JOURNAL OF MAGNETIC RESONANCE **134,** 124–130 (1998)
ARTICLE NO. MN981496

A Neural Network Approach to the Rapid Computation of Rotational Correlation Times from Slow Motional ESR Spectra

Gary V. Martinez and Glenn L. Millhauser[1]

Department of Chemistry and Biochemistry, University of California, Santa Cruz, Santa Cruz, California 95064

Received August 18, 1997; revised April 7, 1998

We explore the use of feed forward artificial neural networks for determining rotational correlation times from slow motional nitroxide electron spin resonance spectra. This approach is rapid and potentially eliminates the need for traditional iterative fitting procedures. Two networks are examined: the radial basis network and the multilayer perceptron. Although the radial basis network trains rapidly and performs well on simulated spectra, it is less satisfactory when applied to experimental spectra. In contrast, the multilayer perceptron trains slowly but is excellent at extracting correlation times from experimental spectra. In addition, the multilayer perceptron operates well in the presence of noise as long as the signal-to-noise ratio is greater than approximately 200/1. These findings suggest neural networks offer a promising approach for rapidly extracting correlation times without the need for iterative simulations. © 1998 Academic Press

Key Words: **neural networks; ESR; slow motional spectra; radial basis network; multilayer perceptron; rotational correlation time.**

INTRODUCTION

Nitroxide spin labels, in conjunction with electron spin resonance (ESR), serve as probes for exploring dynamics in nucleic acids, peptides and proteins (*1–3*). The ESR lineshape is often used to determine the rotational correlation time (τ_R) of the nitroxide, thereby revealing motion at the label site. When τ_R is less than approximately 1 ns, ESR spectra are characterized by three motionally narrowed hyperfine lines, and straightforward lineshape measurement gives accurate values for the correlation time. However, when τ_R is between 1 ns and 100 ns—the so-called slow motional regime—lineshape analysis is substantially more complicated (*4*). The correlation times of most macromolecules fall within this regime and consequently enormous effort has been directed toward extracting dynamic information from slow motional spectra.

Leading efforts in this field have come from Freed and co-workers (*5*) (see also Chapter 3 in Ref. *1*). Throughout the 1970s and 1980s they developed and refined slow motional simulation techniques based on the stochastic Liouville equation. Despite their great success, determining dynamic parameters from slow motional spectra remains a challenge. The essential problem lies with the iterative approach one must employ when simulating spectra. Given an experimental spectrum, one must first guess at a value for the correlation time (as well as other parameters, including motional anisotropy, local ordering, and sample heterogeneity) and then perform a simulation. The result is compared to the experimental spectrum. If the agreement is not satisfactory, the input values are adjusted and another simulation is performed. This process is repeated until good overlap between the experimental and simulated spectra is achieved.

Recently, Budil *et al.* developed a nonlinear least squares approach to automate the procedure of fitting slow motional spectra (*6*). They applied a "model trust region" modification of the Levenberg–Marquardt algorithm and showed that complicated spectra could be successfully simulated using a computer workstation. Nevertheless, their method still relies on iteration.

The difference in effort required for a noniterative approach, such as that used for analyzing fast motional spectra, and the iterative approach required for slow motional spectra is striking. It would be quite helpful if a method could be developed that used a noniterative approach for analyzing slow motional spectra. Toward this goal we explore the use of artificial neural networks (*7*). Neural networks have emerged as remarkable tools for pattern recognition in scientific applications. They have been applied with good success to spectroscopic problems in nuclear magnetic resonance (*8–10*), circular dichroism (*11–13*), and infrared spectroscopy (*14*).

Artificial neural networks were inspired by research into the interplay that takes place among networks of real biological neurons. In an artificial neural network, the synaptic connections between neurons are represented by numerical weights, which measure the strength of a connection, and a transfer function that emulates the firing of the neuron. Training a network involves establishing a set of numerical weights that successfully connect a training input with a desired output. Once trained, an artificial neural network can be an effective tool for recognizing and extracting key features from previously unseen input.

In spectroscopic applications, neural networks have been

[1] To whom correspondence should be addressed. Fax: (408) 459-2935. E-mail: glennm@hydrogen.ucsc.edu

1090-7807/98 $25.00

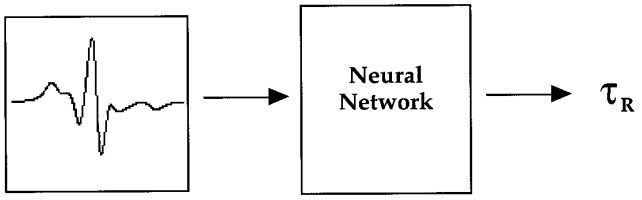

FIG. 1. An overview of the problem. Given an experimental ESR spectrum, what is the rotational correlation time τ_R? The proposal is that the neural network can rapidly predict the τ_R without the need for iteration.

used to identify discrete features such as subspectra arising from molecular components in a mixture or spectral contributions associated with particular molecular conformations. For ESR spectra, the application is somewhat different. Nitroxide spectra vary continuously as a function of correlation time. They are not composed of canonical subspectra and therefore present a new challenge for using neural networks in spectroscopy. A successful neural network will be required to interpolate among a finite set of spectra. We explore the use of multilayer feed forward neural networks, which are among the simplest of the various network schemes. The aim is to be able to feed a spectrum to a neural network and obtain τ_R without iteration. This procedure is shown schematically in Fig. 1.

We examined two particular types of network architectures—radial basis (RB) networks and multilayer perceptrons (MLP)—for their ability to extract isotropic correlation times from nitroxide slow motional spectra (*7*). Our experiments suggest that the RB network trains very rapidly (several minutes on a personal computer) but is overly sensitive to small spectral distortions. In contrast, the MLP (which is often called a back propagation network) trains slowly (approximately 20 min on a workstation and 10–30 h on a personal computer—see Methods) but does an excellent job of extracting accurate correlation times from experimental spectra. The time required for the computation of τ_R from either network is much less than a second. These results suggest that neural networks offer great promise for the noniterative analysis of nitroxide ESR spectra.

THEORY OF NEURAL NETWORKS AS APPLIED TO ESR SPECTRA

The "Neural Network" box in the flow diagram of Fig. 1 represents both the RB network and the MLP. Each network was trained on a series of simulated slow motional ESR spectra (*5*) with correlation times ranging from 1 to 125 ns. The general architecture for the MLP and RB network is shown in Fig. 2. The networks are similar in that each consists of an input layer, followed by a hidden layer of nodes (also called neurons), followed by an output layer. The input layer is a vector that consists of R points sampled from input ESR spectra. At each node the input vector is compared to a weight vector resulting in a scalar that is then passed through a transfer function. The collection of *S1* scalars resulting from the *S1* nodes in the hidden layer is then passed as a vector to the single node output layer. Following comparison to the output layer's weight vector, the resulting scalar is passed through a linear transfer function to yield a value for τ_R. Four features distinguish the RB network from the MLP as implemented here: (1) the method for comparing input and weight vectors, (2) the choice of transfer function employed at each node in the hidden layer, (3) the method for choosing the number of nodes in the hidden layer, and (4) the procedure used for training the network. We now summarize details of each network.

We begin by describing the MLP. In this network, the vector inner product is used to compare the input vector with the weight vector of each node in the hidden layer. At each node a constant bias b is added to the resulting scalar and the value is then passed through a hyperbolic tangent transfer function:

$$O_i = f(\mathbf{W}_i \cdot \mathbf{I} + b), \qquad [1]$$

where O_i is the output of the ith node in the hidden layer, $\mathbf{I}$ is the input vector (a particular spectrum), $\mathbf{W}_i$ is the weight vector for that node, and f is the hyperbolic tangent. Each node (neuron) in the hidden layer will give positive output (fire) as long as $\mathbf{W}_i \cdot \mathbf{I} + b > 0$. The resulting vector $\mathbf{O}$ emerging from the hidden layer is passed to the single-node output layer. The number of nodes in the hidden layer is treated as an empirically adjustable parameter. Networks with a small number of nodes may not successfully train, but in contrast, networks with too many nodes may require excessively long training periods and/or give spurious results. Training uses a set of known spectra and corresponding correlation times along with nonlinear least squares fitting (termed back propagation). The $R \times Q$ matrix composed of known spectra is fed as input to the network. A vector of Q correlation times results, and this vector is compared to the known correlation times. An error is computed according to the sum squared difference between the

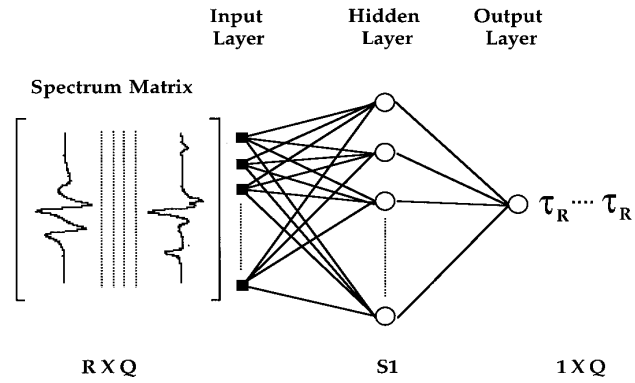

FIG. 2. Network architecture used in batch training mode. The spectrum matrix consists of Q columns of simulated spectra, each represented by R = 255 points. These spectra are presented in batch to the network and the weights of the *S1* hidden layer nodes and the output node are adjusted to correctly predict the Q values of τ_R.

calculated and known τ_R's, and the weights and transfer function parameters are adjusted in an iterative fashion to improve the fit. Each iteration over the entire set of training spectra is termed an epoch.

The RB network differs in several ways from that for the MLP. The Euclidean distance is used to compare $\mathbf{W}_i$ to $\mathbf{I}$ at each node, and f is a Gaussian function characterized by a spread constant sc. Output at each node is thereby given as

$$O_i = f(\|\mathbf{W}_i - \mathbf{I}\| \times \sqrt{\log(2)}/\mathrm{sc}). \qquad [2]$$

Each node (neuron) will give output near unity (fire) as long as $\|\mathbf{W}_i - \mathbf{I}\| \times \sqrt{\log(2)}$ is less than the radius sc. The spread constant is an empirically adjusted parameter typically on the order of unity. The number of nodes is determined dynamically during the training process according to the algorithm of Chen *et al.* (*15*). At the beginning of the training period there is one node in the hidden layer (i.e., $S1 = 1$). An error is computed and if this error exceeds a predetermined limit, a new node is added such that its weight vector is orthogonal to the weight vector of the first node. Each epoch adds a new node at the hidden layer, and Gram–Schmidt orthogonalization is used to maintain orthogonality among the set of hidden layer weight vectors.

RESULTS AND DISCUSSION

We examined both the MLP and the RB network and compared their respective abilities to predict τ_R. The neural networks were trained on 15 simulated ESR spectra, six of which are shown in Fig. 3, and were then applied to a testing set of spectra not present in the training set (see Methods section). Subsequently, the networks were used with experimental data to determine whether they were robust in the presence of noise and other experimental complications.

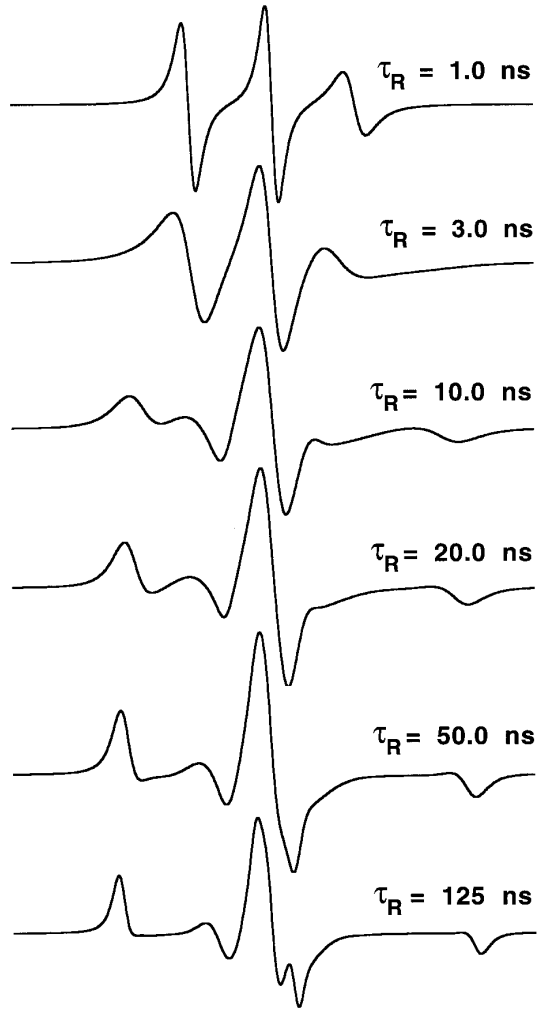

FIG. 3. Six spectra out of the training set of 15 (total sweep width 100 G) shown with corresponding correlation times τ_R. The spectra change dramatically between 1 and 50 ns.

The training histories of each of the two different networks are shown in Fig. 4. The training for the RB network was substantially more rapid than for the MLP. The number of nodes required in the hidden layer of the RB network was only 11 or 13 for error goals of 0.01 and 0.001, respectively. Typically, the network required 1–2 min to reach prescribed sum squared error values. Several values of the spread constant were examined and networks with values either significantly larger or smaller than unity failed to converge. In contrast to the RB network, the MLP required substantially longer training times to reach the desired error goal (20 min on a workstation). MLPs with hidden nodes of 35, 45, and 60 units were trained to error goals of 0.01 and 0.001 (see Methods: Hidden Layer Design for the MLP). The small "spikes" observable in the training history plot represent deviations where the sum squared error increased during a particular new epoch. Empirically we found that if the sum squared error did not drop below 1 within 1200 epochs, the training session would fail to reach the prescribed sum squared error value (i.e., 0.01 or 0.001).

Once training was complete (to a sum squared error of 0.01 for the 45 node MLP and 0.001 for the RB network), the testing set was presented to the network and the τ_R's were compared to the correct values. For each spectrum, the time required to compute τ_R was 50 ms for the RB network and 200 ms for the MLP. The correlation between the correct and output values was used to quantify the goodness of fit. Figure 5a shows the network output plotted against the correct τ_R values for both networks. As a reference, the straight line along the plot diagonal (Output = τ_R) is also shown. Clearly, both networks give excellent results. Residuals are shown in Fig. 5b. Scatter for the MLP is small, with the greatest error on the order of less than 0.3 ns for $\tau_R = 60$ ns. The RB network performs even better, with residuals that never exceed 0.1 ns.

The ability of the networks to predict τ_R from real experimental data is the true test of their validity. Experimental

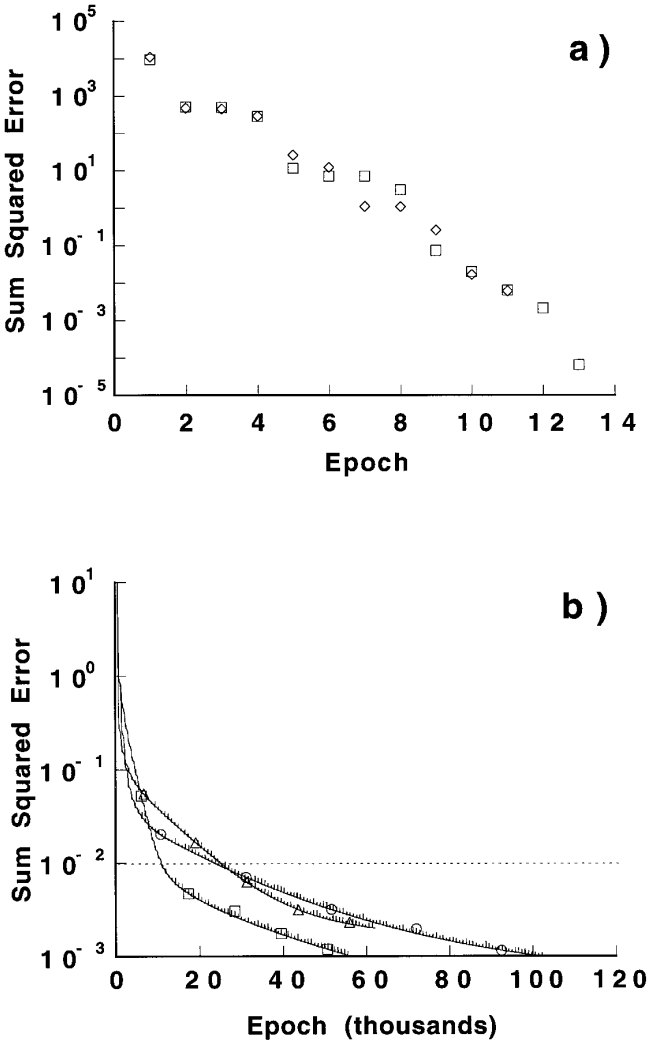

FIG. 4. A plot of the sum squared error as a function of the training cycle in epochs. The RB network is shown in (a) for sum squared error goals of 0.01 (◇) and 0.001 (□). The MLP is shown in (b) with 35 nodes (○), 45 nodes (□), and 60 nodes (△) in the hidden layer.

spectra, from the spin label probe CTPO in 75% v/v glycerol/water, were obtained and processed according to procedures outlined in the Methods section. After network determination of τ_R from each experimental spectrum, the resulting correlation time was used to create a simulation for comparison to that spectrum. The results are shown in Fig. 6. Results for the RB network shown in Fig. 6a indicate that this network does a reasonable job for long correlation times, but the results are less satisfactory for short correlation times. It appears that the RB network emphasizes mainly the large-amplitude features in the middle of each spectrum.

Figure 6b demonstrates that the MLP is able to predict τ_R throughout the slow motional regime. While the RB network fails at shorter correlation times, the MLP clearly does very well. There are some discrepancies between experimental and calculated spectra at longer τ_R, but this may simply be due to inhomogeneous broadening or inadequacy of the assumption of isotropic motion.

When it comes to experimental spectra, the two networks perform differently. It has been asserted that an RB network is as capable as any particular MLP (*7*). This is certainly true in this case when the test spectra are generated from the simulation program. When applied to the testing set, the performance of the RB is excellent. However, experimental spectra contain additional details that may not be captured with a simulation program. Such details include experimental noise, imperfect knowledge of the magnetic tensors, and additional inhomogeneous broadening. In the presence of such distortions, it appears that the RB network is less forgiving than the MLP. RB networks are known to be effective if there are enough nodes to ensure a close match of input to at least one of the weight vectors. Perhaps the addition of noise (see next paragraph) or systematic features of the experimental spectra not present in the simulations results in a lack of proper match with any of the nodes in the trained network. In any case, it appears that the MLP is very robust and quite capable of dealing with both

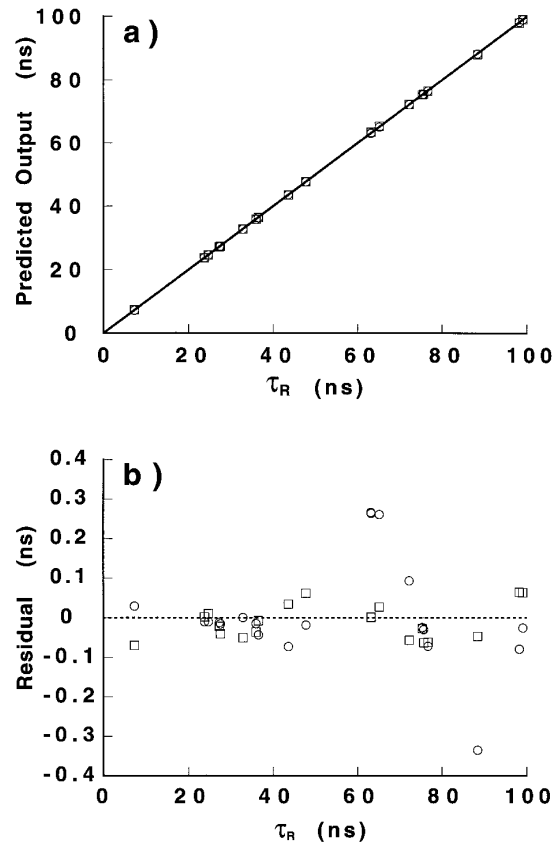

FIG. 5. (a) Predicted rotational correlation time (from the testing set) vs the known correlation time determined by the both RB network (□) and the MLP (○). The correlation times were randomly chosen between 1 and 100 ns for testing the network. The line represents the ideal goal of output = input. The residuals are shown in (b).

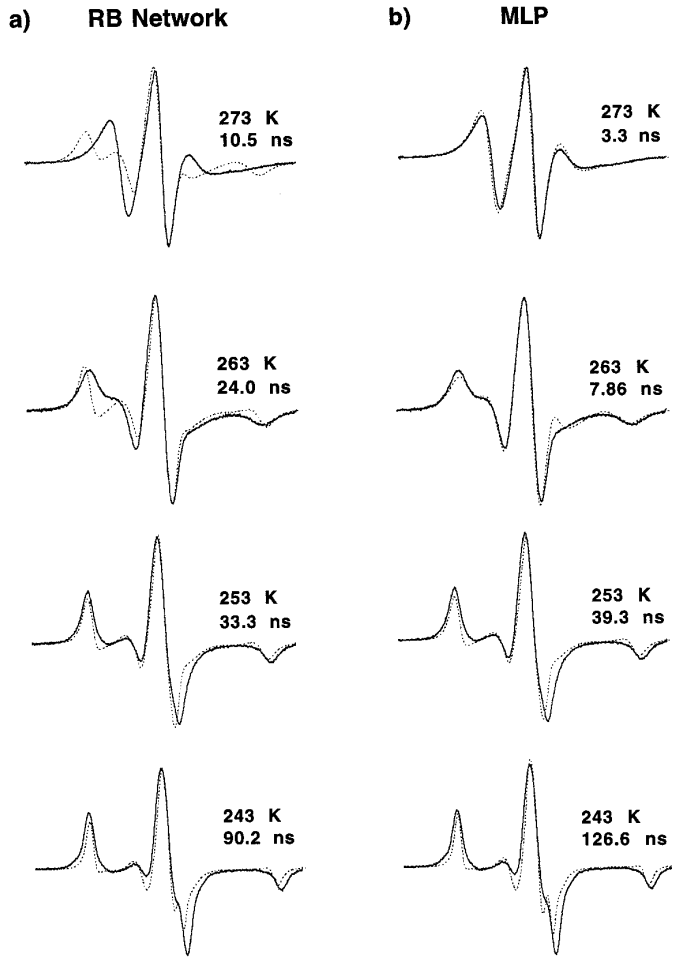

FIG. 6. Overlays of simulated spectra (dashed lines) with experimental CTPO spectra (solid lines) for (a) the RB network and (b) the MLP. Correlation times determined by the networks and temperatures are indicated by each spectrum. The MLP is superior at extracting τ_R in the presence of noise and experimental distortions not present in the training set.

random and systematic differences between the ideal training spectra and experimental spectra.

The signal-to-noise (*S*/*N*) ratio for the experimental spectra in Fig. 6 is approximately 250/1, which is typical for high quality ESR spectra. Given the success of the MLP, it is of interest to explore how this network tolerates the presence of noise. Random noise (Gaussian white noise) was added to the testing set in varying proportions to give *S*/*N* ratios of 100/1, 200/1, and 400/1. These spectra were presented to MLPs with 35, 45, and 60 nodes in the hidden layer, with each trained to sum squared errors of 0.01 and 0.001, as discussed earlier (see also Methods: Hidden Layer Design for the MLP). As was found for the experimental spectra, 45 nodes gave the most reliable output, and the results are shown in Fig. 7. Figure 7a shows the direct output, and Fig. 7b shows the relative % error (defined as $100 \times (\tau_{R\ out} - \tau_{R\ in})/\tau_{R\ in}$). *S*/*N* of 400/1 is well tolerated with an average (RMS) relative error of 9%. Interestingly, the error does not appear to depend on the value of τ_R. *S*/*N* of 200/1 is also reasonably well tolerated, and for this case the average relative error is 16%. However, the largest relative error from this noisy testing set is 32%, which is substantial. Finally, *S*/*N* of 100/1 begins to generate large errors and the output is clearly not a reliable indicator of the true τ_R. We did find that training the MLP to a sum squared error of 0.001 gave a slight improvement in performance for the 100/1 data, but still the RMS error in τ_R was 36%.

Although detection of motional anisotropy was not part of this study, it may certainly be included. The advantages of using a neural network to obtain correlation times may also be used in conjunction with a least squares routine if greater precision is required (*6*). That is, since the neural network can rapidly achieve a good solution, the neural network output can seed a least squares routine which can then refine the fit. Having a MLP serve as a "front end" to a more rigorous nonlinear least squares approach may greatly speed up computation of correlation times and other desired motional and ordering parameters.

In summary, the neural network approach is an effective method for rapidly obtaining rotational correlation times from

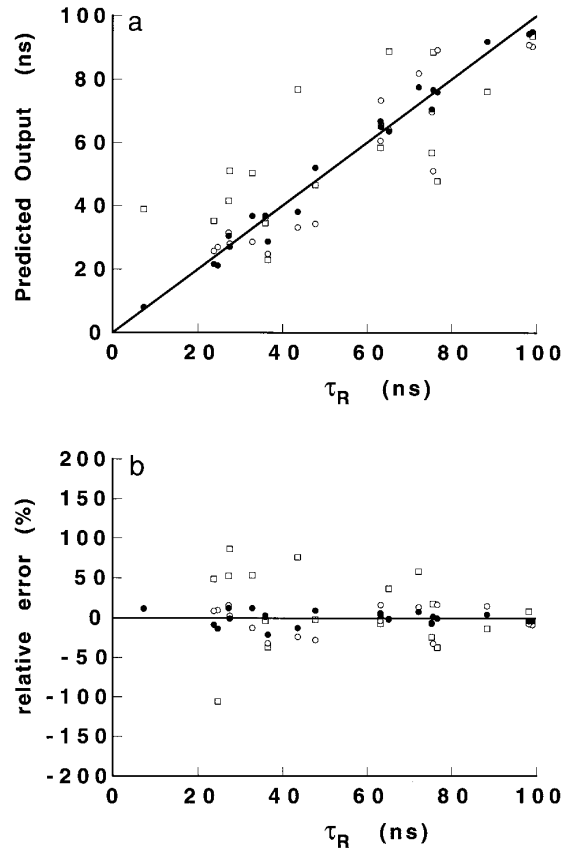

FIG. 7. (a) Predicted rotational correlation time from noisy spectra. Results are from the MLP network with 45 hidden nodes trained to an error goal of 0.01. Noisy input spectra were produced using Gaussian white noise and spectra from the testing set to produce signal-to-noise ratios of 100/1 (□), 200/1 (○) and 400/1 (●). The relative error is shown in (b).

ESR spectra. In particular, the MLP works well with experimental spectra and can tolerate noise as long as the *S*/*N* ratio is greater than approximately 200/1. This computational approach has the advantage that, once the network is trained, correlation times can be extracted almost instantaneously from experimental spectra. The approach outlined here should readily generalize to more complicated cases of anisotropic motion and molecular ordering.

METHODS

Training Set Construction

ESR spectra were simulated using the slow motional program of Schneider and Freed (*5*). Values chosen for the magnetic tensors were $g_{xx} = 2.0086$, $g_{yy} = 2.0066$, $g_{zz} = 2.0022$, $A_{xx} = 6.23$, $A_{yy} = 6.23$, $A_{zz} = 35.7$. (Note that these are the same values used in Todd and Millhauser (*16*), with the exception of g_{zz}, which was adjusted to obtain good agreement between slow motional simulations and rigid limit experimental spectra.) Fifteen spectra were generated with isotropic correlation times ranging from 1 to 125 ns. (Training with fewer than 15 spectra resulted in networks that did not give satisfactory results when applied to the set of test spectra.) Of the 15 spectra, 12 had $\tau_R < 50$ ns since spectra within this range are extremely sensitive to correlation time. Simulated spectra for training and testing were spline fit and resampled at 255 equally spaced points along the field axis of 100 G. A matrix of spectra (Fig. 3) was constructed for batch mode training where all of the training vectors are presented simultaneously. Since spectra are obtained in an optimal experimental setting as first derivatives, we decided to do all training and testing with spectra in this representation. However, we also found (data not reported) that the methods reported here applied well to absorption spectra.

All spectra in the training and testing sets were normalized before input to the networks. Two normalization schemes were tried. In the first scheme the spectra were simply adjusted so that the absorption spectrum integral was unity. Although this approach worked well for the MLP, we found that the RB network did not train successfully. Because spectra are fed as vectors, our second approach used simple vector normalization, and this strategy worked well for both networks.

Network Training

All training and testing was performed in the Matlab environment with the aid of the Neural Network Toolbox (*17*) on either a Sun Ultra 1 Sparc workstation or on a Power Macintosh 7100/66 personal computer. For the RB network, the routines SOLVERB and SIMURB were used for training and inputting spectra, respectively. For the MLP, the routines used were INITFF, TRAINBPX, and SIMUFF, respectively, for initializing the weights and biases, training, and inputting spectra. Training times were on the order of minutes for the RB network when run on the personal computer. For the MLP, training required 20 min on the workstation and from 10 to 30 h on the personal computer to reach the required sum squared error.

The testing set refers to simulated spectra with known correlation times that are not part of the training set. The τ_R's of the testing set were chosen using a random number generator uniformly distributed from 1 to 100 ns. In all, 20 simulated spectra were used to test the generalization ability of each network.

Experimental ESR Spectra

Experimental data were obtained from the spin probe 3-carbamoyl-2,2,5,5-tetramethylpyrrolidin-1-yloxy (CTPO) in a 75% v/v glycerol/water mixture. The experimental spectra were acquired over 100 G at a series of temperatures from 243 to 273 K on a Bruker ESP 380 spectrometer with a modulation amplitude of 0.25 G. Normalization was applied as described previously and spectra were adjusted in the field direction so that the characteristic zero crossing in the middle of each spectrum was placed in the middle of the input vector.

Hidden Layer Design for the MLP

As discussed earlier, a single hidden layer was used for the MLP network. A linear output layer was used with only one output unit corresponding to the rotational correlation time τ_R. The number of hidden units was varied and then network performance was tested. The actual numbers of hidden nodes used were 35, 45, and 60. The network did not reach the required sum squared error goal of 0.01 for 30 hidden nodes or less. The network with 45 nodes demonstrated the best ability to generalize and yielded the best correlation between predicted and known values (see Results and Discussion). The network with 45 nodes was optimized to reach a final sum squared error of 0.001.

ACKNOWLEDGMENTS

The authors thank Dr. Kim Bolin for helpful comments on the manuscript. This work was supported by grants from the National Science Foundation (MCB9408284 to GLM) and the National Institutes of Health (GM46870 to GLM and GM16396 to GVM).

REFERENCES

1. L. J. Berliner, "Spin Labeling: Theory and Applications," Academic Press, New York (1976).
2. G. I. Likhtenshtein, "Spin Labeling Methods in Molecular Biology," Wiley, New York (1976).
3. G. L. Millhauser, Selective placement of electron spin resonance spin labels: New structural methods for peptides and proteins, *Trends Biochem. Sci.* **17,** 448–452 (1992).
4. G. L. Millhauser, W. R. Fiori, and S. M. Miick, Electron spin labels, *in* "Methods in Enzymology" (K. Sauer, Ed.), Vol. 246, pp. 589–610, Academic Press, New York (1995).

5. D. J. Schneider and J. H. Freed, Calculating slow motional magnetic resonance spectra: A user's guide, *in* "Biological Magnetic Resonance: Spin Labeling Theory and Applications" (L. J. Berliner and J. Reuben, Eds.), Vol. 8, pp. 1–76, Plenum Press, New York (1989).
6. D. E. Budil, S. Lee, S. Saxena, and J. H. Freed, Nonlinear-least-squares analysis of slow-motion EPR spectra in one and two dimensions using a modified Levenberg–Marquardt algorithm, *J. Magn. Res. A* **120,** 155–189 (1996).
7. S. S. Haykin, "Neural Networks: A Comprehensive Foundation," Macmillan College Publishing, New York (1994).
8. J. U. Thomsen and B. Meyer, Pattern recognition of the H-1 NMR spectra of sugar alditols using a neural network, *J. Magn. Res.* **84,** 212–217 (1989).
9. J. P. Radomski, H. Vanhalbeek, and B. Meyer, Neural network-based recognition of oligosaccharide H-1 NMR spectra, *Nature: Struct. Biol.* **1,** 217–218 (1994).
10. S. A. Corne, A. P. Johnson, and J. Fisher, An artificial neural network for classifying cross peaks in 2-dimensional NMR spectra, *J. Magn. Res.* **100,** 256–266 (1992).
11. G. Bohm, R. Muhr, and R. Jaenicke, Quantitative analysis of protein far UV circular dichroism spectra by neural networks, *Protein Engineering* **5,** 191–195 (1992).
12. N. Sreerama and R. W. Woody, Protein secondary structure from circular dichroism spectroscopy—combining variable selection principle and cluster analysis with neural network, ridge regression and self-consistent methods, *J. Mol. Biol.* **242,** 497–507 (1994).
13. P. Pancoska, V. Janota, and T. A. Keiderling, Interconvertibility of electronic and vibrational circular dichroism spectra of proteins—a test of principle using neural network mapping, *Appl. Spectrosc.* **50,** 658–668 (1996).
14. K. Tanabe, T. Tamura, and H. Uesaka, Neural network system for the identification of infrared spectra, *Appl. Spectrosc.* **46,** 807–810 (1992).
15. S. Chen, C. F. N. Cowan, and P. M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, *IEEE Trans. Neural Networks* **2,** 302–309 (1991).
16. A. P. Todd and G. L. Millhauser, ESR spectra reflect local and global mobility in a short spin-labeled peptide throughout the α-helix $\rightarrow$ coil transition, *Biochemistry* **30,** 5515–5523 (1991).
17. H. Demuth and M. Beale, "User's Guide: Neural Network Toolbox," The Mathworks, Inc., Natick, MA (1995).