Volume 4, Issue 2
April 2015

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Improvement Estimation Power Flow Using Bayesian Neural Network

[1] Mahdi Sabri, [2] Roshanak Rezaeipour

[1] Department of Electrical Engineering, East Azarbaijan Science and Research Branch, Islamic Azad University, Tabriz, Iran.
[2] Department of Electrical Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran.
E-mail: [1]stu.mahdi.sabri@iaut.ac.ir, [2]rezaeipour@iaut.ac.ir

ABSTRACT

The main purpose of this work is to improvement the estimation method for producing Bayesian Neural Network (BNN) with improvement objective functions. The objective functions modeled by weight functions that have been optimized by Genetic Algorithm (GA). Given power flow estimation in network security, power system operation and Forecasts. Furthermore, reduce the power flow forecasting error is always desired. Now this method is the smart to estimate, because the nonlinear functions can be higher than other methods. In this paper, the Bayesian and Perceptron Neural Networks (B&PNNs) based on GA is used to assess the improvement in power flow estimates. Genetic Algorithm (GA) is used to select the appropriate parameters and more input these neural networks at the same time. This method has been tested in IEEE 30, 118 and 300 bus test systems. The mean absolute error of the estimated 0.57 percent and a response time less than 0.01 seconds of its features in each IEEE bus test system. By comparing the results, we can conclude that Bayesian Neural Network (BNN) approach compromise is calculated between accuracy and speed of response. Simulations are performed on IEEE 30, 118 and 300-bus test systems using Matlab software environment.

Keywords: *estimation; power flow; Bayesian Neural Network (BNN); bootstrap; Genetic Algorithm (GA)*

1. INTRODUCTION

Electrical energy on a large scale could not be saved, and distribution of electrical power supply based on, demand for electrical energy, action planning, and the optimal investment exploitation should be adjusted. Thus, power system load forecasting error rate in planning into the future is important and should be reduced as much as possible. Scheduling and operational planning are often performed in large systems based on the point forecasts of the system's future (unseen targets). As the reliability of these point forecasts is low and there is no indication of their accuracy. They do not require the a-priori model to be assumed or a-priori assumptions to be made about the properties of data [2]. They have been widely employed for modeling, prediction, classification, estimation, and control purposes [3-5]. Paliwal et al [6].

The main prediction methods of power flow are regression analysis and time series. Generally, the relationships are complex and nonlinear between load, especially power flow and its influencing factors. This causes the classical methods such as time series and regression analysis can be modeled as well as the dependency. In addition, most of the refined methods, based on mathematical models and parameters that include several steps such as modeling, identification and estimation of model parameters and confirmation of their authenticity. So, in spite of the long history of use and ease of use, accuracy, and do not have much. In the way of prediction error values are generally higher than the needed. It recently tried the above problems by applying intelligent resolved [1]. Neural networks, including smart methods is due to the ability to learn, complex relationship between input and output vector as well as the model.

* Corresponding author. Tel.: +98 9144113847;
E-mail address: r.a.rezaei@gmail.com (R. Rezaeipour)

In this paper, a method trained to predict based on Bayesian neural network, and estimate flow provided excellent results, that have been stated in practice. In contrast to the usual training methods, Bayesian learning in neural networks cannot be used as a weight category, but a function as the weight distribution is considered. For all their output is calculated and used to calculate the final output of the impact of all weights are considered. This reduces is excessive the probability of compliance. In this paper, Bayesian Neural Network (BNN) and bootstrap method are used to improve the assessment of the estimated power flow through the power system based on Genetic Algorithm (GA). The aim is to reduce the problems of modeling and analysis of fast and yet precise estimates of power flow, also compare the performance of different methods of evaluation to improve the power flow estimation in power systems for terms of speed and accuracy.

This paper is structured as follows. In Section II, we are discussed Bayesian Neural Network (BNN) training using the equations of this type of education .Section III describes BNN design and improves power flow estimates. In addition, the GA is used to select the most effective simultaneous inputs and parameters of the BNN. Simulation results are discussed in Section IV for IEEE 30-bus case studies, and the bootstrap method are applied with estimate power flow. In Section V and VI, the bootstrap methods with estimate power flow are applied to the modified IEEE 118-bus, 300-bus test systems, also the results are presented, and simulations performed in these systems. Furthermore, in Section IV, V and VI, Perceptron Neural Network (PNN) compared with Bayesian Neural Network (BNN) in error result. Section VII concludes the paper with a summary of results.

Volume 4, Issue 2
April 2015

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

2. BAYESIAN NEURAL NETWORK TRAINING

The usual method of propagation is used for training Multi-Layer Perceptron (MLP) neural network. It is possible to statistically estimate the Maximum Likelihood (ML) interpreted [6-16].In this method for a set of training data; the values to the weights of the neural network can be found, so that the cost function (purpose) is reached. Then the parameter of the neural network is used to test mode. The neural network is obtained while the Bayesian method's exit test mode on all the possible values in the parameters. In fact, for each category, selected by an output. Bayesian learning process of neural network is presented below. What follows, is trained in batch mode. It is assumed that "n" data is available for training [7-17]:

$(P_1, T_1), (P_2, T_2) \ldots (P_n, T_n)$

The entrance test is also available as "P_{n+1}." The goal is to predict the output "$\hat{y}=y_{n+1}$" (T and P are the mean vector). A neural network is determined by the weight vector "W" in general mapping from the input space. That is $\hat{y}=f(p, w)$ to an output space. Considering the Gaussian noise, the conditional probability distribution function of the output and input vector with respect to the mapping "f" as follows.

For isolation "p" input with probability "p", instead of "p" input, "x" is placed.

$$(y|x, w) = \frac{1}{(2\pi\alpha^2)^{D/2}} \exp\left(-\frac{|y-f(x,w)|^2}{2\alpha^2}\right) \quad (1)$$

Where "D" is a dimension output vector, "α^2" is the output noise variance (it is assumed that all the output is the same amount of variance). The type of operation being performed ML for the usual method of training. This means that the algorithm based on the error gradient is used as a weight vector and "w" is found that the degree of compliance with the maximum data. This is done by maximizing the following likelihood function.

$$(w) = \sum_{i=1}^{n} logP(y_i|x_i, w) = -\sum_{i=1}^{n} \frac{|y_i-f(x_i,w)|^2}{2\alpha^2} + C \quad (2)$$

Where "C" is a constant that does not depend on "W." Maximize the likelihood function, is the same with the minimum square error. After finding the "w," output "y" the entrance test may be obtained as follows:

$$\hat{y} = f(x^{n+1}, \hat{w}) \quad (3)$$

The Bayesian method is not used as a bunch of weight "w". It is considered a distribution function of "w." For all their output is calculated and used to compute the final output; all the weights are considered. So if a bunch of weight, such as "w_i"slightly better than the rest with the batch weight of the data coincides, then the batch weight "w_i" is only slightly higher than those in the output of influence, while the ML just considered to be "w_i". This process reduces the probability and excessive of compliance. If the training data set "D" show the distribution function of the output test mode as follows:

$$(y_{n+1}|x_{n+1}, D) = \int_{R^N} P(y_{n+1}|x_{n+1}, D, W)\, P(W|D, x_{n+1}) dw \quad (4)$$

"w", "xn+1"are independent of each other so:

$$P(W|D, x_{n+1}) = P(W|D)$$

$$(y_{n+1}|x_{n+1}, D) = \int_{R^N} P(y_{n+1}|x_{n+1}, D, W)\, P(W|D, x_{n+1}) dw \quad (5)$$

"N" is the dimensional vector "w" (all weights of the neural network). The distribution function at the output is P ($y_{n+1}|x_{n+1}$, D); the function " y_{n+1}" will be based. If the output is taken as the average of the last function, "equation (6)" is obtained:

$$= E(P(y_{n+1}|x_{n+1}, D)) = \int_{R^N} f(x_{n+1}, w)\, P(W|D) dw \quad (6)$$

According to "equation (1)," p (y| x, w) is Gaussian function with mean f(x, w). For the calculation of "$\hat{y}$", to P (w |D) (posterior distribution function) is calculated.

$$P(W|D) = \frac{P(W)P(D|W)}{P(D)} \quad (7)$$

P (W) is called prior distribution function, used as the advantages of Bayesian learning, because the use of this function for the values to the weights. After presentation of data "D," the prior function updates and the posterior function are acquired. Assuming the independence from the training data, we can write:

$$(D|W) = P(y_1, \ldots y_n|x_1, \ldots x_n, w) = p(y\,|x, w)$$

$$\Rightarrow P(W|D) = \frac{P(W)\prod_{i=1}^{n} P(y_i|x_i,w)}{P(y|x)} \quad (8)$$

Prior to the Gaussian distribution function "W" with zero mean and variance of "ω^2." According to "equation (9)," all terms of the integral "equation (5)" are known. However, it is difficult to solve this integral.

$$P(W) = \frac{1}{(2\pi\omega^2)^{\frac{N}{2}}} \exp\left(-\frac{|W|^2}{2\omega^2}\right) \quad (9)$$

It is assumed that the target is calculated by the following expression:

$$\hat{g} = \int_{R^N} g(q)p(q)dq \quad (10)$$

Metropolis algorithm, produces a sequence of vectors $q_0, q_1, \ldots$. This sequence is a Markov chain with stationary distribution can be obtained p (q). In this case, "ĝ" approximation is:

$$\hat{g} \approx \frac{1}{M}\sum_{t=I}^{I+M-1} g(q_t) \quad (11)$$

"I", the number of sequence elements that are released early. "M" is the number of samples taken and finally, on the "M" values are averaged. Markov chain synthesis is performed according to the energy function.

$$E(q) = -logp(q) - logZ_E \quad (12)$$

Or

$$p(q) = -Z_E^{-1}\exp(-E(q)) \quad (13)$$

"Z_E"is a constant chosen such that E (q) is simple. Algorithm starts by randomly selecting "q_0" and after each step "t," a candidate for the next state "q_{t+1}," p ($q_{t+1}|q_t$) distributed randomly removed. If the new state energy is lower than the previous case, this case to the candidate state, will be accepted, and if the energy is increased, the probability of "exp (-ΔE)" will be accepted.

$$\Delta E = E(\tilde{q}_{t+1}) - E(q_t) \quad (14)$$

$$q_{t+1} = \begin{cases} \tilde{q}_{t+1} & if\ u < \exp(-\Delta E) \\ q_t & otherwise \end{cases} \quad (15)$$

Volume 4, Issue 2
April 2015

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

"u" is a random number uniformly distributed between zero and one. Now, using the Metropolis algorithm, the integral is calculated. For this purpose, it is enough to change "q" and "w" together. P (q) to P (w |D) and the conversion of this distribution, a sequence of "w_0, w_1… w_M" can be built. g (q) and f (x_{n+1}, w_t) are the same. Thus:

$$\hat{y} = \frac{1}{M}\sum_{t=I}^{I+M-1} f(x_{n+1}, w_t) \quad (16)$$

The output of the neural network according to the "M" value of "W" is calculated and finally; a decision was made on it. "W_t" chosen from the following energy function is performed.

$$E(W) = -\log\big(P(W|D)\big) - logZ_E \quad (17)$$

Using "equations (7) and (8)" and select the appropriate value of "Z_E" of "equation (18)" is obtained.

$$E(W) = \frac{|W|^2}{2\omega^2} + \sum_{i=1}^{n} \frac{|y_i - f(x_i, W)|^2}{2\alpha^2} \quad (18)$$

Due to the lack of detailed knowledge about the distribution function of p (w), the function is considered to be wide. However, the noise variance due to the change in "y_i" of f (x_i, w), is considered to be low.

Function of p (w), which is initially wide with the information collected becomes "D." According to "equation (18)," it is observed that the energy function is L (w) and similar to "equation (2)." Here there is only one additional term. This means there is additional property compensation and prevents excessive compliance.

To improve the speed of the previous algorithm, the gradient information about the network propagation is used to find the next candidate.

Therefore, in Hybrid Monte Carlo (HMC) algorithm, Hamiltonian function definition is as follows:

$$H(q,p) = E(q) + \frac{1}{2}|P|^2 \quad (19)$$

Where "P" is the momentum vector.

The interpretation of the concepts of energy, E (q) can be considered potential energy and kinetic energy of $\frac{1}{2}|P|^2$. In this algorithm, the Markov chain in the form of (q_0, p_0), (q_1, p_1)…. The distribution will be made to prove the following:

$$\begin{aligned}(q,p) &= Z_H^{-1}\exp\big(-H(q,p)\big)\\ &= Z_H^{-1}\exp\big(-E(q)\big).(2\pi)^{\frac{-N}{2}}\exp\left(-\frac{|p|^2}{2}\right)\\ &= p(q).p(p)\end{aligned} \quad (20)$$

(q_{t+1}, p_{t+1}) according to the following, equation is obtained from the candidate state:

$$\begin{cases}\frac{dq}{d\tau} = \frac{\partial H}{\partial p} = \mathrm{P}\\ \frac{dp}{d\tau} = \frac{-\partial H}{\partial q} = -\nabla \mathrm{E(q)}\end{cases} \quad (21)$$

$$(q_{t+1}, p_{t+1}) = \begin{cases}(\tilde{q}_{t+1}, \tilde{p}_{t+1}) & if\ u < \exp(-\Delta H)\\ (q_t, p_t) & Otherwise\end{cases} \quad (22)$$

According to the gradient vector, candidates are always accepted. However, due to the numerical solution may be acceptable in some cases. The discrimination of the "equations (21) and (22)" with a sampling time "ε" , the following equation with Leapfrog Method can be obtained:

$$p\left(\tau + \frac{\varepsilon}{2}\right) = p(\tau) - \frac{\varepsilon}{2}\nabla E(q(\tau)) \quad (23)$$

$$q(\tau + \varepsilon) = q(\tau) + \varepsilon.p(\tau + \frac{\varepsilon}{2}) \quad (24)$$

$$p(\tau + \varepsilon) = p\left(\tau + \frac{\varepsilon}{2}\right) - \frac{\varepsilon}{2}\nabla E(q(\tau + \varepsilon)) \quad (25)$$

"Equations (23) to (25)" and "L" are repeated to produce a candidate. Select "L" and "ε" in this range, the effect on the speed of convergence and will not affect the outcome. After finding the "M+I" position of Markov sequences can be the integral "equation (11)" or "equation (16)" into account.

3. BAYESIAN NEURAL NETWORK DESIGN AND IMPROVE POWER FLOW ESTIMATES

In this paper, a modular Multi-Layer Perceptron (MLP) is used with the back propagation learning algorithms. Each module has the characteristics and parameters of the neural network. Each module is a network with an input layer, an intermediate layer and an output layer. The number of modules of the system power is same in the number of structures; each module has the responsibility to learn about a structure. Each module has one neuron in the output layer and the number of neurons in the input layer is equal to the number of entries is selected. Linear input cells, cells of the intermediate layer and the output activation function of sigmoid function are asymmetric with the following activities:

$$f(\theta) = \frac{1}{1+e^{-\theta}} \quad (26)$$

Sum of Squared Errors (SSE) can be considered as a cost function.

$$SSE = \sum_q (d^q - z^q)^2\ ,\ \ q = 1, \dots, Q \quad (27)$$

Where "Q" is the total number of patterns, "z^q" and "d^q" represent the actual output, and the desired output pattern is "q." Input and output data are normalized in the range of zero and one. The following "equation (28)" weights are used to correct.

$$w_{ij}(n+1) = w_{ij}(n) - \eta\frac{\partial E}{\partial w_{ij}} + \alpha(w_{ij}(n) - w_{ij}(n-1)) \quad (28)$$

Rate of learning in which "η", "α" momentum coefficient, "n" number of repetitions and weights are "w_{ij}" amounts. Learn to prevent or slow down when the output cell values are close to zero or one. In "equation (28)," a value of "0.1" is added to$\frac{\partial E}{\partial w_{ij}}$. The action as sigmoid prime offset added to$\acute{f}(\theta)$.

Neural networks and Genetic Algorithm (GA) are used to select the input parameters, the number of neurons in the middle layer (L_2), learning rate (η) and the momentum factor (α). Depending on the type of input parameters of the neural network and improve the accuracy of estimating power flow impact assessment and select the one according to the choice has an impact of the response, the selection is done simultaneously. Therefore, each chromosome has four parts, "η", "α", "L_2" and the input mask. Binary coding is used to produce chromosome and "Fig. 1," shows the layout of the chromosome.

Volume 4, Issue 2
April 2015

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

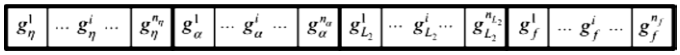


Fig. 1. Chromosome Binary.

In "Fig. 1," $g_\eta^1 \sim g_\eta^{n_\eta}$, $g_\alpha^1 \sim g_\alpha^{n_\alpha}$, $g_{L_2}^1 \sim g_{L_2}^{n_{L_2}}$ respectively, represent the values of the "η", "α" and "L2" parameters. $g_f^1 \sim g_f^{n_f}$ shows the input mask. "n_η", "n_α", "n_{L_2}" and "n_f" respectively, the number of bits, consist of "η", "α", "L2" and input masks. The value of "n_f" is equal to the total number of entries. Based on the precision required for computing the values of "n_η" and "n_α". "n_{L_2}" value will be determined based on engineering experience. In "Fig.2," provider string parameters genotype "η" and "α" to "equation (29)" are transformed phenotype.

$$p = min_p + \frac{max_p - min_p}{2^l - 1} \times d, (p = \eta \; or \; \alpha) \tag{29}$$

Where "p" is a binary string phenotype. Min "p" and max "p", respectively, minimum and maximum values for the parameters, "d" decimal value of the binary string and "l" is the length of the binary string. Phenotype accurately calculates these parameters. The length of this parameter is dependent on the genotype of the provider. It should be noted that the minimum and maximum values of these parameters are determined by the user. In addition, the phenotype of the decimal value to the parameter "L_2" is the binary string. In describing the input mask bits in a value of "1" represents the input selection and bit by bit "0" represents the corresponding input is not selected.

Multi-Layer Perceptron (MLP) neural network cost function and the number of entries selected as the standard Genetic Algorithms (GAs), that have been used for the design as a cost function. So chromosomal error SSE less and less than input selected, will be much less expensive to produce. "Equation (13)"is defined as a function of GA in which the two criteria above, combined with fixed weights and have become a standard.

$$C_f = (w_E \times the\ training\ SSE\ of\ MLP) + (w_F\ No.of\ selected\ features) \tag{30}$$

In "equation (13)," "w_E" is the weight of the standard error of SSE neural network. "WF" weight is related with the number of inputs, and "C_f" is the cost function neural network. The weight values are selected according to user needs.

Initial population size is 1000 and subsequent chromosome to chromosome is 50, and they are considered. If the changes to the cost function are not consecutive 20 generation and otherwise, if the number of generations equal to 100, the algorithm terminates. The combination of the four-point mutation probability is 0.4. This section is intended for simulation. Parent's roulette wheel selection method to be used.

The Bayesian neural network system is applied for mapping between working conditions and the estimated power flow. In addition, the genetic algorithm is used to select the most effective simultaneous inputs and parameters of the Bayesian neural network. At first, some of the most extreme circumstances are responsible for the power system based on the estimated number of power flow selected and responsible learning module's separate data for each of these structures. Flowchart timely evaluation system for the power flow estimate by modules shown in **"**Fig. 2."

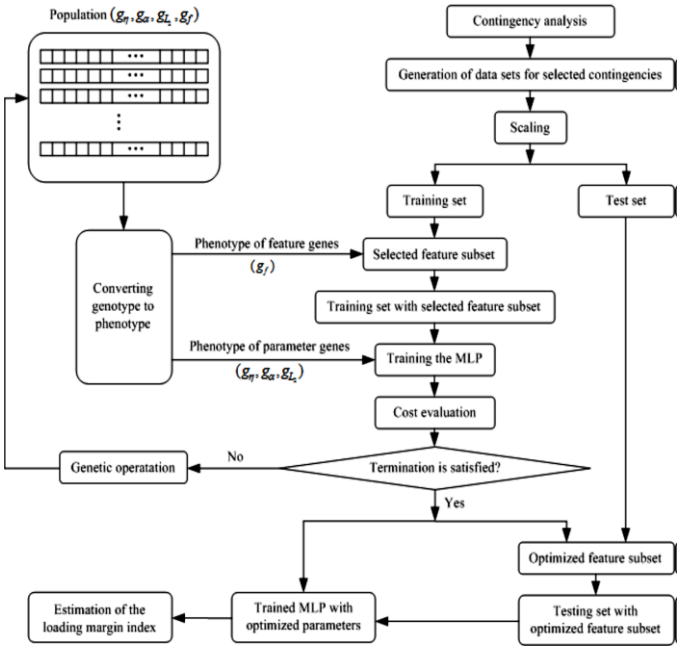


Fig. 2. Flowchart timely assessment estimates power flow through the power system.

Bootstrap method assumes that an ensemble of Neural Network (NN) models will produce a less biased estimate of the true regression of the targets [18]. This method is traditionally called the bootstrap pairs. There exists another bootstrap method, called bootstrap residuals, which resample the prediction residuals. Further information about this method can be found in [19]. As generalization errors of Bayesian Neural Network (BNN) models are made on different subsets the parameter space. The collective decision produced by the ensemble of BNNs is likely to be in error less than the decision made by any of the individual NN models.

The bootstrap methods with estimate power flow are applied to the modified IEEE 30-bus, 118-bus and 300-bus test systems. The results of simulations are presented and performed in these systems. Furthermore, Perceptron Neural Network (PNN) compares with Bayesian Neural Network (BNN) in error result.

4. IEEE 30-BUS TEST SYSTEM

BNN is used to estimation power flow and improves power system operation indices in an IEEE 30-bus test system [20]. Select the appropriate variables as inputs to estimate the weight functions, the ability to predict model is of utmost importance. To assess the validity of the weight functions and the index correlation coefficient (R), root mean square errors (RMSEs) were performed according to "equations (31) and (32)." Best value for these two criteria, respectively, one and zero.

$$R = \frac{\sum_{i=1}^{n} (x_i - \bar{x})\,(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \; \sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{31}$$

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (x_i - y_i)^2} \quad (32)$$

In the above equations, observation and calculation values of "x_i" and "y_i", respectively, "n" is the number of data, "$\bar{x}$" and "$\bar{y}$" are respectively the observed and calculated values. According to Table I, correlation coefficients and root mean square errors between the output functions generated in IEEE 30-bus test system and Underlined numbers in each column represents the best model weight function for each output. Furthermore, Table I, shows the results predicted by the Bayesian Neural Network (BNN) and Perceptron Neural Network (PNN). The estimated parameters of the IEEE 30-bus test system, compared with R and RMSE in PNN and BNN. Note that the variables entered in stepwise to models correctly, predicted increase (decrease RMSE), also active and reactive powers are the injection type. The comparison of the two classifiers/algorithms (PNN & BNN) shows that BNN achieved better accuracy than PNN. This network enables the BNN to attain the state representations by remembering the neural network weights. IEEE 30-bus test system is the best for BNN to estimate the parameters of the bus voltage and reactive power injection.

Table I. R and RMSE compared in PNN & BNN for IEEE 30-bus test system.

Estimated Parameters	Perceptron neural network				Bayesian neural network			
	Train		**Test**		**Train**		**Test**	
	R	RMSE	R	RMSE	R	RMSE	R	RMSE
Bus Voltage	0.68	0.4	0.57	0.7				
P(Active Power)	0.89	0.03	0.78	0.05				
Q(Reactive Power)	0.76	0.01	0.75	0.04				
σ (Load Angle)	0.91	0.02	0.87	0.03				

Bayesian Neural Network (BNN) is trained, using the training data. The test is used to evaluate it. The curves of training state plotted with the actual data's and predicted, by comparing the correlation coefficient (R) and checked for the network carefully. The high accuracy trained to predict the optimal values for the test to show of the neural network. **"**Fig. 3," shows the training state curves for IEEE 30-bus test system with gradient, mu, num and sum squared parameters in three epochs.

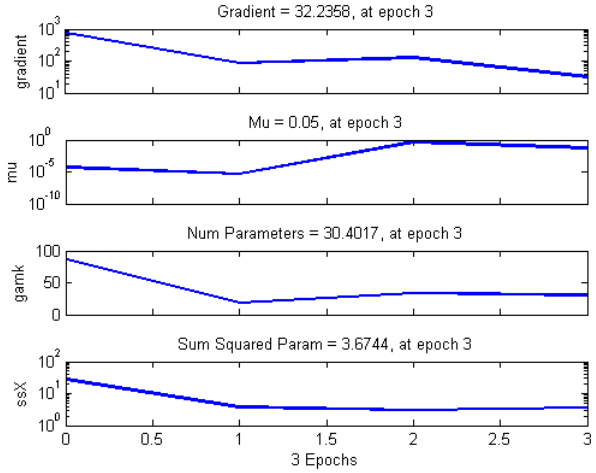


Fig. 3. The training state curves for IEEE 30-bus test system in three epochs, using Bayesian Neural Network (BNN).

The Bayesian neural network was trained with estimated parameter's data, and convergence is achieved in three epochs. In this case, the network was trained for estimated parameters, estimation at the same time. The training, cross validation and testing mean squared training error curves is shown in "Fig. 4." Best validation performance is 0.12055.

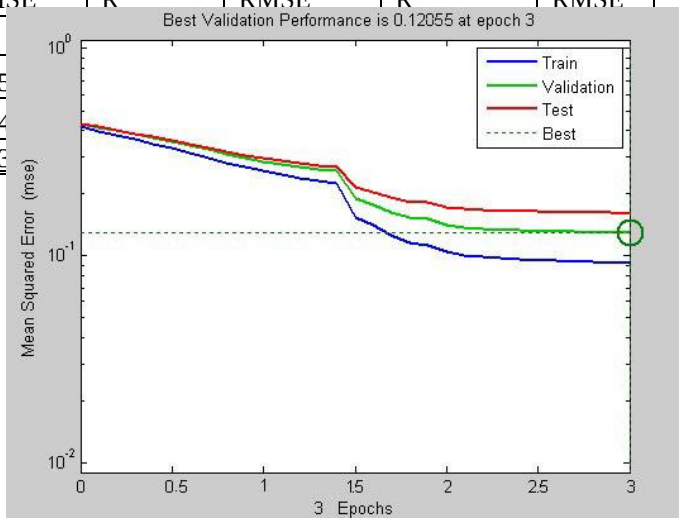


Fig. 4. The training, testing and validation error curves for IEEE 30-bus test system using Bayesian Neural Network (BNN).

5. IEEE 118-BUS TEST SYSTEM

An IEEE 118-bus test system [20] is used to estimation power flow and improves power system operation indices using Bayesian Neural Network (BNN). According to "equations (31) and (32)" best value for R and RMSE to assess the validity of the weight functions was performed.

Table II, shows correlation coefficients (R) and root mean square errors (RMSEs) between the output functions, that generated in IEEE 118-bus test system and Underlined numbers in each column represents the best model weight function for each output. R and RMSE data related to Perceptron Neural Network (PNN) and BNN in IEEE 118-bus test system to IEEE 30-bus test system is more accurate. These neural network weights are trained through estimated parameters in a way to decrease the false positive, false-negative values and increase the overall accuracy.

The estimated parameters of the IEEE 118-bus test system, compared with R and RMSE in PNN and BNN. The variables entered in stepwise to models correctly, predicted increase (decrease RMSE), active and reactive powers are the injection type. IEEE 118-bus test system is the best for Bayesian Neural Network (BNN) to estimate the parameters of the active and reactive power injection. According to BNN based on estimation procedure, IEEE 118-bus test system has

Volume 4, Issue 2
April 2015

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

been training state curves in a better status than the IEEE 30-bus test system.

Table II. R and RMSE compared in PNN & BNN for IEEE 118-bus test system.

Estimated Parameters	Perceptron neural network				Bayesian neural network			
	Train		Test		Train		Test	
	R	RMSE	R	RMSE	R	RMSE	R	RMSE
Bus Voltage	0.59	0.3	0.56	0.6	0.56	0.06	0.54	0.06
P(Active Power)	0.76	0.04	0.72	0.06	0.95	0.01	0.94	0.01
Q(Reactive Power)	0.66	0.02	0.63	0.05	0.48	0.03	0.30	0.03
σ (Load Angle)	0.84	0.03	0.82	0.03	0.55	0.14	0.52	0.20

The overall accuracy of the Bayesian Neural Network (BNN) trained to predict the optimal values for the test. **"**Fig. 5," shows the training state curves for IEEE 118-bus test system with gradient, mu, num and sum squared parameters in 32 epochs.

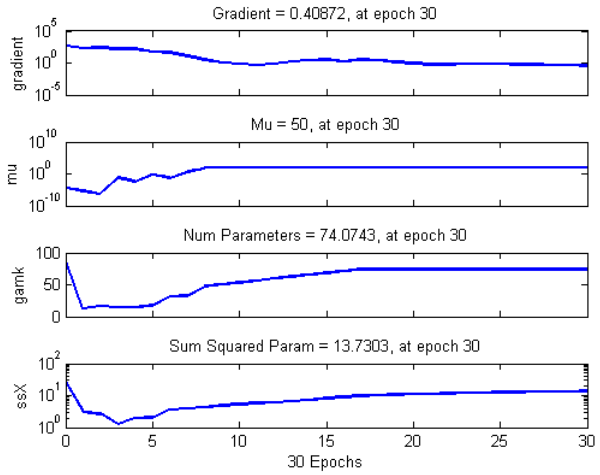


Fig. 5. The training state curves for IEEE 118-bus test system in 32 epochs, using Bayesian Neural Network (BNN).

The BNN was also trained with estimated parameter's data, and convergence is achieved in 32 epochs. In IEEE 118-bus test system, the network was trained for estimated parameters, estimation at the same time. The training, cross validation and testing mean squared training error curves shown in "Fig. 6." Best validation performance is 0.00051055 at 32 epochs using Bayesian Neural Network (BNN).

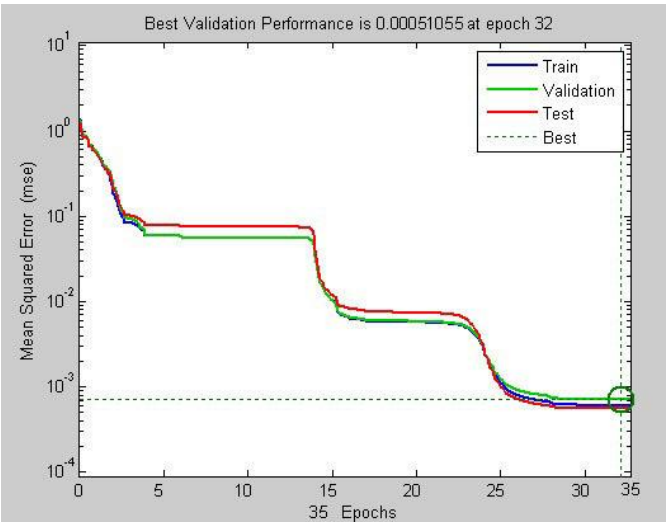


Fig. 6. The training, testing and validation error curves for IEEE 118-bus test system using Bayesian Neural Network (BNN).

6. IEEE 300-BUS TEST SYSTEM

Estimation power flow is used to improve power system operation indices in an IEEE 300-bus test system [20] using Bayesian Neural Network (BNN). The Perceptron Neural Network (PNN) and BNN are used for intelligent processing of estimated parameters**.** The ability to predict model is of utmost importance because, the appropriate variables select as inputs to estimate the weight functions. The result obtained through BNN and PNN are compared with correlation coefficient (R) and root mean square errors (RMSEs).

According to Table III, the output functions generated in IEEE 300-bus test system between correlation coefficients and root mean square errors. It underlined numbers in each column represents the best model weight function for each output. The results predicted by the Bayesian Neural Network (BNN) and PNN. The comparison of the PNN and BNN shows that first neural network (BNN) achieved better accuracy than second neural network (PNN). In this case, network enables the BNN to attain the state representations by remembering the BNN weights. IEEE 300-bus test system is the best for Bayesian Neural Network (BNN) to estimate the parameters of the bus voltage and reactive power injection.

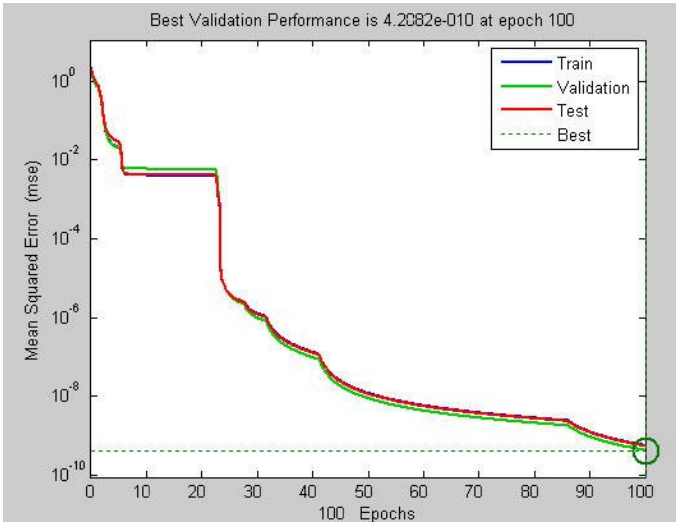
BNN hidden layer consists of a complex of columns is compiled in the input units and units of the class. Per unit of input, which represents a complex combination unit. Between these complex units and units of weight classes, a layer of ordinary Bayesian Neural Network (BNN) is similar to the single-layer network to be trained in teaching methods. This is a complex problem with the column number of units' increases exponentially with their order. Therefore, it is not necessary that the units to which combinations of values that do not occur in the training set and hold, because the categories are not final. This can result in a lower number of units. **"**Fig. 7," shows the training state curves for IEEE 300-bus test system with gradient, mu, num and sum squared parameters in 100 epochs.

Training with estimated parameter's data and convergence is achieved with BNN in 100 epochs. The

network was trained for estimated parameters at the same time in IEEE 300-bus test system. The training, cross validation and testing mean squared training error curves for IEEE 300-bus test system shown in "Fig. 8." Best validation performance is 4.2082e-010 at 100 epochs using Bayesian Neural Network (BNN).

Table III. R and RMSE compared in PNN & BNN for IEEE 300-bus test system.

Estimated Parameters	Perceptron neural network				Bayesian neural network			
	Train		**Test**		**Train**		**Test**	
	R	RMSE	R	RMSE	R	RMSE	R	RMSE
Bus Voltage	0.96	0.06	0.93	0.005	0.97	0.007	0.88	0.004
P(Active Power)	0.98	0.01	0.93	0.008	0.98	0.015	0.95	0.008
Q(Reactive Power)	0.56	0.03	0.52	0.02	0.40	0.02	0.38	0.02
σ (Load Angle)	0.86	0.20	0.71	0.08	0.82	0.19	0.69	0.09



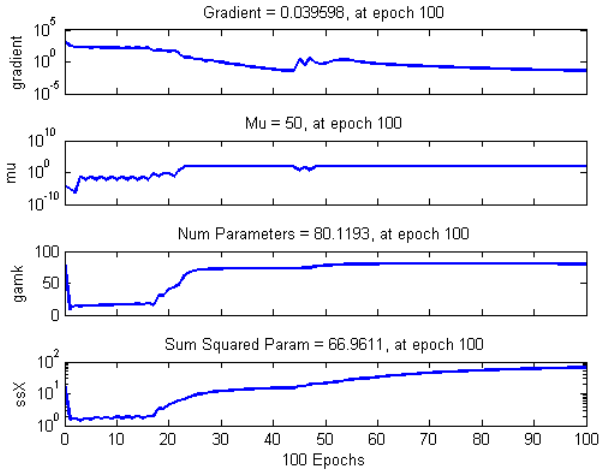


Fig. 7. The training state curves for IEEE 300-bus test system in 100 epochs, using Bayesian Neural Network (BNN).

Fig. 8. The training, testing and validation error curves for IEEE 300-bus test system using Bayesian Neural Network (BNN).

7. Conclusion

Power flow estimation and control through the network are an essential operational requirement. In this paper, a new method based on Bayesian Neural Network (BNN) training in order to increase the accuracy. The predicted distribution of power flow testing that was offered on IEEE 30, 118 and 300-bus test systems.

The results of this study improved and indicate, that this method can predict the extent of power flow used to compare with other methods. RMSE error amounts of training and test data for Perceptron and Bayesian Neural Networks (P&BNNs) decrease with increasing buses of test systems also, R error increases with this state. This decrease and increase are clearly visible on IEEE 300-bus test system. At IEEE 30 and 300-bus test systems, the best estimate for

Bayesian Neural Network (BNN) consists of bus voltage and reactive power injection. The best estimate for the active and reactive power injection is provided on IEEE 118-bus test system that the estimated parameters are both included. Convergence between the training and test data with more input selection in Bayesian and Perceptron Neural Networks (B&PNNs) appropriate parameters is used from Genetic Algorithm (GA). BNN advantages of speed, accuracy and generalization of certified. The results of the calculations for the best performance of the neural network to estimate the power flows in IEEE test systems.

REFERENCES

[1] R.E. Abdel-Aal, "Modeling and Forecasting Electric Daily Peak Loads using Abductive Networks", International Journal of Electrical Power and Energy Systems, vol. 28, pp. 133–141, Feb. 2006.

[2] C. M. Bishop, Neural Networks for Pattern Recognition. Oxford University, Press, Oxford, 1995.

[3] M. Azlan Hussain, "Review of the applications of neural networks in chemical process control – simulation and online implementation," Artificial Intelligence in Engineering, vol. 13, pp. 55–68, 1999.

[4] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," International Journal of Forecasting, vol. 22, no. 3, pp. 443–473, 2006.

[5] B. K. Bose, "Neural network applications in power electronics and motor drives—an introduction and perspective," IEEE Transactions on Industrial Electronics, vol. 54, pp. 14–33, 2007.

[6] M. Paliwal and U. A. Kumar, "Neural networks and statistical techniques: A review of applications," Expert Systems with Applications, vol. 36, no. 1, pp. 2–17, Jan. 2009.

[7] Neal, R.M, "Probabilistic Inference Using Markov Chain Monte Carlo Methods", Technical Report CRG-TR-93-1, Toronto, 1993.

[8] Neal, R.M, "Bayesian Learning via Stochastic Dynamics", Advances in Neural Information Processing Systems 5, pp 475-482, 1993.

[9] Neal, R.M, "Bayesian Training of Back Propagation Networks by the Hybrid Monte Carlo Method", Technical Report CRG-TR-92-1, Toronto, April 1992.

[10] Lampinen, J. and Vehtari, A.,"Bayesian Approach for Neural Networks-Review and case studies", neural networks, pp 7-32, 2001.

[11] Mackay D. J,"Bayesian Methods for Adaptive Models", Ph.D thesis, California Institute of Technology, 1991.

[12] Mackay D. J,"Bayesian Interpolation", Neural Computation, Vol 4, pp 415-447, 1992.

[13] Mackay D. J,"A Practical Bayesian Framework for Back Propagation Networks", Neural Computation, Vol. 4, pp 448-472, 1992.

[14] Neapolitan R. E, "Learning Bayesian Networks", Prentice Hall, 2004.

[15] Neal, R. M, "Prior for Infinite Networks", Technical Report CRG-TR-94-1, Toronto, 1994.

[16] Haggstrom O, "Finite Markov Chains and Algoritmic Applications", Cambridge University Press, 2003.

[17] Müller, P. and Rios Insua, D.,"Issues in Bayesian Analysis of Neural Network Models", Neural Computation, pp 571-592, 1998.

[18] T. Heskes, "Practical confidence and prediction intervals," in Neural Information Processing Systems, T. P. M. Mozer, M. Jordan, Ed., vol. 9, MIT Press, pp. 176–182, 1997.

[19] R. Tibshirani, "A comparison of some error estimates for neural network models," Neural Computation, vol. 8, pp. 152–163, 1996.

[20] Power systems test case archive. http://www.ee.washington.edu.research.pstca[accessed:21.01.08].

AUTHOR PROFILES

Mahdi Sabri was born in Tabriz, Iran, in 1990. He received his B.Sc. degree in electrical engineering from the Islamic Azad University, Tabriz branch, Tabriz, Iran in 2013.He is currently a M.Sc. student in department of electrical engineering in East Azarbaijan Science and Research Branch, Islamic Azad University, Tabriz, Iran.

He is a member of Young Researchers and Elite club in Islamic Azad University, Tabriz Branch. His research interests are power system analysis, neural networks, application of fuzzy control in power system and FACTS devices.

Roshanak Rezaeipour was born in Tabriz, Iran, in 1979. She received her B.Sc. degree in electrical engineering from the Islamic Azad University, Tabriz branch, Tabriz Iran in 2001. She received her M.Sc. and Ph.D. in electrical engineering from the University of Science and Technology, Tehran, Iran in 2008 and 2011, respectively.

She is an Associated Professor in department of electrical engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran. Her research interests are power system analysis, application of fuzzy control in power system and FACTS devices.

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X