Genetic Algorithm Optimized Neural Networks Ensemble as Calibration Model for Simultaneous Spectrophotometric Estimation of Atenolol and Losartan Potassium in Tablets

Dondeti Satyanarayana*, Kamarajan Kannan and Rajappan Manavalan

Department of Pharmacy, Annamalai University, Annamalainagar, Tamil Nadu-608002, India.

Received 30 June 2005; revised 31 October 2005; accepted 10 January 2006.

ABSTRACT

Improvements in neural network calibration models by a novel approach using neural network ensemble (NNE) for the simultaneous spectrophotometric multicomponent analysis are suggested, with a study on the estimation of the components of an antihypertensive combination, namely, atenolol and losartan potassium. Several principal component neural networks were trained with the Levenberg-Marquardt algorithm by varying conditions such as inputs, hidden neurons, initialization, training sets and random Gaussian noise injection to the inputs. Genetic algorithm (GA) has been used to develop the NNE from the trained pool of neural networks. Subsets of neural networks selected from the pool by decoding the chromosomes were combined to form an ensemble. Several such ensembles formed the population which was evolved to generate the fittest ensemble. Ensembling the networks was done with weighted average decided on the basis of the mean square error of the individual nets on the validation data while the ensemble fitness in the GA optimization was based on the relative prediction error on unseen data. The use of a computed calibration spectral data set derived from three spectra of each component has been described. The calibration models were thoroughly evaluated at several concentration levels using spectra obtained for 76 synthetic binary mixtures prepared using orthogonal designs. The ensemble models showed better generalization and performance compared with any of the individual neural networks trained. Although the components showed significant spectral overlap, the model could accurately estimate the drugs with satisfactory precision and accuracy, in tablet dosage with no interference from excipients as indicated by the recovery study results. The GA optimization guarantees the selection of the best combination of neural networks for NNE and eliminates the arbitrariness in the selection of any single neural network model, thus maximizing the knowledge utilization without the risk of memorization or over-fitting.

KEYWORDS

Neural network ensemble, principal components, atenolol, losartan potassium, UV spectrophotometry.

1. Introduction

Neural Networks (NNs) of appropriate architecture have the ability to approximate any function to any desired degree. However, it has been shown that the transfer function must be continuous, bounded and non-constant for a NN to approximate any function.[1] Fundamental background information on NNs can be found elsewhere.[2–4] Research into the theoretical and practical aspects of the use of NNs for calibration and pattern recognition in analytical chemistry has increased rapidly in the last decade. Several papers employing neural networks have been published since then, in practically all areas of chemical research.[5–11] There are some recent reports on the application of NNs for mixture analysis,[12–16] most of which employ a separate network for the estimation of each component of the mixture.

In NN-based modelling, there are many degrees of freedom in selecting the network topology, training algorithm and training parameters. At the end of the training process, a number of trained networks are produced, and then typically one of them is chosen as the best, based on some optimality criterion, while the rest are discarded.[17] The present work attempts to use the pool of trained networks (with potentially useful knowledge) to build an effective neural ensemble, which in consortium may be more effective than single network models in terms of generalization and accuracy.

1.1. Neural Ensemble

Neural network modelling essentially involves an optimization process by training a number of neural networks. Training the same model with the same training data set but with a different initial environment, such as the initial weights, would end up with a slightly different final set of weights and hence final performance. Therefore, one has to consider the intrinsic variance the NN models exhibit. An effective way of reducing the variance of the networks is to combine a number of networks to form an ensemble network.[18] A Neural Network Ensemble (NNE), shown in Fig. 1, is a learning paradigm in which a collection of a finite number of NNs is trained for the same task.[19] It originates from Hansen and Salamon's work[20] which shows that the generalization ability of a neural network system can be significantly improved through an ensemble of a number of neural networks, i.e. training many neural networks and then combining their predictions. The motivation for combining nets in redundant ensembles is that of improving their generalization ability. Combining a set of imperfect estimators can be thought of as a way of managing the recognized limitations of the individual estimators; each component net is known to produce errors, but they are combined in such a way as to minimize the effect of these errors. Since this technology behaves remarkably well, it is being explored in both neural networks and machine learning communities for wider applications.[21] Its implications

* To whom correspondence should be addressed. E-mail: sand60@rediffmail.com

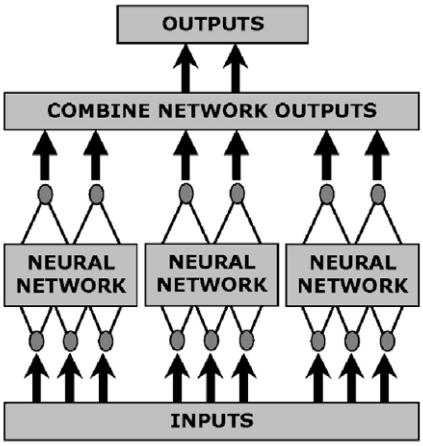


Figure 1 Graphical depiction of a Neural Network Ensemble.

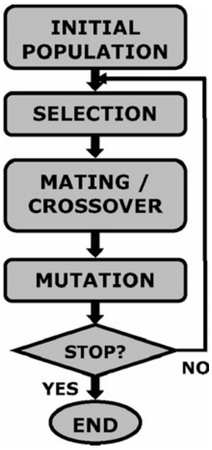


Figure 2 Typical GA.

to other applied areas in chemistry and pharmaceutical science cannot be underestimated.

1.2. Genetic Algorithms

Genetic algorithms[22] (GA), modelled on biological genetics and the law of natural selection, operate by maintaining and modifying the characteristics of a population of solutions (individuals) over a large number of generations. This process is designed to produce successive populations having an increasing number of individuals with desirable characteristics. Like nature's solution, the process is probabilistic but not completely random. The rules of genetics retain desirable characteristics by maximizing the probability of proliferation of those solutions (individuals) who exhibit them.

GA operates on a coding of the parameters, rather than the parameters themselves. Just as the strands of DNA encode all of the characteristics of a human in chains of amino acids, so the parameters of a problem must be encoded in finite length strings which might be a sequence of any symbols, though the binary symbols '0' and '1' are often used. Optimization is performed on a set of strings, where each string is composed of a sequence of characteristics. Given an initial population of strings, a genetic algorithm produces a new population of strings according to a set of genetic rules. This constitutes one generation. The rules are devised so that the new generation tends to have strings that are superior to those in the previous generation, measured by some objective function. Successive generations of strings are produced, each of which tends to produce a superior population. Optimizing a population rather than a single individual contributes to the robustness of these algorithms. Any problem for which an objective function can be defined is a candidate for genetic optimization. A typical implementation of a genetic algorithm is shown in Fig. 2. For fundamental information on GA, one may refer to Goldberg.[22]

1.3. Scope of Work

Given the wide-ranging applicability and uses of neural networks in the field of chemistry, improvements to the NN modelling process are highly desirable. In the process of NN modelling optimization several neural networks are trained with random initialization or with varying calibration data sets. Most of the knowledge of such a network is discarded by employing only one network as a calibration model. According to Chan *et al.*[23] the generalization error of the ensemble network is generally smaller than that obtained by a single network, while at the same time the variance of the ensemble network is smaller than that of a single network, thus becoming a very effective way to improve the prediction ability. Motivated by these characteristics of NNE, the present study attempts to utilize a range of optimally configured and trained neural networks to form an effective ensemble employing the technique of Genetic Algorithms, to compare its performance with a single neural network (the best one as determined by some test data) and ensemble of all the neural networks using atenolol (ATL) and losartan potassium (LST) antihypertensive mixture and also to evaluate its use in the analysis of a combined tablet dosage. In essence, the study was designed to investigate the benefits of neural network ensembles over simple neural models, if any, thereby paving the way for a variety of applications in chemistry, pharmaceutical and allied fields.

2. Experimental

2.1. Chemicals and Reagents

Distilled water served as a solvent for making the stock solutions and all further dilutions of ATL, LST, their standard combinations and the tablet powder. Class A volumetric glassware such as pipettes and volumetric flasks was used for the purpose of making dilutions.

2.2. Instruments and Software

UV absorption measurements were carried out on a Perkin-Elmer Lambda 25 double beam spectrophotometer controlled by UVWINLAB software version 2.85.04, using matched 1.00 cm quartz cells. All masses were measured on an electronic balance with 0.01 mg sensitivity. The spectra of all the solutions were recorded against a blank solution containing no analytes, between 200 and 300 nm and saved in ASCII format. Matlab® version 6.1 was employed for building principal component Levenberg-Marquardt back-propagation neural networks (PC-NN) and neural network ensembles. All computations were carried out on a desktop computer with a Pentium 4, 1.6 GHz processor and 256 MB RAM.

2.3. Preparation of Standard Solutions

Standard solutions of pure ATL and LST were made at different concentration levels ranging from 4 to 20 mg L^{-1} and 5 to 22 mg L^{-1}, respectively, for the purpose of linearity determination and to design the calibration data matrix from their spectra. The analytical levels of 10 mg L^{-1} were chosen for both ATL and LST.

2.4. Calibration Data

Since the absorbances were linearly additive in the desired range and no serious baseline problems or interactions were

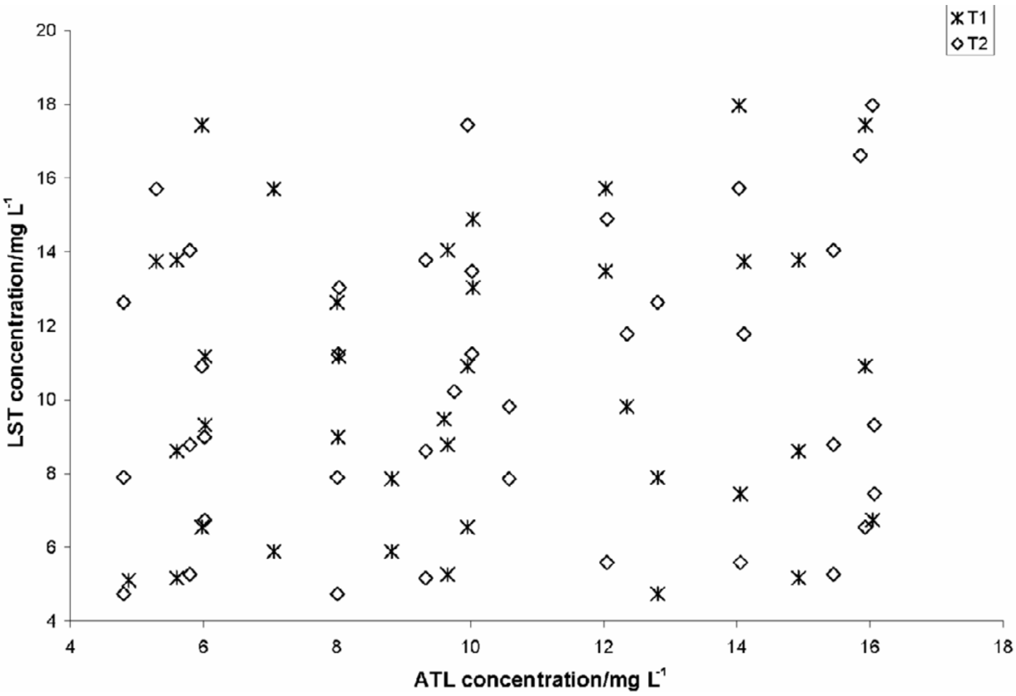


Figure 3 Synthetic binary mixture design for testing the neural networks. Each point represents a mixture of the respective concentrations of the components. 76 mixtures were used. The mixtures were split into two groups T1 (✳) and T2 (◇). The design ensures that the model is thoroughly validated in a well distributed concentration space, especially with regard to the chosen analytical level.

found in our trial studies in the desired range of concentration, the process described below was adopted in the design of calibration data set for training the PC-NN. A set of three spectra for each component at three different concentration levels (low, medium and high) was employed in all possible combinations to provide a fair simulation of calibration data set with some degree of experimental variation. A full factorial design was employed to obtain 49 training pairs from each spectral pair resulting in a total of 441 training pairs (49 × 9) representing the mixture space evenly with target concentrations that are orthogonal. The total of 441 training pairs thus obtained, constituting the complete calibration set, was used to train the PC-NN model. All the target concentrations in the calibration set were then standardized (to a mean of 0 and standard deviation of 1). The spectral region between 215 and 275 nm was chosen on the basis of visual inspection of the spectra. A total of three calibration data sets was obtained using different sets of spectra of pure components.

2.5. Validation Data

Randomized validation data sets were used for the internal validation and terminating the training of the PC-NN at an optimum point to prevent over-fitting and to retain the generalization ability of the network. A validation data set of the same size was also designed from three different pairs of spectra of ATL and LST standard out of which at least two pairs were different from that used in the calibration data set.

2.6. Synthetic Binary Mixtures for Model Evaluation

The synthetic binary mixtures were prepared on different days from fresh stock solutions of pure ATL and LST, by separate weighing each day, in distilled water. Standard mixtures of the components were prepared with the concentrations lying within the known linear absorbance-concentration range by dissolving varying proportions of ATL and LST stock solutions; the concentration of ATL varied between 50 and 160% of the analytical level concentration while that of LST varied between 50 and 180% of its analytical level concentration. A total of 76 mixtures was prepared with the concentrations of components selected to span the mixture space fairly evenly, as show in Fig. 3.

2.7. Processing of Tablet Dosage Form

For the analysis of the active components of the antihypertensive tablet (Losar-Beta, ATL 50 mg and LST 50 mg, Unichem Laboratories Ltd., India, Batch No. LB4005), twenty tablets were accurately weighed, carefully powdered and mixed. Tablet powder corresponding to the equivalent of 20 mg of LST was dissolved in distilled water by sonication for 5 min and made up to 100 mL. The solution was centrifuged and 5 mL of supernatant was diluted to 100 mL. Three replicate dilutions were made from each stock solution, repeating the entire process for a total of five samples of the tablet powder.

For accuracy studies, by recovery, the same tablet powder was used in amounts corresponding to the equivalent of 8 to 22 mg of LST (in order to enable spiking up to desired levels). The powder was then spiked with a known quantity of pure ATL and LST and dissolved in distilled water by sonication and made up to 100 mL with the same solvent. The solution was then centrifuged and 5 mL of supernatant was diluted to 100 mL. A total of five powder samples was spiked to different levels in the range of 50 to 160%, each in three dilution replicates.

2.8. PC-NN Models

Feedforward backpropagation neural network models were used in the study with a single hidden layer consisting of sigmoid neurons. The inputs of the neurons corresponded to the number of principal components for that network. Several PC-NN models were built with varying numbers of input neurons (corresponding to the number of principal components chosen, *viz.* 2 to 5) and the number of hidden neurons. Principal component analysis was carried out by employing custom developed functions in MATLAB using the inbuilt eigenvalue decomposition function (*'eig'*) to obtain the latent vectors (eigenvectors) and the corresponding eigenvalues. The scores obtained by projecting the standardized absorbance values onto these eigenvectors were used as inputs. The PC-NN had two neurons in the output layer corresponding to the two components of interest. The number of neurons in the hidden layer was varied from 2 to 5 neurons for each level of the input neurons chosen. The input layer and output layer nodes had identity and linear transfer functions, respectively, while the hidden layer nodes had sigmoid transfer functions for the PC-NN, decided on the basis of earlier studies on neural calibration models.[16,24,25] All the PC-NN models were trained according to the Levenberg-Marquardt[26] algorithm available in the neural network toolbox for MATLAB through the '*trainlm*' function. The training was terminated when the validation performance as estimated by the mean square error (MSE), for a validation data set, increased continually for more than 10 epochs since the last time it decreased. Five replicate neural network models were trained for a given configuration, and given calibration data set, each with different initialization of weights by the Nguyen-Widrow[27] method. Three different calibration data sets were used in the study to rule out any chance correlations, thus resulting in a total of 15 PC-NN models for each of the configurations explored.

2.9. Neural Ensemble Model

PC-NN models using 3–5 inputs (corresponding to the number of principal components) and having two hidden neurons constituted the pool of networks used for Genetic Algorithm Optimized Neural Network Ensemble (GAONE) model development. This was done in order to keep the topology of the network as simple as possible (a general rule of thumb) and at the same time to explore the usefulness of increasing the input information by choosing a larger number of principal components and hence the number of inputs. Thus, three different configurations of the neural network models were employed to build ensembles. Fifteen PC-NN models were available for each configuration from five replicate training sets (with different initialization of weights) for each calibration data set. Yet another 15 PC-NN models were also trained for each configuration, with fresh injection of random Gaussian noise with a standard deviation of one part per thousand each time for a given calibration data set. By this process, 30 PC-NN models were available for each of the three calibration data sets, making the total PC-NN model population 90 for all three calibration data sets. Five subsets of models (MM1, MM2, MM3, MM4 and MM5) each with 60 models were randomly selected from the available population of 90 models in order to evaluate the consistency in performance of the GAONE and also to rule out any 'chance effect'. Each of this subset of 60 models was subjected to the GA process to build the GAONE. Two different fitness data sets, T1 and T2, were used independently to determine the fitness of individuals in the GA process. Three runs of GA were done for a given subset of models and a given fitness data set each with a different seed (the time in milliseconds as obtained from the computer system at the run time) for the random number generator. In all, a total of 30 runs (5 model subsets × 2 fitness data sets × 3 runs = 30) was made to eliminate all possible chance correlations and effects. The outputs of the constituent PC-NN models in the ensemble were combined on the basis of ensemble weights computed from the mean square error (MSE) on the validation data set of the respective PC-NN model for each of the output neurons.

2.10. Genetic Algorithm Implementation

Standard GA was employed using the Genetic Algorithm Toolbox[28] for Matlab for the purpose of building the GAONE models. Binary coded chromosomes were employed with an initial population of 300. Fitness of the GAONE was estimated by determining the mean percentage relative prediction error (%RPE) of the GAONE for a fitness data set (T1 or T2) employed in the process. The Roulette Wheel Selection[22] scheme was employed in determining the opportunities for individuals to reproduce and recombine to produce offspring. Multi-point crossover was used in the present work as recombination operator. The idea behind multi-point, and indeed many of the variations on the crossover operator, is that the parts of the chromosome representation that contribute most to the performance of a particular individual may not necessarily be contained in adjacent substrings. Further, the disruptive nature of multi-point crossover appears to encourage the exploration of the search space, rather than favouring the convergence to highly fit individuals early in the search, thus making the search more robust. This crossover operation was not necessarily performed on all strings in the population. Instead, it was applied with a probability of 0.7 when the pairs were chosen for breeding (in simple terms, the probability of recombination/crossover occurring between pairs of individuals was 0.7). A further genetic operator, called mutation, was then applied to the new chromosomes, again with a set probability of 0.7. After recombination and mutation, the individual strings were then decoded, if necessary, the objective function evaluated, a fitness value assigned to each individual and individuals selected for mating according to their fitness, and so the process was continued through subsequent generations. In this way, good individuals were preserved and bred with one another and the less fit individuals died out. Six GAONE models were developed from each random pool of 60 PC-NN models by varying the fitness determining data set (*viz.* test data sets T1 or T2) derived from the spectra of binary synthetic mixtures. The GA process was repeated at least three times for each case.

2.11. Evaluation of Models

All trained GAONE models were evaluated by testing with the different sets of spectral data obtained from the synthetic binary mixture designed as described earlier. The mean %RPE (representing the combined error for the entire mixture) was employed as an indicator of performance of the model and was used to compare the performance of the GAONE models with the best single PC-NN model (one from the 60 PC-NN models in the respective subset, yielding the lowest%RPE on a corresponding fitness test data set employed for GAONE development) and the ensemble of all the PC-NN models available in the subset. All the test data sets *viz.* T1, T2 and T (=T1+T2) were employed for the performance comparison.

2.12. Tablet Analysis

Spectra recorded from the tablet solutions were analysed by the GAONE calibration models built from the base pool of all the 90 PC-NN models and the concentrations predicted for each solution were used for calculation of the tablet content. Similarly ATL and LST concentrations in the solutions prepared for recovery study were also obtained from the respective spectra and percentage recovery was calculated to determine the accuracy of the method.

3. Results and Discussion

There are many pitfalls in the use of calibration models, perhaps the most serious being variability in instrument performance over time. Each instrument has different characteristics and on each day and even hour the response may vary. Therefore it is necessary to reform the calibration model on a regular

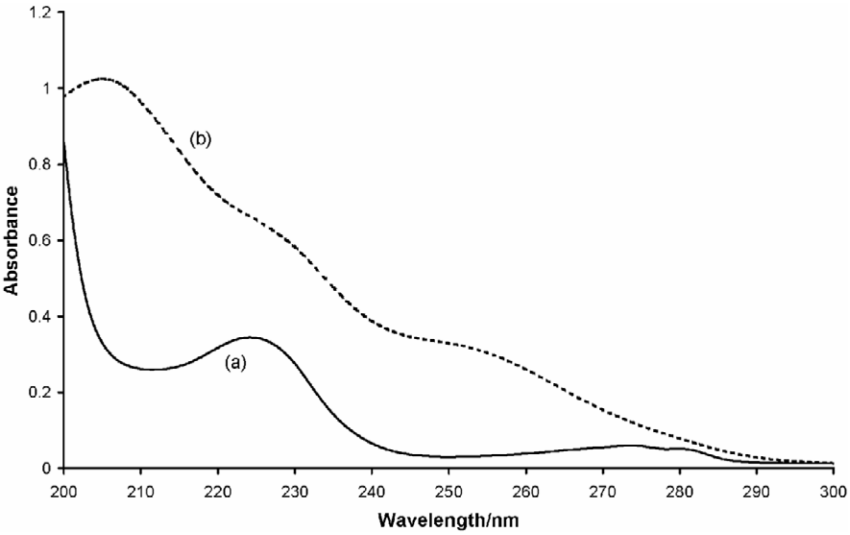


Figure 4 UV spectra of atenolol and losartan potassium. Overlain spectra of (a) ATL at a concentration of 9.319 mg L^{-1} and (b) LST at a concentration of 10.267 mg L^{-1} in distilled water.

basis, by running a standard set of samples.[29] As with other regression methods, there are constraints concerning the number of samples, which at times may be limiting the development of an NN model. The number of adjustable parameters (synaptic weights) is such that the calibration set is rapidly over-fitted if too few training pairs are available, leading to loss of generalization ability. Therefore, calibration sets of several hundred training pairs may often be necessary to get a representative distribution of the concentration across their range. This makes it expensive in time and resources to develop calibration mixtures physically in such large numbers, which is rarely possible in routine laboratory studies and justifies our attempt to use mathematically constructed calibration data sets from individual spectra of components. However, this approach cannot be applied in cases where significant non-linearity is exhibited.

The overlain absorption spectra in Fig. 4 show extensive spectral overlap, which complicates the determination of the individual drug concentrations from a spectrum of a mixture. When considered separately, concentrations between 4 and 20 mg L^{-1} for ATL and 5 and 22 mg L^{-1} for LST were found to be linear over the space of nine concentration levels (absorbances measured at 224 nm for ATL and 220 nm for LST) with r^2 of 0.9985 and 0.9976 for each, slopes of 0.0369 and 0.0678 L mg^{-1}, intercepts of –0.0012 and 0.0205 and residual standard deviation about the regression line being 0.0071 and 0.0185, respectively.

In general, a NNE is constructed in two steps, i.e. training a number of component neural networks and then combining the constituent networks (which together form an ensemble) predictions.[30] For regression tasks, the predictions of constituent neural networks are combined using one of the prevailing approaches, such as simple averaging, weighted averaging[17,31,32] that takes account of the relative accuracies of the nets to be combined or by a generalized ensemble method.[18] In the correlation ensemble method suggested by Chan *et al.*,[23] the weighting of the ensemble was determined by the correlation of the output of the ensemble networks (Y) to the target output (X) as given by $w = X^TY$; the more correlated is the network output, the higher the weighting value it has. In the present study weighted averaging was used.

In the present study, several Principal Component Neural Network (PC-NN) models in replicates were built as described in the experimental section by varying the calibration data sets, validation data sets, number of principal components, number of hidden neurons and injection of random Gaussian noise. The PC-NN model trained rapidly taking less than one minute and fewer than 300 epochs. The mean square error (MSE), calculated according to Equation 1, and the mean percentage relative prediction error (%RPE), computed as given in Equation 2, for both components were the criteria used in the optimization of the neural network configuration.

$$\text{Mean square error (MSE)} = \frac{1}{m}\sum_1^m (C_{\text{act}} - C_{\text{pred}})^2 \qquad (1)$$

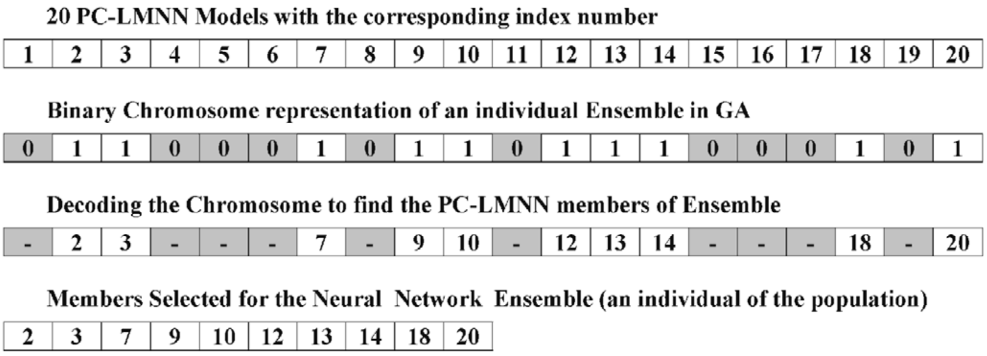
$$\%\text{RPE} = \frac{100 \times \sqrt{\text{MSE}}}{\overline{C}}, \qquad (2)$$

where C_{act} is the desired target, C_{pred} is the output produced by the network for each input vector, $\overline{C}$ is the mean concentration of the component and m is the number of input vectors or samples. It was found that PC-NN models with an input of 3–5 neurons, an output of two neurons, both having linear transfer function and a hidden layer with 2–5 neurons with sigmoid transfer function showed no significant difference in their performance as determined by ANOVA[33] on their mean %RPE as shown in Table 1. Only PC-NNs with two hidden neurons were used in

Table 1 Optimization of PC-NN models.

PCs [a]	Hidden Neurons	Mean %RPE [b]	Standard deviation
2	2	3.6297	0.0968
2	3	3.6520	0.0840
2	4	3.6649	0.1134
2	5	3.6511	0.1144
3	2	1.3488	0.0288
3	3	1.3845	0.1021
3	4	1.5681	0.3195
3	5	1.4905	0.2041
4	2	1.4901	0.1503
4	3	1.4980	0.1508
4	4	1.7846	0.7528
4	5	1.8973	0.9476
5	2	1.4336	0.1312
5	3	1.4716	0.1803
5	4	1.5405	0.2326
5	5	1.6202	0.4491

[a] Principal components used for the input.
[b] Average of 15 PC-NN models.



20 PC-LMNN Models with the corresponding index number

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Binary Chromosome representation of an individual Ensemble in GA

0	1	1	0	0	0	1	0	1	1	0	1	1	1	0	0	0	1	0	1

Decoding the Chromosome to find the PC-LMNN members of Ensemble

-	2	3	-	-	-	7	-	9	10	-	12	13	14	-	-	-	18	-	20

Members Selected for the Neural Network Ensemble (an individual of the population)

2	3	7	9	10	12	13	14	18	20

Figure 5 Illustrative coding and decoding of chromosome in building the GAONE model.

NNE as described under '*Neural Ensemble Model* in section 2.9, since it was desirable always to minimize the topology of the network. However, the inputs were varied in order to explore the usefulness of any extra information in terms of principal component scores.

Though most ensemble approaches in engineering applications have been to employ all of the networks available to constitute an ensemble, recently it has been reported that an ensemble of many of them may be better than an ensemble of all the available neural networks.[30] However, excluding those 'bad' neural networks from the ensembles is not an easy task as we may have imagined. It is not as simple as combining a few selected best performing networks, but combining networks that make error diversely, which as an ensemble perform better than any single NN in the population. Selection of nets for effective combination is to reduce the number of shared failures that a set of nets will produce. The extent to which they exhibit coincident failures can be determined only through a process of testing the performance of the selected ensembles.[34] If there are N trained networks, the number of possible combinations would be 2^N–1 which would become enormous as the value of N increased.

It was reported that varying the data on which NNs are trained is more likely to result in a set of nets that can be combined more effectively than, for instance, varying the set of initial conditions from which they are trained, or topology.[34] However, both approaches have been adopted in the present study as explained under '*Neural Ensemble Model'* in section 2.9 using 90 PC-NN models which formed the base pool. In order to investigate the effectiveness of an ensemble and also the fact that many of the NNs may be better than an ensemble of all available neural networks, 60 models were randomly selected to form a set of models. Five such subsets of models (MM1, MM2, MM3 MM4 and MM5) were created and used in the present study. Thus the possible number of combinations of NNs into ensembles equals 2^{60}–1 for one subset of models and the task of selecting the best one may be computationally very intensive. Hence the GA approach was considered since it has been shown to be a powerful optimization tool[22] to pick the best ensemble from a pool of NNs with the use of a selection criterion. Genetic algorithms actively create a population of ensembles and search for the best ensemble which generalizes well. The standard genetic operators, crossover and mutation, were used to create new individuals from an initial set. The fit members then constitute the parents who reproduce to create the next generation, and the process was repeated until a stopping criterion was reached.

Genetic Algorithm Optimized Neural Network Ensemble (GAONE) model development here was realized by utilizing the standard genetic algorithm[22] with a binary coding scheme that represents each ensemble of neural networks. The process of coding and decoding in GA implementation is illustrated in Fig. 5 using a single chromosome and a subset of 20 NN models that combine to form ensembles. An initial population of 300 neural network ensembles was evolved by GA to build the GAONE model.

In building ensembles the weighted average approach was preferred over the simple averaging because of the fact that one should believe accurate models more than inaccurate ones. In this approach the predictions of the networks were achieved by taking a weighted sum of the output of each network, where each weight was based on the validation-set accuracy of the network. The present one being a multi-output case, an optimal combination of weight vectors for each output was computed separately. The weights for combining the networks in the ensemble were defined by Equation 3 (N equals the number of networks, $\hat{o}$ is the ensemble output, wi, and o_i are the weight and output for the i^{th} network):

$$\hat{o} = \sum_{i=1}^{N} w_i . o_i \text{ with the constraint that } \sum_{i=1}^{N} w_\mathrm{i} = 1 . \tag{3}$$

The mean square error (MSE) was chosen as the criterion for determining the weights in combining the NNs into an ensemble since it is a measure of both the accuracy and the variance. If $\hat{y}$ is an estimator of an unknown quantity, μ the bias for $\hat{y}$ is defined by $bias(\hat{y}) = E\ (\hat{y}) - \mu$, where E is the mean over infinitely many replications. The squared bias, $bias(\hat{y})^2 = \{E\ (\hat{y}) - \mu\}^2$ is a measure of the accuracy of $\hat{y}$. The variance of $\hat{y}$, $variance(\hat{y}) = E\ \{\hat{y} - E\ (\hat{y})\}^2$ measures the precision of $\hat{y}$. Good accuracy in itself is not enough for a good estimator because $\hat{y}$ may be very variable yet still accurate. Good precision is not in itself enough for a good estimator, because $\hat{y}$ may be very precise, yet may fail the target y most of the time. Hence an overall measure of the quality of $\hat{y}$ is defined by MSE $(\hat{y}) = E\ \{\hat{y} - \mu\}^2$ the mean square error for $\hat{y}$. The MSE is the sum of precision and accuracy as can be seen from the following decomposition:

$$\begin{aligned}\text{MSE}\ (\hat{y}) &= E\ \{\hat{y}\ - \mu\}^2\\ &= E\ \{\hat{y} - E\ (\hat{y}) + E\ (\hat{y}) - \mu\}^2\\ &= E\ \{\hat{y} - \mathrm{E}\ (\hat{y})\}^2 + \{\mathrm{E}\ (\hat{y}) - \mu\}^2 + 2\mathrm{E}\ \{\hat{y} - \mathrm{E}\ (\hat{y})\}\ \{\mathrm{E}\ (\hat{y}) - \mu\}\\ &= \mathrm{E}\ \{\hat{y} - \mathrm{E}\ (\hat{y})\}^2 + \{\mathrm{E}\ (\hat{y}) - \mu\}^2 \quad \text{since the term E } \{\hat{y} - \mathrm{E}\ (\hat{y})\}\\ &\quad \text{is } \textit{zero.}\\ &= variance(\hat{y}) + bias(\hat{y})^2.\end{aligned}$$

Since the NN models were multi-output type, the optimal combination-weights vector for each output was done separately on the basis of the MSE for each output. The exact mechanism for the determination of weights in the present study is given below:

$$E_i = \text{MSE} / \sum_{j=1}^{m} \text{MSE}_j \tag{4}$$

$$AE_i = 1 - E_i \tag{5}$$

$$W_i = AE / \sum_{j=1}^{m} AE_j \tag{6}$$

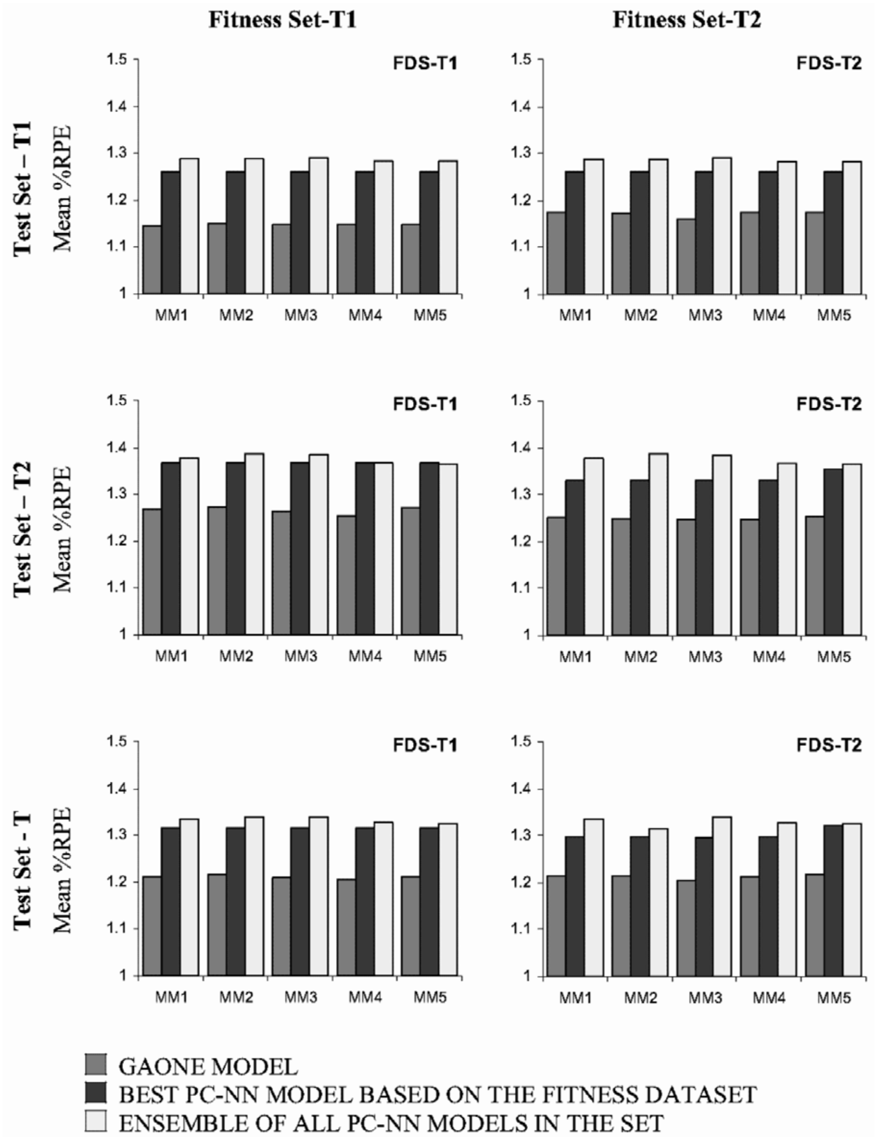


Figure 6 Comparison of GAONE models with other types of calibration models with respect to mean relative prediction error. FDS = fitness data set.

where E_i is the normalized MSE for a member of NN in the ensemble, m is the number of NNs in the ensemble, AE_i is the adjusted MSE (which is necessary to provide more weight to NNs exhibiting lower MSE) and W_i is the weight for the i^{th} NN. Ensemble weights were obtained individually for each of the output neurons by employing matrix operations in Matlab using Equations (4) to (6) above.

All the ensembles (individuals) thus formed were evaluated for their fitness using the mean %RPE obtained for an unseen fitness data set (T1 or T2). The individual having the lowest mean %RPE was considered the fittest individual and fitness ranking was assigned in ascending order of mean %RPE (the individual with lowest mean %RPE was ranked first). The parents were selected according to a probabilistic function (Roulette Wheel Selection[22]) based on relative fitness. In other words, those individuals with higher relative fitness are more likely to be selected as parents. *N* children were created *via* recombination from the *N* parents. The *N* children were mutated and survive, replacing the *N* parents in the population. Mutation flips bits with some small probability (here it was 0.7), and is often considered to be a background operator. Recombination (multi-point crossover), on the other hand, was emphasized as the primary search operator. The GA was terminated when the highest ranking individual's fitness had reached a plateau such that 10 successive iterations were no longer producing better results (individuals).

The entire process was repeated with five different subsets of models (MM1, MM2, MM3 MM4 and MM5) and different fitness data sets (T1 or T2) to obtain the GAONE models, including a minimum of three replicate runs of GA for each combination of models set and fitness data set. With a few exceptions, the GA found the same GAONE model for a given combination of model set and fitness data set in the replicate runs of GA. The run time ranged from 25 minutes to 38 minutes and the number of generations ranged from 36 to 61. The performances of the GAONE models were compared with the best NN model in the subset, and ensemble of all NN models in the respective model subset. The mean %RPE with the corresponding fitness data set was the criterion used for determining the best NN model(s). The results are shown in Fig. 6. The bar diagrams clearly indicate the performance edge of the GAONE models over other model types, irrespective of the models set from which it was derived or the fitness data set used in the GA process when tested against all test data sets. Though, expectedly, different error levels were observed for any given combination of the fitness data set and test data set on all model types, the relative performance of the

Table 2 Atenolol prediction characteristics of NNE calibration models (regression of the actual versus predicted concentrations).

Fitness data set	Test data set	%RPE	Slope	Intercept/ mg L^{-1}	Res. S.D. [a]	R^2
T1	T (T1+T2)	1.124	1.010	–0.092	0.115	0.999
T2	T (T1+T2)	1.171	1.011	–0.102	0.120	0.999
T1	T2	1.101	1.012	–0.118	0.115	0.999
T2	T1	1.208	1.008	–0.069	0.122	0.999

[a] % Residual standard deviation.

Table 3 Losartan potassium prediction characteristics of NNE calibration models (regression of the actual versus predicted concentrations)

Fitness data set	Test data set	%RPE	Slope	Intercept/ mg L^{-1}	Res. S.D. [a]	R^2
T1	T (T1+T2)	1.293	1.005	–0.098	0.128	0.999
T2	T (T1+T2)	1.254	1.005	–0.093	0.125	0.999
T1	T2	1.429	1.004	–0.090	0.139	0.999
T2	T1	1.138	1.007	–0.107	0.115	0.999

[a] % Residual standard deviation.

GAONE model was always superior. The difference was found to be significant as inferred by ANOVA in all cases (*P* value = 0). In a few instances of replicate runs of GA, the GAONE found was different but their performance relative to other forms of ensembles and independent NN models remained the same. In all cases, the single best NN model selected from respective sets of models for a given test data, performed relatively poorly when compared with GAONE models for the respective subset of models, thereby confirming the hypothesis that ensembles of selected NNs perform better than the single best NN model.

Since the GAONE models consistently outperformed the stand-alone PC-NN models, irrespective of the subset of 60 models used, and proved their superiority, the GAONE model was built from the entire pool of 90 models and their prediction characteristics were studied. Further, it was employed in the analysis of tablets and the accuracy studies thereafter. The performance characteristics of the GAONE model are summarized in Tables 2 and 3 for each ATL and LST, respectively.

Spectra obtained from 30 tablet solutions (including replicates) prepared from 5 different samples as described in the experimental section were analysed by the GAONE model (built from the entire pool of 90 PC-NN models) and the average content was calculated. The results are summarized in Table 4.

The accuracy of the method for the analysis of tablets was further investigated using the recovery studies as described in the experimental section. The mean percentage recovery and its relative standard deviation obtained by the GAONE models for both ATL and LST were found to be excellent as indicated in Table 5.

In developing neural network models for multivariate calibration, several networks are usually trained since it is known that they exhibit intrinsic variance. Hence, retaining only one neural network model and rejecting others may not be a good idea since many workers have found ensembles of neural networks to be an effective way of reducing the variance, improving generalization and accuracy.

Based on the reports that 'many may be better than all',[30] GAONE models were successfully built and tested in this study which has clearly proved that they were always better than any given single best neural network model or ensembles of all such models. This technique also eliminates the pitfalls of arbitrariness in the selection of any single neural network as a calibration model and discarding all other trained neural networks. The GAONE models developed in this study performed well in

Table 4 Analysis of tablet samples by GAONE model.

	Mass/ mg	
	ATL	LST
Sample 1	47.93	44.17
Sample 2	49.72	45.29
Sample 3	49.47	45.92
Sample 4	48.47	45.82
Sample 5	47.95	45.78
Mean tablet content	48.71	45.40
Standard deviation	0.843	0.729
Relative standard deviation	1.731	1.605
Amount on the label	50.000	50.000
% of the reported content	97.42	90.79

Table 5 Recovery studies of ATL and LST in tablets using the GAONE model.

Spiked sample	Mass of atenolol (ATL)/mg			Mass of losartan potassium (LST)/mg		
	Actual	Found	% Recovery	Actual	Found	% Recovery
1	10.17	10.22	100.48	9.67	9.67	100.06
2	15.38	15.21	98.87	14.60	14.77	101.17
3	21.55	21.80	101.15	20.46	20.54	100.40
4	26.61	26.16	98.29	25.36	24.90	98.16
5	31.56	30.83	97.68	30.17	30.00	99.44
Mean			99.29			99.85
R.S.D.			1.480			1.134

estimating ATL and LST simultaneously when tested with spectra recorded on different days and exhibited ruggedness even when different sets of constructed calibration data were used in the model development as indicated by the prediction results.

The performance of the GAONE model was better than even the best performing PC-NN model in terms of the mean %RPE. The accuracy of the GAONE model was also established in the analysis of the combined tablet dosage. The study indicates that in neural network calibration modelling, it may be more worthwhile to build neural network ensembles than rely on an independent neural network model. The optimization and manual selection of the neural network model becomes redundant since the GA process does the task automatically from a given pool of trained neural networks of different configurations. The work presents a technique that might have wide repercussion to any work associated with neural network modelling in chemistry.

Acknowledgements

The authors are grateful to the All India Council for Technical Education, New Delhi, for providing financial assistance under an R&D scheme to carry out this work. We are grateful to Dr Reddy's Laboratories Ltd., Hyderabad, India, for the generous donation of drug samples used in this study.

References

1 K. Hornik, *Neural Netw.*, 1991, **4**, 251–257.

2 D.E. Rumelhart, G.E. Hinton and R.J. Williams, *Distributed Parallel Processing: Explorations in the Microstructures of Cognition,* (D.E. Rumelhart and J.L. MacClelland, eds.), vol. 1, MIT Press, Cambridge, MA, USA, 1986, pp. 318–362.

3 L. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms and Applications,* Prentice-Hall, NJ, USA, 1994, pp. 1–330.

4 R.J. Schalkoff, *Artificial Neural Networks,* McGraw-Hill, New York, USA, 1997, pp. 1–182.

5 S.D. Brown, S.T. Sum, F. Despagne and B.K. Lavine, *Anal. Chem.*, 1996, **68**, 21R–61R.

6 B.K. Lavine, *Anal. Chem.*, 1998, **70**, 209R–228R.

7 J. Zupan, *Acta. Chim. Slov.*, 1994, **41**, 327–352.

8 J. Zupan, M. Novic and I. Ruisanchez, *Chemom. Intell. Lab. Syst.*, 1997, **38**, 1–23.

9 F. Despagne and D.L. Massart, *Analyst,* 1998, **123**, 157R–178R.

10 J. Zupan and J. Gasteiger, *Neural Networks in Chemistry and Drug Design,* 2nd edn., Wiley-VCH, New York, USA, 1999, pp. 1–154.

11 S. Agatonovic-Kustrin and R. Beresford, *J. Pharm. Biomed. Anal.*, 2002, **22**, 717–727.

12 C. Yin, Y. Shen, S. Liu, Q. Yin, W. Guo and Z. Pan, *Comput. Chem.*, 2001, **25**, 239–243.

13 Y. Ni, C. Liu and S. Kokot, *Anal. Chim. Acta,* 2000, **419**, 185–196.

14 G. Absalan and M. Soleimani, *Anal. Sci.,* 2004, **20**, 879–882.

15 C. Balamurugan, C. Aravindan, K. Kannan, D. Satyanarayana, K. Valliappan and R. Manavalan, *Indian J. Pharm. Sci.*, 2003, **65**, 274–278.

16 D. Sathyanarayana, K. Kannan and R. Manavalan, *Indian J. Pharm. Sci.*, 2004, **66**, 745–752.

17 S. Hashem, *Neural Networks,* 1997, **10**, 599–614.

18 M.P. Perrone and L.N. Cooper, *Artificial Neural Networks for Speech and Visi*on, (R.J. Mammone, ed.), Chapman & Hall, New York, USA, 1993, pp. 126–142.

19 P. Sollich and A. Krogh, *Advances in Neural Information Processing Systems,* (D.S. Touretzky, M.C. Mozer and M.E. Hasselmo, eds.), volume 8, MIT Press, Cambridge, MA, USA, 1996, pp. 190–196.

20 L.K. Hansen and P. Salamon, *IEEE Trans. Pattern Analysis Machine Intelligence,* 1990, **12**, 993–1001.

21 A. Sharkey, *Combining Artificial Neural Nets: Ensemble and Modular Multi-net Systems,* Springer-Verlag, London, UK, 1999, pp. 1–30.

22 D.E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning,* Addison-Wesley, Reading, MA, USA, 1989, pp. 1–256.

23 L.-W. Chan, *Proc. IEEE Intern. Joint Conf. Neural Networks,* Washington, DC, USA, 1999, pp. 1393–1396.

24 J. R. Long, V. G. Gregoriou and P. J. Gemperline, *Anal. Chem.*, 1990, **62**, 1791–1797.

25 D. Satyanarayana, K. Kannan and R. Manavalan, *Acta Chim. Slov.*, 2005, **52**, 138–144.

26 M.T. Hagan and M.B. Menhaj, *IEEE Trans. Neural Networks,* 1994, **5**, 989–993.

27 D. Nguyen and B. Widrow, *Proc. Intern. Joint Conf. Neural Networks,* San Diego, CA, USA, 1990, **3**, 21–26.

28 A. Chipperfield, P. Fleming, H. Pohlheim and C. Fonseca, *Genetic Algorithm Toolbox, Version 1.2,* Department of Automatic Control and Systems Engineering, University of Sheffield, UK, 1994.

29 R. G. Brereton, *Analyst,* 2000, **125**, 2125–2154.

30 Z-H. Zhou, J. Wu and W. Tang, *Artificial Intelligence,* 2002, **137**, 239–263.

31 A. Krogh and J. Vedelsby, *Advances in Neural Information Processing Systems,* (G. Tesauro, D.S. Touretzky and T. Leen, eds.), vol. 7, MIT Press, Cambridge, MA, USA, 1995, pp. 231–238.

32 D.W. Opitz and J.W. Shavlik, *Connection Science,* 1996, **8**, 337–354.

33 R. Caulcutt, *Statistics in Research and Development,* Chapman & Hall, London, UK, 1983, pp. 240–260.

34 A.J.C. Sharkey, N. E. Sharkey and G.O. Chandroth, *Neural Computing and Applications,* 1996, **4**, 218–227.