Applications of Neural Networks in Chemistry. 1. Prediction of Electrophilic Aromatic Substitution Reactions†

DAVID W. ELROD* and GERALD M. MAGGIORA

Computational Chemistry, Upjohn Laboratories, Kalamazoo, Michigan 49001

ROBERT G. TRENARY

Department of Computer Science, Western Michigan University, Kalamazoo, Michigan 49008

Received May 23, 1990

A back-propagation neural network was trained on connection table representations of monosubstituted benzenes to predict the products of electrophilic aromatic substitution. Ten of 13 unknown test reactions and all 32 of the training cases were correctly predicted. With an alternative charge vector representation, 8 of 13 test cases were predicted correctly. Neural networks differ from expert systems by constructing reactivity rules implicitly from examples rather than by explicitly including rules in the expert system. The results obtained by using neural networks were comparable to those obtained from an existing chemical expert system and to predictions made by synthetic organic chemists.

INTRODUCTION

Computer-aided organic synthesis (CAOS) methods have made considerable progress during the past 20 years but are seldom used by laboratory chemists.[1] One of the reasons is that chemists are still better at solving many of the problems in chemistry than are computers. This especially applies to problems that require judgement, extrapolation from analogous situations, or the application of multiple, poorly defined or conflicting rules. Such situations are encountered in designing the synthesis of new compounds or in predicting the reactivity of complex chemical systems. Traditional computer methods excel at solving problems that require many numerical calculations or the generation of hundreds of possible solutions, but are poor at drawing inferences or making generalizations from limited data. In contrast, artificial intelligence (AI) based approaches can make limited judgments and inferences by applying a set of "rules" or heuristics modeled on those used by expert chemists.

The AI-inspired CAOS programs generally use one of three methods to apply chemical knowledge to synthesis problems: (1) a library of reactions or transforms, as in the LHASA[2] program, (2) mathematical methods to generate all possible products or precursors, as in IGOR[3] and SYNGEN,[4] or (3) mechanistic rules governing reaction types, as in CAMEO.[5] All of these methods have had some success, but are limited by the requirement that rules governing reactivity must be stated explicitly or heuristics must be devised to reduce the number of possibilities to manageable size. Rule-based systems have the advantage that one may query the system to determine how the rules have been applied to achieve a given result. However, generation of these rules and heuristics from the chemical literature or from interviews with expert chemists is a tedious process that may limit the rate at which progress can be made. A more "automated" method of extracting this knowledge would benefit not only CAOS methods but also the synthetic chemists who ultimately use them.

Neural networks[6] are a promising new method for solving chemical problems by virtue of their ability to construct an internal representation of the problem which allows predictions to be made for similar problems. Neural networks may be particularly useful in cases where it is difficult to specify exact rules governing reactivity or where several overlapping or seemingly opposing rules apply. Examples of such cases are seen in retrosynthetic analysis, and in the prediction of reaction products and metabolic transformations where steric, electronic, resonance, and solvent effects compete with or reinforce each other. In many chemical processes there is a wealth of experimental data but a scarcity of rules to define the process. In such cases neural networks, which employ learning procedures to develop internal representations from examples,[7] may be able to discern patterns in the data which would allow reasonable predictions to be made.

Although neural networks have not yet gained widespread use in chemistry, they have been extensively studied by computer and cognitive scientists.[6-8] Reported applications of neural networks in chemistry have been limited to prediction of secondary structure in proteins,[9] pattern recognition in NMR spectroscopy,[10,11] classification of mass spectral patterns using a less powerful type of neural network called a Perceptron,[12] and the prediction of drug safety from physical and chemical parameters.[13]

The general goals of the present study are twofold: to extend the use of neural networks to the prediction of chemical reactions and to investigate the representation of chemical information in neural networks. Specifically, electrophilic substitution was chosen since it is a well-studied reaction and since there exists a reasonably complete and consistent set of data for the reaction.

Electrophilic aromatic substitution (EAS) involves the substitution of a hydrogen atom on an aromatic ring by an electrophile such as nitric acid, to give ortho-, meta-, or para-substituted products, as shown in Figure 1. The ratio of isomers formed depends mainly on the nature of the substituent X and to a lesser degree on the electrophile Y. Substituents may be divided into two classes, ortho–para directors, which tend to be electron donors, and meta directors, which tend to be electron acceptors. Resonance effects of the substituent can reinforce or oppose these inductive effects and thereby affect the product ratio. Steric hindrance by large substituents also can affect the reaction by blocking the adjacent ortho positions.

Chemists can usually predict the major products in EAS simply by looking at the substituent and classifying it as either ortho–para or meta directing without any attempt to predict percentages of each product. It is usually implied that the products formed are only ortho–para or only meta, but it is

† A preliminary report on these studies was presented at the 197th National Meeting of the American Chemical Society, Dallas, TX, April 12, 1989; Abstract COMP 27.

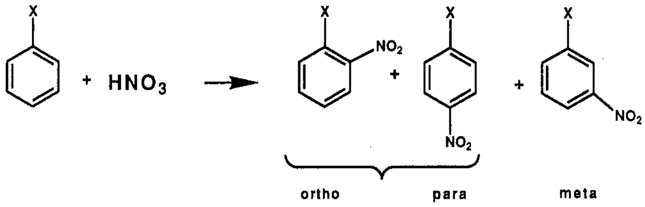

Figure 1. Electrophilic aromatic substitution.

apparent from examining the chemical literature that many intermediate cases exist which give mixtures of all three products. In these cases, it would be useful to have a prediction of the ortho–para to meta product ratio.

When investigating the applicability of neural network methods in general and in chemistry in particular, two issues must be addressed at the outset: the choice of network paradigm and the type of data representation. In the present work, a back-propagation network (vide infra) was chosen based on its pattern mapping ability. Further discussion of back-propagation is found under Neural Networks. Data representation is crucial as an inappropriate data representation is likely to severely reduce the predictive ability of a network. Since no prior knowledge of chemistry is present in an untrained network, the network must extract relevant features from training examples, encoded in numeric form, in order to make predictions. Consequently, finding a scheme for encoding the relevant chemical information into a format which is suitable for use in a back-propagation network was an important feature of the current investigation, and one that lies at the heart of essentially all applications of neural networks to chemistry.

NEURAL NETWORKS

Neural networks, also known as neurocomputers, parallel distributed processing (PDP) models, or connectionist models, are a computational method based on a idealized model of the brain, where a large number of simple processing units, analogous to neurons, are extensively interconnected to form a highly parallel computer. Humans have the ability to perform certain tasks, such as recognition of faces or "calculating" the velocity and trajectory of a tennis ball and then determining the body movements required to return a tennis volley, much faster than supercomputers. It may take a digital computer billions of processing steps to accomplish what the brain can do in about 100 steps. The key to human superiority at these tasks seems to reside in the brain's ability to integrate information in a parallel manner.[14] Much of the current interest in neural networks stems more from their attempt to exploit this parallelism than from providing a realistic model of the brain. Most applications of neural networks have employed simulations run on single processor computers because highly parallel hardware implementations are not readily available. Hence, the benefits obtainable from parallelism have not been fully realized. Nevertheless, considerable interesting and novel work has been accomplished.

Besides parallelism, there are several other properties of neural networks that make them useful in chemistry. One property is their ability to represent information by a pattern of activity distributed over many processing units, instead of by allocating a single computational unit for each item of information, as is the case in a local data representation.[8a] This distributed representation leads to the ability to generalize to new situations and fosters adaptability. Adaptation or learning is another property that is a major benefit of neural network methods. The ability to improve performance with increased training is an important feature of neural networks which is quite valuable in chemical applications where new knowledge and examples can be incorporated in the network relatively easily. Neural nets are thus significantly different from both traditional computer programming in procedural languages like Fortran and from AI programming where rules, frames, and schema are constructed. In a neural net based expert system the rules are not explicitly programmed but rather are generated by the network directly from the data. Neural nets are also related to statistical data analysis methods[8] but have the added advantages of being adaptive and nonparametric.[7]

Network Architecture. Three types of components comprise a neural net, the processing elements or nodes, the topology of the connections between the nodes, and the learning rule by which new information is encoded in the net. The information contained in a neural network is "stored" entirely in the strengths of the connections between the processing units and not in the units themselves.

Many different network topologies have been explored.[7,8] Figure 2 illustrates the type of network used in the present study. This multiple-layer network topology has been found to be the most useful one for the bulk of practical applications.[9-13] The processing units are arranged in distinct layers, designated by the subscript λ, with each unit connected to every unit in the adjacent layers. There are no connections within layers and no connections that bypass the intervening layers. Each connection is represented by a specific weight, W_{ij}, which initially is chosen randomly and subsequently changed by the learning rule to improve network performance. A zero weight implies no effect, i.e., no connection between nodes. Positive weights correspond to excitatory connections and negative weights to inhibitory connections. The bottom or input layer ($\lambda = 1$) receives the input signal from the environment and passes it on to the next layer. The middle layer ($\lambda = 2$) is called the hidden layer because it is "hidden", i.e., not connected, to anything outside of the network. It is

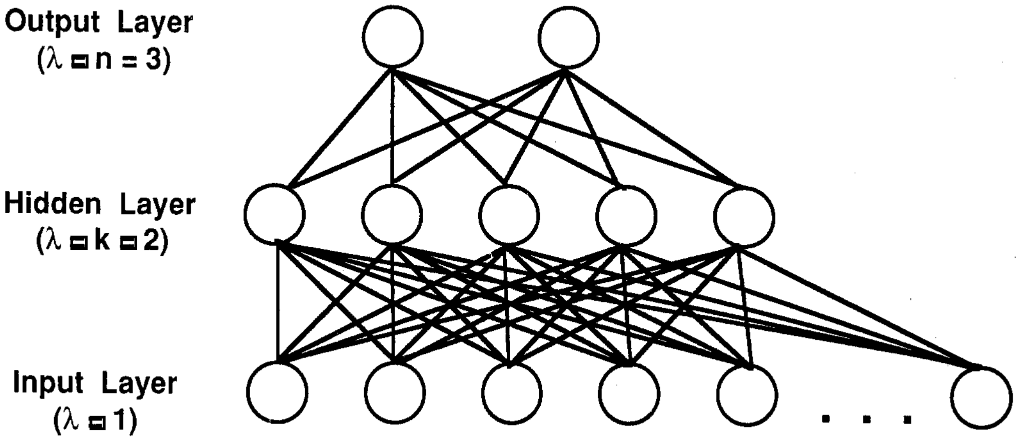

Figure 2. Three-layer feed-forward neural network.

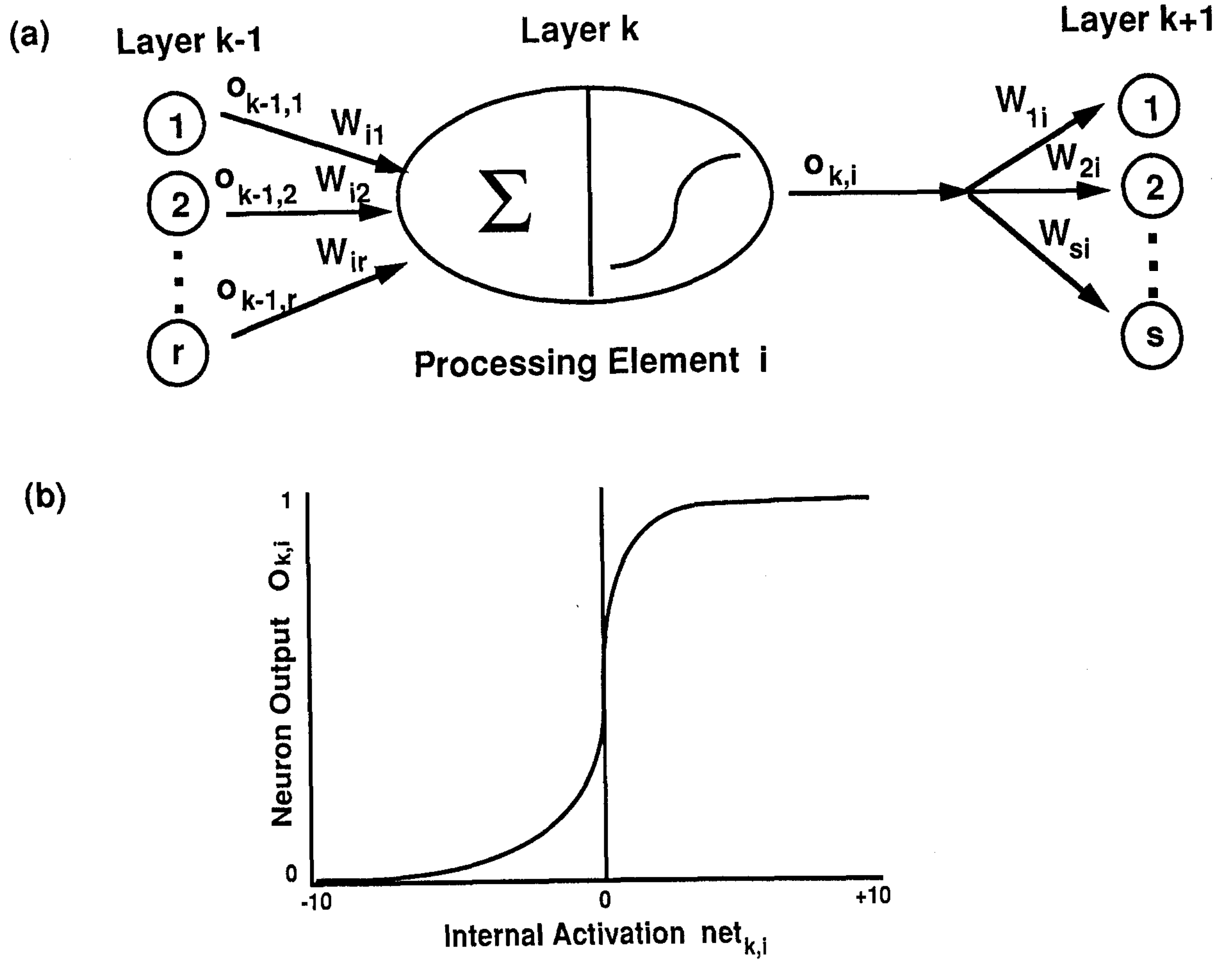

Figure 3. (a) Computational element or neuron whose output, $O_{k,i}$, is a function of the weighted sum of its inputs. (b) Plot of the logistic sigmoid activation function used to compute $O_{k,i}$ in part a.

the hidden layer and the nonlinearity of the activation function, defined in eq 2, that give neural networks much of their power. The top layer ($\lambda = n = 3$) is the output layer which displays the result computed by the network. Information flow in the network depicted in Figure 2 is from bottom to top: networks of this type are termed feed-forward networks.

Without a hidden layer, the network in Figure 2 becomes equivalent to a type of two-layer pattern classifier neural network called a Perceptron,[8] which has been shown to be limited to problems that are linearly separable. This constraint would eliminate the solutions to many interesting problems in chemistry. Even a problem as simple as the exclusive OR (XOR)[8a] problem cannot be solved by a Perceptron. In the XOR problem, the input patterns 0 0 and 1 1 both give an output of 0 while input patterns 0 1 and 1 0 both give an output of 1. If the four input patterns were displayed as the points (0,0), (1,1), (0,1), and (1,0) in two-dimensional space, there is no line that can separate them according to their output values and thus they are not linearly separable.

A node, as shown in Figure 3a, is analogous to an electronic capacitor in that it accumulates or sums its inputs until a threshold is reached at which point it fires or outputs a signal. The net input to node i in layer λ, $\mathrm{net}_{\lambda,i}$, as given by eq 1 is

$$\mathrm{net}_{\lambda,i} = \sum_j W_{ij} O_{\lambda-1,j} \tag{1}$$

a linear-weighted sum of the products of its inputs $O_{\lambda-1,j}$, which are the outputs of the previous layer, times the weights W_{ij} on those connections, but it is not restricted to that form. The activation function, f, maps the net input to the unit to an output value $O_{\lambda,i}$. Figure 3b shows a plot of the most commonly used activation function, the logistic function, which is given in eq 2.

$$O_{\lambda,i} = f(\mathrm{net}_{\lambda,i}) = 1/(1 + e^{-\mathrm{net}_{\lambda,i}}) \tag{2}$$

The logistic function is a continuous function whose value is close to zero or close to one over most of its domain. Thus it is a continuous and differentiable approximation to a threshold function. The necessity for the activation function being differentiable will be seen in the following section. In the first layer, the inputs to node i correspond to the elements of the input vector, and for subsequent layers, the inputs correspond to the outputs $O_{\lambda-1,j}$ of the previous layer. In the neural network model used here, the result of the activation function f is passed directly as the output $O_{\lambda,i}$, but other mappings are possible. A node sends the same output, usually scaled between 0 and 1, to all of its output lines.

Training methods used for neural nets can be split into two types: supervised and unsupervised. In unsupervised methods, input patterns are presented once to the network and the network settles or converges to a final state. Supervised methods, on the other hand, use the desired or target output value(s) to determine the amount of error in the actual net output for each input pattern. This error signal is used as a basis for making corrections to the weights so as to minimize the errors. The processing of presenting each training example to the network is continued until the error over all of the training patterns is minimized. The set of connection weights thus obtained optimizes the mapping of the input patterns to their corresponding output patterns.

Back-Propagation Network. Back-propagation (BP) is a supervised learning method for multiple-layer nets that seems

Table I. EAS Literature Data Ranked by Percent of Meta Product[a]

substituent	% meta	substituent	% meta	substituent	% meta
$PhNH_2$	0	$C{\equiv}CCOOH$	8	$CH_2N^+H_2CH_3$[c]	60
$CH{=}CHCOOH$	0	CH_2NH_2[c]	10	CCl_3	64
F	0	CH_2OCH_3	12	$COOCH_2CH_3$	68
$CHMe_2$	0	$CH_2SO_2O^-$	14	$COOCH_3$[c]	68
O^-	0	CH_2CN[b]	14	$N^+H_2CH_3$	70
OH[b]	0	CH_2Cl	16	$CONH_2$[b]	70
CH_2CH_3[b]	0	CH_2F	18	$COCH_3$	72
Cl	1	CH_2COOH[c]	22	CHO	72
Br[b]	1	$CH_2SO_2NH_2$	31	N^+HMe_2	78
I	2	$CHCl_2$	34	COOH	80
$NHCOCH_3$	2	$SiMe_3$[b]	40	CN	82
OCH_3	2	NH_3^+[b]	42	NO_2	93
$CH{=}CHNO_2$	2	CH_2SO_2Cl	51	S^+Me_2[b]	95
$CH_2CH_2OCH_3$	3	CH_2NO_2[b]	56	CF_3	100
CH_3	4	SO_3^-	60	SO_2CH_3	100

[a] Data from refs 16a–f. [b] Test set. [c] Test set homologues.

to be particularly good at pattern classification and at discovering complex relationships between input variables.[8] It is the one used in the present study, with the network topology shown in Figure 2. Back-propagation networks have their units arranged in three or more layers with an input layer, an output layer, and one or more "hidden" layers which generate the solution to the problem. The hidden layers act as feature detectors by becoming activated in response to certain features or combinations of features in the input.

For a given input pattern p the network computes an output $O^{(p)}$ based on its current set of weights. The delta $\delta_{\lambda,j}{}^{(p)}$ or error for pattern p at unit j in layer λ is determined by subtracting the output pattern $O_{\lambda,j}{}^{(p)}$ from the target output $t_{\lambda,j}{}^{(p)}$ as shown in eq 3. The delta rule,[15] given in eq 4 changes the

$$\delta_{\lambda,j}{}^{(p)} = (t_{\lambda,j}{}^{(p)} - O_{\lambda,j}{}^{(p)}) \qquad (3)$$

$$\Delta^{(p)}W_{ji} = \eta\delta_{\lambda,j}{}^{(p)}O_{\lambda-1,i}{}^{(p)} \qquad (4)$$

weight between the input unit i and the output unit j by an amount proportional to the δ (eq 3) of the output unit and the inputs to unit j. The constant of proportionality η is called the learning rate. The delta rule assumes that each processing unit's contribution to the total error is proportional to that unit's activation, and thus the weights on the connections from the most active units are changed the most.

The generalized delta rule[8] is used to achieve learning in networks, such as BP nets, which have hidden layers. In the generalized delta rule the error $\delta_{\lambda,j}{}^{(p)}$ used in eq 4, is calculated differently for output units and hidden units. For units in the output layer ($\lambda = n$), $\delta_{n,i}{}^{(p)}$ is given by eq 5, where $f'(\mathrm{net}_{n,i}{}^{(p)})$

$$\delta_{n,i}{}^{(p)} = (t_{n,i}{}^{(p)} - O_{n,i}{}^{(p)})f'(\mathrm{net}_{n,i}{}^{(p)}) \qquad (5)$$

is the derivative of the activation function with respect to the net input.

For internal units ($\lambda = 2, 3, ..., n - 1$), a specific target value $t_{\lambda,i}{}^{(p)}$ is not known and the error $\delta_{\lambda,j}{}^{(p)}$ is computed in terms of the errors in the units in the next layer forward, as in eq 6.

$$\delta_{\lambda,i}{}^{(p)} = (\sum_j \delta_{\lambda+1,j}{}^{(p)}W_{ji})f'(\mathrm{net}_{\lambda,i}{}^{(p)}) \qquad (6)$$
$$\lambda = 2, 3, ..., n - 1$$

The index j is over all of the units to which unit i sends a signal. For eqs 5 and 6 to hold, the activation function must be differentiable. In order to calculate the δ values, the pattern information is fed forward through the net, obtaining an error with respect to the desired target. This error is then used to compute the weight corrections layer by layer, backwards through the net, hence the name back-propagation. The supervised training process is repeated until the error for all of the training data is minimized. The delta rule makes weight changes based on local information, namely the inputs and outputs of each pair of nodes, but has been shown to implement a gradient descent on the global system error surface[8c] analogous to energy minimization in molecular mechanics.

METHODS

Data. Product ratios for electrophilic substitution reactions of monosubstituted benzenes taken from the literature were used to train and test the neural network.[16a-f] The same electrophile, nitric acid, was used in all of the reactions except for phenol and phenoxide, where *tert*-butyl hypochlorite was used.[16c] The compounds were split into a training set of 32 compounds and a test set of 13 compounds, shown in Table I. The test set was chosen to be representative of both the types of substituents and the range of product ratios. Nine of the 13 test compounds were from the literature, the other 4, indicated in Table I, were constructed by making trivial structural changes in four of the training set compounds to produce homologues; for example, methyl benzoate was constructed from ethyl benzoate.

Data representation is an important aspect of any study involving neural nets. Two different approaches were used in the present work. The first, more successful approach employed information derived from a connection table representation of the reactants. The second one employed quantum mechanically calculated partial charges in an effort to model electronic effects on the aromatic ring due to the substituent.

Connection Table Representation. Chemists use their intuition, which is developed through training and experience, to perceive relevant features from structural diagrams of chemical reactions. The connectivity matrix or connection table provides a convenient method for representing information related to chemical structure for computer manipulation.[17] In the connectivity matrix, shown in Figure 4a for acetanilide (Figure 4d), the atoms are displayed horizontally and vertically along the edges. The entries correspond to the bonds in the molecule. The connection table, shown in Figure 4b, has two parts. The first has a row for each atom and its properties, while the second is a list of the pairs of bonded atoms and bond types.

A compact connection table, shown in Figure 4c for the acetamido substituent, was developed as an input representation for the neural net. Since the aromatic ring is the same for all of the reactants, only information on the substituents is included here. Atoms, excluding hydrogens, were numbered from the aromatic ring out in a breadth-first manner, with atom number 1 being the atom in the aromatic ring where the substituent was attached. The connection table was kept as concise as possible while preserving the connectivity information by citing only the bonds from higher numbered atoms

(a) Connectivity Matrix

	1	2	3	4	5	6	7	8	9	10
	C	N	C	O	C	C	C	C	C	C
1 C	0	1	0	0	0	2	0	0	0	1
2 N	1	0	1	0	0	0	0	0	0	0
3 C	0	1	0	2	1	0	0	0	0	0
4 O	0	0	2	0	0	0	0	0	0	0
5 C	0	0	1	0	0	0	0	0	0	0
6 C	2	0	0	0	0	0	1	0	0	0
7 C	0	0	0	0	0	1	0	2	0	0
8 C	0	0	0	0	0	0	2	0	1	0
9 C	0	0	0	0	0	0	0	1	0	2
10 C	1	0	0	0	0	0	0	0	2	0

(b) Connection Table

Atoms		Bonds		
1	C	1	2	1
2	N	2	3	1
3	C	3	4	2
4	O	3	5	1
5	C	1	6	2
6	C	6	7	1
7	C	7	8	2
8	C	8	9	1
9	C	9	10	2
10	C	1	10	1

(c) Connection Table for Neural Net

7	2	1	1	0
6	3	2	1	0
8	4	3	2	0
6	5	3	1	0
0	0	0	0	0

(d) Acetanilide

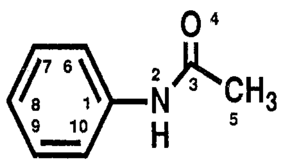

Figure 4. (a) Connectivity matrix for acetanilide (shown in part d). Entries *ij* correspond to bond order between atom *i* and atom *j*: 0 = not bonded, 1 = single, 2 = double, 3 = triple. (b) Connection table for acetanilide. Atom section has atom numbers and atom types. Bond section has atom numbers and bond type for each bonded pair of atoms. (c) Connection table used for neural net has one row for each non-hydrogen atom in the acetamido substituent on aromatic ring. Column 1 is the atomic number. Column 2 is the atom number of the atom in column 1. Column 3 is the atom to which the atom is bonded. Column 4 is the bond type. Column 5 is the charge.

to lower numbered atoms. One row is used for each non-hydrogen atom in the substituent. The first column contains the atomic number of the higher numbered atom in a bonded pair. The atomic number provides a chemically relevant means of differentiating atom types with a single number since, as noted under Introduction, a neural network requires numeric values as input data. Bonds are specified with the higher numbered atom in the second column and the lower numbered atom in the third column. The fourth column indicates the bond order: 1 for single bonds, 2 for double bonds, and 3 for triple bonds. The charge on the atom, either −1, 0, or +1, is placed in the fifth column. Specifying the charge on the atom allows hydrogen atoms to be ignored by using charge to indicate differences from normal valence. This results in a very compact 5 × 5 connection table. For substituents with less than five atoms, the remaining rows are filled out with zeros to ensure that all the input patterns have the same number of elements. Other variants of this connection table format were found to require longer learning times and gave poorer results.

Charge Vector Representation. Instead of representing the substituent directly, the effect of the substituent on the aromatic ring was represented by the charge at each carbon atom on the ring. The carbons were numbered starting at the atom to which the substituent was attached and proceeding clockwise around the ring. MOPAC,[18] a semiempirical quantum mechanics program, was used to calculate the Mulliken partial charges on each carbon atom.

Network Configuration. The best network paradigm and configuration for a particular problem must be found empirically. However, some decisions can be made on the basis of the characteristics of the various network types. A three-layer back-propagation (BP) network was chosen here because of its pattern mapping properties.[7] The BP net constructs an internal representation of the task, in this case predicting the products of the EAS reaction, which can be applied to new reactants not in the training set. The network which used the connection table as input (Figure 5) had 25 units in the input layer, a hidden layer of 5 units, and an output layer of 2 units. The 25 input units correspond to a 5 × 5 connection table representation. Five hidden units gave the best network performance. A larger number of hidden units caused the network to fit the input examples very well but gave poor generalization for the test set of reactants. Fewer hidden units gave poor predictions for both the training set and the test set. The two output units represent the fraction of ortho–para product and the fraction of meta product produced in the reaction. Only six input units were required when the partial charges on the aromatic ring were used as input. Ten hidden units were found to give the best predictions in this case. Two outputs units were also used for the charge vector network.

Network Training and Testing.[19] Training a network is an iterative process that involves repeatedly presenting the training examples, adjusting the weights according to the learning rule, and modifying the learning parameters. Thirty-two of the reactants were used to train the network and the remaining 13 compounds were kept for testing. Starting with a set of randomly chosen connection strengths, the weights were changed by the supervised BP algorithm to minimize the root mean squared (RMS) error between the net's predicted output values and the experimental ones. Typically it required presenting the entire set of training examples for more than 100 000 repetitions to obtain a set of weights that reduced the RMS error to less than 0.05. The learning rate η (eq 3), which

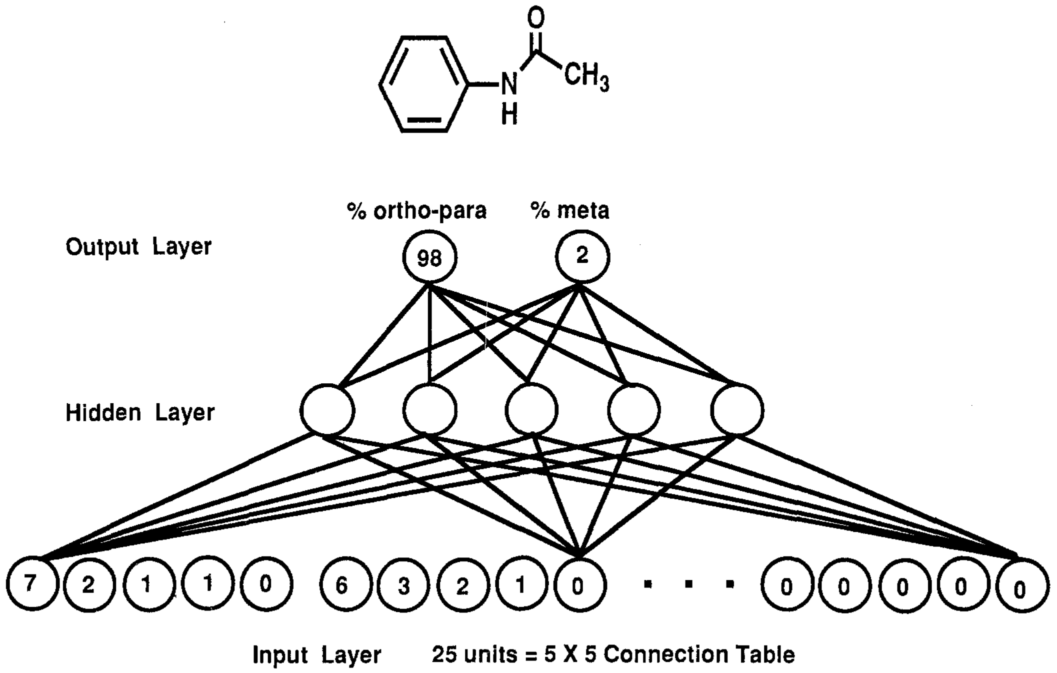

Figure 5. Three-layer back-propagation neural net with 25-unit connection table input, 5 hidden units, and 2 output units corresponding to percent of ortho–para and meta products for acetanilide.

is the proportion of the error that is used to adjust the weights, was reduced from 0.1 to 0.005 over the course of training.

It is not possible to determine whether the set of weights obtained by the BP procedure represents the global minimum on the error surface for the system. Two approaches were combined in the training regimen in an attempt to avoid being trapped in local minima. In the first approach, random values are added to the weights after the total RMS error has stabilized. This allows the network to explore other parts of the error surface. Training was continued until the RMS error reached another plateau. This process of "damaging" the weights by adding a random component and retraining the net was repeated several times until no further improvement in the RMS error was obtained. The second approach to improving network performance adds random noise, which gradually decays to zero, to the input patterns. This process can make the network more robust to the effects of noisy data and may also serve to help avoid some local minima. Both of these approaches were repeated several times during the training process in order to obtain the best network predictions.

Unlike the training process, which can take a very long time, testing the network can be done rapidly in a single pass through the data. In the testing phase a set of input data is presented to the network without providing the target output values and without changing the weights. The output values computed by the net are determined by the set of weights that were obtained in the training process. Network predictions for both the training set and the test set were obtained separately so that the ability of the net to learn how the patterns of input features correlate with the product ratios in the training data could be measured as well as the ability of the net to make generalizations about new cases on which it had not been trained.

Comparison with Other Prediction Methods. Both networks were compared to predictions made by the widely used chemical expert system CAMEO[20] and by three experienced synthetic organic chemists. CAMEO applies mechanistic rules to make predictions about a broad range of organic reactions, of which EAS is only one. Moreover, CAMEO, like the organic chemists it emulates, predicts either ortho–para products or meta products but does not quantitate those predictions. Thus, CAMEO predicts either 0% meta or 100% meta products. Reactions which yield less than 20% or more than 80% of meta product will have small errors, while reactions which form between 20% and 80% meta product will have correspondingly larger errors.

Three organic chemists were given the percentages of ortho, para, and meta products for the 32 reactants in the training set and were asked to predict the percent of the three types of products for the 13 unknowns in the test set. Based on the range of predictions by the chemists and also on the variation in product ratios due to experimental conditions, a prediction was deemed correct if it was within 20% of the average of the experimental values. The accuracy of the prediction for the fraction of meta product was used as a basis for determining the overall performance of a network.

RESULTS AND DISCUSSION

Figure 6 shows the magnitude of the errors in the network predictions of the amount of meta product for each example in the training set. The vertical axis displays the 32 training compounds, and the horizontal bars correspond to the size of the errors between the predicted percent of meta product and the experimental value. The best connection table network, with a 25-unit input layer, a 5-unit hidden layer, and 2-unit output layer was trained to a total RMS error of 0.022. All 32 of the training compounds were correctly predicted with an average error per compound of 0.3%. The best charge vector network, with 6 input units, 10 hidden units, and 2 output units (final RMS error 0.078) also correctly predicted 32 of 32 training compounds but with an average error of 5.2%. CAMEO predicted only 22 of 32 of the training set correctly, with an average error of 18%. CAMEO is at a disadvantage in this type of comparison because it predicted all meta or no meta product, while the networks gave quantitative predictions.

A more difficult problem is making predictions on the test set, as shown in Figure 7. The connection table network correctly predicted 10 of 13 reactions of the test compounds with an average error of 12%. The charge vector network predicted only 8 of 13 correctly with an average error of 20%. CAMEO, partly due to the all or nothing nature of its predictions, was correct in 7 of 13 cases. The same 13 reactants were given to three synthetic organic chemists who made correct

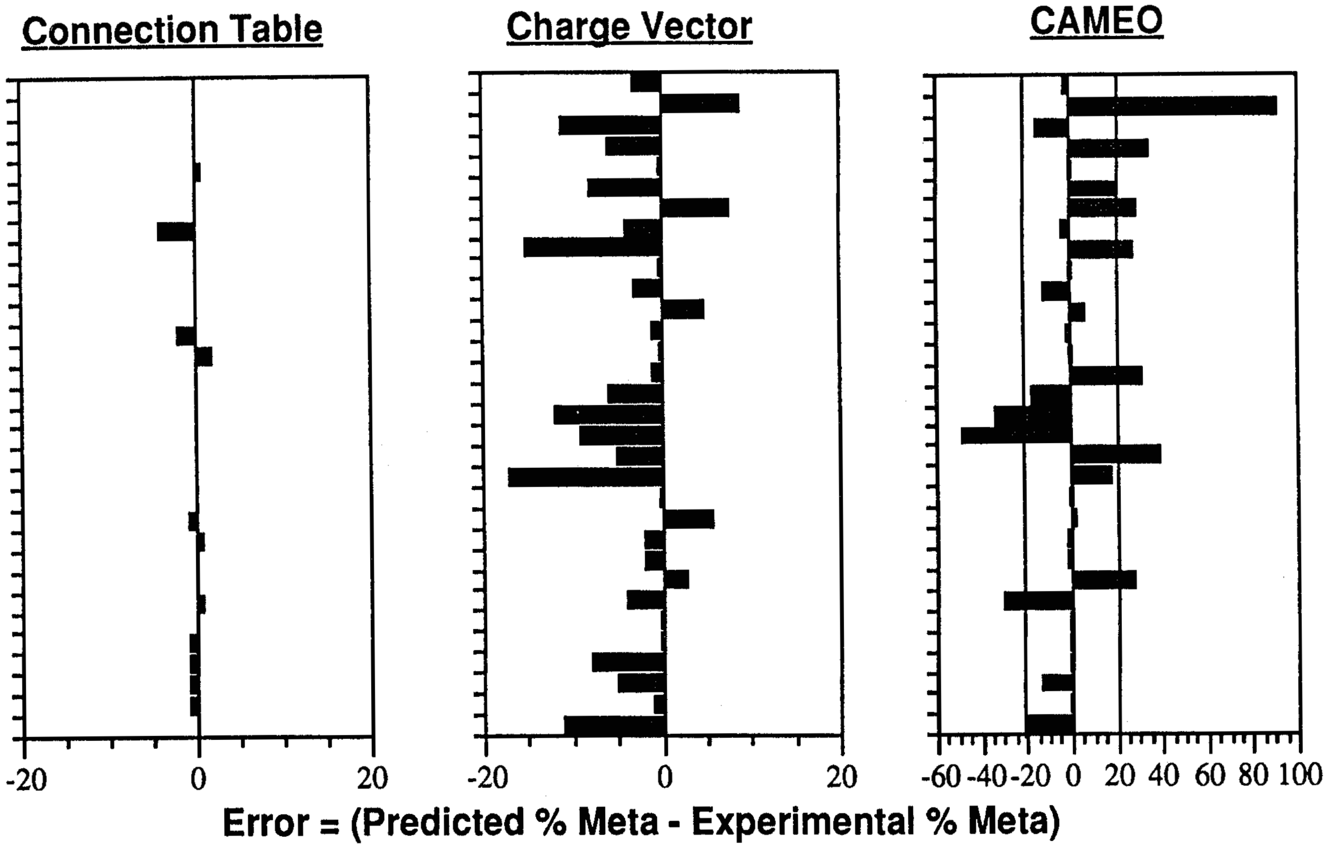

Figure 6. Training set predictions for EAS reaction. Vertical axis represents 32 training set reactants; horizontal axis is the magnitude of errors in prediction of percent of meta product. The connection table network correctly predicted 32 of 32 with an average error of 0.3%, the charge vector net predicted 32 of 32 with an average error of 5.2%, and CAMEO predicted 22 of 32 with an average error of 18%.

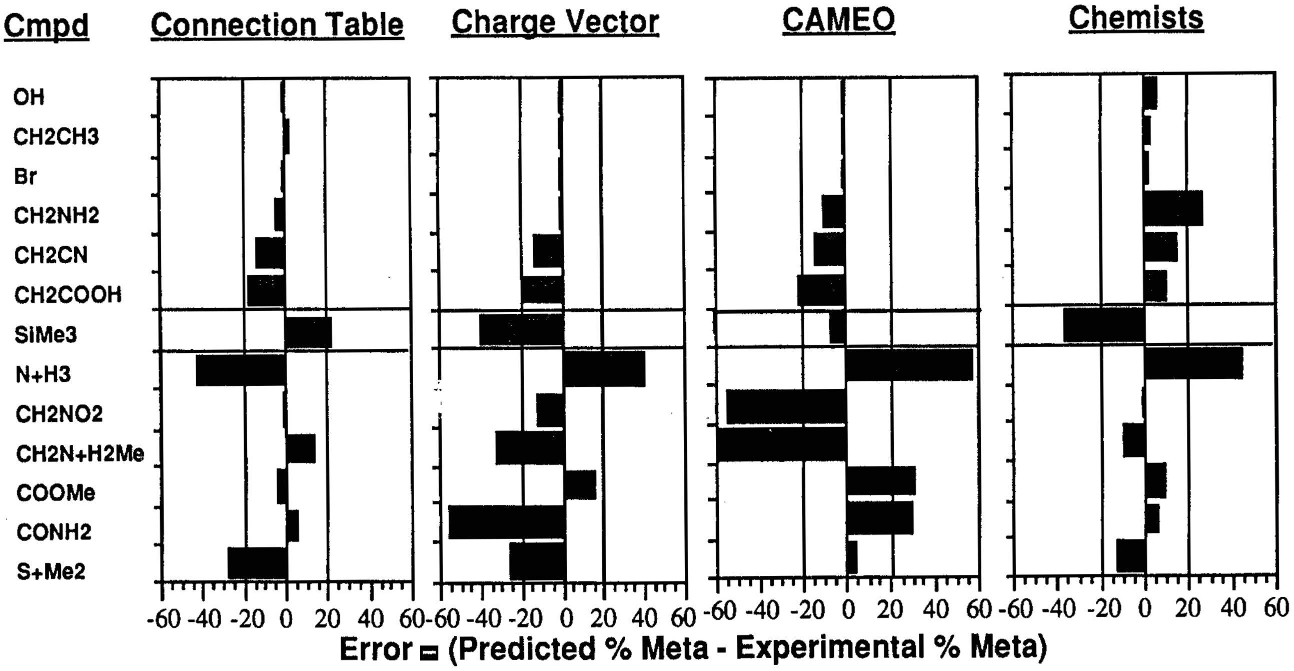

Figure 7. Test set predictions for EAS reaction. Vertical axis is 13 test reactants; horizontal axis is the magnitude of errors in prediction of percent of meta product. The connection table network correctly predicted 10 of 13 with an average error of 12.1%, the charge vector net was correct for 8 of 13 with an average error of 19.8%, CAMEO was correct on 7 of 13 with an average error of 22.6%, and the synthetic chemists were correct for 10 of 13 with an average error of 14.7%.

predictions for 10 of 13 reactants with an average error of 15%.

Of particular interest is the case in which the substituent is a trimethylsilyl (TMS) group, which was reported to yield 40% of meta product.[16b] None of the training examples contained any silicon atoms, so the network was forced to make a generalization for this substituent. The connection table network gave a reasonable prediction of 63% meta even though it had not been trained on any examples of silyl substituents. The charge vector network gave a very poor prediction of 0% meta while CAMEO predicted 33% meta. The compound with a TMS substituent was the only one of the 45 in the data set for which CAMEO predicted all 3 products: ortho, para, and meta. It appears that CAMEO has a rule that handles a TMS group as an exception. The three chemists did not do as well as the network on this reactant. They predicted that the TMS substituent would yield only 2% of meta substitution. When asked why they made this prediction they said that a silicon substituent was not part of their "training set" and so they used methyl or *tert*-butyl as a model. These both produce less than 10% of meta compound so neither is a good model for the reaction with a TMS substituent.

The two networks compare well with the CAMEO expert system program and with the chemists in the number of correct predictions. To within 20% error, CAMEO predicted 69% of the training set correctly and 54% of the test cases. The connection table network was somewhat better than either the charge vector network or the CAMEO program and performed about as well as the chemists. The connection table network predicted 100% of the training set correctly and 77% of the unknowns while the charge vector was correct for 100% of the training set and 62% of the test set. The chemists correctly predicted 77% of the test set.

Results comparing the connection table and the charge vector representations used in this study indicate that the connection table representation provides better predictions. The charge vector representation focuses on a specific aspect of the reactants, the distribution of charge in the aromatic ring, which is only one of the factors which determine the relative reactivity of the ortho, para, and meta positions. The more general connection table representation is a richer source of information for the network and provides a means of comparing atoms on the basis of atomic number, the number and types of bonds, and other properties such as charge or number of hydrogens.

CONCLUSIONS

The ability of neural networks to predict a chemical reaction as well as chemists or an expert system has been demonstrated for the EAS reaction: the neural net was able to learn by example to make predictions for cases on which it had not been trained. Quantitative predictions were made by the network in contrast to the expert system which made qualitative predictions. Also, the neural network did not require formulating rules about reactivity in order to make useful predictions, but was able to form an internal model of the reaction by extracting information directly from examples of the reaction. The method used for representing the chemical information for the neural network is a major factor in determining the predictive ability of the network. A more descriptive, less specific representation derived from connectivity information gave better results than the representation that employed specific charge information.

Neural networks may be useful as an adjunct to expert systems in cases where concrete rules describing reactivity can not be easily formulated. Databases of chemical reactions could provide sufficient examples to train a set of neural nets to predict a number of reactions. However, one of the key questions that needs further exploration is how best to represent chemical information so that a neural network can make useful predictions. The connection table used in this study is not sufficiently general for use with other reactions. Further work is underway to find a more general structure-based representation that would allow extension of the current neural network approach to a wider range of reaction types.

REFERENCES AND NOTES

(1) Barone, R.; Chanon, M. In *Computer Aids to Chemistry*; Vernin, G., Chanon, M., Ed.; Ellis Horwood Limited: Chichester, U.K. 1986; Chapter 1, pp 19–102.

(2) Corey, E. J.; Long, A. K.; Rubenstein, S. D. *Science* **1985**, *228*, 408–418.

(3) Bauer, J.; Herges, R.; Fontain, E.; Ugi, I. *Chimia* **1985**, *39*, 43–53.

(4) Hendrickson, J. B.; Grier, D. L.; Toczko, A. G. *J. Chem. Inf. Comput. Sci.* **1984**, *23*, 171–177.

(5) Gushurst, A. J.; Jorgensen, W. L. *J. Org. Chem.* **1988**, *53*, 3397–3408.

(6) Klimasauskas, C. C. *PC AI*, 1988, *4*, 26–30.

(7) Lippmann, R. P. *IEEE ASSP Mag.* **1987**, *4*, 4–22.

(8) (a) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. In *Parallel Distributed Processing*; Rumelhart, D. E., McClelland, J. L., Eds.; MIT Press: Cambridge, MA, 1987; Vol. 1, pp 319–362. (b) McClelland, J. L.; Rumelhart, D. E. *Explorations in Parallel Distributed Processing*; MIT Press: Cambridge, MA, 1988; pp 121–159. (c) Hopfield, J. J. *Proc. Natl. Acad. Sci. U.S.A.* **1984**, *81*, 3088–3092. (d) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. *Nature* **1986**, *323*, 533–536. (e) Domany, E. *J. Stat. Phys.* **1988**, *51*, 743–775.

(9) (a) Qian, N.; Sejnowski, T. J. *J. Mol. Biol.* **1988**, *202*, 865–884. (b) Bohr, H.; Bohr, J.; Brunak, S.; Cotterill, R. M. J.; Lautrup, B.; Norskov, L.; Olsen, O. H.; Petersen, S. B. *FEBS Lett.* **1988**, *241*, 223–228. (c) Holley, L. H.; Karplus, M. *Proc. Natl. Acad. Sci. U.S.A.* **1989**, *86*, 152–156. (d) McGregor, M. J.; Flores, T. P.; Sternberg, M. J. E. *Protein Eng.* **1989**, *2*, 521–526. (e) Liebman, M. 197th National Meeting of the American Chemical Society, Dallas, TX, April 12, 1989; Abstract COMP 29.

(10) Thomsen, J. U.; Meyer, B. J. *Magn. Reson.* **1989**, *84*, 212–217.

(11) Aoyama, T.; Suzuki, Y.; Ichikawa, H. *Chem. Pharm. Bull.* **1989**, *37*, 2558–2560.

(12) Kowalski, B. R.; Jurs, P. C.; Isenhour, T. L.; Reilly, C. N. *Anal. Chem.* **1969**, *41*, 695–700.

(13) Stubbs, D. 197th National Meeting of the American Chemical Society, Dallas, TX, April 12, 1989; Abstract COMP 30.

(14) Recce, M.; Treleavan, P. *New Sci.* **1988**, 26 May, 61–64.

(15) Jones, W. P.; Hoskins, J. *Byte* **1987**, October, 155–162.

(16) (a) Carey, F. A.; Sundberg, R. J. *Advanced Organic Chemistry*, 2nd ed.; Plenum Press: New York, 1984, Part A, pp 481–503. (b) De La Mare, P. D. B.; Ridd, J. H. *Aromatic Substitution: Nitration and Halogenation*; Academic Press: New York, 1959; pp 80–93, 236–237. (c) Harvey, D. R.; Norman, R. O. C. *J. Chem. Soc.* **1961**, 3606–3611. (d) Hoggett, J. G.; Moodie, R. B.; Penton, J. R.; Schofield, K. *Nitration and Aromatic Reactivity*; Cambridge University Press: London, 1971; pp 166–183. (e) Norman, R. O. C.; Taylor, R. *Electrophilic Substitution in Benzenoid Compounds*; Elsevier Publishing Company: Amsterdam and New York, 1965; pp 72–85. (f) Patai, S. *The Chemistry of the Amino Group*; Wiley-Interscience: London, 1968; pp 250–257.

(17) (a) Lynch, M. F.; Harrison, J. M.; Town, W. G.; Ash, J. E. *Computer Handling of Chemical Structure Information*; American Elsevier Inc.: New York, 1971; pp 12–35. (b) Ash, J. E.; Chubb, P. A.; Ward, S. E.; Welford, S. M.; Willett, P. *Communication, Storage and Retrieval of Chemical Information*; Ellis Horwood Limited: Chichester, U.K. 1985; pp 128–156.

(18) MOPAC, A General Molecular Orbital Program Package V4.0, QCPE, Department of Chemistry, Indiana University, Bloomington, IN, 47405.

(19) The BP networks were run using the ANSIM (SAIC Inc., San Diego, CA) neural network simulator on a 20-MHz 80386/80387 PC. Each component of the input and output patterns was normalized over the entire data set and scaled to the range −0.5 to +0.5.

(20) The 1988 version of the CAMEO program was used. See ref 5.