The Feasibility of Using Neural Networks to Obtain Cross Sections from Electron Swarm Data

W. Lowell Morgan

***Abstract*—Although still more a curiosity than an accepted technique in computational modeling, the very new field of neural computing is beginning to find applications in physics. Presented is some background on neural computing and a discussion on the use of neural networks to obtain electron-impact cross sections from measured drift velocities, characteristic energies, and other swarm data. This is what is known as an "inverse problem," a class of problems for which neural networks may be frequently superior to other numerical algorithms. Momentum transfer cross sections obtained for a model problem and for xenon using a neural network are presented.**

I. Introduction

THE use of theory to obtain collision cross sections from electron transport data, one of the "inverse problems" of physics, was pioneered by Townsend and by Ramsauer in the 1920's. The method used in such early analyses involved measuring the drift velocity of electrons in a gas as a function of E/p (electric field strength divided by gas pressure) and inverting the integral relating the drift velocity and the momentum transfer cross section using an approximate expression for the energy distribution of the electrons. This technique has increased in sophistication over the years. In the 1960's, Phelps and various collaborators applied electronic computation to the problem and developed algorithms for solving Boltzmann's equation for the transport of electrons in a weakly ionized plasma to obtain an accurate electron energy distribution function valid at higher fields and in the presence of inelastic and, even, superelastic collisions. This began an era that has given us very accurate momentum transfer and lower energy (rotational and vibrational) inelastic cross sections that have been derived from measurements of the drift and diffusion of electron swarms in gases. This methodology is reviewed in [1]–[4]. This has become an increasingly active field in recent years due, for example, to the desire for cross-sectional data on molecules such as CH_4, CF_4, SF_6, SiH_4, and SiF_4 that are used in semiconductor plasma processing and switching applications.

The cross sections are fundamental quantities depending only on the energy of the incident electron and the particular initial and final atomic or molecular states. The swarm parameters, such as drift velocity, diffusion coefficients, and excitation coefficients, are derived quantities, depending on the local environment of the electron swarm. They are integrals over the product of a cross section and the energy distribution $f(\epsilon)$ of the electrons, which is the solution to Boltzmann's transport equation.

The process of inverting the swarm data to obtain cross sections has involved inserting cross-section models in the collision terms of Boltzmann's equation, calculating $f(\epsilon)$ and the swarm coefficients, altering the model cross sections, and iterating until an acceptable match between measured and computed swarm coefficients is found. Clearly, this iterative process is very labor intensive. The experience of the researcher plays an important role, comparable to that of the specific computational techniques used. The use of numerical optimization algorithms has been investigated as a means of aid in obtaining cross sections from electron swarm data. The desirability of such a procedure has been noted by Garscadden [5]. The results obtained for He, Ar, and CH_4 using conventional numerical optimization algorithms, such as the creeping or downhill simplex and simulated annealing [6], are presented elsewhere [7]. In this paper is described the use of a *neural network* to explore the mapping between swarm coefficients and cross sections. Results are presented for a model problem and for the xenon momentum transfer cross section.

The pertinent equations in this problem are Boltzmann's equation for electron transport in a gas and its energy integral, which leads to the definitions of the various transport coefficients for electrons in a gas. We can see how the various aspects of this problem are related to each other by examining the so-called two-term expansion of Boltzmann's equation and the various electron transport coefficients. If we take the general form for Boltzmann's equation:

$$\left(\partial/\partial t + \mathbf{v}\cdot\nabla_r + \frac{e\mathbf{E}}{m}\cdot\nabla_v\right) f(\mathbf{r},\mathbf{v},t) = (\partial f/\partial t)_{\text{collisions}} \quad (1)$$

neglect the spatial and temporal dependence of the distribution function $f(\mathbf{r},\mathbf{v},t)$, and express $f = f(\mathbf{v})$ as the first two terms of a spherical harmonic expansion, that is:

$$f(\mathbf{v}) = f_0(v) + \frac{\mathbf{v}}{v}\cdot \mathbf{f}_1(v) \quad (2)$$

then we obtain (see [2] for details of the derivation) the following scalar equation for $f_0(\epsilon)$(where $\epsilon = mv^2/2$):

$$\begin{aligned}&\frac{1}{3}(eE/N)^2 d/d\epsilon\{\epsilon/\sigma_m df_0/d\epsilon\} + d/d\epsilon\\ &\{(2m\sigma_m/M)\epsilon^2[f_0(\epsilon) + kt\, df_0/d\epsilon]\}\\ &+\sum_i[(\epsilon+\epsilon_i)\sigma_i(\epsilon+\epsilon_i)f_0(\epsilon+\epsilon_i) - \epsilon\sigma_i(\epsilon)f_0(\epsilon)] = 0\,. \quad (3)\end{aligned}$$

Here, we have assumed that the populations of the excited levels, labeled by i, are small enough that superelastic collisions and transitions among excited states are unimportant. The electron impact cross sections involved are σ_m, the momentum transfer cross section, and $\{\sigma_i\}$, the set of cross sections for transitions from the ground state to the various excited states i. This equation does a remarkably good job of describing the transport of electrons under the influence of an electric field in most gases. This, or more a sophisticated version of it that goes beyond the two-term expansion, is the equation that is repeatedly solved in the iterative process, whereby one develops a set of cross sections given a collection of transport coefficients.

Manuscript received June 22, 1990; revised September 28, 1990. This work was supported by the Wright Research and Development Center, Wright-Patterson Air Force Base.

The author is with Kinema Research, 18720 Autumn Way, Monument, CO 80132

IEEE Log Number 9143147.

0093-3813/91/0400–0250$01.00

Boltzmann's equation, via the probability density function $f_0(\epsilon)$, is a microscopic description of the behavior of electrons in a gas. We need to relate $f_0(\epsilon)$ to some macroscopic quantities that can be measured. The two most commonly measured transport coefficients are the drift velocity, v_d, and the tranverse diffusion coefficient, D_T, which are related to $f_0(\epsilon)$ and the momentum transfer cross section, $\sigma_m(\epsilon)$ by the following:

$$v_d \approx \int [\sigma_m(\epsilon)]^{-1}(df_0/d\epsilon)\epsilon d\epsilon, \quad D_T \approx \int [\sigma_m(\epsilon)]^{-1} f_0(\epsilon)\epsilon d\epsilon. \tag{4}$$

We see that the drift velocity and diffusion coefficient sample different aspects of $f(\epsilon)$ and hence represent two somewhat independent pieces of information. Generally, the quantity D_T/μ, the *characteristic energy*, is reported in the literature rather than D_T itself. For a Maxwellian distribution of electrons, where $f_0(\epsilon) \approx \exp(-\epsilon/kT_e)$ for which an electron temperature, T_e, can be defined, Einstein's relation $D_T/\mu = kT_e = 2 < \epsilon > /3$ holds. Since the mean electron energy $< \epsilon >$ is not a measureable quantity (it is usually computed by solving Boltzmann's equation), the characteristic energy is generally the only measure of electron energy that we have. We see that comparison with measured D_T/μ values gives us another constraint on the cross sections as does comparison with measured rate coefficients $k_i \approx \int \sigma_i(\epsilon) f_0(\epsilon)\epsilon d\epsilon$ and spectral data where they are available.

The relationship between the cross sections and the transport coefficients via the distribution function $f_0(\epsilon)$ is highly nonlinear. We have a mapping:

$$\begin{Bmatrix} \sigma_m(\epsilon) \\ \{\sigma_i(\epsilon)\} \end{Bmatrix} \Rightarrow \begin{Bmatrix} v_d(E/N) \\ D/\mu(E/N) \\ \{k_i(E/N)\} \end{Bmatrix} \tag{5}$$

and we want to find the reverse mapping, given the transport data. It has been claimed in the literature (see [8], for example) that the reverse mapping may not be unique. It seems likely that the more transport data we have available, the more likely it is that the reverse mapping is going to be unique.

There are numerous techniques [6], [9], [10] that might be used in inverting electron transport data to obtain a collision cross section. The use of conventional numerical optimization algorithms is described elsewhere [7]. Artificial neural networks (ANN's), which are still more a curiosity than an accepted technique in computational modeling, do not fit into any of the usual categories for numerical algorithms. This approach is very new and largely unknown in applications to physical problems. Initially the author applied the technique to a model problem and then extended it to estimate the momentum transfer cross section of xenon from the measured drift velocities and characteristic energies of electrons in xenon.

II. Neural Networks

Neural networks [11]–[14], which consist of layers of simulated "neurons" with associated activation functions, transfer functions, and weighting functions for the "synapse" connections to other neurons, have been shown [10], [12], [15], [16] to be capable of computing decisions in optimization problems. Such networks have a "learning" capability in that the weights associated with connections between pairs of neurons can be modified (strengthened or weakened) in response to the network's successes and failures so as to optimize in favor of the network's successful strategies. Jeffrey and Rosner [16] find neural networks to be much faster at function minimization and solving an integral equation than simulated annealing and to be competitive with the conjugate gradient algorithm. Aarts and Korst [17] have found that on some graph problems, the neural network approach is from 20 to 400 times faster than the simulated annealing method, another optimization algorithm.

One kind of neural network consists of a network of layers of simulated neurons as shown in Fig. 1. The key elements are an input layer, one or more "hidden" layers, and an output layer. Each neuron has a transfer function associated with it that gives an output value that is some nonlinear function of the sum of the input values, and each pair of neurons has a weight value associated with it. The hidden layers give the network a high degree of nonlinearity and, from the point of view of the network as a pattern matcher, provide an internal representation of the correlation between the input and output patterns. The concept behind this kind of network (feed-forward, back-propagation) is that it can "learn" to associate a set of output patterns with a set of input patterns by adjusting the weights that connect together the network of nonlinear devices. One might also think of a neural network as a highly flexible, many parameter interpolation and hence extrapolation algorithm.

The usual transfer function used in such networks is the sigmoid $T(x) = 1/(1+e^{-x})$ (there is an equivalent arctan function also). If the output of a neuron j is o_j and w_{ij} is the weight connecting neurons i and j, then the output of neuron i is:

$$o_i = T\left(\sum_j w_{ij} o_j\right) = 1/\left(1 + e^{-\sum_j w_{ij} o_j}\right) \tag{6}$$

where the sum is over all neurons j having outputs that feed into neuron i. The network is trained by running a number of cases of known {input,output} sets through it and adjusting the weights to minimize the sum of the squares of the differences between the desired result and the computed result. This quadratic function is the so-called *energy*, *cost*, or *objective* function. The weights are adjusted using what is known as the generalized delta rule [14], [18]. The energy function, E, can be written:

$$E = \sum_p \sum_i [t_i(p) - o_i(p)]^2 \tag{7}$$

where $t_i(p)$ is the value of the training output for output neuron i and training data set p; and $o_i(p)$ is the corresponding network output. E is minimized using an iterative procedure whereby the weights w_{ij} are adjusted according to:

$$\Delta w_{ij} = \sum_p \epsilon \delta_i(p) o_j(p) \tag{8}$$

where $\delta_i(p) = [t_i(p) - o_i(p)]o_i(p)[1 - o_i(p)]$ for output layer neurons, and $\delta_i(p) = o_i(p)[1 - o_i(p)] \sum_j w_{ij}\delta_j(p)$ for hidden layer neurons. The parameter ϵ is called the training rate coefficient. The weights are adjusted, starting with the neurons in the output layer and moving back a layer at a time toward the input layer.

The ANN will assimilate and correlate data given it in the process of "training" so that when given input vectors outside of the training set, it will produce reasonable values for the corresponding output vectors. The matrix of weights, w_{ij} clearly represents the mapping between the set of input vectors and the set of output vectors.

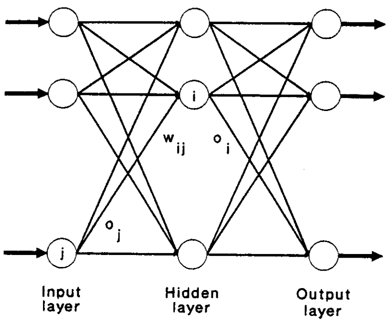

Fig. 1. Three-layer feed-forward, back-propagation neural network. The neurons, denoted by circles, in a given layer operate via a transfer function on the weighted sum of the output from neurons in the previous layer to produce an ouput signal that is passed on to the neurons in the next layer. That is, $o_i = T\left(\sum_j w_{ij} o_j\right)$.

A. Application to the Problem of Obtaining Cross Sections from Swarm Data

In order to explore the feasibility of using neural networks on this problem, the author has used a commercial neural net simulator called BRAINMAKER [19]. This is one of a number of such commercially available programs (see [20] and [21]). In this case, the input neurons represent swarm coefficients as functions of E/N, and the output neurons represent the momentum transfer cross section as a function of electron energy. The neural network is schematically diagrammed in Fig. 2.

Initially the author chose to study a model problem. The two obvious examples to use as models are those of electrons drifting in a gas having either constant collision frequency or constant cross section. Here, the electron energy distributions are Maxwell-Boltzmann and Druyvesteyn, respectively, with easily calculable E/N dependent drift velocities. Any method should be able to use the v_d (E/N) curve and the appropriate $f_0(\epsilon)$ and recover $\sigma \approx 1\sqrt{\epsilon}$ or $\sigma \approx \sigma_o$, respectively, for these two special cases. The author investigated the capability of the neural network to reproduce the constant collision frequency cross section, $\sigma = \sigma_o/\sqrt{\epsilon}$ using the drift velocity, v_d (E/N) and characteristic energies D/μ (E/N) associated with that cross section. The resulting electron energy distribution functions, $f_0(\epsilon), v_d, D/\mu$, and $< \epsilon >$ are all analytic functions [22].

The author wrote a program to generate cross-section sets of the form $\sigma(\epsilon) = \sigma_0/\epsilon^P$, where σ_0 and p are chosen from uniform random numbers in (10^{-17}, 10^{-14}) and (0,1) respectively, and then compute for a range of E/N the distribution function $f(\epsilon)$ and the associated drift velocities, v_d, and characteristic energies, D/μ. He then set up a training set for BRAINMAKER that consisted of the sets $\{v_d\}$ and $\{D/\mu\}$ for ten values of E/N and the cross section $\sigma(\epsilon)$ at nine energies from which the swarm data were computed. The input layer of the network then consists of 20 neurons, one for each value of $v_d(E/N)$ or $D/\mu(E/N)$. The output layer comprises 9 neurons, one for each cross section point $\sigma(\epsilon_i)$, i=1 to 9. The network has two hidden layers of 25 neurons each. The network is summarized in Table I. The training tolerance was 5%, meaning that for the network to be acceptable the cost/energy/objective function of the difference between the $\{\sigma(\epsilon)\}$ defined by the values of the output neurons and the "data" given to the network as part of the training pattern had to be less than or equal to 0.05.

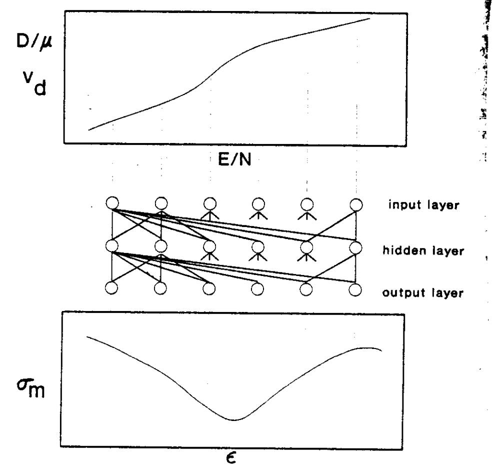

Fig. 2. Schematic diagram of the relationship between the swarm data input, the neural network, and its cross section ouput.

TABLE I

Layer	*Neurons*	*Weights*
1	20	0
2	25	525
3	25	650
4	9	234

Once the network was "trained" it was given another file of sets $\{v_d\}$ and $\{D/\mu\}$ corresponding to different sets $\{\sigma(\epsilon_i)\}$ computed with random σ_r and p to see what it predicted for the cross sections. These results are shown in Fig. 3 for the three best (out of 11) cases. These illustrate several things: First, the results denoted by the circles are very good. They all diverge at high energy, because the highest E/N that was used (3.0 Td) was too small for v_d and D/μ to be adequately sensitive to the high energy part of the cross section. In addition, it was observed that the results for large cross sections are generally better than the results for small cross sections. The author thinks that this is due to a numerical range problem with BRAINMAKER that is attributed to its being single precision; it was not really designed for computation on scientific problems where the range of numbers to be dealt with may cover many orders of magnitude.

B. Application to a Real Gas

In order to test the feasibility of using this kind of neural network to find the pattern in the mapping between $\{v_d(E/N), D/\mu(E/N)$, etc.$\}$ and $\{\sigma(\epsilon)\}$ BRAINMAKER was trained on 25 sets of $\{v_d(E/N), D/\mu(E/N)\}$ data for cross sections of the form: $\sigma(\epsilon) = \sigma_o\epsilon^P$, where $-1 \leq p \leq +1$. That is, we have some cross sections that increase with energy, and some that decrease with energy. The author then constructed an input

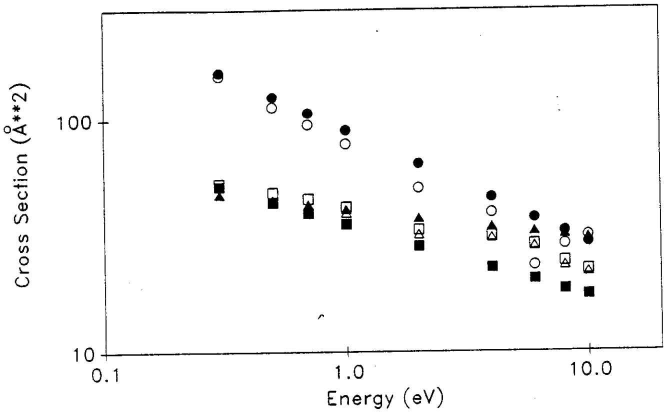

Fig. 3. Neural network results for the model problem. The filled symbols are the cross section "data" of the form, $\sigma(\epsilon) = \sigma_o/\epsilon^P$ from which the swarm parameters were computed that were fed into the network, and the open symbols are the output from the network. The values of (σ_o, p) are: Circles, (90, 0.49); triangles, (48, 0.12); squares, (36, 0.31).

set for Xe with $\{v_d(E/N)\}$ from Hunter, *et al.* [23] and $\{D/\mu\,(E/N)\}$ from Koizumi, *et al.* [24]. Unfortunately, neither paper presented *both* drift velocity *and* characteristic energy data. It was found that a reasonable result was obtainable with a network of only three layers (see Table II). How many neurons and layers one needs for a given problem is discussed in the concluding section.

The cross section that the neural network returned in the output layer for xenon in the energy range around the Ramsauer minimum is shown in Fig. 4, along with the $\sigma_m(\epsilon)$ from [23]–[25]. These author's results were derived from either drift velocity or characteristic energy measurements. We see that the neural network gives a respectable estimate of the cross section, even though the number of E/N values is small and the energy grid is very coarse. This is a stringent test, because of the Ramsauer minimum; the cross section is not monotonically decreasing or increasing, as are the training data.

III. Conclusion

The use of an artificial neural network as an optimization technique for treating the inverse problem of obtaining electron collision cross sections from electron transport data has been explored. The neural network approach to problems such as these is very new and has not yet had much application to the problems of applied physics. This approach has demonstrated some capability in addressing the swarm data inversion problem.

The author believes that the neural network approach is worthy of further exploration. The next step would be to write a network for this problem with larger numbers of neurons (the input and output of version 1.5 of BRAINMAKER used in these calculations was very tedious, which is why such a small number of input and output neurons were used); double precision arithmetic; a capability for having different transfer functions for different layers; allowing different convergence criteria for different energy ranges; and, perhaps, using the Boltzmann training algorithm [14], [26], [27], which is an application of simulated annealing to adjustment of the weights of the connections in the network. Such a network is called a *Boltzmann machine* and has the capability of escaping a local minimum in the energy function should the network settle into one.

TABLE II

Layer	*Neurons*	*Weights*
1	18	0
2	20	380
3	9	189

There is little in the way of previous experience in applying neural networks to the problems of physics to serve as guidence in constructing neural networks for problem such as this. One finds in the literature rules of thumb, such as rarely needing more than two hidden layers or using for the number of hidden layer neurons the average of the number of input layer and ouput layer neurons. There are numerous training algorithms that take varying amounts of processor time and memory, but the only way of determining which ones work best is to try them. The calculations in this paper were performed on a 25 MHz 80386/87 personal computer. Typical training times were between a half hour and an hour, which, depending on the number of neurons in the network, corresponds to hundreds to several thousand cycles through the set of training data. Of course, for a given input the computation of the network output takes only a fraction of a second.

It would be interesting to investigate how a neural network would perform when trained on a completely artificial data set (probably using more sophisticated functions for the training cross sections than I have used here), as compared to training by feeding it a large set of data on real atoms and molecules. There is also the issue of how well a neural network will perform for a molecule having a momentum transfer cross section and one or more inelastic cross sections. This problem has been addressed [7] using more conventional optimization algorithms and has given very encouraging results for the momentum transfer cross section and a vibrational excitation cross section in CH_4.

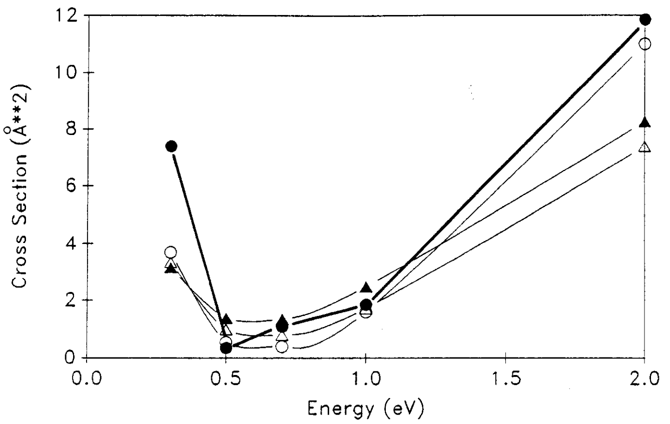

Fig. 4. Xe momentum transfer cross section. The filled circles are the output from the neural network; the open triangles are from Hunter *et al.* [23]; the open circles are from Koizumi *et al.* [24]; and the filled triangles are from Frost and Phelps [25].

Ultimately, we may find that a neural network is a good means of getting a rough estimate of a cross section $\sigma(\epsilon)$ that can then be refined using another numerical optimization algorithm. The conventional wisdom has been that neural networks are useful for only very rough solutions and not for accurate scientific calculations, but some authors, such as Lapedes and Farber [18] refute that point of view. Indeed, the claim has been made [28] that neural networks are "formally capable of solving any conventional computational problem." As this area of research is very much in its infancy, we can expect many new developments in the understanding of neural networks and in their applications to the problems of physics in the future.

ACKNOWLEDGMENT

This research was suggested by Dr. A. Garscadden, who has had a long standing interest in this kind of approach to swarm studies. The electron swarm and cross-section data used in this research came from the atomic data base of the Atomic Collisions Data Center at the Joint Institute for Laboratory Astrophysics of the University of Colorado and the National Institute of Standards and Technology.

REFERENCES

[1] A. V. Phelps, "Rotational and vibrational excitation of molecules by low energy electrons," *Rev. Mod. Phys.*, vol. 40, pp. 399–410, 1968.
[2] I. P. Shkarofsky, T. W. Johnston, and M. P. Bachynski, *The Particle Kinetics of Plasmas.* Reading, MA: Addison-Wesley, 1966.
[3] L. G. H. Huxley and R. W. Crompton, *The Diffusion and Drift of Electrons in Gases.* New York: Wiley, 1974.
[4] A. V. Phelps "Relations between electron-molecule scattering and swarm experiments and analysis," in *Swarm Studies and Inelastic Electron Molecule Collisions*, L. C. Pitchford, B. V. McKoy, A. Chutjian, and S. Trajmar, Eds. New York: Springer-Verlag, 1987, pp. 127–141.
[5] A. Garscadden, "Current applications and opportunities in swarm studies," in *Swarm Studies and Inelastic Electron Molecule Collisions*, L. C. Pitchford, B. V. McKoy, A. Chutjian, and S. Trajmar, Eds., New York: Springer-Verlag, 1987, pp. 409–423.
[6] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes.* Cambridge, UK: Cambridge Univ. Press, 1986.
[7] W. L. Morgan, "The use of numerical optimization algorithms to obtain cross sections from electron swarm data," *Phys. Rev. A*, to be published.
[8] M. J. Kushner, "A model for the discharge kinetics and plasma chemistry during plasma enhanced chemical vapor deposition of amorphous silicon," *J. Appl. Phys.*, vol. 63, pp. 2532–2551, 1988.
[9] W. Jeffrey and R. Rosner, "On strategies for inverting remote sensing data," Ap. J., vol. 310, pp. 463-472, 1986.
[10] W. Jeffrey and R. Rosner, "Optimization algorithms: simulated annealing and neural network processing," *Astrophys. J.*, vol. 310, pp. 473–481, 1986.
[11] J. A. Anderson and E. Rosenfeld, Eds., *Neurocomputing: Foundations of Research.* Cambridge, MA: MIT Press, 1988.
[12] J. Denker, Ed., *Neural Networks for Computing.* New York: Amer. Instit. Phys., 1986.
[13] R. Hecht-Nielsen, *Neurocomputing.* Reading, MA: Addison-Wesley, 1990.
[14] P. D. Wasserman, *Neural Computing.* New York: Von Nostrand Reinhold, 1989.
[15] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biological Cyber.*, vol. 52, pp. 141–152, 1985.
[16] W. Jeffrey and R. Rosner, "Neural network processing as a tool for function optimization," in *Neural Networks for Computing*, J. S. Denker, Ed., New York: Amer. Instit. of Phys., 1986, pp. 241–246.
[17] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines.* New York: Wiley, 1989.
[18] A. Lapedes and R. Farber, "How neural nets work," in *Neural Information Processing Systems*, D. Z. Anderson, Ed. New York: Amer. Instit. Phys., 1988, pp. 442–456.
[19] California Scientific Software, 10145 Evening Star Dr. #6, Grass Valley, CA 95945.
[20] "Neural networks: theory and practice," *Byte*, pp. 244–245, Aug. 1989.
[21] D. Stubbs, "Software review," *AI Expert*, pp. 61–63, June 1990.
[22] B. M. Smirnov, *Physics of Weakly Ionized Gases.* Moscow: MIR, 1981.
[23] S. R. Hunter, J. G. Carter, and L. G. Christophorou, "Low energy electron drift and scattering in Kr and Xe," *Phys. Rev. A*, vol. 38, pp. 5539–5551, 1988.
[24] T. Koizumi, E. Shirakawa, and I. Ogawa, "Momentum transfer cross sections for low energy electrons in Kr and Xe from characteristic energies," *J. Phys. B*, vol. 19, pp. 2331–2342, 1986.
[25] L. S. Frost and A. V. Phelps, "Momentum transfer cross sections for slow electrons in He, Ar, Kr, and Xe from transport coefficients," *Phys. Rev.* vol. 136, pp. A1538–A1545, 1964.
[26] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
[27] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in

Boltzmann machines," in *Parallel Distributed Processing*, vol. 1, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 282–317.

[28] Y. S. Abu-Mostafa, "Neural networks for computing?," in *Neural Networks for Computing*, J. S. Denker, Ed. New York: Amer. Instit. Phys., 1986, pp. 1–6.

W. Lowell Morgan was born in Royal Oak, MI in 1946. He received the B.S. degree in engineering physics from the University of Michigan, the M.A. degree in physics from Wayne State University, and the Ph.D. degree in atomic and molecular physics from the University of Windsor. He conducted postdoctoral research on nonequilibrium low-pressure plasmas at the Joint Institute for Laboratory Astrophysics (JILA) of the University of Colorado from 1976 through 1978.

From 1978 through 1987 he was a Physicist at the Lawrence Livermore National Laboratory, where he variously worked on modeling the plasma chemistry of excimer lasers; was a Group Leader for non-LTE atomic modeling in the X-ray laser program; and worked on simulating physical and chemical processes on surfaces in the Chemistry and Materials Science Department. He was also a Lecturer in statistical mechanics in the Department of Applied Science of the University of California, Davis. He has been a Visiting Scientist at AT&T Bell Laboratories (1986) and at JILA (1987–1988). He was the Acting Director of the Atomic Collisions Data Center at JILA during 1988–1989. In 1989 he founded Kinema Research as an outgrowth of his research and consulting activities in applied and computational physics. He is also an Adjunct Professor of physics at the University of Denver. He specializes in modeling and simulating complex nonequilibrium physical and chemical processes in gases, plasmas, liquids, and on surfaces. His current research involves the application of neural networks and of cellular automata to problems in physics and chemistry.