Hierarchical Neural Network Modeling for Infrared Spectra Interpretation of Modified Starches

Ludmila Dolmatova,[†] Valery Tchistiakov,[†] Cyril Ruckebusch,[‡] Nathalie Dupuy,[‡] Jean-Pierre Huvenne,*[,‡] and Pierre Legrand[‡]

Laboratory of Computer-Aided Molecular Design, Institute of Physiologically Active Compounds, Russian Academy of Sciences, 142432 Chernogolovka, Russia, and Laboratoire de Spectrochimie Infrarouge et Raman, LASIR, CNRS, Bât. C5, Ecole Universitaire d'Ingénieurs de Lille, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq Cedex, France

Received March 20, 1999

The goal of this research was to demonstrate a new approach to multilevel hierarchical modeling using neural networks. In a situation where the data set is small and error-prone, it is necessary to build multiple models of input−output relationships, combining these models into a hierarchical structure. This allows utilization of the best aspects of every model, eliminating the need to choose "the best one", when the general sample itself is not known exactly. At the first stage of modeling we used feed-forward artificial neural networks for spectra interpretation. Then at the second stage a simultaneous recurrent network was used to correct the predictions made at the first stage. This architecture enables enhancement of the generalization ability according to the similarity of the input patterns. The suggested method was applied to the interpretation of spectra of modified starches. This has definite practical value, since authentication of food is very important for the consumers and the food industry at each level of the food chain from raw materials to finished products. It is demonstrated that multilevel modeling offers accuracy advantages compared to choosing the best model or model averaging.

1. INTRODUCTION

We used feed-forward artificial neural networks (ANNs) with resilient propagation (RPROP) algorithm[1] for supervised learning on a training set (35 samples) which included native or modified starches of different origin. The goal of modeling was to build input−output relationships between infrared spectra of starches under study and the vectors of their origin and modifications. Several neural network models were built, two of which are discussed in this paper. The performance of trained nets was tested on the test set, which contained patterns not included in the training set (21 samples). The quality of recognition for the test set was satisfactory in both cases, with definite peaks corresponding to the observed modifications.

Nevertheless, we have explored the possibility of enhancing the quality of modeling using a second level of neural network modeling. At this level, the results of prediction made by both networks were used as the inputs of another neural network, a simultaneous recurrent network (SRN). The known vectors of origins/modifications for the training set were used as target outputs. According to this information the second level network was able to produce an error-correction procedure which allowed improvement of the overall quality of modifications and origin recognition.

Presently, there are no standards for the method of analysis of raw samples for the purpose of such identification; therefore we decided to use infrared (IR) spectra as the basic source of information. The mid-IR spectra have proved to contain enough information to permit successful classification and clustering of data.[2,3] Moreover, Fourier transform infrared (FT-IR) spectroscopy is faster and less expensive than other physical methods. During our previous research[4] we have studied the application of self-organizing ANNs and feed-forward ANNs to the problem of recognition of corn starch modifications. However, in that case most attention was paid to the recognition of the most important (P and W) modifications, and the problem was restricted to only one origin (corn). It was important to extend the possibility of modification recognition and analysis to data sets that contained starches of different origin—corn, potato, rice, etc. It was also necessary to model the relationships between infrared spectra and more complex sets of modifications—up to 3 at a time.

2. EXPERIMENTAL TECHNIQUE AND DATA MODELING METHODS

This work was performed on 56 starches of different origins: corn, potato, and rice. Some starches have been submitted to structure modifications. Properties, codes, and chemical structures of these treatments are described in Table 1. The 56 coded experimental data are then presented as training and test samples in Table 2. The samples were collected by CREALIS, and they represent the raw material of that research center.

2.1. Experimental Technique. The ATR spectra were obtained with a horizontal ZnSe crystal accessory. A reference spectrum of the ZnSe crystal was collected before

* To whom correspondence should be sent. Phone: 33 3 20 43 49 02. Fax: 33 3 20 43 67 55. E-mail address: Jean-Pierre.Huvenne@univ-lille1.fr.
[†] Russian Academy of Sciences.
[‡] Université des Sciences et Technologies de Lille.

10.1021/ci990442y CCC: $18.00
Published on Web 10/05/1999

Table 1. Definition of the Modifications Applied to the Starches under Study

property or modification	code	structure
hydroxypropyl distarch phosphate	R	$StOCH_2CHOHCH_3$, $StOPOSt$ (O, ONa)
waxy	W	pure amylopectine
pregelatinized	P	adduct of H_2O
acetylated distarch adipate	H	$StOC(CH_2)_4COSt$, $StOCCH_3$
acetylated distarch phosphate	J	$StOPOSt$, $StOCCH_3$
distarch phosphate	I	$StOPOSt$
nonmodified	N	amylose + amylopectine

Table 2. Definition of the Learning and Test Sets of Samples[a]

	no. of samples			no. of samples	
origin/ modification	in the training set	in the test set	origin/ modification	in the training set	in the test set
corn/N	2	2	rice/N	3	2
corn/W	4	2	rice/W	2	1
corn/RW	3	2	rice/WP	2	1
corn/RWP	2	2	rice/WJ	1	1
corn/WH	6	5	potato/N	1	
corn/WPH	4	2	potato/R	2	1
corn/P	1		potato/P	1	
corn/PI	1				

[a] Capital letters refer to Table 1.

processing of each sample. Briefly, the powder was impregnated by 0.5 mL of dry acetone, and the spectrum was recorded after total solvent evaporation. The addition of nondissolving liquid improves the refractive component of the sample by removal of air present in the powder. The spectra were recorded on a Fourier transform infrared Perkin-Elmer spectrometer (model PE 2000 g) run under a SPECTRUM operating system. Each sample was routinely recorded 3 times. We co-added 200 scans of symmetrical interferograms which are transformed in absorbance spectra in the 1800 to 700 cm^{-1} range with apparent resolution of 1.928 cm^{-1}. To minimize the spectral disturbance due to some residue of acetone, we delete three spectral regions 1697−1729, 1342−1372, and 1205−1225 cm^{-1}. Then the spectra were differentiated in order to eliminate the influence of base line variation. A nine point moving average smoothing was performed on the profiles; this avoids a decrease in the signal to noise ratio during the differentiation. The differentiation and the smoothing were performed simultaneously according to the algorithm developed by Savitzky and Golay.[5] These data transformations give rise to a final spectrum of 528 points. Fingerprint regions of the ATR spectra of some starch samples are shown in Figure 1. The spectra seem to be similar because the different samples contain the same main components (amylose, amylopectine) and the IR absorption is essentially sensitive to the changes in short-range structure in C−C and C−O stretching regions.[6]

2.2. Data Organization. Any measurement which initially contains N wavelengths can be reduced to a vector with $k < N$ features by projection on a smaller orthogonal subspace. Furthermore, any base of orthogonal functions can be selected to transform the data. For instance, the Hadamard transformation[7] enables decomposition of the signal in a series of block functions. An IR spectrum can be reconstructed by 2, 4, 8, ..., 2^n Hadamard coefficients, where n is an integer. For this purpose, the dimension of the vibrational spectra was lowered to 512 variables by truncation of the lowest wavenumbers. This data compression has enabled the reduction of the dimension of the input space from 512 to 64 coefficients. It was the highest degree of data compression which still allowed successful recognition of the origins and modifications of starch.

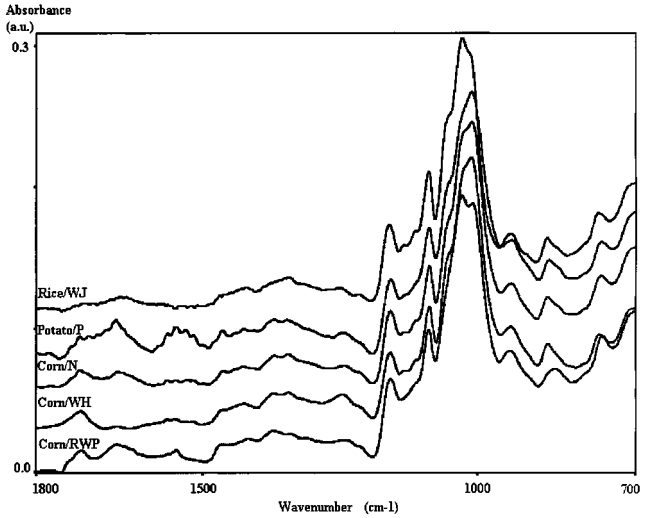

Figure 1. Infrared spectra of some starch samples in the informative region.

We represented each sample as a set of input−output data. The input data vector consisted of the 64 numerical values associated with the Hadamard coefficients. The nine-dimensional output vector was constructed as follows: the first three positions were used to represent the origin (corn, potato, or rice), and the next six positions indicated the modifications (R, W, P, H, J, or/and I) defined in Table 1. Since each modification was coded as a binary variable, the number of possible combinations of modifications was very large in our study: $2^6 = 64$. For all three origins the minimal number of samples sufficient to represent all possible combinations of origin and modifications was $3 \times 64 = 192$, more than the available amount of data.

Therefore, instead of predicting different class labels for all possible combinations, it seemed to be advantageous to predict the set of observed modifications directly from the spectra of starches with stand-alone modifications. Thus the nine-component output vectors were coded in the following way. First, these vector components were all set to zero. Second, we inserted a "1" into one of the first three positions corresponding to the origin. Third, we put a "1" into the rest of the position(s) corresponding to modification(s). The selection of the test samples has been done randomly among those samples that had at least one example of such modifications in the training set, as can be seen in Table 2.

2.3. Data Preprocessing. Artificial neural networks have proved themselves as a powerful and adaptable tool for representation of all kinds of dependencies: ⟨structure−activity⟩ or ⟨stimuli−response⟩, both qualitative and quantitative. When ANNs first appeared on the horizons of chemistry, their ease of use in comparison with traditional nonlinear methods gave rise to a widespread misconception of ANNs' omnipotence. This, in turn, encouraged some researchers to eliminate the preliminary treatment of input data (unfortunately, this kind of mistake happens also

nowadays). Extensive research in the field of neural networks resulted in a collection of rules and recommendations for the optimal choice of the dimensions of the vector of input data, the ratio of the number of data points to the number of adjustable parameters (weights), etc. These issues were discussed widely both from theoretical[8] and practical[9] points of view; however there is no strict solution of this problem, and people basically follow empirical procedures, according to the specifics of their data. In this study, since the ratio of samples/weights is rather critical, it is obvious that adjusting a few hundred parameters to the outputs of 35 objects carries a rather high amount of uncertainty even if specific algorithms are used. Nevertheless, one should keep in mind that we do not claim to predict completely unknown objects but samples which have their properties in the dynamical range spanned by the characteristics of the calibration objects. The ability to generalize, which is the capability we are looking for, expresses how neural networks can handle inputs which have not been learned, but which are similar to the set of inputs seen during the training session.

2.4. Choice of Neural Network Architectures, Learning Algorithm, and Error Function. Artificial neural networks represent massively parallel structures consisting of nonlinear processing nodes connected by links with different weights. Such structures can represent an arbitrary complex input−output mapping, as it was theoretically stated in Kolmogorov's theorem;[10] furthermore, the actual form of transfer functions for multilayer neural networks was concretized.[11] Therefore, ANNs are well-suited both for pattern recognition, classification, or clustering and for function approximation, thus allowing quantitative modeling. Feed-forward neural networks with learning algorithms mainly evolved from error back-propagation[12] are extremely useful for building complex relationships between input and output sets of data. Currently resilient propagation[1,13] seems to be the most efficient algorithm for supervised learning.[14] It is a local adaptive scheme where the signs of partial derivatives of error measure are used for determining the directions of weight update. We currently use the version of RPROP algorithm without restoring the previous weights after the jump over local minimum[13] implemented in the Stuttgart neural network simulator (SNNS).[15]

One of the useful features of the RPROP algorithm consists of the fact that it does not minimize the sum of least squares, but the more general function

$$E = \sum(t_i - o_i)^2 + 10^{-\alpha}\sum w_{ij}^2$$

where t_i means target (desired) output, o_i means calculated output, w_{ij} means weight of the link between neurons i and j, and α stands for weight decay coefficient. One can use this function in a straightforward manner, as was first suggested;[16] however there exists more advanced forms, which zero outweighs, that are only 1 standard deviation or so away from zero.[17] The weight decay provided by the modified error function leads to a systematic decrease of the unnecessary weights, thus pruning the network and reducing the number of adjustable parameters. It leads to a more structured network with strong connections between a smaller number of neurons. Such concentration of information is extremely helpful in the analysis of the resulting models. The adequate selection of weight decay coefficient can also increase the generalization ability of the networks.[18] In fact, this approach is quite similar to that of ridge regression described in Dempster's review.[19]

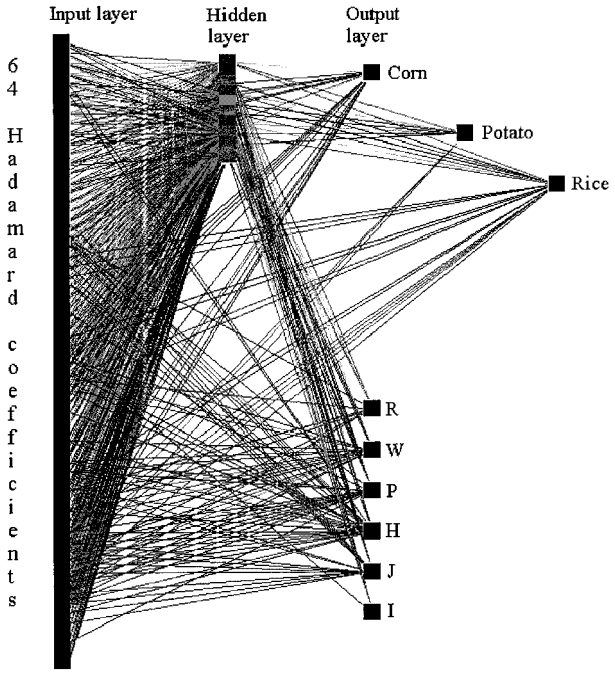

Figure 2. (64 × 10 × 9) neurons structure of the FC-SC feed-forward neural network.

Modified error functions help to solve one of the most important problems in neural network training: how to avoid overdetermination of the model, when it has too many adjustable parameters (weights). The specifics of the weight decay procedure consists of a systematic decreasing of weights while minimizing the second term in the error function, thus zeroing all of them, except the necessary ones. The ⟨necessary weights⟩ are those which receive reinforcement during back-propagation of error value through the network. So, formally we start from the full set of random weights, and during the learning process this set is modified: the most important weights, with high values of derivatives of the error function with respect to them, are increased, while the weights which do not change significantly the value of the error function are decreased to zero. When training is finished, instead of several hundred almost equivalent weights, which we could have using the simplest error function such as the squared difference between target and calculated outputs, we have at most a few dozen significant weights. It is still more than we would prefer, even with compressed data; however this complexity truly reflects the difficulty of the problem, with almost equivalent spectra for different modifications. The structure of ANN with shortcut connections is shown in Figure 2. Although connections between all neurons are formally allowed, most of the links are not visible at the selected level of resolution (it means, the values of weights for those links are less than 10% of the values for shown links). Such iterative pruning of weights increases the generalization ability of the neural network.

2.5. Hierarchical Modeling. Selecting the best model for data analysis involves many trials on a huge number of different methods. Moreover even the optimization of the parameters used inside a chosen method, for example the

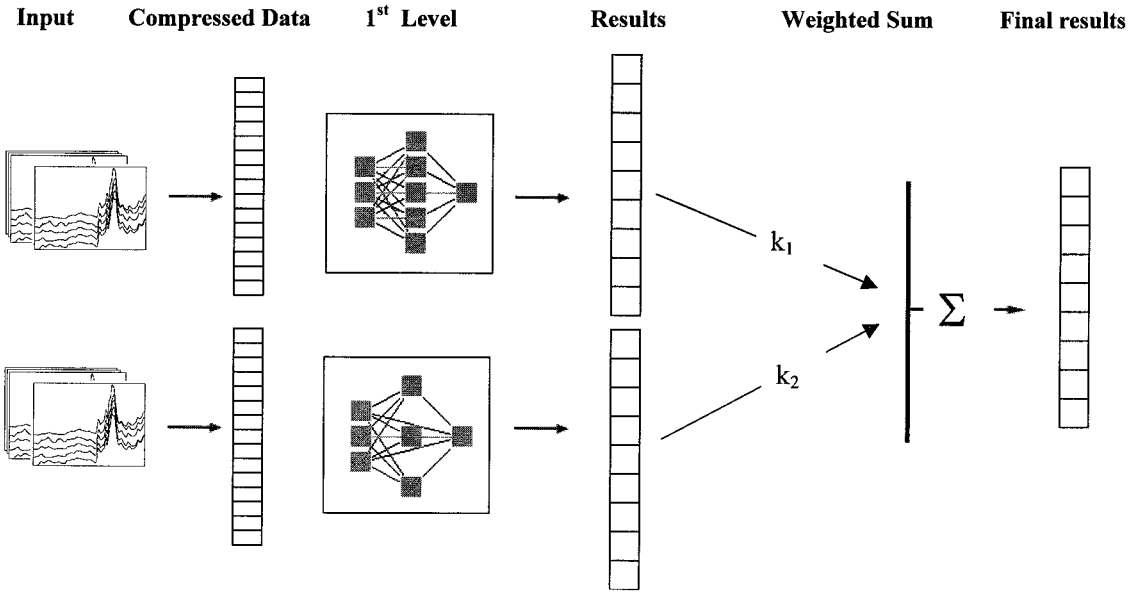

Figure 3. Linear combining of models. First level is composed of FC and FC_SC networks.

number of hidden units and the learning rate in case of ANNs, is far from obvious. It has been proved both theoretically[20,21] and empirically[22] that the accuracy of the quantitative modeling can be improved using the combination of several models. The hierarchical combining of their outputs leads to a new model and a decreasing of prediction error, both for the training and test data sets.

Given an infinite data set, the consistent model can provide an arbitrary degree of generalization; however in real life one usually deals with a restricted set of the training patterns. It is therefore necessary to build several models to select the best one. This selection is very critical, because rejection of the competitive models leads to the loss of potentially important information that was produced during the learning.[23,24] Even when the selected model strictly dominates the others, it is not obvious that it would be the best for the novel patterns. When the models are comparable in terms of fitting quality, the problem of choice becomes essentially more difficult. In order to include into consideration as much relevant information as possible, the outputs of all possible models, which we will call predictors from now, should be pooled into a new model before the final decision is made. There are several techniques for combining ANNs outputs, varying from the simple averaging[21,25] to the weighted averaging[23] using different methods of weights calculation. It is worth mentioning that in a common context of applied statistics, averaging of the models is a well-established practice, including averaging based on the cross-validation results.[25]

Nevertheless, all these techniques use some linear combinations of outputs, a procedure similar to that shown in Figure 3. We propose a nonlinear combining of results of the nonlinear predictors, where the outputs of low-level predictors serve as the inputs of a higher-level predictor (Figure 4). For the second level of modeling we needed a network that allowed us to extract the dependencies between input data, i.e. the network which is able to solve a subtype of a general problem of spatial pattern memorizing; thus we have selected the architecture named in SNNS as Elman network[27] which uses multiple iterations (not just one feed-forward pass) in order to represent the association between input and output data.[28] This network is an example of simultaneous recurrent networks (SRN), which are very popular in some fields of neural modeling. They allow the memorizing of spatial patterns in the space of input data. The design and learning algorithms of time-delayed recurrent networks and the networks with simultaneous recurrence were first described by Werbos.[26]

3. DISCUSSION OF AN EXAMPLE OF MODIFIED STARCHES

The importance of starches in the food industry has been well-recognized during the long centuries. They are greatly available, are easy to use, and provide an indispensable source of energy. Starch comes from many sources—cereals like corn, wheat, and rice and root plants like potato and tapioca. Since these crops can be planted every year, starch is an abundant and naturally renewable resource. Nowadays the manufacturing of native and modified starch is measured in hundred million tons. The modern food industry uses this product as an additional component in meat products, sausages, yogurts, etc. Technological requirements determine the necessity of starch structure modifications that preserve the main nutritional features of starch, allowing an easier way of industrial treatment. A brief description of the most widely used modifications is presented in Table 1; more extensive information can be found in refs 29 and 30. Since the legislation concerning food in the European community is becoming stricter and the standards of quality are established, it is important to determine and control these modifications according to the industrial needs and food security.

3.1. Analysis of the Results Using First-Level Modeling. 3.1.1. Choice of the Network Structure. A simple feed-forward neural network composed of three layers was selected for the modeling of the relationship between the data and their origin and modifications. The size of the input layer was determined by the Hadamard maximal possible degree of compression of the spectral data; the size of the

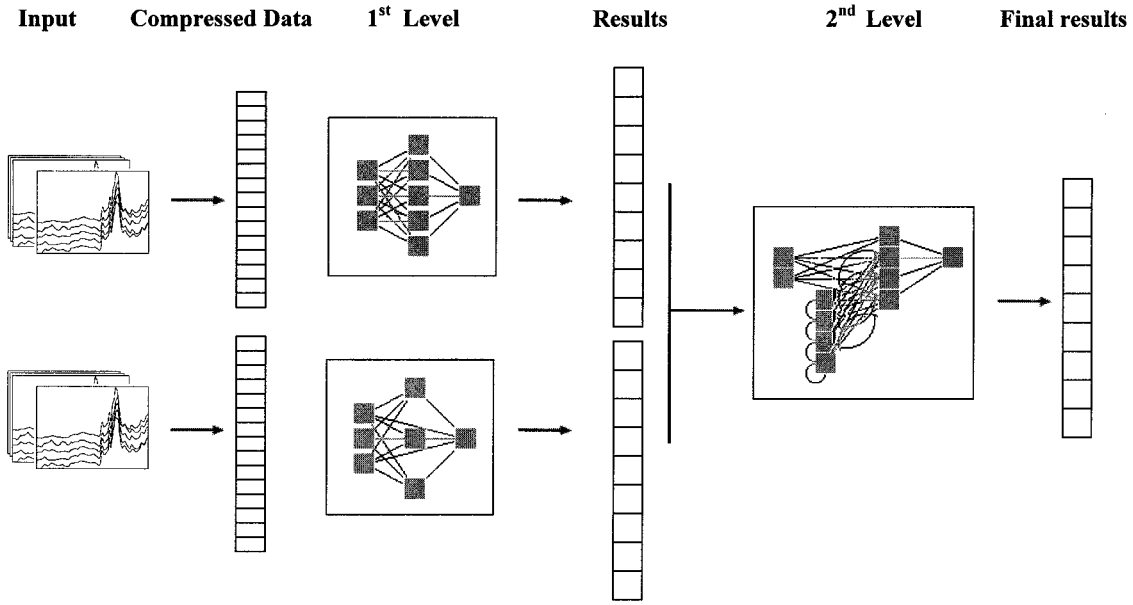

Figure 4. Multilevel hierarchical modeling. First level is composed of FC and FC_SC networks. Second level consists of the MetaANN network.

output layer was equal to the sum of the maximal number of considered origins and modifications. Therefore, only the number of hidden neurons had to be adjusted.

The problem of selecting the most suitable network architecture and the parameters of the algorithm for training of the feed-forward network was discussed extensively both in theoretical and applied publications.[8,9] In our previous papers[3,4] one can find some recommendations as well. Empirically with the number of neurons in a hidden layer below 10 the network was unable to compress the input vector and to produce a noncontradictory input−output mapping. Therefore the number of neurons in the hidden layer was chosen to be 10 in order to minimize the number of adjustable parameters (weights). Thus, the network architecture can be described as $64 \times 10 \times 9$ with 64 input neurons, 10 hidden neurons, and 9 output ones. Additional reduction of the number of weights for better generalization ability was made possible in this study using the modified error function, described above in section 2.4.

Two different possibilities of modeling were practised: the fully-connected ANN without direct links between nonadjacent layers (FC) and the ANN with direct connections from the input to the output layer in addition to full-connected adjacent layers (FC-SC). Formally the results of prediction for the 21 test samples (Table 3) may be considered nearly the same for both the full-connected network and the one with shortcut connections enabled. Although for some examples in the test set the prediction were better when we used the FC-SC network, the FC ANNs perform generally better. This discrepancy gave us an additional opportunity to analyze how the hierarchical modeling works in the case of asymmetrically biased models.

3.1.2. Quality Assessment of the First-Level Modeling. Prediction results for the network with shortcut connections, as well as the results for the full-connected network, were reasonably good but not clear enough, allowing multiple interpretations. Undoubtedly, the quality of modeling is determined to a high degree by the quality of the data; however it is necessary to understand why the predictions for some combinations of the modifications are not as accurate as those for the others and what to do in order to improve the quality of modeling. The obvious solution would be to reduce the noise, inevitably present in the input data. It could also be very interesting that the output labeling take into account variable degrees of modification since the treatment of a sample may be partial, i.e. different from 0 or 100%. This involves codification of the output using continuous instead of binary variables. In fact, the nonavailability of this information and the industrial restrictions prevent this here. In any case, real data (especially for natural compounds, like starches) make necessary the deal with such limitations. A more realistic way presumes the multilevel modeling, which helps to extract maximal information from available data even if they are rough and noisy. Having reasonably good but not exact predictions, it is necessary to analyze all possible causes of prediction errors. Such an analysis suggests the directions of the data set modification and the recommendations for the choice of the learning algorithm and the error function in order to achieve better results. Usually such an analysis is done by the human expert using the formal domain knowledge and the appropriate empirical rules. The rules for the correction of prediction should be induced from the training set itself by the analysis of the links between formal attributes and chemical information and applied to the test set like a kind of "renormalization procedure".

In the data set we have at our disposal, H modification is exclusively associated with a corn origin, never with potato or rice starches. Consequently, the network correlates H modification to a corn origin and, moreover, non-H modification to a potato origin. This is illustrated by the good prediction of all the corn/WH samples, more distinguished from potato or rice ones than the unmodified corn starch. These wrong associations are the consequences of the fact that the data set is little representative and could have been hardly avoided.

In the case of W-modification the artificial neural network was not able to discriminate corn/N (nonmodified) and corn/

Table 3. Summary of the Prediction Results for the Test Set[a]

	model	corn	potato	rice	R	W	P	H	J	I
1. corn/WH	exptl	1	0	0	0	1	0	1	0	0
	FC	0.954	0.004	0.02	0.106	0.972	0.003	0.847	0.008	0
	FC-SC	0.995	0.002	0.009	0.089	0.998	0	0.818	0.03	0
	averaging	0.975	0.003	0.015	0.098	0.985	0.001	0.832	0.019	0
	MetaANN	0.999	0.002	0.001	0.002	1	0.004	0.99	0.004	0
2. corn/WH	exptl	1	0	0	0	1	0	1	0	0
	FC	0.979	0.007	0.008	0.107	0.984	0.003	0.936	0.006	0
	FC-SC	0.999	0.001	0.002	0.086	1	0.001	0.954	0.026	0
	averaging	0.989	0.004	0.005	0.096	0.992	0.002	0.945	0.016	0
	MetaANN	0.999	0.003	0.001	0.001	1	0.004	0.996	0.004	0.002
3. corn/RWP	exptl	1	0	0	**1**	1	1	0	0	0
	FC	0.987	0.171	0.002	**0.929**	0.763	0.994	0.113	0	0.017
	FC-SC	0.992	0.004	0.012	**0**	0.814	0.998	0.025	0	0.077
	averaging	0.99	0.087	0.007	**0.465**	0.788	0.996	0.069	0	0.047
	MetaANN	0.996	0.004	0.001	**0.992**	0.99	0.993	0.004	0	0.002
4. corn/WPH	exptl	1	0	**0**	0	1	1	1	0	0
	FC	0.766	0	**0.401**	0.003	0.971	0.999	0.771	0.043	0.065
	FC-SC	0.721	0	**0.109**	0	1	0.999	0.909	0	0.071
	averaging	0.743	0	**0.255**	0.001	0.986	0.999	0.840	0.022	0.068
	MetaANN	0.996	0.002	**0.008**	0.002	0.999	0.994	0.992	0.002	0.005
5. rice/WP	exptl	**0**	0	**1**	0	1	1	0	0	0
	FC	**0.465**	0	**0.453**	0.026	0.948	0.955	0.087	0.003	0.009
	FC-SC	**0.507**	0	**0.71**	0	0.967	0.972	0.049	0	0.009
	averaging	**0.486**	0	**0.581**	0.013	0.958	0.963	0.068	0.001	0.009
	MetaANN	**0.071**	0.001	**0.953**	0.006	0.998	0.994	0.004	0.003	0
6. corn/N	exptl	1	0	0	0	0	0	0	0	0
	FC	0.783	0.018	0.048	0.173	0.922	0.01	0.174	0	0
	FC-SC	0.968	0.008	0.018	0.019	0.968	0.003	0.192	0.001	0
	averaging	0.875	0.013	0.033	0.097	0.945	0.007	0.183	0.001	0
	MetaANN	0.996	0.001	0.006	0.019	0.998	0.003	0.012	0.001	0.002
7. corn/W	exptl	1	0	0	0	1	0	0	0	0
	FC	0.971	0.024	0.005	0.138	0.963	0.108	0.202	0	0.001
	SC	0.992	0.004	0.011	0	0.967	0.129	0.167	0	0
	averaging	0.981	0.014	0.008	0.069	0.965	0.118	0.184	0	0.001
	MetaANN	0.997	0	0.004	0.013	0.998	0.008	0.014	0.001	0.002
8. corn/RW	exptl	1	0	0	1	1	0	**0**	0	0
	FC	0.845	0.102	0.02	0.849	0.947	0.006	**0.357**	0.001	0
	FC-SC	0.966	0.048	0.003	0.695	0.974	0	**0.368**	0.010	0
	averaging	0.905	0.075	0.011	0.772	0.961	0.003	**0.363**	0.005	0
	MetaANN	0.996	0.008	0.001	0.991	0.998	0.004	**0.062**	0	0.001
9. potato/R	exptl	**0**	1	0	**1**	**0**	0	0	0	0
	FC	**0.792**	0.936	0.008	**0.909**	**0.427**	0.26	0.015	0	0.001
	FC-SC	**0.802**	0.823	0	**0.285**	**0.333**	0.146	0.02	0	0
	averaging	**0.797**	0.879	0.004	**0.597**	**0.38**	0.203	0.016	0	0.001
	MetaANN	**0.268**	0.937	0.001	**0.961**	**0.005**	0.011	0	0	0.001
10. rice/N	exptl	0	0	1	0	0	0	0	0	0
	FC	0.046	0	0.979	0.045	0.187	0.052	0.003	0.131	0
	FC-SC	0.004	0.002	0.991	0.006	0.104	0	0	0.013	0
	averaging	0.025	0.001	0.985	0.026	0.14	0.026	0.002	0.072	0
	MetaANN	0.002	0.011	0.994	0.001	0.002	0.004	0	0.006	0
11. corn/W	exptl	1	0	0	0	1	0	0	0	0
	FC	0.973	0.027	0.005	0.128	0.967	0.111	0.229	0	0.001
	FC-SC	0.994	0.004	0.007	0	0.979	0.151	0.252	0	0.001
	averaging	0.983	0.015	0.006	0.064	0.973	0.131	0.24	0	0.001
	MetaANN	0.997	0.001	0.003	0.01	0.998	0.009	0.035	0.002	0.002
12. rice/W	exptl	0	0	1	0	1	0	0	0	0
	FC	0.041	0	0.974	0.058	0.799	0.027	0.033	0.135	0
	FC-SC	0.013	0	0.984	0.003	0.801	0	0.007	0.082	0
	averaging	0.027	0	0.979	0.03	0.8	0.013	0.02	0.109	0
	MetaANN	0.003	0.001	0.997	0.002	0.985	0.004	0	0.005	0
13. corn/WH	exptl	1	0	0	0	1	0	1	0	0
	FC	0.948	0.006	0.01-9	0.135	0.98	0.003	0.831	0.005	0
	FC-SC	0.996	0.002	0.007	0.07	0.999	0	0.854	0.025	0
	averaging	0.972	0.004	0.013	0.103	0.989	0.002	0.843	0.015	0
	MetaANN	0.999	0.003	0.001	0.002	1	0.004	0.991	0.003	0.002
14. corn/WH	exptl	1	0	0	0	1	0	1	0	0
	FC	0.956	0.004	0.018	0.126	0.979	0.003	0.829	0.006	0
	FC-SC	0.996	0.001	0.008	0.07	0.999	0	0.866	0.031	0
	averaging	0.976	0.007	0.013	0.098	0.989	0.002	0.848	0.019	0
	MetaANN	0.999	0.003	0.001	0.002	1	0.004	0.991	0.004	0.002

Table 3 (Continued)

	model	corn	potato	rice	R	W	P	H	J	I
15. corn/WPH	exptl	1	0	0	0	1	1	1	0	0
	FC	0.992	0	0.004	0.1	1	0.935	0.995	0.004	0.003
	FC-SC	0.98	0.019	0	0.021	1	0.925	1	0.003	0.002
	averaging	0.986	0.01	0.002	0.061	1	0.93	0.997	0.003	0.003
	MetaANN	0.999	0.003	0.001	0.002	1	0.991	0.997	0.003	0.005
16. corn/RWP	exptl	**1**	0	0	**1**	1	1	0	0	0
	FC	**0.625**	0.005	0.13	**0.491**	0.923	0.966	0.092	0.001	0.006
	SC	**0.745**	0.001	0.081	**0**	0.956	0.965	0.091	0	0.008
	averaging	**0.685**	0.003	0.105	**0.245**	0.943	0.965	0.092	0	0.007
	MetaANN	**0.968**	0.001	0.023	**0.832**	0.998	0.995	0.013	0	0.002
17. corn/N	exptl	*1*	0	0	0	0	0	0	0	0
	FC	*0.001*	0.138	0.971	0.178	0.492	0.003	0.003	0	0
	FC-SC	*0*	0.53	0.901	0.224	0.003	0.003	0	0.001	0
	averaging	*0.001*	0.334	0.936	0.201	0.247	0.003	0.001	0	0
	MetaANN	*0*	0.111	0.989	0.003	0.002	0.003	0	0.001	0
18. corn/RW	exptl	**1**	0	0	1	1	0	0	0	0
	FC	**0.618**	0.027	0.1	0.893	0.804	0.005	0.036	0.002	0
	FC-SC	**0.643**	0.063	0.043	0.916	0.745	0	0.03	0.052	0
	averaging	**0.63**	0.045	0.072	0.904	0.774	0.003	0.033	0.027	0
	MetaANN	**0.984**	0.004	0.005	0.998	0.993	0.005	0.001	0	0.001
19. rice/WJ	exptl	0	0	1	0	*1*	0	0	1	0
	FC	0.001	0	1	0.015	*0*	0	0	0.986	0
	FC-SC	0.002	0	1	0.018	*0.001*	0	0	0.926	0
	averaging	0	0	1	0.016	*0*	0	0	0.956	0
	MetaANN	0.005	0.007	0.997	0	*0.018*	0.007	0	0.983	0.001
20. rice/N	exptl	0	0	1	0	0	0	0	0	0
	FC	0.052	0	0.983	0.029	0.197	0.022	0.003	0.202	0
	FC-SC	0.003	0.001	0.997	0.009	0.075	0	0	0.034	0
	averaging	0.027	0.001	0.99	0.019	0.136	0.011	0.002	0.118	0
	MetaANN	0.002	0.01	0.995	0.001	0.002	0.004	0	0.008	0
21. corn/WH	exptl	1	0	0	0	1	0	1	0	0
	FC	0.991	0.02	0.002	0.217	0.993	0.003	0.943	0.002	0
	FC-SC	1	0.001	0.001	0.078	1	0.001	0.985	0.02	0
	averaging	0.995	0.01	0.002	0.148	0.997	0.002	0.964	0.011	0
	MetaANN	0.999	0.003	0.001	0.002	1	0.004	0.995	0.002	0.002

[a] FC means no direct input−output connections for the network. FC-SC stipulates that these kinds of links are allowed. Bold text: for evident improvement with recurrent networks (MetaANN). Italic text: for wrong predictions.

W. The distinction between the nonmodified and waxy samples is in the amylose/amylopectine ratio (20/80 for nonmodified samples and 0/100 for waxy (W) samples). It might be possible that such a quantitative difference is not adequately reflected in IR spectra, but it can also be a consequence of the dominance of the waxy samples in the training set. The tendency to assign the W-property to the nonmodified samples shows up also for the prediction of rice nonmodified samples.

Another example of sample-specific error in prediction is the recognition of the corn/N starches which are respectively identified as corn/W and rice/N. We believe that the origin of the failure comes from the training data themselves. Since we dispose of only two samples for training, in case they are slightly different, the network is unable to merge them into a new category, i.e. corn/N, and identifies them wrongly. An answer we have found to this problem is to separate arbitrarily the corn/N category in two groups in the training set, corn/N1 and corn/N2. Each of them contains only one training sample. Hence, the learning performs better and the test samples are predicted successfully as corn/N1 or corn/N2, that is to say corn/N for our purpose. We do not develop this possibility any further here.

These observations can help to explain some biases in prediction. They let us resolve which modifications are predicted incorrectly due to the lack of learning or due to the inconsistency of the data set and which modifications are less likely to be predicted correctly because of the specifics of the problem.

3.2. Analysis of the Results Using a Second Level of Neural Modeling. As one can see from the previous section, the analysis of the prediction resulting for the training set is easier than a direct interpretation of spectra. Besides, the results of such analysis present better evidence for the expert in the consumption quality, and also they are more useful for the manufacturer. The next question follows: Is it possible to extract these facts (and perhaps other ones) directly from the results of learning? The answer is ⟨yes⟩. Moreover, it is not necessarily done by a human expert but by another artificial neural network.

The simultaneous recurrent network (SRN) used in this study combined the prediction results of the two first-level networks (FC and FC-SC) at the input layer. Therefore SRN had 18 input neurons, 20 hidden neurons that were connected by recurrent links with the same number of context units, and 9 output neurons that represented the same set of origins/modifications as those for used for the first-level networks. This second-level network, called hereafter MetaANN, used the same training and test sets. The general structure of this network is shown in Figure 5.

The predictions of the modifications of the test samples by MetaANN are presented in Table 3, which also displays

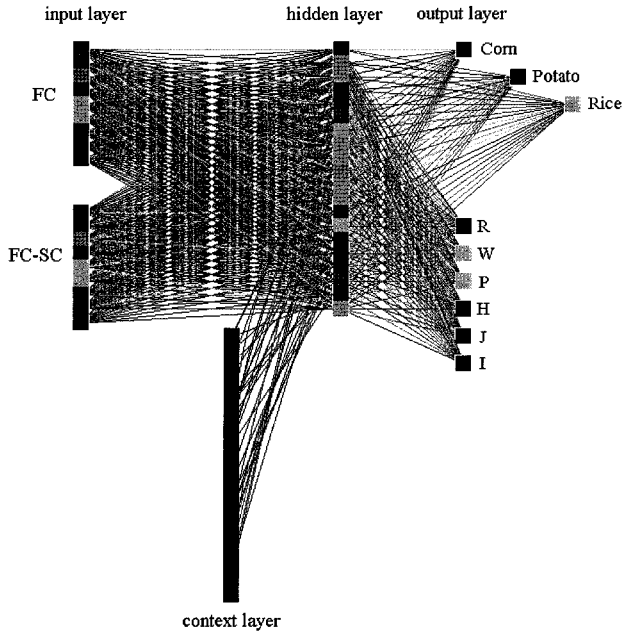

Figure 5. Structure of the second-level recurrent neural network (MetaANN).

the prediction results of FC and FC-SC, previously discussed. Preliminarily we can see that in most cases MetaANN increases the value of the output parameter if it is high for both first-level networks and decreases this value if it is low. However, this hypothesis is oversimplified and, fortunately, wrong. It is the most common case, and then there is no need in the MetaANN: a simple logical rule would be enough to escape "the zone of uncertainty", from 0.2 to 0.7 for predicted value, where the interpretation of prediction by an expert is highly subjective and dependent on his a priori knowledge of the data sets.

But we can also demonstrate a lot of counterarguments to this simple rule. MetaANN modifies first-level predictions even when they agree, as in the potato/R example. In this case both first-level networks predicted a corn origin, but the error correcting procedure of the second-level network changed it to potato. More examples of such "asynchronic" corrections can be found in Table 3. The task of MetaANN is to increase the prediction accuracy. This is done by analyzing and comparing the outputs from the two first-level networks with the target values. This analysis involves some recurrent combining of the input signals, which are performed thanks to the contextual layer of the network. It seems that MetaANN is able to find out information that thwarts some initial predictions.

If we compare the prediction results for the recognition of the RW-modified corn starch in the test set, and the results of the prediction of the same modification in the training set, we can see that the first-level networks reproduce the quality achieved in the training set that is far from the ideal. MetaANN is able to recognize that predicted values of origins/modifications are similar to the values learned for the training set, which were approximate. In this case it is possible to improve the quality of the prediction, implicitly taking into account the impurity of the samples and increasing the degree of recognition. A human expert in this situation could hardly find better interpretation.

At this level, we can carry on the comparison between a human expert and recurrent network reasoning. In the case of poor learning of the rice/WJ prediction in the test set, an expert can suggest two alternative explanations: either this sample is really WJ-modified (and then it is a data error) or it is nonmodified rice, because traces of WJ are typical for rice starch. In fact, MetaANN neglects the WJ-modification when predicting sample 19 of the test set. Effectively, on one hand, the three rice/N samples in the training set are declared as if they had shown traces of the WJ-modification. On the other hand, the unique rice/WJ example in this set sees its modification not fully accepted by both the two first-level networks—maybe because the effect of the J-modification on IR spectra are very light. To resume, in three cases, low levels of WJ-modification are present and should not be and the logic found out by the network is that the trace of WJ should be neglected whatever the origin of the starch. It demonstrates that when there is logic in learning data, MetaANN is able to extract it (corn/WH, corn/W, rice, etc.), and then even an "unsuccessful" prediction of rice/WJ follows the same logic scheme. However, in this case the sample from the test set is more instructive in some way: the WJ-modifications are expressed stronger in the test set sample than in the training set. In fact, it seems that MetaANN is not worse than a human expert in such analysis: it is faster and more transparent, and it does not use a priori knowledge about data.

As it was described in our previous publication[31] for the case of the glass refraction coefficient modeling (a standard benchmark problem[32,33]), the hierarchical neural network modeling really leads to a better prediction than averaging or weighted averaging. Similarly, the results described in this study cannot be achieved using a linear combination of predictors because such a tool uses a priori, constant values of the parameters. For instance, if one of the networks regularly predicts an output below target value, and another one predicts higher, the best linear combination in this case would slightly increase the first output and slightly reduce the second. Thus, variable weights, adaptive with training, are really necessary. The context layer of the partial recurrent network, which is calculated each time a training sample is presented, memories a kind of summary of the prior events and makes enhancement of the predictions possible. It is the only aspect that differentiates recurrent networks from more classical back-propagation algorithm inspired networks. Thus, some more sophisticated behavior can be assessed by the recurrent network.

When the second-level network[31] was only the simple feed-forward network, it was not free from the usual problems with the premature saturation, convergence to different local minima depending on different initialization, etc. On the contrary, simultaneous recurrent neural networks provide better possibilities for modeling of interrelationships between input data. It is theoretically clear that SRN can imitate feed-forward architecture without backward and recurrent connections; however in this case such connections were certainly active. Our experiments with second-level modeling using feed-forward networks have shown that overtraining began very fast, when MetaANN is still not well-trained. With SRN it was possible to avoid this problem.

3.3. Stopping Criteria and Descriptive Statistics for the Models. Taking into consideration the accuracy of the input data, the process of the neural network learning was stopped when the mean squared error (MSE) on the training set

Table 4. Descriptive Statistics for the Models[a]

model	FC	FC-SC	averaging	MetaANN
MSE_corn	0.108	0.103	0.105	0.051
MSE_potato	0.003	0.015	0.007	0.001
MSE_rice	0.068	0.044	0.054	0.047
MSE_R	0.062	0.168	0.096	0.049
MSE_W	0.165	0.153	0.156	0.141
MSE_P	0.052	0.051	0.051	0.047
MSE_H	0.062	0.060	0.061	0.047
MSE_J	0.004	0.001	0.002	0
MSE _I	0	0.001	0	0
MSE_Sum	0.525	0.595	0.531	0.384

[a] MSE_corn (MSE_potato, MSE_rice) stands for the mean squared error in the origin prediction; MSE_x, where x denotes the modification, is the mean squared error for this particular modification prediction; MSE_Sum means the sum of MSEs for all output parameters—origins and modifications—for the particular model.

decreased below 0.10 for FC and FC-SC and below 0.005 for SRN, the second-level network.

The MSEs on the test set for all these models are presented in Table 4. As we already mentioned, the statistics is significantly better for multilevel modeling then for any of the first-level models or their average.

4. CONCLUSION

The method of recognition of starches modifications using feed-forward artificial neural networks described in this paper has proven to be a reliable and effective tool. It produces the unified model for the vast variety of possible modifications since it memories the images of different modifications simultaneously. The generalization ability of the feed-forward ANN makes it possible to recognize the combinations of several modifications having separately modified training samples. The combinations of models demonstrate that the quality of modeling can be significantly improved by the processing of the results of prediction with another artificial neural network, this time exploiting recurrence.

The suggested approach to the combining of the outputs of different models into a multilevel model using simultaneous recurrent neural network for correction of separate predictions seems to be more powerful than a conventional selection of the best model or than the linear combining of outputs, although the results of computer experiments should be theoretically verified using the theory of statistical decisions.

ACKNOWLEDGMENT

The authors wish to thank CREALIS (Centre de Recherches et d'Etudes Alimentaires), DANONE, Z. I. du Teinchurier, 19100 BRIVE, France, for support of this research, and especially Dr. C. D. Ta and Dr. C. Wojciechowski for their assistance.

REFERENCES AND NOTES

(1) Riedmiller, M.; Braun, H. A Direct Adaptive Method for Faster Backpropagation Learning: the RPROP Algorithm. *Proceedings of the IEEE International Conference on Neural Networks*, San Francisco, 1993; IEEE: Piscataway, NJ, 1993; pp 586−591.

(2) Dupuy, N.; Huvenne, J. P.; Duponchel, L.; Legrand, P. Classification of Green Coffees by FT-IR Analysis of Dry Extract. *Appl. Spectrosc.* **1995**, *49*, 580−585.

(3) Dolmatova, L.; Ruckebusch, C.; Dupuy, N.; Huvenne, J. P.; Legrand, P. Quantitative Analysis of Paper Coatings Using Artificial Neural Networks. *Chemom. Intell. Lab. Syst.* **1997**, *36*, 125−140.

(4) Dolmatova, L.; Ruckebusch, C.; Dupuy, N.; Huvenne, J. P.; Legrand, P. Identification of Modified Starches Using Kohonen Mapping of Infrared Spectra. *Appl. Spectrosc.* **1998**, *52*, 329−338.

(5) Savitsky, A.; Golay, M. J. E. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Anal. Chem.* **1964**, *36*, 1627−1679.

(6) Van Soest, J. G.; Tournois, H.; Dewit, D.; Vliegenthar, J. F. G. Short-range structure in (partially) crystalline potato starch determined with attenuated total reflectance Fourier-transform IR spectroscopy. *Carbohydr. Res.* **1995**, *279*, 201−214.

(7) Novic, M.; Zupan, J. Investigation of infrared spectra−structure correlation using Kohonen and counterpropagation neural networks. *J. Chem. Inf. Comput. Sci*. **1995**, *35*, 454−466.

(8) Bishop, C. M. *Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, U.K., 1995.

(9) Nelson, M. C.; Illingworth, W. T. *A Practical Guide to Neural Nets*; Addison-Wesley: Reading, MA, 1991.

(10) Kolmogorov, A. N. On the representation of continuous function of many variables by superposition of continuous functions of one variable and addition. *Dokl. Akad. Nauk SSR* **1957**, *114*, 953−956. English translation: *Math. Soc. Transl.* **1963**, *28*, 55−59.

(11) Kurkova, V. Kolmogorov's Theorem and Multilayer Neural Networks. *Neural Networks* **1992**, *5*, 501−506.

(12) Werbos, P. J. Beyond regression: New tools for predictions and analysis in the behavioural sciences. *Ph.D. dissertation*; Harvard University Press: Cambridge, MA, 1974.

(13) Riedmiller, M. Advanced Supervised Learning in Multilayer Perceptrons−from Backpropagation to Adaptive Learning Techniques. *Int. J. Comput. Stand. Interfaces* **1994**, *16*, 265−278.

(14) Schiffmann, W. Optimisation of the backpropagation algorithm for training multilayer perceptrons. *Technical Report*; Department of Physics, University of Koblenz: Koblenz, Germany, 1994 (ftp://archive.cis.ohiostate.edu/pub/schiff.bp_speedup.ps.Z).

(15) Zell, A.; Mamier G.; Vogt, M. Stuttgart Neural Network Simulator. *User Manual* (Version 4.1); Report No. 6/95, University of Stuttgart, IPVR: Stuttgart, Germany, 1995.

(16) Werbos, P. J. Learning How the World Works: Specifications for Predictive Networks in Robots and Brains. *Proceedings of the SMC Conference*; IEEE Press: New York, 1987.

(17) White, D. In *Handbook of intelligent control*; Sofge, D., Ed.; Van Nostrand Reinhold: New York, 1992.

(18) Ripley, B. D. Statistical aspects of Neural Networks. In *Networks and Chaos-Statistical and Probabilistic Aspects*; Barndorff-Nielsen, O. E., Jensen, J. L., Kendall, W. S., Eds.; Chapman & Hall: London, 1993; p 40.

(19) Dempster, A. P.; Shatzoff, M.; Wermuth, N. A simulation study of alternatives to ordinary least squares. *J. Am. Stat. Assoc.* **1977**, *72*, 77−91.

(20) Turner, K.; Ghosh, J. Analysis of Decision Boundaries in Linearly Combined Neural Classifiers. *Pattern Recognit.* **1996**, *29*, 341−348.

(21) Tumer, K.; Ghosh, J. A Framework for Estimating Performance Improvements in Hybrid Pattern Classifiers. *Proc. World Congr. Neural Networks* **1994**, *3*, 220−225.

(22) Lincoln, W. P.; Skrzypek, J. Synergy of Clustering Multiple Backpropagation Networks. In *Advances in Neural Information Processing System*; Touretzky, D. S., Ed.; Morgan Kaufmann: San Mateo, CA, 1990; Vol. 2, pp 650−657.

(23) Wolpert, D. H. Stacked Generalisation. *Neural Networks* **1992**, *5*, 241−259.

(24) Perrone, M. P.; Cooper, L. N. In *Artificial Neural Networks for Speech and Vision*; Mammone, R. J., Ed.; Chapman & Hall: London, 1993; pp 126−142.

(25) Stone, M. Cross-Validatory Choice and Assessment of Statistical predictions (with discussion). *J. Royal Statist. Soc. B* **1974**, *36*, 111−147.

(26) Werbos, P. J. Backpropagation: Past and Future. *Proceedings of the IEEE International Conference on Neural Networks*; IEEE Press: New York, 1988; pp 343−353.

(27) Elman, J. L. Finding Structure in Time. *Cognit. Sci.* **1990**, *14*, 179−211.

(28) Omlin, C. W.; Thronber, K. K.; Giles, C. L. Fuzzy Finite-State Automata Can Be Deterministically Encoded into Recurrent Neural Networks. *Technical Report CS-TR-3599*; University of Maryland: College Park, MD, 1996.

(29) Radley, J. A. In *Starch Production Technology*; Radley, J. A., Ed.; Applied Science Publishers Ltd.: London, England, 1976.
(30) Forrest, B.; Cove, L. Identification and quantification of hydroxypropylation of starch by FTIR. *Starch* **1992**, *44* (5), 179−183.
(31) Dolmatova, L. M. The architecture of multimodular system with optimal quality of modeling. *Proceedings of JCKBSE'96: Second Joint Conference on Knowledge-Based Software Engineering*; Dochev, D., Stefanuk, V., Ueno, H., Eds.; Sozopol, Bulgaria, Sep 21−22, 1996; pp 214−217.
(32) Pretchel, L. Proben1-A set of neural network benchmark problems and benchmarking rules. *Technical report 21/94*; Universitat Karlsruhe: Karlsruhe, Germany, 1994 (ftp.ira.uka.de/pub/neuran/probenl.tar.gz).
(33) Ripley, B. D. In *Pattern Recognition and Neural Networks*; Cambridge University Press: Cambridge, U.K., 1996.

CI990442Y