Correction of Temperature Variations in Kinetic-Based Determinations by Use of Pruning Computational Neural Networks in Conjunction with Genetic Algorithms

César Hervás*

Department of Computer Science, University of Córdoba, E-14004 Córdoba, Spain

José Antonio Algar and Manuel Silva

Department of Analytical Chemistry, Faculty of Sciences, University of Córdoba, E-14004 Córdoba, Spain

Received September 16, 1999

The joint use of genetic algorithms and pruning computational neural networks is shown to be an effective means for selecting the number of inputs required to correct temperature variations in kinetic-based determinations. The genetic algorithm uses a pruning procedure based on Bayesian regularization and is highly efficient as a feature selector; it provides quite good results in the generalization process without the need to use a validation set. The fitness function is defined as the sum of two subfunctions: one controls the learning ability of the network and the other its complexity. The training, pruning, and generalization processes were initially tested with simulated data in order to acquire preliminary information for the ensuing work with real data. The performance of the proposed method was assessed by applying it to the determination of the amino acid L-glycine by its classical spectrophotometric reaction with ninhydrin. A straightforward network topology including temperature as input (40+T:2:1 with 19 connections after the pruning process) was used to estimate the L-glycine concentration from kinetic curves affected by temperature variations over the range 60−75 °C, using kinetic data acquired up to only 1.5 half-lives. The trained network estimates this concentration with a standard error of prediction for the testing set of ca. 8%, which is much smaller than those provided by a classical parametric method such as nonlinear regression (even if kinetic data acquired at longer half-lives are used). Finally, a kinetic interpretation of the pruning process is provided in order to better demonstrate its potential for kinetic analysis.

INTRODUCTION

Kinetic methods of analysis are currently major choices available to the analytical chemist for addressing individual and joint determinations of species in a wide variety of real samples.[1] Data processing is a relevant step of kinetic-based determinations on account of the large amount of information provided by analytical instrumentation, information that can currently be processed most readily thanks to the low cost of powerful computer systems. Thus, new data-processing methods are useful for the treatment of complex kinetic processes and hence closer to real problems, even though they rely on more complex mathematical algorithms. The marriage between kinetics and chemometrics is consolidating day after day.[2] Computational neural networks (CNNs) have lately played a prominent role in kinetic analysis, where they have been successfully used to estimate kinetic analytical parameters[3] and to resolve nonlinear multicomponent systems.[4,5]

The development of measurement/data-processing methods to improve the ruggedness of kinetic-based determinations—error-compensated methods—is a major but still scarcely explored field for kinetic analysis. These methods overcome the experimental deficiencies of kinetic methods relative to equilibrium methods and are thus intended to reduce the strong dependence of experimental variables (temperature, pH) on methods based on time-dependent signals. In these approaches, data collected in the transient region of the process are used to compute the response that would be measured if the process were to be monitored to equilibrium.[6] These systems pose two main drawbacks: first, kinetic data must be acquired over a substantial reaction time domain (usually more than three half-lives, $t_{1/2}$, are required for implementation); second, they rely on a fixed kinetic model, so the corresponding differential equations must be known. These problems can be overcome by using powerful chemometric tools. We chose to use CNNs in this work to improve the ruggedness of kinetic-based determinations affected by variations in temperature taking into account the suitability for this type of kinetic problem.

Multilayer feed-forward CNNs based on different versions of a back-propagation (BP) learning algorithm have been used by several authors as highly powerful tools to solve a great variety of problems in analytical chemistry.[7−10] These models, designed for regression or classification, must be trained using formal stabilization by adding a penalty term to the original error function of the BP algorithm; this raises the complexity of the model. On the other hand, current pruning techniques decrease network complexity by modifying not only connection weights but also the network structure during the training step. Thus, unnecessary nodes or connections are gradually removed from the network

* Corresponding author. Phone: +34-957-218349. E-mail: ma1hemac@uco.es.

10.1021/ci9901284 CCC: $19.00
Published on Web 03/09/2000

architecture.[11,12] Pruning techniques combine the advantages of training large networks (i.e., learning speed and avoidance of local minima[13]) and those of training small ones (i.e., improved generalization[14]); however, removal of weights is highly sensitive to some pruning parameters on which the standard error of prediction (SEP) for the training and testing sets, and network size, depend on them. To solve these problems, we used genetic algorithms associated with weight regularization pruning. This association allows one to construct a genetic algorithm that uses a population of two-layer networks with enough nodes in the hidden layer. This simple structural topology provides a hill-climbing method amenable to trapping at structural local minima;[15,16] however, it is simpler, computationally inexpensive, and easier to interpret.

This work deals with the use of pruning CNNs in conjunction with genetic algorithms to correct temperature variations affecting kinetic data acquired after a short reaction time and compares the results obtained with those provided by standard CNNs in order to justify use of the pruning process. The strong and weak points of the proposed approach are examined in light of computer-generated synthetic data and experimental data obtained by monitoring the classical spectrophotometric reaction between L-glycine and ninhydrin.

THEORY

Our approach to network minimization involves successively removing weights after the network has been trained to satisfactory performance. This learning process is carried out by using the back-propagation learning procedure EDBD,[15] where the pruning algorithm is independent of the particular training procedure. The steps involved in running the algorithm are as follows:

1. Creating an initial random population. The networks used are encoded so as to create an initial population of N_P networks, starting with an overparametrized network and considering some copies of this network as having variable initial weights and pruning parameters but the same architecture. The neural network is coded or represented by a local scheme[16] where weights and bias are encoded in a string of real numbers.

2. At every iteration, a new population is formed using genetic operators of reproduction, crossover, and mutation over the old population. Each member of the population is decoded as a feedforward network and evaluated. The purpose of the reproduction operator is to create a new population based on the evaluation (fitness) of the networks in the old population. We used the roulette wheel selection scheme[17] with fitness linear scaling[18] to avoid premature convergence. Wheel selection is a fitness-proportional procedure that often places too much emphasis on the "exploitation" of highly fit strings at the expense of "exploration" of other regions of the search space. The purpose of fitness scaling is to control the number of copies that the members with high fitness values will receive in the next population. In conjunction with this selection mechanism, we used two heuristics, viz., the elitism strategy and the generation gap.[19] The elitism strategy forces the genetic algorithm to retain some of the best individuals in each generation. Such individuals can be lost if they are not selected to reproduce or if they are destroyed by crossover or mutation. In our algorithm, we introduced the best individual in each generation. The generation gap (G_p) is a parameter for the control of the percentage of the population to be replaced during each generation. In this way, $N_P(1 - G_p)$ members are chosen at random to survive unmodified in the next generation. The loss of critical alleles due to selection pressure results in a lack of diversity in the population and in premature convergence on local optima. This can be avoided by using a crossover operator. The crossover procedure involves choosing both network elements to be crossed with P_{cross} probability. We used a crossover-first-layer-nodes operator.[20,21] This method is a particular case of parametrized uniform crossover[22] in which an exchange takes place with a probability P_{recom} at each bit. Next, we used a mutation operator that affords random searching. These operators change groups of weights feeding into a node. This is called mutate-nodes.[21] The mutation is produced for every individual with a probability P_{mut}. If mutation occurs, it is affected by the addition of a random quantity to each weight set, following this pattern

$$w_j' = w_j + 2(\xi - 0.5)R_m \qquad j=1, ..., n_W \tag{1}$$

where w_j is the jth weight, ξ a uniform random variable in [0, 1], R_m a parameter, and n_W the number of weights.

3. Fitness evaluation. We defined the following modified error function[23]

$$E_W = \frac{1}{2} n_T \log \sum_{p=1}^{n_T} (y_p - o_p)^2 + \lambda_{EA} n_W \log \sum_{k=1}^{n_W} |w_k| \tag{2}$$

where the first term on the right-hand side is the sum of the squared errors between the actual output values (o_p) and the target output values (y_p) and n_T is the number of patterns for the training set. The second term represents the complexity of the network as a function of the absolute value of the weight magnitudes, w_k, where n_W is the number of weights. The evaluated function is based on the hypothesis that the "a priori" distribution of network weights follows a Laplacian distribution, so the initialization of the weights is as follows

$$w_j = \pm c \log(\eta) \tag{3}$$

where η is a uniform random variable over the interval [0, 1], c is a scale parameter, and the sign is chosen at random. The adaptive parameter λ_{EA} is a weighting factor that determines the significance of network complexity against network performance on the training set, thereby stimulating the reproduction of good trainable networks with a small number of connections. Several authors have developed procedures for determining the λ_{EA} adaptively during the training process.[24]

4. The connections associated with weights that have not been previously pruned are deleted once a present number of training epochs have been reached and the algorithm converged provided the next two conditions are fulfilled[23]

$$|w_{ij}| < \text{ghiw}(\bar{w}_a/(1 + S_{w_a}^2)) \tag{4}$$

$$\left|\frac{\partial E_W}{\partial w_{ij}}\right| < g\text{hider}(\bar{d}_w/(1 + S_{d_w}^2)) \tag{5}$$

Table 1. Parameter Values Used by the Algorithm

algorithm	parameters
training EDBD[15]	$\theta = 0.7$, $\kappa_\eta = 0.095$, $\gamma_\eta = 0.1$, $\phi_\eta = 0.1$, $\kappa_\mu = 1.0$, $\gamma_\mu = 0.05$, $\phi_\mu = 0.01$
pruning	ghiw = 1.1, ghider = 1.1
genetic	$G_p = 0.9$, $P_{cross} = 0.8$, $P_{recomb} = 0.5$, $P_{mut} = 0.1$, $R_{mut} = 0.1$, $\gamma = 0.9$, maxgen = 200, $N_P = 100$, $\lambda_{EA} = 0.6$

where $\bar{w}_a$ and $S^2_{w_a}$ are the mean and variance of the absolute values of the network weights, $\bar{d}_w$ and $S^2_{d_w}$ are the mean and variance of the absolute values of the derivatives of network weights, d_w, and ghiw and ghider are heuristic parameters to be determined.

5. To justify the resulting increase in the error due to the weight removal, we trained the pruned network for a number of additional epochs, using the EDBD algorithm. This pruning-learning process can be done once, not at all, or twice so as to avoid premature removal of weights.

6. The convergence of the genetic algorithm was determined by the following stop decision rule:

If $\Delta\bar{F}_h/\Delta\bar{F}_{h-1} < 1$, or the genetic algorithm reaches a maxgen number of generations.

$\Delta\bar{F}_h$ is the average improvement at step h, defined by the recursive equation

$$\Delta\bar{F}_h = \gamma\Delta\bar{F}_{h-1} + (1 - \gamma)\,\Delta F_h \qquad (6)$$

ΔF_h is the improvement at the hth generation, defined as the average fitness at the hth generation over the average fitness at the $(h - 1)$th generation, i.e., the ratio of the average fitness of each individual in generation h to the average fitness in generation $h - 1$. γ is a constant, usually chosen heuristically and very close to 1.0.

EXPERIMENTAL SECTION

Computations were performed by using the EDBD rule. Sigmoidal and linear functions were used for hidden and output layers, respectively. The algorithm software, written in C language, was run on an IRIS Release 6.5 in an Origin 2000. Table 1 shows the parameter values used to ensure optimal application of the algorithm.

Experimental kinetic data were obtained by spectrophotometric monitoring of the pseudo-first-order reaction between L-glycine and ninhydrin at different temperatures. Thus, an overall set of 112 real kinetic data containing uniformly distributed amino acid concentrations over the range $(9.0 \times 10^{-6}) - (7.2 \times 10^{-5})$ M were prepared by using the following procedure: 1.5 mL of ninhydrin reagent (9.3×10^{-2} M in ninhydrin and 1.3×10^{-2} M in hydrindantin, prepared in methylcellosolve), 0.5 mL of 0.8 M sodium acetate/acetic acid buffer (pH 5.0), and 2.0 mL of distilled water were placed in a 4.0 mL thermostated reaction vessel. The reaction was started by adding different amounts of L-glycine (in the above-mentioned range); then, the reaction mixture was briefly stirred. The time between the reaction and the beginning of data collection (1 min) was the same in all experiments. Kinetic runs were obtained at fixed temperature ranging from 60 to 75 °C, and absorbance vs time readings were monitored at 572 nm at 6 s intervals for 350 s. We used an instrumental setup consisting of a Metrohm 662 spectrophotometer furnished with an immersion probe that was interfaced to a Pentium 100-MHz PC-compatible computer equipped with a PC-Multilab PCL-812PG 12-bit analog-to-digital converter.

RESULTS AND DISCUSSION

Preliminary Studies. The use of computer-generated synthetic data to implement chemometric approaches is a very interesting alternative to obtaining a prior knowledge on their performance in the problem at hand. Although these data provide only approximate results, it may be of interest with the intent to minimize computational and laboratory costs in the ensuing work with real data. This option was used to study the kinetic problem addressed in this work, namely, the correction of variations in the estimation of the total signal change (S_∞)—proportional to the analyte concentration—in first-order reactions due to changes in temperature from kinetic data acquired after a short reaction time. The integrated rate equation for this type of reaction is

$$S_{t_j} = S_\infty(1 - e^{-kt_i}) \qquad (7)$$

where S_{t_i} is the signal at time t_i and k the first-order rate constant. On the other hand, the relationship between reaction energetics and chemical kinetics is described by various equations, the simplest of which is the Arrhenius equation:

$$k = Ae^{-E_a/RT} \qquad (8)$$

A is the collision frequency, E_a the Arrhenius activation energy, R the gas constant, and T the temperature. This relation accurately describes changes in rate constant with temperature. The combination of both equations allows one to determine S_∞ at variable reaction temperatures by using kinetic data once A and E_a are known.

To generate a simulated kinetic data set closer to a real one, we carried out a preliminary kinetic study to establish the dependence on temperature of the rate constant in the reaction model used; viz., the pseudo-first-order reaction of L-glycine with ninhydrin. Thus, seven L-glycine samples containing an amino acid concentration of 1.5×10^{-5} M were reacted with ninhydrin by using the procedure described in the Experimental Section while temperature was changed over the range 50−80 °C. By plotting ln k (estimated from the kinetic curve by using nonlinear regression) against the reciprocal of the absolute temperature, based on Arrhenius equation, the collision frequency ($A = 6.6 \times 10^6$ s) and activation energy ($E_a = 61.2$ kJ mol^{-1}) were obtained on the assumption that $R = 8.31$ J K^{-1} mol^{-1}. Using these parameters, we generated an overall set of 112 simulated kinetic data with uniformly distributed concentrations of L-glycine and temperatures over the ranges $(9.0 \times 10^{-6}) - (7.2 \times 10^{-5})$ M and 65−75 °C, respectively. An amount of synthetic noise similar to the instrumental background signal was added to these data. A similar real data set was prepared in the laboratory by using the procedure described in the

Table 2. Effect of the Reaction Time Region Used for the Estimation of S_∞ in Kinetic-Based Determinations Affected by Temperature Variations Using NLR

reaction time region, $t_{1/2}$	standard error of prediction, %	
	simulated data	real data
0.7	47.3	74.0
1.0	10.2	33.4
1.5	3.2	22.1
2.0	2.5	15.1

Experimental Section. Both data sets were split into two groups, each randomly comprising 80% of the elements for training and 20% for generalization, respectively.

The reaction time region used to estimate S_∞, or the analyte concentration, was an important aspect in this study. Thus, reported methods for improving the ruggedness of kinetic-based determinations use a substantial reaction time domain—more than three half-lives—for implementation.[6] Methods using shorter reaction time are therefore required in order to reduce time and hence costs. In this work, we explored three reaction time regions corresponding to 0.7, 1.5, and 2.0 $t_{1/2}$; taking into account that signal vs time readings were acquired at 6 s intervals, the above-mentioned time domains corresponded to a number of signal data S_{t_i} of 20, 40, and 57, respectively, which were used as input to the CNNs.

As stated above, CNNs possess a high potential in kinetic analysis if properly used. In fact, their advantages over alternative methods (e.g., statistical parametric methods) in solving specific kinetic problems must be unequivocally established. The classical parametric nonlinear regression (NLR) method is a useful choice for estimating the total signal change in first-order reactions as the kinetic model is known. However, some deviations can be found in practice that entail processing experimental data at several half-lives in order to obtain accurate results. In addition, the problem addressed in this work is more complex because the kinetic profile depends on temperature variations whereas S_∞ remains virtually constant (kinetic vs equilibrium behavior) at a fixed analyte concentration. It is therefore of interest to evaluate the performance of the NLR method in correcting temperature variations in kinetic-based determinations. For this purpose, simulated and real data acquired in different reaction time regions (0.7−2.0 $t_{1/2}$) were processed by NLR. Table 2 gives the results obtained for both data sets, as SEP, calculated from

$$\mathrm{SEP} = \frac{100}{\bar{S}_i}\sqrt{\frac{\sum_{i=1}^{n}(S_i - \hat{S}_i)^2}{\mathrm{n}}} \qquad (9)$$

where S_i and $\hat{S}_i$ are the found and expected values for S_∞, respectively, $\bar{S}_i$ is its average value; and n is the number of patterns used ($n = 112$). The expected S_∞ values for the real data set were obtained by allowing the reaction to proceed for a long time. As can be seen, SEP value increased with decrease in the reaction time used to estimate S_∞. The dependence was stronger for real data: errors were ca. 20%, even if an appreciable reaction time region (2.0 half-lives) was used.

From these results it follows that the NLR method is not a good choice for solving this kinetic problem, so a more powerful chemometric approach (CNNs) was adopted. To decrease network complexity and hence minimize the number of training patterns needed to obtain low enough SEP values, standard CNNs were pruned by using genetic algorithms and the performance of both choices was compared.

CNN Experiments with Simulated and Real Data. Initially, simulated data were used to study the behavior of standard and pruning CNNs—obtained by using genetic algorithms—to correct temperature variations in kinetic-based determinations. Thus, the previous data set (112 samples) was split into two groups comprising 80% of the elements for training and 20% for testing, which will henceforth be designated (train/test: 90/22). The network topologies tested in all experiments (over five runs) were as follows: (1) the input layer consisted of a number of S_{t_i} values (neurons) dependent on the time domain used (viz., 0.7, 1.5, and 2.0 $t_{1/2}$, which correspond to 20, 40, and 57 inputs, respectively). Inclusion of the temperature in this layer was also considered in order to examine its influence on network performance. (2) One or two nodes in the hidden layer were tested. (3) The output layer consisted of one node, the estimated S_∞ value for the analyte concentration (i.e., the L-glycine concentration in the sample).

As can be seen from Table 3, pruning is an effective way of reducing network complexity with a network topology where the temperature is included in the input layer and two nodes are used in the hidden layer. Under these conditions, the time domains studied slightly affected percent SEP for the testing process (3.4−5.6%). Figure 1 shows the effect of temperature, number of nodes in the hidden layer, and time domain used on the generalization ability of pruning CNNs. As expected, percent SEP for the testing set decreased with increase in reaction time. On the other hand, the effect of temperature was more pronounced than that of the number of nodes in the hidden layer as time domain decreased. On the basis of these results, the network topology used to deal with real data should include the temperature in the input layer and two nodes in the hidden layer.

An overall 112 kinetic curves were recorded at different concentrations of L-glycine [$(9.0 \times 10^{-6}) - (7.2 \times 10^{-5})$ M] and temperatures (60−75 °C) under the experimental conditions described in the Experimental Section. Kinetic data acquired in different time domains (0.7, 1.5, and 2.0 $t_{1/2}$) were processed by CNNs with different topologies (dependent on the reaction time region used) and relative sizes of the training and testing sets. The results are shown in Table 4. As with simulated data, pruned CNNs provided results (percent SEP for the testing set) similar to those of standard ones. The 1.5 $t_{1/2}$ time domain (network topology 40+T:2:1) was seemingly the most suitable, with no appreciable detriment to accuracy. With this network topology, the performance of the CNN is not significantly affected by the relative size of the training and testing sets.

Selection of Network Topology and Data Set Size. To choose appropriate values for these variables on the basis of statistical and practical considerations, analysis of variance (ANOVA)—a technique typically used to ascertain the statistical significance of observed differences among means—[25,26] was applied to the results provided by CNNs. The ANOVA involved the use of a linear regression model where the percent SEP values for the training and testing sets were the dependent variables, and network topology was

Table 3. Accuracy of the Algorithm Used with Various Network Topologies as Applied to Simulated Data[a]

network topology	mean			SD		
	SEP train	SEP test	connections	SEP train	SEP test	connections
57+T:2:1[b]	3.41	4.80	121.0	0.37	0.37	0.0
40+T:2:1[b]	3.42	5.35	87.0	0.62	1.24	0.0
20+T:2:1[b]	3.49	5.34	47.0	0.20	0.30	0.0
57+T:2:1	3.52	4.60	18.3	0.14	0.25	4.4
40+T:2:1	3.68	5.17	16.0	0.23	0.23	1.3
20+T:2:1	4.07	5.58	12.0	0.22	0.40	0.6
57+T:1:1	7.29	10.06	29.0	0.43	0.71	7.4
40+T:1:1	9.96	13.33	18.6	0.45	0.66	6.7
20+T:1:1	11.91	15.01	12.0	0.51	0.64	4.4
57:2:1	7.46	9.02	29.6	0.21	0.85	16.6
40:2:1	9.64	12.05	32.4	1.98	1.83	6.8
20:2:1	18.48	21.86	23.6	2.14	1.98	4.6
57:1:1	7.36	8.85	29.8	0.30	0.63	6.5
40:1:1	8.96	10.37	25.0	1.33	1.33	5.3
20:1:1	19.26	20.64	13.8	2.30	3.14	3.5
network topology	best			worst		
	SEP train	SEP test	connections	SEP train	SEP test	connections
57+7:2:1[b]	2.96	4.34	121	3.84	5.25	121
40+T:2:1[b]	2.96	4.45	87	4.19	7.05	87
20+T:2:1[b]	3.28	5.04	47	3.74	5.74	47
57+T:2:1	3.38	4.25	16	3.48	5.06	18
40+T:2:1	3.35	4.79	16	3.87	5.34	16
20+T:2:1	3.87	4.92	12	3.88	6.03	13
57+T:1:1	6.55	8.91	35	8.00	11.15	34
40+T:1:1	9.61	12.43	17	10.03	14.21	14
20+T:1:1	11.18	14.02	14	12.25	15.74	14
57:2:1	7.38	8.47	30	7.62	10.49	14
40:2:1	8.86	10.43	38	13.34	15.22	28
20:2:1	15.65	10.88	25	20.30	24.64	16
57:1:1	7.14	7.86	32	8.02	10.01	35
40:1:1	8.12	9.22	30	11.5	12.90	16
20:1:1	16.82	17.69	17	22.54	26.29	7

[a] The results are averages of five runs. Mean, SD, best, and worst indicate the mean value, standard deviation, and best and worst value, respectively. [b] Standard CNNs.

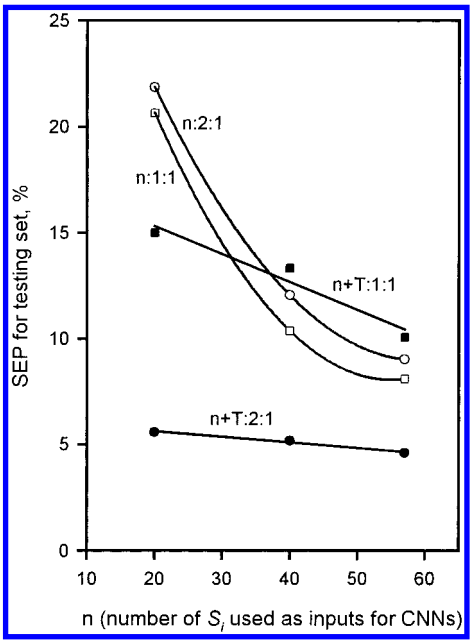

Figure 1. Influence of network topology on the percent standard error of prediction for the testing set obtained by using simulated data.

the independent variable. The results of the one-way ANOVA (percent SEP for the training and testing sets) are shown in Table 5. As can be seen, there was a significant statistical dependence on network topology [*p*-values were much smaller than the standard significance level ($\alpha = 0.05$)] on account of the inputs (time domain) and relative size of the training and testing sets. The purpose of this study was thus to rank percent SEP values in order to choose, on the basis of statistical considerations, the most suitable network topology (viz., that providing the smallest percent SEP for the training and testing sets).

On the basis of the mean percent SEP values for the training and testing sets listed in Table 4, topologies were ranked as shown in Table 6. These significant values must be interpreted carefully since multiple comparisons between the same data must be made. We used many comparison tests, based on the post-hoc method: the Student−Newman−Keuls (SNK) test. Table 6 shows the homogeneous subsets provided obtained by the SNK test. Considering $\alpha = 0.05$, it is smaller than the *p*-value for each topology subset, so these are homogeneous (i.e., there is no significant difference between the mean percent SEP values for each subset). From the results of SNK test it follows that the 57+T:2:1 (90/22) standard CNN for the training process is the best topology as it provides a percent SEP value significantly smaller than those yielded by the other topologies. The second and third best topologies are the pruned network topologies having 57+T and 40+T inputs, respectively. Virtually the same results are obtained by using the latter topology with 56, 78, or 90 patterns for the training set. With regards to the

Table 4. Accuracy of the Algorithm Used with Various Network Topologies and Data Set Sizes as Applied to Real Data[a]

network topology	data set train/test	mean			SD		
		SEP train	SEP test	connections	SEP train	SEP test	connections
57+T:2:1[b]	90/22	5.12	6.79	121.0	0.14	0.08	0.0
40+T:2:1[b]	90/22	5.70	8.30	87.0	0.50	0.30	0.0
20+T:2:1[b]	90/22	6.53	10.40	47.0	0.18	0.38	0.0
57+T:2:1	90/22	5.65	8.13	29.4	0.28	0.41	13.5
40+T:2:1	90/22	6.14	8.57	19.4	0.36	0.60	3.0
20+T:2:1	90/22	7.74	10.50	12.6	0.11	0.28	2.0
40+T:2:1	78/22	5.97	8.92	16.2	0.39	0.82	5.0
40+T:2:1	56/22	5.75	9.90	18.6	0.43	0.79	2.7
40+T:2:1	78/34	6.11	7.96	19.0	0.38	0.13	10.4
network topology	data set train/test	best			worst		
		SEP train	SEP test	connections	SEP train	SEP test	connections
57+T:2:1[b]	90/22	4.97	6.67	121	5.36	6.92	121
40+T:2:1[b]	90/22	5.27	7.93	87	6.40	8.68	87
20+T:2:1[b]	90/22	6.40	13.5	47	68.85	14.22	47
57+T:2:1	90/22	5.39	7.80	18	5.43	8.92	26
40+T:2:1	90/22	5.93	7.98	22	5.56	9.63	20
20+T:2:1	90/22	7.54	10.05	9	7.79	10.85	10
40+T:2:1	78/22	5.92	7.87	23	5.44	9.89	17
40+T:2:1	56/22	6.53	9.14	18	5.84	11.04	20
40+T:2:1	78/32	5.52	7.75	18	6.37	8.15	9

[a] For the meaning of abbreviations, see Table 3. [b] Standard CNNs.

Table 5. Percent SEP Results for the Training and Testing Sets Obtained by ANOVA[a]

	percent SEP for training set					percent SEP for testing set				
source	SS	DF	MS	*F*-value	*p*-value	SS	DF	MS	*F*-value	*p*-value
topologies	21.762	8	2.720	20.576	0.000	60.723	8	7.590	26.935	0.000
error	4.795	36	0.132			10.145	36	0.282		
total	26.521	44				70.868	44			

[a] SS = total sum of squares; DF = degrees of freedom; MS = mean square.

Table 6. Homogeneous Subsets of Network Topologies Provided by the SNK Test

	percent SEP for training test						percent SEP for testing test				
		subsets for $\alpha = 0.05$						subsets for $\alpha = 0.05$			
topology	(train/test)	1	2	3	4	topology	(train/test)	1	2	3	
57+T:2:1[a]	(90/22)	5.118				57+T:2:1[a]	(90/22)	6.794			
57+T:2:1	(90/22)		5.648			40+T:2:1	(78/34)		7.964		
40+T:2:1[a]	(90/22)		5.704			57+T:2:1	(90/22)		8.130		
40+T:2:1	(56/22)		5.748			40+T:2:1	(90/22)		8.300		
40+T:2:1	(78/22)		5.966	5.996		40+T:2:1[a]	(90/22)		8.566		
40+T:2:1	(78/34)		6.114	6.114		40+T:2:1	(78/22)		8.918		
40+T:2:1	(90/22)		6.136	6.136		40+T:2:1	(56/22)			9.904	
20+T:2:1[a]	(90/22)			6.53		20+T:2:1[a]	(90/22)			10.398	
20+T:2:1	(90/22)				7.744	20+T:2:1	(90/22)			10.500	
p-value		1.000	0.299	0.085	1.000			1.000	0.054	0.192	

[a] Standard CNNs.

testing process, the best topology is again the 57+T:2:1 (90/22) standard CNN, followed by 57+T with pruned CNNs and 40+T with or without pruning (using 78/34 or 90/22 as relative data sets).

On the basis of these results, is it necessary and useful to prune standard CNNs in order to solve the kinetic problem addressed? We believe so because the best topology provided by the SNK test requires 57+T inputs (to acquire kinetic data up to 2 $t_{1/2}$ over 342 s) in order to obtain a SEP value of 6.8% for the testing process. In addition, this topology requires the estimation of 121 connections by using a training set with 90 patterns, which leads to less reliable estimates and raises the computational cost. On the other hand, the use of pruned CNNs with a network topology of 40+T inputs only requires the acquisition of kinetic data up to 1.5 $t_{1/2}$ (over 240 s) and a smaller training set (78 patterns are enough) to obtain similar % SEP values, about 8%. This slight increase in percent SEP can be ascribed to the fact that only 19 connections are needed, so estimates are more reliable and the computational cost lower, at the expense of less homogeneous results: SD values from 0.13 to 0.60 for the testing set with 34 and 22 patterns, respectively.

Kinetic Interpretation of the Pruning Process. Once the network topology has been chosen, it is interesting to relate the inputs selected by the pruning process to the kinetic information they provide. In fact, the 40+T:2:1 pruning

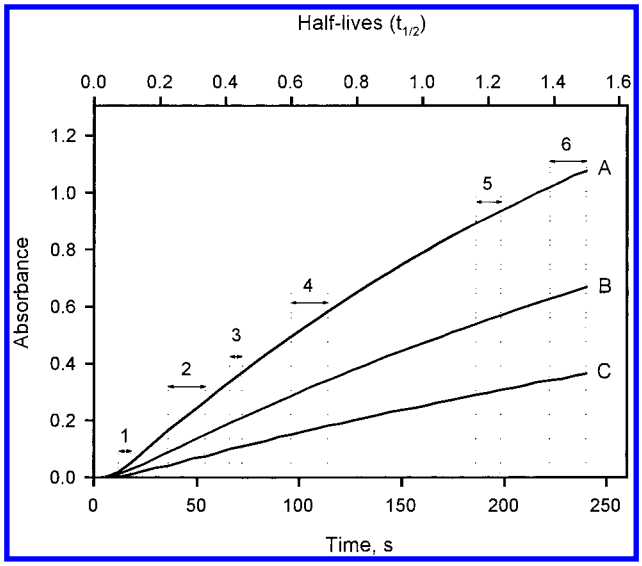

Figure 2. Typical absorbance vs time profiles obtained at different L-glycine concentrations and temperatures. The selected zones (1−6) provided by the pruning algorithm for acquisition of kinetic data to be used as input to the CNNs are shown. L-glycine concentration: 6.0, 4.8, and 2.7 × 10^{-6} M in A, B, and C, respectively. Temperature: 70, 65, and 60 °C in A, B and C, respectively.

network topology uses only ca. 19 connections distributed over the selected time domain (1.5 $t_{1/2}$) as input to the CNNs. This simpler network topology implies that only 7−8 values of S_{t_i} (kinetic data), in addition to temperature, are used as input for the CNNs. Figure 2 illustrates the selection of these kinetic data over the time domain used by the pruning algorithm. In fact, the reaction-time region is split into six zones by the algorithm, among which the seven to eight values of S_{t_i} are distributed. As can be seen, the zones 1−4 (up to ca. 0.7 $t_{1/2}$) provide more information for solving this kinetic problem; that provided by zones 5 and 6 (near 1.5 $t_{1/2}$) is also very useful, however.

At this point, it is of interest to consider earlier results of our research group regarding the use of CNNs in kinetic analysis. We had previously solved a similar but easier kinetic problem. In fact, our first use of CNNs in kinetic analysis was for the estimation of kinetic analytical parameters, specifically S_∞ after a short reaction time.[3] On the basis of kinetic data obtained at a fixed temperature over a time domain of only 0.66 $t_{1/2}$, a 5:3:1 network topology was heuristically optimized and used to estimate the analyte concentration with a SEP value for the testing set of only 2.14%. Obviously, the problem addressed in this work is more complex because, as stated above, variations in temperature provide different kinetic profiles but not a different S_∞ value at a fixed analyte concentration.

From these studies we can conclude that the inputs ascribed to zones 1−4 in Figure 2 may suffice to estimate S_∞ provided a constant temperature is used. There is a clear similarity between the number of inputs (4−5) and the time domain (0.6−0.7 $t_{1/2}$) where they were acquired in both studies. With regards to the problem addressed in this work, kinetic data from zones 5 and 6 and, especially, the use of temperature as input, provide the information required by the CNNs to discriminate among several kinetic curves all having the same S_∞ value but exhibiting slight differences in their kinetic profiles. In summary, pruned CNNs have been shown to possess a high potential for kinetic analysis and, in general, serve as an analytical tool for deriving quality information with substantial savings in time and in experimental and computational costs.

ACKNOWLEDGMENT

The authors thank Spain's Dirección General Interministerial de Ciencia y Tecnología (DGICyT) for funding this research within the framework of Project ALI98-0676-CO2-02 (Department of Computer Science, University of Córdoba) and Project PB96-0984 (Department of Analytical Chemistry, University of Córdoba).

REFERENCES AND NOTES

(1) Crouch, S. R.; Cullen, T. F.; Scheeline, A.; Kirkor, E. S. Kinetic determinations and some kinetic aspects of analytical chemistry. *Anal. Chem.* **1998**, *70*, 53R−106R.
(2) Pérez-Bendito, D.; Silva, M. Recent advances in kinetometrics. *Trends Anal. Chem.* **1996**, *15*, 232−240.
(3) Ventura, S.; Silva, M.; Pérez-Bendito, D.; Hervás, C. Artificial neural networks for estimation of kinetic analytical parameters. *Anal. Chem.* **1995**, *67*, 1521−1525.
(4) Ventura, S.; Silva, M.; Pérez-Bendito, D.; Hervás, C. Computational neural networks in conjunction with principal component analysis for resolving highly nonlinear kinetics. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 287−291.
(5) Hervás, C.; Ventura, S.; Silva, M.; Pérez-Bendito, D. Computational neural networks for resolving nonlinear multicomponent systems based on chemiluminescence methods. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 1119−1124.
(6) Pardue, H. L. Unified view of kinetic-based analytical methods with emphasis on ruggedness. A Review. *Analyst* **1996**, *121*, 385−390.
(7) Zupan, J.; Gasteiger, J. *Neural networks for chemists. An introduction*; VCH: Weinheim, 1993.
(8) Schulz, H. Neural networks in analytical chemistry—A new method or only a passing fashion? *GIT. Fachz. Lab.* **1995**, *39*, 1009−1010.
(9) Zupan, J.; Novic, M.; Ruisanchez, I. Kohonen and counterpropagation artificial neural networks in analytical chemistry. *Chemom. Intell. Lab. Syst.* **1997**, *38*, 1−23.
(10) Despagne, F.; Massart, D. L. Variable selection for neural networks in multivariate calibration. *Chemom. Intell. Lab. Syst.* **1998**, *40*, 145−163.
(11) Le Cun, Y.; Denker, J.; Solla, J. *Optimal brain damage. Advances in neural information processing systems 2*; Touretzky, S., Ed.; Morgan Kaufmann Publishers: San Mateo, CA, 1990; pp 598−605.
(12) Hassibi, B.; Stork, D. Second-order derivatives for networks pruning: optimal brain surgeon. In *Advances in neural information process systems 5*; Hanson, S. J., Cowan, J. D., Giles, C. L., Eds.; Morgan Kaufmann Publishers: San Mateo, CA, 1993; pp 164−171.
(13) Hush, D.; Horne, B. Progress in supervised neural networks. *IEEE Signal Process. Mag.* **1993**, *10*, 8−39.
(14) Castellano, G.; Fanelli A. M.; Pelillo M. An iterative pruning algorithm for feedforward neural networks. *IEEE Trans. Neural Networks* **1997**, *8*, 519−531.
(15) Minai, A. A.; Williams, R. J. *Back-propagation heuristics: A study of the extended delta-bar-delta*; IEEE International Joint Conference on Neural Networks, San Diego, CA, 1990; pp 595−600.
(16) Montana D.; Davis L. Training feedforward neural networks using genetic algorithms. Proceedings of the 11th International Joint Conference on Artificial Intelligence; Morgan Kaufmann Publishers: San Mateo, CA, 1989; pp 762−767.
(17) Holland, J. H., *Adaptation in natural and artificial systems*, second ed.; MIT Press: Cambridge, MA, 1992.
(18) Baker, J. E., Adaptive selection methods for genetic algorithms. Proceedings of the First International Conference on Genetic Algorithms; Morgan Kaufmann Publishers: San Mateo, CA, 1985; pp 101−111.
(19) Grefenstette, J., Optimization of control parameters for genetic algorithms. *IEEE Trans. Syst. Man Cybernetics* **1986**, *16*, 122−128.
(20) Bebis G.; Georgiopoulos M.; Kasparis, Takis. Coupling weight elimination with genetic algorithm to reduce network size and preserve generalization. *Neurocomputing* **1997**, *17*, 167−194.
(21) Miller, G.; Todd, P.; Hegde, S. Designing neural networks using genetic algorithms. Proceedings of the Third International Conference on Genetic Algorithms; Morgan Kaufmann Publishers: San Mateo, CA, 1989; pp 379−384.

(22) Spears, W. M.; De Jong, K. A. On the virtues of parametrized uniform crossover. Belew, R. K., Booker, L. B. Eds.; Proceedings of the Fourth International Conference on Genetic Algorithms; Morgan Kaufmann Publishers: San Mateo, CA, 1991; pp 230−236.
(23) Williams, P. M. Bayesian regularization and pruning using a Laplace prior. *Neural Comput.* **1995**, *7*, 117−143.
(24) Weigend, A. Rumelhart, D Huberman, B. Generalization by weight elimination with application to forecasting. *Adv. Neural Inform. Process. Syst.* **1991**, *3*, 875−882.
(25) Zar, J. H. *Biostatistical analysis;* Prentice Hall: Englewood Cliffs, NJ, 1974.
(26) Keppel, G. *Design of analysis;* Prentice Hall: Englewood Cliffs, NJ, 1991.

CI9901284