittance matrix contains the information of the transformers and capacitors bank.

The initial data were based on Wu (2008) paper, on what the function is given by Eq. (31).

$$\mathrm{F}(x, a, b, r) = \sum W_3 . f((W_1{}' . x + W_2) + W_3). \tag{31}$$

Being the values of W_1 and W_3 withdrawn of a matrix β, which one is updated, according to neural network process, and initially, generated randomly and defined according to Eq. (32) on what N represents the number of neurons. The random values of β are limited by [-ε_1,ε_1] on $\varepsilon_1$varied depending on each system and activation function studied. The value of W_2 is the bias and is one-unit vector defined by Eq. (34). The values of W_1 and W_3 are determined from the Eq. (33) and Eq. (34). From this, it is possible to obtain values of the exposed weights on Jacobian. A function *f* is the activation function, that will be discussed later.

To NLM, was considered the parameters of activation function, defined by P_1 and P_2, that are started by random ways, according to Eq. (36) – which α is limited by [0,ε_2], and ε_2 is different for each type of activation function and type of system. How P_1 e P_2 is calculated is defined in Eq. (37) and Eq. (38). From this P_1 e P_2 will also be changed conform the network update. These parameters are exclusives of each function, conform were presented on subitem 3.1. The Elliot function, for didn't present parameters on your original form, was updated only according to the weights W_1, W_2 and W_3.

$$\beta = random\ values\ (3 \cdot N + 1). \tag{32}$$

$$W_1 = \beta\begin{bmatrix} 1 \\ \vdots \\ N \end{bmatrix}. \tag{33}$$

$$W_2 = \begin{bmatrix} 1 \\ \vdots \\ N \end{bmatrix}. \tag{34}$$

$$W_3 = \beta\begin{bmatrix} N+1 \\ \vdots \\ 2.N \end{bmatrix}. \tag{35}$$

$$\alpha = random\ values(3 \cdot N + 1). \tag{36}$$

$$P_1 = \alpha\begin{bmatrix} 1 \\ \vdots \\ N \end{bmatrix}. \tag{37}$$

$$P_2 = \alpha\begin{bmatrix} N+1 \\ \vdots \\ 2.N \end{bmatrix}. \tag{38}$$

Based on this, the neural network was also updated following these parameters, from the Eq. (39) and Eq. (40).

$$\beta = \beta + h_{lm}'. \tag{39}$$

$$\alpha = \alpha + h_{lm}'. \tag{40}$$

When the values of conductance and susceptance were found and were structured according to Eq. (29) and Eq. (30), were defined the input of the LM and NLM algorithm and the output, V and Θ, were associated with each input, respectively. The model was set with a hidden layer and activation functions according to item 3.1. The scheme used on the neural network can be described in Fig. 3.

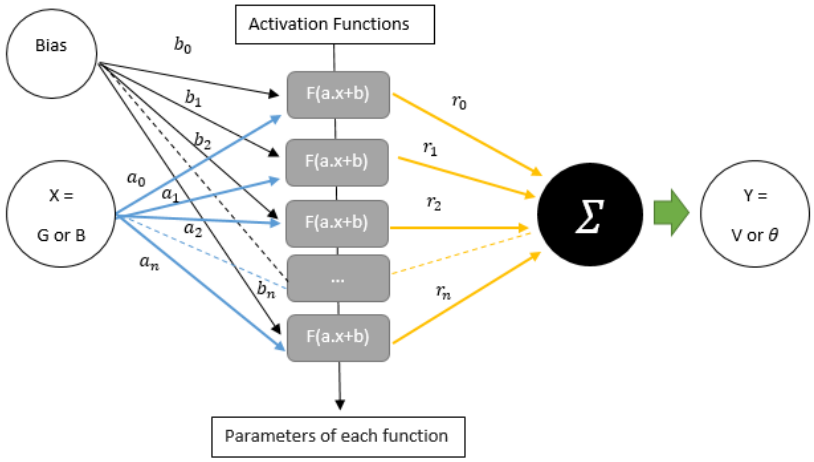

Figure 3. Schematic diagram of the Novel Levenberg-Marquart

Based on that, the algorithm was described based on these follow stages:

Step 1 – Initialize Power Flow data:

1. Import information from the PF bus on MATPOWER (Zimmerman et al., 2011) and use the Newton-Raphson algorithm to obtain values of V (voltage magnitude) and θ (voltage phase);
2. Utilize Eq. (29) and Eq. (30) to find the value of G_d and B_d.

Step 2 – Initial date of the Neural Network:

1. Import values of x (the values of G_d to calculate tension and the values of B_d to calculate phase, obtained from the previous step) and Y (V or θ);
2. Define the number of neurons, the parameter of damping "μ" and the maximum and minimum limits of α and β of neural network, choosing $\varepsilon_1$and ε_2;
3. Initialize parameters β and α to calculate vectors W_1, W_2, W_3, P_1 e P_2;
4. Define the activation function according item 3.1.

Step 3 – Update β values

1. Find the initial value of activation function in accord of Eq. (31);
2. Obtain the first medium quadratic error;
3. Calculate the Jacobian function algebraically;

4. Get the value of g conform Eq. (23) and then h_{lm} using Eq. (22);
5. Update β and α using Eq. (39) and Eq. (40) to calculate new values of W_1, W_2, W_3, P_1 and P_2.

Step 4 – Correction Procedure

1. Calculate correction procedure using Eq. (27);
2. Using the fundamentals of Berger et al. (2010), if:
   i. ∂ <0.2 → calculate μ=2μ;
   ii. ∂ >0.8 → μ=0.5 μ;
   iii. ∂ <0.0005 → μ=0.0005;
   iv. 0.2< ∂ <0.8 → μ= μ;
3. Calculate new error;
4. If ∂ be on same interval over 3 iterations, a new random value of β and α must be calculated;
5. If $E < 10^{-3}$ || Iterations > 100 → End.

Step 5 – Analysis of results

1. If the new value of $x_{new}(reference\ bus) - x(reference\ bus) \leq 10^{-3}$ and the values of V are inside the imposed limits → Calculus of technical losses and verification of reactive losses, if they are inside the imposed limits in each bus → End of the code. If not succeed → Step 1.

5 RESULTS

In this paper was presented two algorithms that use the Levenberg-Marquardt principle to reduce losses. Being one, using a traditional method found in references, and after, the code used with the methodology described in Section 4. From the developed code, were trained four different types of system (IEEE-6, IEEE-30, IEEE-118 e IEEE-300) bus, varying the activation functions (Bi-hyperbolic Tangent, Gauss, Sigmoid, Elliot e Ackley) and limits forced by ε_1 and ε_2, verifying, if the newly obtained results reduce losses of the system inside the specified technical limits. The algorithms were rolled in MATLAB® via video card of a notebook with an operational system Windows 10, 1 Tb HD, 8 Gb RAM, processor Intel I5 (5th generation) with video card GEFORCE 920M.

Table 1 presents results of losses of the training with the LM traditional method. The values represent the best hypothesis of calculation, and best implementation of each function, with the numbers of neurons and the selection of parameters, which presented the smallest losses calculated. The results exposed on Table 1 show a significant reduction in losses of the four systems using LM. For the systems IEEE-6 and IEEE-30, the Bi-hyperbolic tangent accomplished the biggest reduction, with possible results of just 3.23 MW. On this system, the other functions didn't get a significant reduction, keeping the losses well close just using NR. On the system IEEE-118, the Elliot function succeeded to reduce the losses on 3.53 MW, while the other functions behaved like NR. Lastly, for IEEE-300, the best optimization was calculated from the Gauss function, obtaining a 9.79 MW reduction, the Hyperbolic Tangent, Elliot and Sigmoid functions presented significant losses closer, while the other functions didn't show a significant variation on their losses.

Table 1. Technical Losses using Levenberg-Marquardt neural network

LM Traditional Method	**Newton Raphson (MW)**	**Hyperbolic Tangent (MW)**	**Bi-Hyperbolic Tangent (MW)**	**Elliot (MW)**	**Gauss (MW)**	**Sigmoid (MW)**	**Ackley (MW)**
IEEE-06	11.12	10.63	3.23	10.98	11.11	11.11	11.11
IEEE-30	9.22	8.21	4.28	9.20	9.21	9.20	9.22
IEEE-118	117.36	116.40	117.23	113.83	117.34	117.35	117.35
IEEE-300	6162.99	6158.63	6162.97	6158.48	6153.20	6158.18	6162.99

On this training, was required caution to values of voltage magnitude and phase, were physically possible applicated, further being inside the limit imposed by Eq. (6) and Eq. (7), with modification of some parameters of the system. This means that too small losses, close to zero or training which the values of θ were stabled (in other words, a three-phase without phase legging) were discarded. On execution of the training set, was realized that increases the medium error of the system, created a bigger flexibility on reduction electrical losses, however, also reduced the technical limit, that the system could exercise in each bus. The values of technical losses, found in Table 1, were adjusting parameters and errors to obtain the smallest possible result. It was realized that in LM traditional method cases, modifying the values of weights and lambda, didn't change significantly final results, in term of electrical losses or medium quadratic error.

The Tables 2 and 3 are showing, for the Bi-Hyperbolic Tangent and Ackley, the chosen parameters, to achieve this result (defined with w to neural network weights and λ to coefficient of damping, EV and Eθ the medium quadratic errors calculated from tension and angle, ΔE and Δθ the difference between the values of reference bus). It was possible to observe that tests of IEEE-118 and IEEE-300 bus elaborated with Bi-hyperbolic tangent and Ackley functions saturated, independently the number of iterations and type of parameters selected. This means that the results of the flow study with these activation functions always had close results. Also realizing difficulties in training the system of IEEE-06 and IEEE-30 bus. It was necessary to use a bigger number of interactions to the algorithm, mostly for the solution, the results obtained weren't technically possible.

After the training with LM traditional was applied NLM to verify how the results would behave themselves. From this solution was built Table 4. The results in Table 4 show a significant reduction in losses of the four systems. To System IEEE-06, the hyperbolic tangent got the biggest reduction, coming to 1.67 MW of losses. Using Ackley function is getting a result of 4.87 MW of losses; however, all the other functions didn't get optimize the system. On system IEEE-30 all the functions succeeded significant reduction, being a Bi-hyperbolic tangent, the one which presented the best result, reducing 6.12 MW of losses. On system IEEE-118, as it was to LM method, the Elliot function decrease losses to 115.31 MW. The other functions behaved similarly, presenting close values to these losses. The only exception was using Gauss function, which in all training realized, the system didn't get being optimized. Finally, to IEEE-300 the best optimizations were calculated from the Ackley function, obtaining a bigger reduction, than any other system, coming to 305.61 MW less than using NR. The Elliot function with 178.76 MW of reduction comparing with de NR. Hyperbolic Tangent and Bi-Hyperbolic Tangent only obtain reductions of 12.16 MW and 7.8 MW, respectively, while the other functions don't get loss reduction. The results that are traced means, that it wasn't

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

possible to find results that reduce the losses. The product of this interactions determines bigger losses as with NR only.

Table 2. Difference between the training sets inputs and the training time for each system with Bi-Hyperbolic Tangent function.

	Bi-Hyperbolic Tangent									
IEEE (Bus)	**EV**	**wV**	**E Θ**	**wΘ**	**Training Time**	**ΔV**	**ΔΘ**	**Neuron**	**λ (V)**	**λ (Θ)**
06	$1.7\cdot10^{-4}$	10^{-3}	$3.2\cdot10^{-4}$	10^{-3}	$1.8\cdot10^{-4}$	$3.7\cdot10^{-4}$	$4.5\cdot10^{-4}$	V26|Θ23	0.1	10^{-4}
30	$1.2\cdot10^{-4}$	10^{-3}	$2.3\cdot10^{-4}$	10^{-3}	$2.2\cdot10^{-4}$	$6.0\cdot10^{-4}$	$3.8\cdot10^{-4}$	V11|Θ49	10^{-3}	0.1
118	$2.4\cdot10^{-4}$	10^{-3}	$2.5\cdot10^{-4}$	10^{-3}	$5.3\cdot10^{-1}$	$4.8\cdot10^{-6}$	$4.5\cdot10^{-6}$	V30|Θ21	10^{-3}	0.01
300	$2.7\cdot10^{-6}$	10^{-3}	$1.1\cdot10^{-5}$	10^{-3}	$3.5\cdot10^{-1}$	$2.1\cdot10^{-6}$	$1.1\cdot10^{-5}$	V44|Θ32	0.1	0.1

Table 3. Difference between the training sets inputs and the training time for each system with Ackley function.

	Ackley									
IEEE (Bus)	**E V**	**w V**	**E Θ**	**w Θ**	**Training Time**	**Δ V**	**Δ Θ**	**Neuron**	**λ (V)**	**λ (Θ)**
06	$8.7\cdot10^{-5}$	10^{-3}	$1.2\cdot10^{-6}$	10^{-3}	$3.8\cdot10^{-4}$	$8.7\cdot10^{-5}$	$1.2\cdot10^{-6}$	V45|Θ46	0.1	0.1
30	$5.1\cdot10^{-4}$	10^{-3}	$2.2\cdot10^{-4}$	10^{-3}	$1.7\cdot10^{-1}$	$5.9\cdot10^{-5}$	$6.5\cdot10^{-7}$	V23|Θ28	0.1	10^{-4}
118	$4.7\cdot10^{-06}$	10^{-3}	$6.1\cdot10^{-7}$	10^{-3}	$6.6\cdot10^{-1}$	$4.7\cdot10^{-6}$	$6.1\cdot10^{-7}$	V19|Θ29	0.1	10^{-4}
300	$3.4\cdot10^{-07}$	10^{-3}	$1.6\cdot10^{-7}$	10^{-3}	$9.9\cdot10^{-2}$	$3.4\cdot10^{-7}$	$1.6\cdot10^{-7}$	V06|Θ06	10^{-2}	0.01

For the Hyperbolic Tangent function, using IEEE-6, IEEE-30, IEEE-118 and IEEE-300 bus was obtained, a voltage error from: $1.23\text{x}10^{-4}$, $5.95\text{x}10^{-4}$, $3.62\text{x}10^{-4}$, $4.72\ \text{x}10^{-4}$, respectively; phase error: $1.86\text{x}10^{-4}$, $21.7074\text{x}10^{-2}$, $7.74\text{x}10^{-4}$, $1.021\text{x}10^{-3}$ with, wV and wΘ equal to 0.01, using 39, 44, 05 and 47 neurons for voltage magnitude; and 38, 32, 10 and 39 neurons for voltage phase. The training time for these set was $3.2\text{x}10^{-4}$, $2.8\text{x}10^{-4}$, $11.493\text{x}10^{-2}$ and $42.558\text{x}10^{-2}$ s. The difference between the optimized voltage and the voltage reference bus is $7.01\text{x}10^{-6}$, $5.13\text{x}10^{-4}$, $2.61\text{x}10^{-6}$ and $4.72\text{x}10^{-4}$ with λ (V): 0.1, 0.1, 0.01, 0.1. And considering the difference between the optimized phase and phase of reference bus is $4.67\text{x}10^{-6}$, $4.01\text{x}10^{-5}$, $2.88\text{x}10^{-6}$ and $10.83\text{x}10^{-4}$ with λ (Θ): 0.01, 0.01, 0.1, 0.01. For the Hyperbolic Tangent (New method) function was obtained, respectively, a voltage error from: $27.07\text{x}10^{-4}$, $17.52\text{x}10^{-3}$, $70.23\text{x}10^{-4}$, $29.11\ \text{x}10^{-3}$, respectively; phase error: $17.19\text{x}10^{-3}$, $70.33\text{x}10^{-3}$, $98.61\text{x}10^{-4}$, $37.84\text{x}10^{-3}$ with, wV: 0.001, 0.001, 0.001, 0.001, pV: 0.001, 0.001, 0.1, 0.1, w Θ: 1, 1, 0.001, 0.001, pΘ: 0.01, 0.01, 0.001, 0.1, using 16, 32, 08 and 07 neurons for voltage magnitude; and 24, 48, 07 and 38 neurons for voltage phase. The training time for these set was $86.14\text{x}10^{-3}$, $70.33\text{x}10^{-3}$, $17.22\text{x}10^{-8}$ and $19.97\text{x}10^{-2}$ s. The difference between the optimized voltage and the voltage reference bus is $7.91\text{x}10^{-4}$, $1.04\text{x}10^{-4}$, $7.41\text{x}10^{-4}$ and $5.79\text{x}10^{-4}$ with λ (V): 0.1, 0.1, 0.01, 0.1. And considering the difference between the optimized phase and phase of reference bus is $4.43\text{x}10^{8}$, $3.65\text{x}10^{-4}$, $9.52\text{x}10^{-4}$ and $8.37\text{x}10^{-4}$ with λ (Θ): 0.01, 0.01, 0.1, 0.01.

For the Elliot function, using IEEE-6, IEEE-30, IEEE-118 and IEEE-300 bus was obtained, a voltage error from: $3.44x10^{-4}$, $5.02x10^{-4}$, $3.18x10^{-4}$, $3.60\ x10^{-4}$, respectively; phase error: $3.97x10^{-4}$, $2.16x10^{-4}$, $49.43x10^{-4}$, $4.89x10^{-4}$ with, wV and wΘ equal to 0.001, using 36, 47, 13 and 50 neurons for voltage magnitude; and 44, 39, 06 and 48 neurons for voltage phase. The training time for these set was $1.76x10^{-4}$, $2.14x10^{-4}$, $84.336x10^{-3}$ and $49.9871x10^{-2}$ s. The difference between the optimized voltage and the voltage reference bus is $8.26x10^{-5}$, $5.91x10^{-4}$, $6.08x10^{-4}$ and $1.62x10^{-4}$ with λ (V): 0.1, 0.001, 0.001, 0.01. And considering the difference between the optimized phase and phase of reference bus is $5.78x10^{-5}$, $8.05x10^{-4}$, $2.38x10^{-4}$ and $4.42x10^{-4}$ with λ (Θ): 0.1, 0.001, 0.001, 0.01. For the Elliot (New Method) function, using IEEE-30, IEEE-118 e IEEE-300, was obtained, a voltage error from: $21.58x10^{-4}$, $10.05x10^{-2}$, and $10.36x10^{-2}$, respectively; phase error: $13.66x10^{-3}$, $27.44x10^{-3}$, and $32.84x10^{-3}$, with, wV and wΘ equal to 0.0001, 0.001, 0,00001, pV equal to 1, 0.01 and 0.1, using 06, 32, and 08 neurons for voltage magnitude; and 29, 16, and 43 neurons for voltage phase. The training time for these set was $24.53x10^{-2}$, $23.78x10^{-8}$ and $20.04x10^{-2}$ s. The difference between the optimized voltage and the voltage reference bus is $10.32x10^{-5}$, $3.17x10^{-5}$ and $2.58x10^{-5}$ with λ (V): 0.001, 0.001, 0.01. And considering the difference between the optimized phase and phase of reference bus is $6.65x10^{-4}$, $2.03x10^{-5}$and $1.07x10^{-5}$ with λ (Θ): 0.001, 0.001, 0.01.

For the Gauss function, using IEEE-6, IEEE-30, IEEE-118 and IEEE-300 bus was obtained, a voltage error from: $2.84x10^{-5}$, $5.06x10^{-4}$, $2.01x10^{-5}$, $7.98\ x10^{-4}$, respectively; phase error: $4.14x10^{-6}$, $2.17x10^{-4}$, $5.94x10^{-6}$, $2.63x10^{-5}$ with wV and wΘ equal to 0.001, using 46, 15, 32 and 40 neurons for voltage magnitude; and 43, 49, 45 and 27 neurons for voltage phase. The training time for these set was $10.85x10^{-4}$, $25.07x10^{-4}$, 1,104 and $30.94x10^{-2}$ s. The difference between the optimized voltage and the voltage reference bus is $2.84x10^{-5}$, $5.2x10^{-5}$, $1.5x10^{-5}$ and $7.63x10^{-4}$ with λ (V): 0.01, 0.1, 0.1, 0.01. And considering the difference between the optimized phase and phase of reference bus is $4.14x10^{-6}$, $3.84x10^{-6}$, $2.31x10^{-6}$ and $1.67x10^{-5}$ with λ (Θ): 0.0001, 0.01, 0.01, 0.01. For the Gauss (New method) function, using IEEE-30 and IEEE-300 bus was obtained, respectively, a voltage error from: $23.36x10^{-3}$ and $27.90x10^{-3}$; phase error: $96.34x10^{-2}$ and $15.17x10^{-2}$, with, wV and wΘ equal to 0.01, pV equal to 0.1, using 37 and 20 neurons for voltage magnitude; and 11 and 36 neurons for voltage phase with pθ equal to 1 and 0.1. The training time for these set was $15.08x10^{-2}$ and $19.92x10^{-2}$ s. The difference between the optimized voltage and the voltage reference bus is $95.07x10^{-5}$ and $68.77x10^{-5}$ with λ (V): 0.1, 0.01. And considering the difference between the optimized phase and phase of reference bus is $4.09x10^{-4}$ and $4.33x10^{-4}$ with λ (Θ): 1, 0.01.

For the Sigmoid function, using IEEE-6, IEEE-30, IEEE-118 and IEEE-300 bus was obtained, a voltage error from: $3.87x10^{-5}$, $5.13x10^{-4}$, $3.87x10^{-5}$, $3.97\ x10^{-4}$, respectively; phase error: $2.41x10^{-5}$, $2.17x10^{-4}$, $2.44x10^{-5}$, $1.82x10^{-4}$ with wV and wΘ equal to 0.001, using 34, 37, 41 and 48 neurons for voltage magnitude; and 33, 16, 36 and 39 neurons for voltage phase. The training time for these set was $8.34x10^{-4}$, $1.39x10^{-3}$, 0.93 and 0.45 s. The difference between the optimized voltage and the voltage reference bus is $3.85x10^{-5}$, $5.9x10^{-5}$, $5.57x10^{-6}$ and $3.19x10^{-4}$ with λ (V): 0.01, 0.1, 0.1, 0.01. And considering the difference between the optimized phase and phase of reference bus is $2.5x10^{-5}$, $1.17x10^{-6}$, $2.88x10^{-6}$ and $1.51x10^{-4}$ with λ (Θ): 0.0001, 0.1, 0.1, 0.01. For the Sigmoid (New method) function, using IEEE-30, IEEE-118 and IEEE-300 bus was obtained, respectively, a voltage error from: $3.22x10^{-3}$, $2.11x10^{-1}$ and $9.70x10^{-2}$; phase error: $1.18x10^{-2}$, $2.11x10^{-1}$ and $2.16x10^{-1}$, with wV and wΘ equal to 0.00001, 0.001 and 0.0001, pV equal to 1, 0.01, 0.1, using 15, 43 and 15 neurons for voltage magnitude; and 26, 41 and 38 neurons for voltage phase with pθ equal to 0.1, 0.01 and 0.1. The training

time for these set was $2.46x10^{-1}$, $2.96x10^{-7}$ and $2.00x10^{-1}$ s. The difference between the optimized voltage and the voltage reference bus is $1.00x10^{-4}$, $7.75x10^{-4}$ and $2.07x10^{-6}$ with λ (V): 0.1, 0.1, 0.01. And considering the difference between the optimized phase and phase of reference bus is $2.68x10^{-5}$, $1.41x10^{-4}$ and $1.38x10^{-4}$ with λ (Θ): 0.1, 0.1, 0.01.

Table 4. Technical Losses using Novel Levenberg-Marquardt Neural Network

Novel LM Method	**Newton Raphson (MW)**	**Hyperbolic Tangent (MW)**	**Bi-Hyperbolic Tangent (MW)**	**Elliot (MW)**	**Gauss (MW)**	**Sigmoid (MW)**	**Ackley (MW)**
IEEE-06	11.12	1.67	-	-	-	-	4.87
IEEE-30	9.22	5.34	3.10	7.30	4.78	7.30	4.64
IEEE-118	117.36	115.96	116.06	115.31	-	115.12	115.27
IEEE-300	6162.99	6150.83	6155.19	6163.81	5984.23	6163.79	5857.61

From the training of each function, perceived smaller errors, than which were obtained with the traditional method. The errors obtained, parameters selected and a number of neurons ideal are showing, for Bi-Hyperbolic Tangent and Ackley function, in Tables 5 and 6.

Table 5. Difference between the training sets inputs and the training time for each system with Bi-Hyperbolic Tangent function (New method).

	Bi-Hyperbolic Tangent (New method)									
IEEE (Bus)	**E V**	**wV**	**pV**	**E Θ**	**wΘ**	**pΘ**	**Traning Time**	**Δ V**	**Δ Θ**	**Neuron**
6	-	-	-	-	-	-	-	-	-	-
30	$1.0\cdot10^{-2}$	10^{-4}	1	$1.2\cdot10^{-2}$	10^{-4}	1	$2.5\cdot10^{-1}$	$9.7\cdot10^{-3}$	$9.4\cdot10^{-04}$	V24|Θ05
118	$3.0\cdot10^{-2}$	10^{-3}	10^{-2}	$2.8\cdot10^{-2}$	10^{-3}	10^{-2}	$2.5\cdot10^{-7}$	$8.4\cdot10^{-4}$	$4.9\cdot10^{-04}$	V17|Θ14
300	$3.9\cdot10^{-2}$	10^{-3}	10^{-1}	$6.6\cdot10^{-1}$	10^{-3}	10^{-1}	$2.0\cdot10^{-1}$	$3.9\cdot10^{-3}$	$8.8\cdot10^{-04}$	V08|Θ49

Table 6. Difference between the training sets inputs and the training time for each system with Ackley function.

	Ackley (New method)									
IEEE (Bus)	**E V**	**wV**	**pV**	**E Θ**	**wΘ**	**pΘ**	**Traning Time**	**Δ V**	**Δ Θ**	**Neuron**
6	$1.2\cdot10^{-2}$	10^{-3}	10^{-2}	$1.7\cdot10^{-1}$	10^{-1}	$1.\cdot10^{-2}$	$2.2\cdot10^{-1}$	$5.5\cdot10^{-4}$	$-6.5\cdot10^{-4}$	V43|Θ34
30	$5.2\cdot10^{-3}$	10^{-4}	10^{-2}	$7.4\cdot10^{-2}$	10^{-1}	$1.\cdot10^{-2}$	$2.5\cdot10^{-1}$	$9.4\cdot10^{-4}$	$-1.0\cdot10^{-4}$	V16|Θ11
118	$2.3\cdot10^{-2}$	10^{-3}	10^{-2}	$6.6\cdot10^{-2}$	10^{-3}	$1.\cdot10^{-2}$	$2.9\cdot10^{-7}$	$4.4\cdot10^{-4}$	$7.4\cdot10^{-4}$	V09|Θ20
300	$1.2\cdot10^{-1}$	10^{-5}	10^{-1}	$1.2\cdot10^{-1}$	10^{-5}	$1.\cdot10^{-1}$	$2.\cdot10^{-1}$	$3.4\cdot10^{-4}$	$9.4\cdot10^{-4}$	V28|Θ31

The values of λ (V) are 0.001 for IEEE-30 and 0.1 for the other two systems. For λ (Θ), for all systems were determined 0.1. The values of λ (V) and λ (Θ) are the same for Ackley function being 0.001 for 6 buses, 0.1 for 30 and 118 buses and 0.01 for 300 buses. These results showed

a behavior similar, when compared to the choice of activation function, with LM traditional method with a new method to study of Bi-hyperbolic Tangent and Gauss to systems of IEEE-30 and IEEE-300, respectively, showed the best functions in both cases. Except for the IEEE-118 system, the new method of training got reduce more the power losses, not only on best activation function but also on most others trained functions. Once again, to systems of IEEE-06 and IEEE-30, the training occurred with more number of interactions, with many results, which diverge from the ideal answer. Also, many values found being discarded due to the impossibility of implementation, because of the high errors.

To IEEE-06 system, obtained on (Oliveira and Picanco, 2014), that describes a set of capacitors, transformers, generators and lines of an electrical system. The Fig. 4 and Fig. 5 show the difference between the values obtained using NR and the usual LM method, for angle and tension of each bus. Figure. 6 and Fig. 7 show the same comparison, but between NR and the NLM method. The result of 3.23 MW of loss using LM traditional was gotten with the training of 11 neurons for calculating voltage magnitude and 50 neurons for the voltage phase, using Bi-hyperbolic Tangent activation function. Then for NLM was obtained 1.67 MW losses using 26 neurons to calculate voltage magnitude and 24 neurons to the voltage phase.

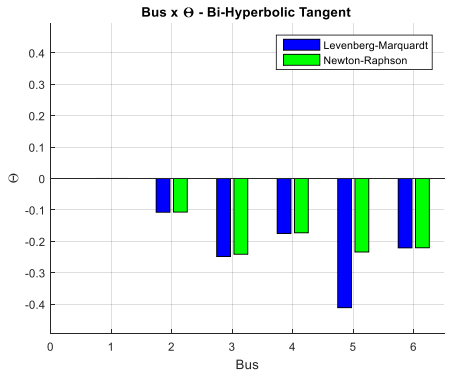

Figure 4. Bus x Θ (Traditional LM)

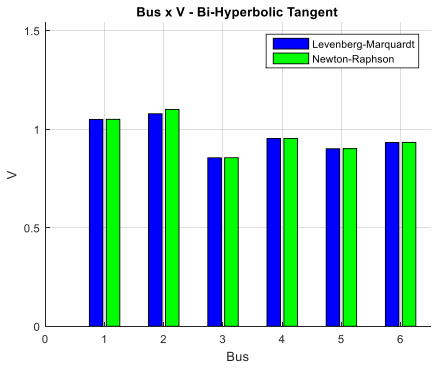

Figure 5. Bus x V (Traditional LM)

To the case of IEEE-06 it is possible to see the levels of voltage magnitude calculated by NR method and the use of LM neural network, are practical equals. The bus 2, which represented the biggest gap, is seen on electrical system as a generation bus. This allows that the bus can control its voltage levels to improve the system. An analysis of the phase values obtained was verified a significant difference between the methods in bus 5 and 6. This can be associated with capacitor banks coupled on the bus, which allows controlling the reactive that is injected into the system, modifying this way, the values of phases obtained.

Using the new method, where was obtained results on voltage magnitude, basically equals, to those calculated using NR method. However, the values of phases suffered massive changes in all the buses to this system. How this is a small system, it can be considered that change on shunt of the capacitors would be able of modifying the next bus of the system. On IEEE-30 Bus system, obtained from Christie (1993), acquired reduce losses to 4.27 MW, using the LM training with Bi-hyperbolic tangent with 11 neurons to calculate tension and 49 neurons to calculate angles. In case of IEEE-30 bus, it is possible to see that, like the IEEE-06, there is no significant variation in magnitude voltage values, which determines the losses reduction. To

optimization of voltage phases, it is possible to realize that in bus 5, a derivation bus, receives synchronous condensers where magnitude tension and angles suffered significant changes on that values. But the biggest difference in bus occurred in bus 12, which is fed by three transformers with an equivalent winding with 132:33 kV and feeds a bus directly with a condenser and a three-phase motor. This bus is directly connected to the synchronous condenser, allowing a control of the phases of this system, to reducing losses.

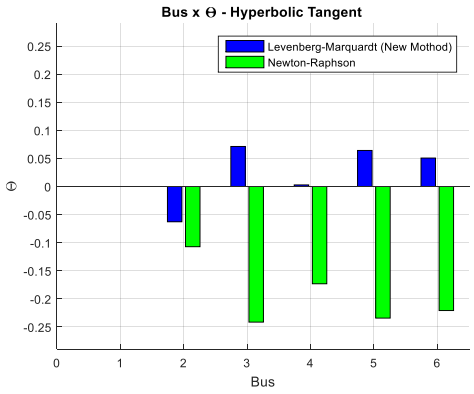

Figure 6. Bus x Θ (New LM method)

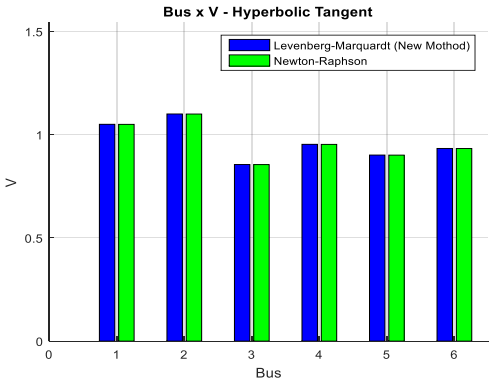

Figure 7. Bus x V (LM new method)

Based on this information, it was verified, how significant is changing the angle of phase in bus 12 and how it would interfere with the results of losses. All the angles and tensions were kept on constant, in exception of bus 12 which was adopted the value calculated on neural network. From this, were obtained losses coming to 4.523 MW. The same analysis was done using NLM, with using Bi-hyperbolic tangent activation function with 24 neurons, to calculate tension and five neurons to phase, acquiring a reduction on 6.12 MW on electrical losses.

With this training, again was obtained values of tension well close to those trained with NR, however, the values of phases changed a lot. Observed the biggest changes were in buses 4, 5, 9, 11, 12 and 13, those buses are connected to synchronous condensers and transformation substations of 132:33 kV, where exist capacitors bank installed as a set. This can explain the reduction of phase values and propagation of this on another bus.

The results obtained from the IEEE-118 system, Fig. 8, obtained from Christie (1993), were training with 13 neurons to calculate tension and six neurons to calculate phase, using the traditional LM code. For the study of this case, it was possible to analyze, one more time, which the voltage values, when some were modified don't change the results. Also, the system becomes to be more stable, so it has a smaller percentage reduction than the systems with a lower number of the bus. It was possible to observe that the biggest difference on angles consists of buses, which are connected to synchronous condensers, generators or charges, with reactive power control due to element shunt. It was noticed too that generation buses, besides the important optimization on values of Θ, the values of V, also suffered a small change in your value. The optimization brought significant changes, also on a set of distribution changes and the arrival of lines. On these points are many inputs/outputs of systems, besides being connected to many different charges, and synchronous condensers. Once again, the optimization points found a deal with electrical equipment, that can be controlled, enabling an easier way the results. Using NLM with Elliot function was required 32 neurons to the optimization of voltage values

and 16 neurons to phase values. The values of voltage and phase found some were optimized. It was gotten to obtain a difference superior to 0.30 pu on the phase in just 17 buses of voltage and phase. To the voltage, the bus with the biggest difference between the NR and the optimization with NLM are 4, 6, 14, 22, 29, 40, 42, 50, 58, 65, 76, 78, 86, 94, 101, 112 and 114. These buses are connected to generators or transformers step-down/step-up of voltage, making them possible to be implemented. To the optimization of phases, the bus are 1, 2, 4, 21, 22, 24, 41, 42, 44, 62, 64, 81, 82, 84, 101, 102, and 104. These buses are all connected to synchronous condensers or in buses close to these condensers. This allows affirming the technical possibility of controlling the phase.At this case, to reduce losses, the adjust, as on voltage magnitude as on phase, must be done together. Otherwise, the losses would increase. With voltage magnitude constant and optimizing just the voltage phase was obtained a loss of 117.60 MW. Making the opposite, the losses were 118.07 MW.

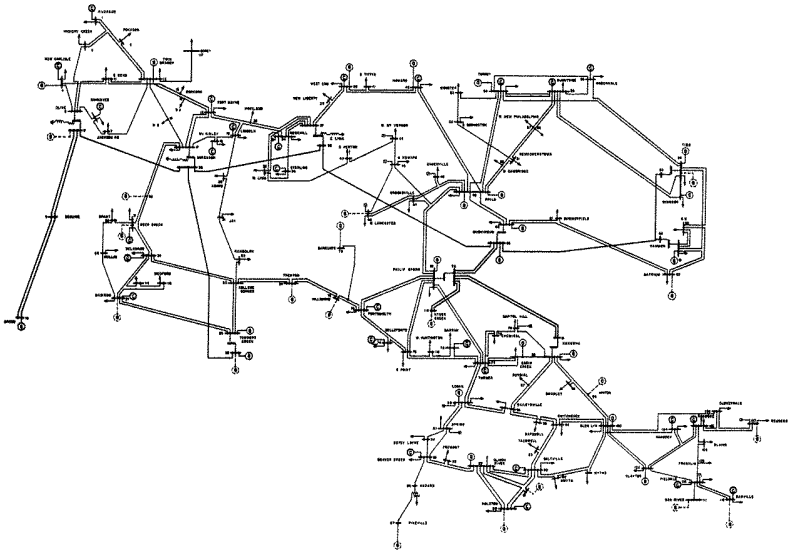

Figure 8. IEEE-118 Power System

By far, to IEEE-300 system, obtained from Christie (1993), the results for LM have received training with 40 neurons to calculate tension and 23 neurons to calculate angles. To the IEEE-300 system, using LM method, it is possible to see a bigger variation in tension than on angles of the system. On the neural network, training was obtained, those optimal neurons to reduce losses were those, that kept their values of phase close to those calculated by the traditional method. The bus, with a gap on the system voltage, where the generations bus, that had a transformer coupled. The bus that also had just one generator present shows a significant difference between the methods. Once again, the neural network was able to identify points, that located transformers, reduction or raising the tension, making it possible to control them to system taps variation and just with this control, it's possible reducing losses.

Using NLM code with Ackley function, achieved the best solution with 28 neurons on calculating tension and 31 to calculate phase. For this case, different from the one calculated previously, the voltage values differ much less than the phase, however, the variation occurs on most bus in similar ways. The bus that suffered the biggest changes is those, who have some generation coupled with them. The bus that suffered the greatest gap on phase is those who are on substation of transformation. The use of NLM with Ackley and Gauss function got excellent results to the entire system, from about 5% and 3% power loss decreases.

6 CONCLUSION

The power flow study allows planning and operating the electrical system in the best way, reducing technical losses of the system and guaranteeing that it can work in any state of the system as most stable as possible. The results obtained demonstrated, that using Neural Network, more specifically the Levenberg-Marquardt traditional algorithm and the new one proposed, become as an alternative to optimization a power system calculated by NR method. It was exposed that to each type of system a kind of activation function should behave as a better alternative, as on the question of reducing energy losses, as on time of convergence of the results.

From the results found, it can be concluded that to LM the parameter Ɵ has an importance much more significant, than any other changes in values of tensions to small systems. This behavior was noticed on tests of IEEE-06, IEEE-30, and IEEE-118, where the values of tension didn't affect as much on the reduction of electrical losses. To IEEE-300 system, the values of tension varied much more significant, than the values of phase, despite this variation is small, when compared to the other systems changes. With these numerical results, it was also verified, that they were associated with some electrical element, which could be controlled by transformers, generators, condensers and capacity banks. This turns the implementation of the values obtained possibly.

The Bi-hyperbolic Tangent function brought excellent results to the study of IEEE-06 and IEEE-30 bus. It's this variable way, with fixed parameters, allowed a quick convergence with small technical losses, besides a real identification of the points to control the flow. To the IEEE-118 system, the best behavior came from the Elliot sigmoidal activation function, that a more complex system, identified the control points and achieved reduce system losses significantly compared to the other functions. Finally, on IEEE-300 system, the Gauss activation function brought results different from the others. This time the tension values behaved more significant than those of phase. Comparing to bigger systems, control values of Ɵ are more difficult.

On optimization, using NLM obtained better results than with the traditional LM, in exception of the IEEE-118 system. This alternative, of varying the internal parameters of the function, confirming there is a change of the weights on neural network, enabled the networks to find new values that could be possible to implement. On systems of IEEE-30, IEEE-118 and IEEE-300, using the same activation functions, brought similar reduction to the system. However, the optimization with NLM permitted functions, that with fixed parameters didn't obtain a great optimization, became to have great results. For this algorithm, it was only possible to optimize IEEE-06 system with Ackley and Hyperbolic Tangent functions. The other functions increased the system losses.

Due to exposed facts, conclude that, with the smallest power flow systems, comes more difficulties to find the greatest answers. Many times the results are starting to diverge, increasing more the medium quadratic errors and also the values found were discarded for not being inside the technical limits, to enable implementation. To IEEE-118 and IEEE-300 this problem was minimized. The answer of the training was faster and easier to be adjusted, with the results always found inside imposed technical limits. Besides, on optimized systems with LM, can be observed, that activation functions, that didn't achieve reduce significantly, results happened to behave with an approach to values obtained by NR algorithm. To NLM, when the

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering
P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

algorithm didn't get reducing losses, the values found became many times higher than those with NR.

REFERENCES

Abdelfatah, N., & Brahim, G. (2011). Novel power flow problem solutions method's based on genetic algorithm optimization for banks capacitor compensation using a fuzzy logic rule bases for critical nodal detections. *Advances in Electrical and Electronic Engineering*, *9*(4), 150.

Afolabi, O. A., Ali, W. H., Cofie, P., Fuller, J., Obiomon, P., & Kolawole, E. S. (2015). Analysis of the Load Flow Problem in Power System Planning Studies. *Energy and Power Engineering*, *7*(10), 509.

Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

Beaufays, F., Abdel-Magid, Y., & Widrow, B. (1994). Application of neural networks to load-frequency control in power systems. *Neural Networks*, *7*(1), 183-194.

Berger, J., Brakhage K.-H., & Ludescher T. (2010). Levenberg-Marquart-Verfahren, Aachen.

Berhold, M. H. (1973). The use of distribution functions to represent utility functions. *Management Science*, *19*(7), 825-829.

Chopade, P., & Bikdash, M. (2011, March). Minimizing cost and power loss by optimal placement of capacitor using ETAP. In *System Theory (SSST), 2011 IEEE 43rd Southeastern Symposium on* (pp. 24-29). IEEE.

Christie, R. (1993). Power Systems Test Case Archive,". [Online]. Available: https://www2.ee.washington.edu/research/pstca/ [Access on 28 November 2016].

Fathabadi, H. (2016). Power distribution network reconfiguration for power loss minimization using novel dynamic fuzzy c-means (dFCM) clustering based ANN approach. *International Journal of Electrical Power & Energy Systems*, *78*, 96-107.

Grainger, J. J. S., Grainger, W. D. J. J., & Stevenson, W. D. (1994). *Power system analysis*.

Haykin, S. S. (2001). *Neural networks: a comprehensive foundation*. Tsinghua University.

Lagacé, P. J., Vuong, M. H., & Kamwa, I. (2008, July). Improving power flow convergence by Newton Raphson with a Levenberg-Marquardt method. In *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE* (pp. 1-6). IEEE.

Lockett, A. J., & Miikkulainen, R. (2011, July). Real-space evolutionary annealing. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation* (pp. 1179-1186). ACM.

Madsen, K., Nielsen, H. B., & Tingleff, O. (2004). Methods for non-linear least squares problems.

Mhaskar, H. N., & Micchelli, C. A. (1992). Approximation by superposition of sigmoidal and radial basis functions. *Advances in Applied mathematics*, *13*(3), 350-373.

Miguez, G. A. (2012). *Otimizaçao do Algoritmo de Backpropagation Pelo Uso da Funçao de Ativaçao Bi-Hiperbólica* (Doctoral dissertation, Universidade Federal do Rio de Janeiro).

Moré, J. J. (1978). The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis* (pp. 105-116). Springer, Berlin, Heidelberg.

Muller, H. H., Rider, M. J., Castro, C. A., & Paucar, V. L. (2005, June). Power flow model based on artificial neural networks. In *Power Tech, 2005 IEEE Russia* (pp. 1-6). IEEE.

Nakawiro, W., & Erlich, I. (2009, March). A combined GA-ANN strategy for solving optimal power flow with voltage security constraint. In *Power and Energy Engineering Conference, 2009. APPEEC 2009. Asia-Pacific* (pp. 1-4). IEEE.

Olamaei, J., Niknam, T., & Arefi, S. B. (2012). Distribution feeder reconfiguration for loss minimization based on modified honey bee mating optimization algorithm. *Energy Procedia*, *14*, 304-311.

Oliveira, A., & Picanco, A. (2014). Análise dos métodos de rede neural artificial com aplicação em controle de reativos de fluxo de potência. In *Instituto Federal De Educação, Ciência E Tecnologia Da Bahia, Campus De Vitória Da Conquista Coordenação Do Curso De Engenharia Elétrica*.

Oliveira, A., & Picanco, A. (2015). Analysis of Artificial Neural Network Methods with Application in Reactive Power Flow Control. In *Proceedings of the XXXVI Iberian Latin-American Congress on Computational Methods in Engineering.* Rio de Janeiro.

Pourbagher, R., & Derakhshandeh, S. Y. (2016). Application of high-order Levenberg–Marquardt method for solving the power flow problem in the ill-conditioned systems. *IET Generation, Transmission & Distribution*, *10*(12), 3017-3022.

Puppala, V., & Chandrarao, P. (2015). Analysis of Continuous Power Flow Method, Model Analysis, Linear Regression and ANN for Voltage Stability Assessment for Different Loading Conditions. *Procedia Computer Science*, *47*, 168-178.

Ramesh, L., Chowdhury, S. P., Chowdhury, S., Natarajan, A. A., & Gaunt, C. T. (2009). Minimization of power loss in distribution networks by different techniques. *International Journal of Electrical Power and Energy Systems Engineering*, *2*(1), 1-6.

Sibi, P., Jones, S. A., & Siddarth, P. (2013). Analysis of different activation functions using back propagation neural networks. *Journal of Theoretical and Applied Information Technology*, *47*(3), 1264-1268.

Suratgar, A. A., Tavakoli, M. B., & Hoseinabadi, A. (2005). Modified Levenberg-Marquardt method for neural networks training. *World Acad Sci Eng Technol*, *6*(1), 46-48.

Wu, J. M. (2008). Multilayer potts perceptrons with Levenberg–Marquardt learning. IEEE Transactions on Neural Networks, 19(12), 2032-2043.

XAVIER, A. E. (2005). Uma função de ativação para Redes Neurais Artificiais mais flexível e poderosa e mais rápida. *Learning and Nonlinear Models–Revista da Sociedade Brasileira de Redes Neurais (SBRN)*, *1*(5), 276-282.

Yu, H., & Wilamowski, B. M. (2011). Levenberg–marquardt training. *Industrial Electronics Handbook*, *5*(12), 1.

Zimmerman, R. D., Murillo-Sánchez, C. E., & Thomas, R. J. (2011). MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on power systems*, *26*(1), 12-19.