



# Journal of Artificial Intelligence

ISSN 1994-5450

**science**  
alert

**ANSI***net*  
an open access publisher  
<http://ansinet.com>

## Effect of Weight Updates Functions in QSAR/QSPR Modeling using Artificial Neural Network

Mohammad Asadollahi-Baboli

Department of Science, Babol University of Technology, P.O. Box 71167-47148, Babol, Mazandaran, Iran

### ABSTRACT

There are many parameters in artificial neural network which must be optimized for developing an acceptable model. One of these parameters is Weight Update Functions (WUFs) which imply how the weights are re-calculated in artificial neural networks. To show the importance of weight update functions in modeling, four different common WUFs were selected: (1) Basic Back Propagation (BBP), (2) Conjugate Gradient (CG), (3) Quasi-Newton (Q-N) and (4) Levenberg-Marquart (L-M) algorithms. Then the effects of these WUFs were studied on four different data sets (two QSPR and two QSAR sets). It is shown that selecting the most favorable weight update function is a key role in ANNs models developing and the performance of developed ANNs model is directly related to the selected weight update function. Moreover it is shown in this article that the levenberg-Marquart algorithm is the best WUF for developing ANNs models. The accuracy and prediction ability of ANNs models was illustrated using leave-one-out and leave-multiple-out cross validation techniques. High  $R^2$  and low Root Mean Square Error (RMSE) values for leave-one-out ( $Q^2_{LOO} > 0.75$  in all four sets) and leave-multiple-out ( $R^2_{L25\%O} > 0.70$  in all four sets) procedures revealed that the levenberg-Marquart algorithm is a good algorithm for developing robust ANNs models.

**Key words:** Quantitative structure activity/property relationship, levenberg-marquart algorithm, partition coefficient, intrinsic viscosity, inhibition

### INTRODUCTION

Artificial neural networks are widely used in chemometrics studies as a non-linear method and many researchers used ANNs in many fields in last two decades. Many parameters are important in ANNs affecting the prediction ability, reliability and consistency of developed models. One of these parameters is weight update functions which indicate the way that weights are changed during the learning process. Weight update functions are specification of the networks which have a key role in developing ANNs models (Khanale and Chitnis, 2011). Many different weight update functions such as Basic Back Propagation (BBP), Conjugate Gradient (CG), Quasi-Newton (Q-N) and Levenberg-Marquart (L-M) algorithms can be used in ANNs models. To show the importance of weight update functions, four different data sets have been chosen. Two data sets are QSPR: (1) modeling of partition coefficient (Log P) and (2) modeling of intrinsic viscosity ( $\eta$ ). The other two data sets are QSAR: (3) Inhibition of tyrosine kinase and (4) Inhibition of cyclooxygenase (COX) enzyme. In each of this data set, the effect of weight update functions has been surveyed in this study.

A partition coefficient (log P) is a measure of differential solubility of a compound in two solvents. Partition coefficient is very important molecular properties that influence the release, transport and the extent of absorption of drugs in the body. In toxicology, partitioning is critical to

understand the tendency of chemical to cross biological membranes. Thus, the ability to predict a chemical behavior in biological or environmental systems is largely dependent upon the knowledge of its partition coefficient. For these reasons, there have been a number of attempts to use various algorithms to predict log P (Caron and Ermondi, 2005). While log P can be determined experimentally using reversed-phase high-pressure liquid chromatography, this requires the prior synthesis of the compounds. Schnackenberg and Beger (2005) attempted to predict log P values using the differences in NMR chemical shifts for some compounds in water and in methanol. However, their QSPR study was restricted to the partial least squares model and they just consider linear dependency in data set. The data set 1 studied in this work consists of 162 molecules for a diverse set of compounds which is extracted from a recent work reported by Schnackenberg and Beger (2005).

The design of new materials with the optimal thermo physical, mechanical and optical properties is a challenge for computational chemistry. One of these properties is intrinsic viscosity ( $\eta$ ) which is important factor in material design with desired properties. There are some empirical equations for predicting intrinsic viscosity (Kacem *et al.*, 2008). However, these equations are not suitable for predicting the diverse sets of molecules. The limitations of empirical equations can be avoided with the use of QSPR approach. Afantitis *et al.* (2006) developed a linear QSPR model for predicting of intrinsic viscosity of some polymers and their QSPR study was restricted to the MLR model. The data set 2 studied in this work consists of 65 polymer-solvent combinations.

Quantitative-Structure Activity Relationship (QSAR) has been demonstrated as a capable tool for the investigation of bioactivity of various classes of compounds. In addition, QSAR studies are usually essential in drug design. Using such an approach one could predict the activities of newly designed compounds before a decision is being made whether these compounds should be really synthesized and tested (Sharma and Sharma, 2011). Two data sets of drug-like compounds were selected in this study. Inhibitors of tyrosine kinase (data set 3) and Cyclooxygenase (COX) enzymes (data set 4) reported by Lu *et al.* (2004) and Shi *et al.* (2007). They used MLR-Antcolony approach for developing linear model and variable selection. The data set 3 consists of 61 analogues of 4-(3-bromoanilino)-6,7-dimethoxyquinazoline (Lu *et al.*, 2004) and the data set 4 consists of 42 diarylimidazole derivatives (Shi *et al.*, 2007).

In this study, we have used four different weight update functions to develop non-linear models between chemical/biochemical properties and reported descriptors. For choosing the best WUF, different models were developed using each weight update function. However, the main aim of this work was making comparison between different WUFs in ANNs modeling technique for predicting and describing chemical/biochemical properties of considered compounds. The results of this study are promising and show that Levenburg-Marquart algorithm has a better performance among different WUFs. The predictability and reliability of the developed models were investigated using leave-one-out and leave-multiple-out cross validation techniques and also Y-randomization technique.

## MATERIALS AND METHODS

**Artificial neural network:** The theory of the artificial neural networks has been adequately described elsewhere (Dastorani *et al.*, 2010). Artificial Neural Networks (ANNs) consider not only linear correlations but also non-linear correlations. As chemical/biochemical properties may have non-linear characteristics, ANNs techniques can discover the possible relationship between the input descriptors and output chemical/biochemical properties. One important step for developing

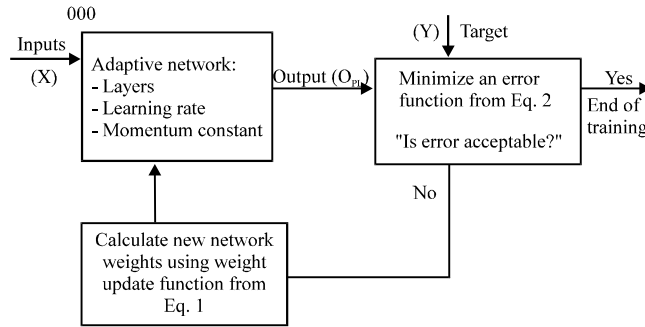


Fig. 1: Weight update function in ANNs

ANN model is training of network. The goal of training the networks is to change the weights between the layers in a direction to minimize the output errors. This procedure is shown in Fig. 1. The changes in the values of the weights can be obtained using Eq. 1:

$$\Delta W_{ij,n} = F_n + \alpha \Delta W_{ij,n-1} \tag{1}$$

where,  $\Delta W_{ij}$  is the changes in the weight factor in each network node;  $\alpha$  is the momentum factor; and  $F$  is a weight update function which indicates how weights are changed during the learning process. Various types of algorithms have been found to be effective for most practical purposes. However, four different weight update functions of Basic Back Propagation (BBP), Conjugate Gradient (CG), quasi-newton (Q-N) and Levenberg-Marquardt (L-M) algorithms are the common ones which are described below.

**Basic back propagation algorithm:** In this type of training algorithm, the weights are changed in the direction of the negative gradient to minimize an objective function (error function) is called gradient decent algorithm. The most popular error function is the sum-of-squares that is given by Eq. 2:

$$E = \frac{1}{2} \sum_{p=1}^p \sum_j (t_{pj} - o_{pj})^2 \tag{2}$$

where,  $t_{pj}$  is the target value (desired output) and  $o_{pj}$  is the actual output. In basic back propagation training algorithm the function  $F_n$  in Eq. 1 is defined by Eq. 3 and 4:

$$g_n = \eta \frac{\partial E_n}{\partial w_{ij,n}} \tag{3}$$

$$F_n = -g_n \tag{4}$$

where,  $w_{ij}$  are the weights connected between neurons  $i$  and  $j$  and  $\eta$  is learning rate.

**Conjugate gradient algorithms:** These techniques not only focus on the local gradient of the error function but also make use of its second derivative. The first derivative measures the slope

of the error surface at a point while second one measures the curvature of the error surface at the same point. This information is very important for determining the optimal update direction. As these methods make use of the second derivatives of the function to be optimized, they are typically referred to as second-order methods. In particular, the conjugate gradient method is commonly used in training BP networks due to its speed and simplicity. Conjugate gradient algorithms can be defined by using Eq. 5 and 6:

$$F_0 = g_0 \tag{5}$$

$$F_n = g_n + \beta_n \times F_{n-1} \tag{6}$$

Various versions of conjugate gradients can be distinguished by the manner in which the constant  $\beta_n$  is computed. In the case of Polak and Ribiere update,  $\beta_n$  can be estimated using Eq. 7:

$$\beta_n = \frac{g_{n-1}^T \times g_n}{g_{n-1}^T \times g_{n-1}} \tag{7}$$

This is the inner product of the previous change in the gradient with the current gradient divided by the norm squared of the previous gradient.

**Quasi-Newton algorithm:** Newton technique is defined as Eq. 8, where,  $A_n$  is the Hessian matrix of the performance index at the current values of the weights and biases:

$$F_n = g_{n-1} + A_n^{-1} \times F_{n-1} \tag{8}$$

Because computation of Hessian matrix is complex, often approximate Hessian matrix is computed in each iteration. Quasi-Newton algorithms usually converge faster than conjugate gradient methods.

**Levenberg-Marquardt algorithm:** This algorithm is somehow similar to Quasi-Newton for second derivative optimization without having to compute the Hessian matrix. In this algorithm, the update function,  $F_n$ , can be calculated using Eq. 9 and 10:

$$F_0 = -g_0 \tag{9}$$

$$F^n = -[J^T \times J + \mu I]^{-1} \times J^T \times e \tag{10}$$

where,  $J$  is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and  $e$  is a vector of network errors. The parameter  $\mu$  should be multiplied by some factor ( $\lambda$ ) whenever a step would result in an increased  $E$ . On the other hand, when a step reduces  $E$ ,  $\mu$  is divided by  $\lambda$ .

### Data sets and descriptors

**Data set 1:** This data set consisted of 162 diverse compounds was taken from Schnackenberg and Beger (2005). The partition coefficients, expressed as Log P (water/Methanol) of these compounds,

Table 1: Selected descriptor for developing ANN models in each data set

Data set	Descriptors using in ANN models <sup>1</sup>
1	$\Delta C$ , $V_m$ , NHD and NHA
2	HOMO <sub>S</sub> , LUMO <sub>S</sub> , TIndx <sub>S</sub> , PMIX <sub>S</sub> , DPLL <sub>P</sub> , MS <sub>P</sub> , LUMO <sub>P</sub> and MW <sub>P</sub>
3	Shadow-XYfrac, shadow- YZfrac, LUMO, PMI-X, $\delta_Y^-$ , B1 <sub>Y,7</sub> and B1 <sub>X,3</sub>
4	Jurs-FPSA-3, Jurs-RASA and S-sNH <sub>2</sub>

<sup>1</sup>Definitions of descriptors are stated in text

was used as dependent parameter (Y) in developing ANNs models. The numerical descriptors are responsible for encoding important chemical features of the molecular structures. These descriptors were used as independent parameters (X) in developing ANNs models. Four meaningful descriptors were calculated and selected in Schnackenberg and Beger (2005) work (Table 1). These descriptors are: (a) the differences in chemical shift of <sup>13</sup>C NMR ( $\Delta C$ ) between two solvents, (b) The molar volume ( $V_m$ ), (c) Number of hydrogen bond donor (NHD) and (d) Number of hydrogen bond acceptors (NHA).

**Data set 2:** This data set consisted of 65 polymer-solvent combinations was taken from Afantitis *et al.* (2006). Intrinsic viscosity ( $\eta$ ) in polymer Solutions was used as dependent variable and eight descriptors selected in Afantitis work were considered as independent variables. Four of these descriptors are solutions descriptors and the other four descriptors are polymer descriptors (Table 1). These descriptors are: (a) HOMO energy in solution (HOMO<sub>S</sub>), (b) LUMO energy in solution (LUMO<sub>S</sub>), (c) Molecular topological index in solution (TIndx<sub>S</sub>), (d) Principal moment of inertia X in solution (PMIX<sub>S</sub>), (e) Dipole length of polymer (DPLL<sub>P</sub>), (f) Connolly molecular area of polymer (MS<sub>P</sub>), (g) LUMO energy in polymer (LUMO<sub>P</sub>) and (h) Molecular weight of polymer (MW<sub>P</sub>)

**Data set 3:** This data set of 61 drug-like compounds of 4-(3-bromoanilino)-6,7-dimethoxyquinazoline derivatives (Fig. 2a) on epidermal growth factor receptor tyrosine kinase (Lu *et al.*, 2004). Inhibitory of these derivatives was modeled using Ant Colony-MLR technique and seven descriptors describing inhibitory (pIC<sub>50</sub>) were selected. The activity parameter IC<sub>50</sub> is a measure of antiviral potency and refers to the molar concentration of each compound required to reduce the concentration of tyrosine kinase viral by 50% respect to the level measured in an infected culture. pIC<sub>50</sub> was used as dependent parameter in developing of ANNs model. The selected descriptors are (Table 1): Surface area projection (shadow-XYfrac and shadow- YZfrac), LUMO energy (LUMO), Principal moment of inertia (PMI-X), the Y substituent effect on aromatic system ( $\Delta_Y^-$ ) and Velooop's sterimol parameter (B1<sub>Y,7</sub> and B1<sub>X,3</sub>).

**Data set 4:** This data set of 42 diarylimidazole derivatives on cylcooxygenase (COX) enzyme (Fig. 2b) (Shi *et al.*, 2007) and three descriptors were selected using Ant Colony-MLR technique. These descriptors were selected as inputs for ANNs and pIC<sub>50</sub> was used as output for ANNs. The selected descriptors are (Table 1): Jurs charged partial surface area descriptors (Jurs-FPSA-3, Jurs-RASA) and S-sNH<sub>2</sub>. All of descriptors in four data sets were summarized in Table 1.

**ANN generation with four different weight update functions:** The networks for each data set were generated using the descriptors as networks' inputs and chemical/bio-chemical properties as networks' outputs. Each data set was divided into three groups: training set (about 50% of data),

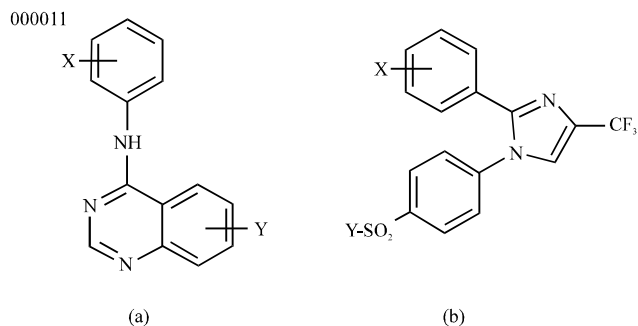


Fig. 2(a-b): Structure of (a) 4-(X-bromoanilino)-Y-quinazolines and (b) Diarylimidazole derivatives

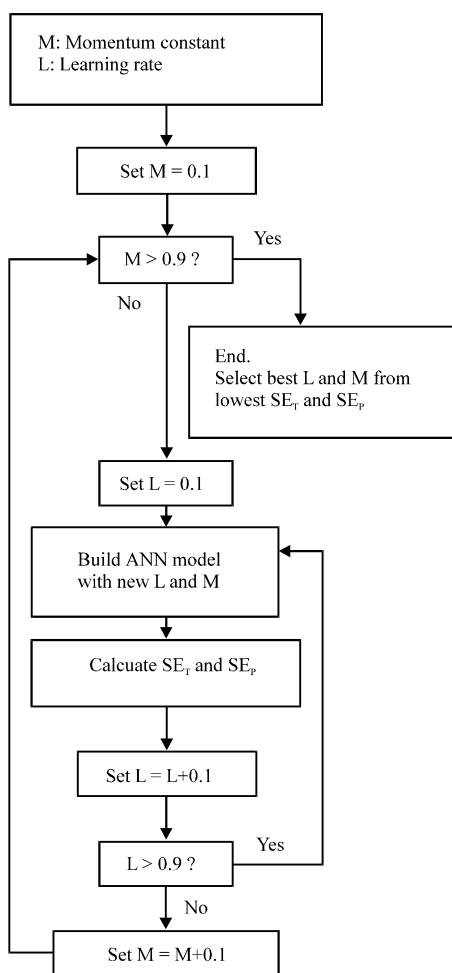


Fig. 3: Optimization procedure for learning rate and momentum constants to select the minimum RMSE

test set (about 25% of data) and prediction set (about 25% of data). The training set was used for the model generation; however, the test set was used to take care of the over-training and over-fitting. The prediction set was used to evaluate the generated model. A layered network

structure with a sigmoid transfer function was designed for ANNs models. Before training the networks, the inputs and output values were normalized between -1 and 1. The initial weights were selected randomly between -0.5 and 0.5. The initial values of biases were also set to be 1. The ANNs program was coded in MATLAB 7 for windows. The network was then trained using four different WUFs strategies for optimization of the weights and bias values. All networks parameters (number of hidden layer, momentum constant and learning rate) were optimized based on obtaining the minimum standard error of training ( $SE_T$ ) and standard error of prediction ( $SE_p$ ). In addition, the strategy of optimization of learning rate and momentum constant has been shown in Fig. 3. The proper number of nodes in the hidden layer was determined by training the network with different number of nodes in the hidden layer. The Root-Mean-Square Error (RMSE) value measures how good the outputs are in comparison with the target values. It should be noted that for evaluating the over-fitting, the training of the network must stop when the RMSE of the test set begins to increase while RMSE of calibration set continues to decrease. Therefore, training of the network was stopped when overtraining began.

All of above processes were repeated using basic back propagation, conjugate gradient, quasi-newton and levenberg-marquardt weight update functions for each data set. To select the best weight update function, three statistical methods were considered for evaluating the developed models: leave-one-out cross-validation ( $Q^2_{LOO}$ ), leave-multiple-out cross-validation ( $R^2_{LMO}$ ) and prediction standard error of estimation ( $SE_p$ ). The statistics together with the specifications of ANNs are given in Table 2-5 for the four different data sets.

**Cross-validation techniques:** Cross-validation techniques are often used to evaluate the consistency and reliability of developed models. Two main strategies of Leave-One-Out (LOO) and Leave-Multiple-Out (LMO) can be carried out in this method. The theories behind LOO and LMO

Table 2: Statistical and optimized parameters for four different weight update functions in data set 1 and comparison with PLS model

Update weight function	Optimized learning rate	Optimized momentum constant	LOO		L25%O <sup>b</sup>	
			Q <sup>2</sup>	SE <sub>p</sub>	R <sup>2</sup>	SE <sub>p</sub>
Basic back propagation	0.3	0.9	0.830	0.673	0.802	0.681
Conjugate gradient	0.2	0.8	0.855	0.644	0.810	0.646
Quasi-Newton	-	-	0.901	0.521	0.895	0.570
Levenberg-Marquardt <sup>a</sup>	-	-	0.958	0.432	0.956	0.503
PLS	-	-	0.870	0.602	0.860	0.698

<sup>a</sup>Initial values:  $\mu = 0.01$  and  $\lambda = 10$ . calculation of  $R^2_{LMO}$  was based on 1000 random selections of 25% of samples

Table 3: Statistical and optimized parameters for four different weight update functions in data set 2 and comparison with stepwise-MLR model

Update weight function	Optimized learning rate	Optimized momentum constant	LOO		L25%O <sup>b</sup>	
			Q <sup>2</sup>	SE <sub>p</sub>	R <sup>2</sup>	SE <sub>p</sub>
Basic back propagation	0.4	0.9	0.701	44.01	0.622	46.73
Conjugate gradient	0.3	0.8	0.722	43.31	0.707	45.97
Quasi-Newton	-	-	0.806	41.95	0.753	44.56
Levenberg-Marquardt <sup>a</sup>	-	-	0.895	36.68	0.846	38.22
Stepwise-MLR	-	-	0.680	45.21	0.612	47.40

<sup>a</sup>Initial values:  $\mu = 0.01$  and  $\lambda = 10$ . calculation of  $R^2_{LMO}$  was based on 1000 random selections of 25% of samples



Table 4: Statistical and optimized parameters for four different weight update functions in data set 3 and comparison with Ant colony-MLR model

Update weight function	Optimized learning rate	Optimized momentum constant	LOO		L25%O <sup>b</sup>	
			Q <sup>2</sup>	SE <sub>p</sub>	R <sup>2</sup>	SE <sub>p</sub>
Basic back propagation	0.2	0.7	0.684	0.827	0.631	0.859
Conjugate gradient	0.4	0.8	0.702	0.810	0.655	0.846
Quasi-Newton	-	-	0.718	0.743	0.674	0.804
Levenberg-Marquardt <sup>a</sup>	-	-	0.789	0.612	0.761	0.672
Antcolony-MLR	-	-	0.625	0.878	-	-

<sup>a</sup>Initial values:  $\mu = 0.01$  and  $\lambda = 10$ . calculation of  $R^2_{LMO}$  was based on 1000 random selections of 25% of samples

Table 5: Statistical and optimized parameters for four different weight update functions in data set 4 and comparison with MLR-Ant colony model

Update weight function	Optimized learning rate	Optimized momentum constant	LOO		L25%O <sup>b</sup>	
			Q <sup>2</sup> <sub>LOO</sub>	SE <sub>p</sub>	R <sup>2</sup>	SE <sub>p</sub>
Basic back propagation	0.1	0.8	0.824	0.746	0.771	0.773
Conjugate gradient	0.3	0.7	0.851	0.720	0.785	0.718
Quasi-Newton	-	-	0.870	0.709	0.845	0.711
Levenberg-Marquardt <sup>a</sup>	-	-	0.922	0.586	0.889	0.620
MLR-Antcolony	-	-	0.810	0.755	-	-

<sup>a</sup>Initial values:  $\mu = 0.01$  and  $\lambda = 10$ . calculation of  $R^2_{LMO}$  was based on 1000 random selections of 25% of samples

methods have been adequately described elsewhere (Jalali-Heravi and Asadollahi-Baboli, 2009) and we just bring their equations here. In LOO Strategy, the  $Q^2$  can be easily calculated by Eq. 11:

$$Q_{LOO}^2 = \frac{PRESS}{SSY} = 1 - \frac{\sum_{i=1}^n (y_{exp} - y_{pred})^2}{\sum_{i=1}^n (y_{exp} - \bar{y})^2} \quad (11)$$

where, PRESS is prediction error sum of squares and SSY is sum of squares of deviations of the experimental values from their mean.

In the case of LMO, M represents a group of randomly selected data points which would leave out at the beginning and would be predicted by the model which was developed using the remaining data points. So, M molecules are considered as prediction set. The  $R^2_{LMO}$  can be calculated by Eq. 12:

$$R_{LMO}^2 = \frac{PRESS}{SSY} = 1 - \frac{\sum_{i=1}^{test} (y_{exp} - y_{pred})^2}{\sum_{i=1}^{test} (y_{exp} - \bar{y}_{train})^2} \quad (12)$$

In present study, Calculation of  $R^2_{LMO}$  was based on 1000 random selections of groups of about 25% of data. This algorithm is shown in Fig. 4. The higher  $Q^2_{LOO}$  or  $R^2_{LMO}$  the higher predictive power of the model.

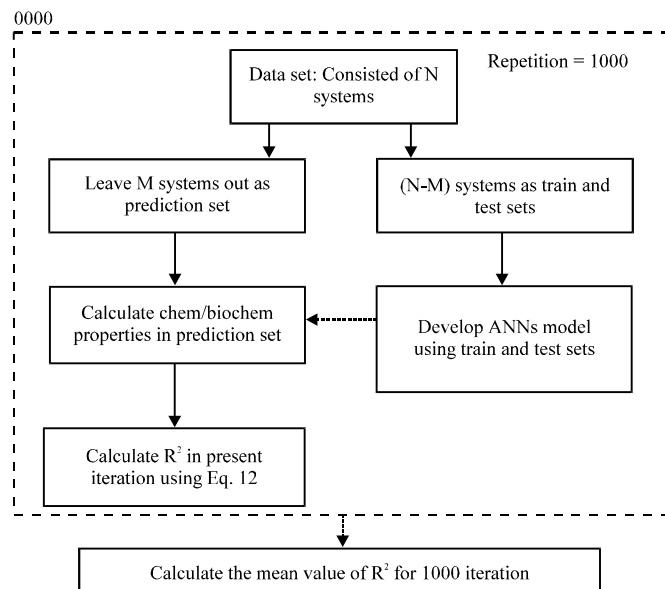


Fig. 4: Scheme of leave-multiple-out algorithm used in this study

## RESULTS AND DISCUSSION

As it can be seen from Eq. 1, there is a term called weight update function, which indicates the way that weights are changed during the learning process. This study focuses on investigating the role of weight update function which is an important specification of the networks. The four different weight update functions of BBP, CG, Q-N and L-M algorithms were used to develop ANNs models. The statistical parameters of these ANNs models are given in Table 2-5. In particular, the leave-one-out (LOO), leave-25%-out (L25%O) and  $SE_p$  procedures were utilized in this study to evaluate consistency, reliability and reproducibility of models. A calculation of  $R^2_{L25\%O}$  was based on 1000 random selections of molecules from the training set. Inspection of Table 2-5 reveals the importance of the role of algorithms in obtained results. The importance of four weight update functions for each different data set was shown in Fig. 5 and 6. These Fig. 5 and 6 also show the  $Q^2_{LOO}$  and  $R^2_{L25\%O}$  for each data set, respectively.

In the data set 1, Table 2 shows the superiority of L-M algorithm over the BBP, CG and Q-N algorithms. The results from L-M ANNs model are superior to the results of PLS model from previous work (Schnackenberg and Beger (2005)). Table 2 also shows the  $SE_p$  of the model is improved from 0.602 (PLS) to 0.432 (L-M ANNs). These results reveal the standard error of prediction for log P is reduced about 30% because of choosing appropriate weight update function. Also, Fig. 5 and 6 show that  $Q^2_{LOO}$  and  $R^2_{L25\%O}$  are at their highest values when L-M algorithm is used. It is clear from Table 2 that selecting BBP or CG weight updates functions do not improve the results of  $R^2$  and standard errors. Therefore, one may inaccurately conclude that there is not any non-linearity in data if BBP or CG weight updates functions had been selected.

Figure 5 shows that in the data set 2 the superiority of L-M algorithm over the BBP, CG and Q-N algorithms. The  $Q^2_{LOO}$  was improved from 0.701 (in BBP algorithm) to 0.895 (in L-M algorithm). As Table 3 shows the  $SE_p$  is reduced about 20% after implying the L-M weight update function. From this table we also can conclude that BBP, CG and Q-N are not suitable for ANNs models because they show poor results in comparison with L-M algorithms. The acceptable value

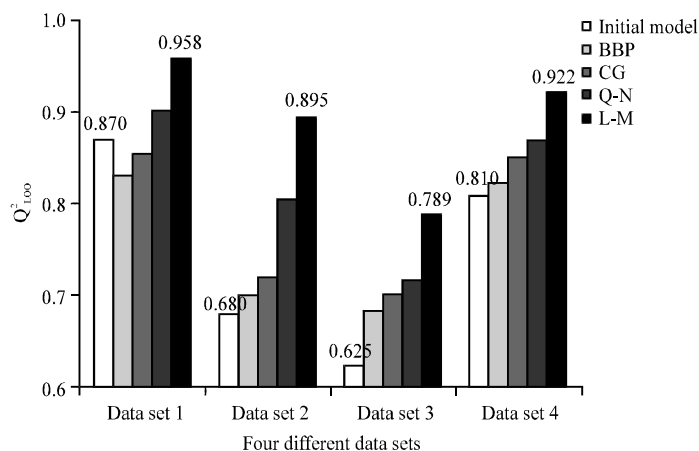


Fig. 5: Values of  $Q^2_{LOO}$  in four different data sets using different weight update functions

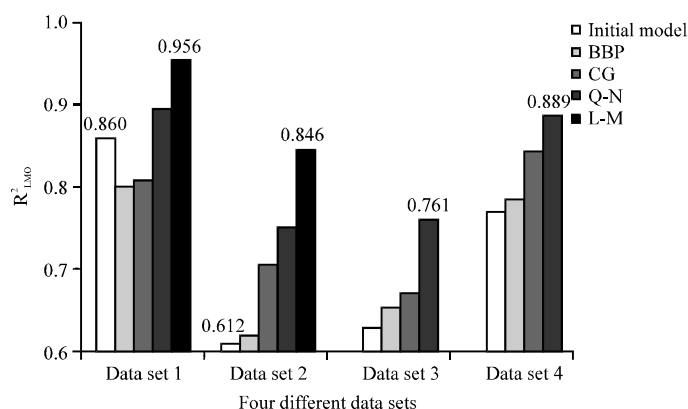


Fig. 6: Values of  $R^2_{L25\%O}$  in four different data sets using different weight update functions

of  $R^2_{L25\%O}$  ( $= 0.846$ ) reveals that the L-M ANNs model shows superiority over the other WUFs algorithms by accounting 84.6% of the variances of the intrinsic viscosity.

In developing ANNs models using the data set 3 and data set 4, the best weight update function to predict inhibitors activities was L-M algorithm (Table 4 and 5). In data set 3, the  $Q^2_{LOO}$  value was increased from 0.701 (in BBP algorithm) to 0.895 (in L-M algorithm). In data set 4, the L-M ANNs model has higher  $Q^2_{LOO}$  and  $R^2_{L25\%O}$  values and lower  $SE_P$  (Table 5, Fig. 5 and 6). Using ANNs method in third and fourth data sets, the  $SE_P$  was reduced about 20-25% compare to the  $SE_P$  in Antcolony-MLR method. It is also clear that the results (both  $R^2_{L25\%O}$  and  $Q^2_{LOO}$ ) are not satisfactory enough in BBP and CG algorithms.

In order to assess the robustness of the developed models, the Y-randomization test was applied in this contribution. The dependent variable vector (pIC50, Log P or  $\eta$ ) was randomly shuffled and a new QSAR/QSPR model was developed using the original variable matrix. The new QSAR/QSPR model is expected to show a low value for  $R^2_{L25\%O}$  and  $Q^2_{LOO}$ . The p value was calculated in Y-randomization test using Eq. 13:

$$p = \frac{\text{sum}(I(Q_{\text{LOO}}^2 > Q_{\text{LOO}}^2(n)))}{N} \quad (13)$$

where,  $I()$  is an indicator function that takes value 1 if the inequality in the parenthesis is true and takes value 0 otherwise.  $N$  is the total number of  $Y$ -randomization test.  $Q_{\text{LOO}}^2(n)$  is value of  $Q_{\text{LOO}}^2$  in iteration  $n$  (where  $n = 1, \dots, N$ ). The  $p$  value is the percentile of  $Q_{\text{LOO}}^2$  in the sequence of  $N$   $Q_{\text{LOO}}^2(n)$ . In the case of L-M algorithm, the  $p$  value for 100 iterations was equal to 1 in four different data sets. The  $p$  value indicates that the good results for L-M ANNs models are not due to a chance correlation or structural dependency of the training set.

It is shown in this article that the L-M weight update function algorithm is the best weight update function in four different data sets. However, in two cases (data sets 1 and 4) the Q-N algorithm also shows relatively acceptable results. Therefore, it is obvious that weight update functions have a key role to find the best relationship between dependent and independent variables.

## CONCLUSIONS

A survey of four different data sets of QSPR/QSAR methodologies has been applied successfully to establish a mathematical relationship between the chemical/biochemical property and selected descriptors. By focusing on the role of the weight update functions, we realized that the algorithms of weight update functions are important in the performance of the networks. It is shown that while all neural network parameters are optimized, the best weight update function is L-M algorithms in our study. The L-M algorithm has been testified to be an effective algorithm for developing ANNs models using the cross-validation techniques of leave-one-out, leave-multiple-out and also  $Y$ -randomization. Comparing the results of BBP algorithm, CG and Q-N algorithm with those for L-M algorithm reveals that the latter algorithm develops the best models to predict the partition coefficients, intrinsic viscosity,  $pIC_{50}$  of tyrosine kinase inhibitors and  $pIC_{50}$  of cyclooxygenase enzyme inhibitors.

## REFERENCES

- Afantitis, A., G. Melagraki, H. Sarimveis, P.A. Koutentis, J. Markopoulos and O. Igglessi-Markopoulou, 2006. Prediction of intrinsic viscosity in polymer-solvent combinations using a QSPR model. *Polymer*, 47: 3240-3248.
- Caron, G. and G. Ermondi, 2005. Calculating Virtual log P in the alkane/water system ( $\log P_{\text{alk}}^N$ ) and its derived parameters  $\Delta \log P_{\text{oct-alk}}^N$  and  $\log D_{\text{alk}}^{\text{pH}}$ . *J. Med. Chem.*, 48: 3269-3279.
- Dastorani, M.T., A. Talebi and M. Dastorani, 2010. Using neural networks to predict runoff from ungauged catchments. *Asian J. Applied Sci.*, 3: 399-410.
- Jalali-Heravi, M. and M. Asadollahi-Baboli, 2009. Quantitative structure-activity relationship study of serotonin receptor inhibitors using modified ant colony algorithm and adaptive Neuro-Fuzzy Inference System. (ANFIS). *Eur. J. Med. Chem.*, 44: 1463-1670.
- Kacem, I., H. Majdoub and S. Roudesli, 2008. Fraction of soluble polysaccharides from *Inula crithmoides* by sequential extraction. *J. Applied Sci.*, 8: 2442-2448.
- Khanale, P.B. and S.D. Chitnis, 2011. Handwritten devanagari character recognition using artificial neural network. *J. Artificial Intell.*, 4: 55-62.

- Lu, J.X., Q. Shen, J.H. Jiang, G.L. Shen and R.Q. Yu, 2004. QSAR analysis of cyclooxygenase inhibitor using particle swarm optimization and multiple linear regression. *J. Pharm. Biomed. Anal.*, 35: 679-687.
- Schnackenberg, L. and R.D. Beger, 2005. Whole-molecule calculation of log P based on molar volume, hydrogen bonds and simulated  $^{13}\text{C}$  NMR spectra. *J. Chem. Inf. Model.*, 45: 360-365.
- Sharma, M.C. and S. Sharma, 2011. 2D QSAR study of 7-methyljuglone derivatives: An approach to design anti tubercular agents. *J. Pharmacol. Toxicol.*, 6: 493-504.
- Shi, W.M., Q. Shen, W. Kong and B.X. Ye, 2007. QSAR analysis of tyrosine kinase inhibitor using modified ant colony optimization and multiple linear regression. *Eur. J. Med. Chem.*, 42: 81-86.