Predicting Phosphorus Nuclear Magnetic Resonance (NMR) Shifts Using Neural Networks. 3. Element-Value Optimizing (EVO) Network Architectures

Geoffrey M. J. West

School of Computing and School of Sciences, Staffordshire University, Staffordshire ST5 2JE, England

Received November 7, 1994[⊗]

This work describes further refinements to a method which predicts NMR shifts using neural networks and graph templates. A network is taught to predict NMR shifts when it is given a numeric code representing the focal environment. This code is obtained by first translating environments onto a graph template to produce the symbolic form of the code. The element symbols are then replaced by a numeric associated "property". This work introduces specialized architecture(s) where the network derives its own internal set of element associated values. These element value optimizing or EVO networks were trained on separate subsets of phosphorus compounds. The results indicate further similarities between template based prediction and additivity rules methods and nominate it as a graph based version of the additivity rules.

INTRODUCTION AND BACKGROUND

Predicting nuclear magnetic resonance (NMR) shifts from radial environments now plays a crucial part in most procedures for solving compound structure. The relationship is so complex, however, that at present only empirical methods have any widespread use. While more accurate techniques exist,[1,2] most empirical predictions are made using either additivity rules[3-5] or substructure codes,[6-8] as these methods can be applied across the entire domain of topologies. Additivity rules approximate the structure-shift relationship by piece-wise linear modeling. Here the topological domain is factored into realms, and a set of additive parameters is derived for each. These allow the structure-shift relationship to be linearly approximated within the realm. This approach works well but requires much initial effort to derive the parameter sets. Maintenance is also necessary as parameters can become obsolete when new compounds are added to a realm. Predictions based on substructure codes depend upon the implicit correlations contained in large quantities of {environment, shift} pairs. If there is sufficient data, this approach also works well and, being data-driven, is readily automated. Predicted shifts are obtained by fetching lexicographically matching or most similar environments from the databank. This approach can suffer from relatively slow prediction times and is sensitive to data errors. There is also a fundamental problem concerning how environmental similarity is measured, as any interpolation between the symbolic representations has no physiochemical relevance. If a hybrid prediction method could be devised retaining the favorable aspects of both the existing methods then this would offer considerable benefits. The minimum features of a hybrid would be (a) data driven allowing easy automation, (b) quick production of accurate shifts, (c) a degree of chemical relevance in any similarity measurements, and (d) a degree of fault tolerance. In the search for the components with which to build a hybrid system, artificial neural networks seem a good choice as (a) they can take into account any implicit correlations within input,output pairs, (b) on modest workstations medium sized networks can make thousands of predictions every second, (c) interpolation is an inherent characteristic, and (d) network performance declines gracefully with respect to the number of data errors. A further advantage is that the learning and prediction infrastructures are identical, and thus no separate prediction subsystem needs constructing.

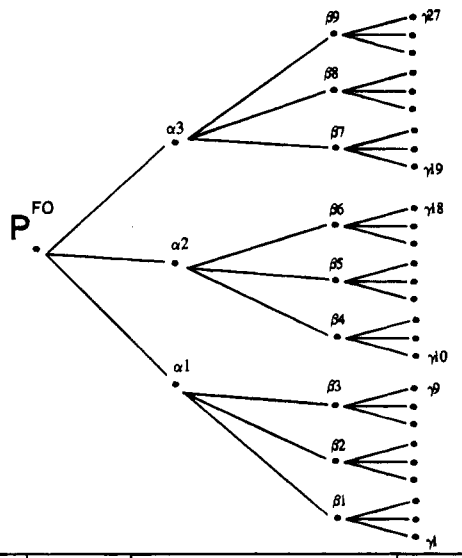
During the last decade, the popularity of neural networks has increased immensely due to the (re) invention of the back-propagation training algorithm.[9-12] Unlike previous algorithms, back-propagation[9] and its many variants[13-16] allow training of networks with multiple layers. These hidden layers bestow the capability of learning representations of nonlinear relationships, which has expanded the potential of neural networks enormously. Several schemes demonstrating the ability of multilayer networks to predict ^{13}C NMR shifts have appeared.[17-21] These either retain a categorical description of the environment by representing it with simple binary vectors[17-19,21] or categorize the networks output.[20] However, the NMR shift and the physiochemical properties that influence it are usually measured on metric scales. This implies that more accuracy would be obtained if metric data were used, with the network acting as a function approximator rather than as a classifier. The theoretical abilities of neural networks to learn to approximate continuous functions has been known for some time.[22-25]

A more detailed treatment of ***template based*** prediction has been given in a previous paper.[26] While the method was originally developed for 31-phosphorus (^{31}P) NMR, from the outset the intent was to develop as generic a technique as possible. Using ^{31}P does make quantitative assessment of the method difficult as no other (nonheuristic) ^{31}P shift prediction methods exist. Thus, while work on ^{31}P NMR continues, the method has also been applied to ^{13}C NMR,[27] using the Lindeman and Adams data set.[28] The potential for applying the method to ^{13}C NMR was suggested by (a) the similarities to other neural network based methods[17-21], (b) a reliance on α-δ substituent effects,[3,26] and (c) the similarities of template based prediction and ^{13}C additivity rules.[29] An outline of template based prediction, its terminology, and the results of previous work follow.

Template Based Prediction. The method is named after the graph templates used to standardize focal environment

[⊗] Abstract published in *Advance ACS Abstracts,* July 15, 1995.

0095-2338/95/1635-0806$09.00/0



linear	α1, α2, α3	β1, β2, β3, β4, β5, β6, β7, β8, β9	γ1 −γ27
branch	α1, β1 −β3, γ1 −γ9	α2, β4 −β6, γ10 −γ16	α3, β7 −β9, γ19 −γ27
random	..... no relationship between template nodes and input vector positions		

Figure 1. Template $33\alpha\beta\gamma$.

sizes. This is necessary because the kind of neural network used has a static architecture and hence requires a fixed size of input vector. The environmental radius, focal coordination, and maximum allowed coordination of any substituents determine which model of template is used. This work uses hydrogen explicit environments which have three coordinate three valent (3c3v) ^{31}P foci and contain α, β, and γ substituents. This, together with a maximum coordination of four for any α and β substituents, determines that the template$_{33\alpha\beta\gamma}$ model which is shown in Figure 1 is used to ***translate*** environments. Starting from, but not including, the focus, any directly attached substituents of the focus or "current atom" are ordered and translated onto this template. The initial concerns with the Connectionist aspects were made at the expense of those of graph theory, and Morgan algorithm extended connectivity (EC) values[30] are used to order the substituents. As network learning is fault tolerant, and any misorderings introduced by the Morgan algorithm constant factors, the use of Morgan EC values has been retained to allow comparisons to previous work. After the focus, the ordered α substituents each have their turn as "current atom", and translation continues, processing breadth first. Translation terminates for template$_{33\alpha\beta\gamma}$ when the directly attached substituents of (focal) substituent β_9, i.e., focal substituents γ_{25-27} , are ordered and translated. Template nodes which have no atom translated onto them are filled with the unoccupied symbol "−". Once complete a template representation is converted into a ***molecular abstract graph space*** (MAGS) code. There are three ***formats*** of this substructure code: linear, branch, and random, each defined by the relationship between the template nodes and substructure code positions. The relationships for the three code formats are also shown in Figure 1.

The actual constructs used to train a network are obtained by applying three procedures to a set of MAGS codes. Here it is assumed that every code in such a set has an identical format and was derived by using the same template. The three procedures are as follows:

1. Eliminating any one to many mappings from the original set, creating a ***unique set*** where every instance of MAGS code is unique. The shift values associated with these unique codes are obtained by averaging the shifts of any removed duplicates.

2. Replacing the element symbols in the code with a numeric "property" such as electronegativity values. The replacement values are contained in ***symbol replacement files*** (SRF), with the files used in this work given in Table 1.

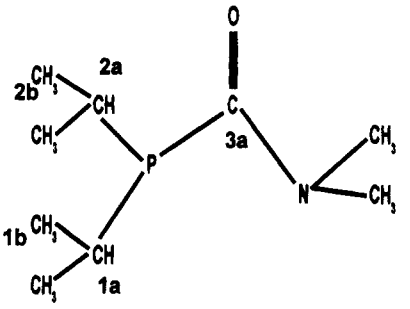
3. Scaling the shift range spanned by the unique set to the output range of the transfer function used by the neuron in the output layer. This work uses only the $(1/[1 + e^{-NET}])$ transfer function which has a [0 to 1] output range. To avoid the extremes of this range, shift values are transformed to a [0.1 to 0.9] range from [−455.5 to 293.7], the range spanned by the 3c3v class. Here a class is defined by the coordination and valency of the ^{31}P focus, and a subclass is defined as a group with one or more topological features in common, derived from a class.

Applying these procedures produces a unique set which contains one particular ***type*** of ***compound vector***. Each compound vector has two parts, an ***output vector*** containing the transformed shift and an ***input vector*** containing the environment data. A compound vector's type is identified from (a) a label identifying the format of the MAGS code within, (b) preceding this label, a superscript identifying the template, and (c) following the label, a subscript identifying the SRF file used. Thus a $^{33\alpha\beta\gamma}$linear$_{map-eneg}$ type has a linear format of MAGS code, was obtained by template$_{33\alpha\beta\gamma}$ translation, and has had its code symbols replaced by their [0 to 0.9] scaled electronegativity values. Figure 2 shows a compound, the linear format of MAGS code obtained by template$_{33\alpha\beta}$ translation, and the $^{33\alpha\beta}$linear$_{map-eneg}$, $^{33\alpha\beta}$linear$_{row,col}$ (Periodic Table row and column valves), and $^{33\alpha\beta}$linear$_{19E-LRFU}$ types that result when the code symbols are replaced using the SRF$_{map-eneg}$, SRF$_{row,col}$, and SRF$_{19E-LRFU}$ files, respectively. The SRF$_{19E-LRFU}$ replacements are discussed later. For clarity full electronegativity values are given in Figure 2.

The Results of Previous Work. Networks learn when given ***structured*** representations of the environment. Here structured refers to types with linear and branch formats, where the substituent positions are ordered. Each unique set of structured types has the ***positional equivalence property*** (PEP), which refers to the equivalence of the code positions across the set. This equivalence is with respect to the compound topologies. The importance of PEP is shown by training networks with random format types, which have the same data content as their structured counterparts but have had their positions randomized. Training with random types results in performances equivalent to training with types

Table 1. Description and Subscript Identifiers of the Different Symbol Replacement Files (SRF)

the "property" contained within the SRF file	SRF file subscript identifier	description of the result of using the SRF to replace the MAGS substructure code symbols
mapped electronegativity	map_eneg	element symbol replaced by its electronegativity value scaled to a [0 to 0.9] range
mapped atomic number	map_atno	element symbol replaced by its Atomic number scaled to a [0 to 0.9] range
global 1.0 replacement	all_1.0	all element symbols replaced by a value of 1.0
none	19E_LRFU	element symbol replaced by the appropriate 19 component binary vector shown in Table 2



MAGS code

C C C C C H C C H N O -
1a 2a 3a 1b 2b

	19C_LRFU
C	00100000000000000000
C	00100000000000000000
C	00100000000000000000
C	00100000000000000000
C	00100000000000000000
H	10000000000000000000
C	00100000000000000000
C	00100000000000000000
H	10000000000000000000
N	00010000000000000000
O	00001000000000000000
-	00000000000000000000

	map_eneg	(Periodic Table) row,col	
C	2.5	2	4
C	2.5	2	4
C	2.5	2	4
C	2.5	2	4
C	2.5	2	4
H	2.3	1	1
C	2.5	2	4
C	2.5	2	4
H	2.3	1	1
N	3.07	2	5
O	3.5	2	6
-	0.0	0	0

Figure 2.

which contain no environmental data. Such zero format types have zeros in every code position, i.e., all symbols have been treated as if they were the unoccupied symbol "−". The importance of PEP is further demonstrated by cross-evaluating between the different structured formats. Here networks trained with linear format types are monitored using the comparable branched format type and vice versa. A "comparable type" is one which is equivalent in all respects apart from the format. Such cross-evaluations show that a network's predictive capability is restricted to types with the same structured format as those used to train it.

Research into the factors governing the predictive ability used different environment sizes and ***variant*** monitoring sets. A variant set is created by modifying the original monitoring set of a ***partition***. In this work a single partition is defined as the random division of one unique set into one training and one monitoring set. The modifications involve removing data from substituent ***regions*** within the monitoring set types, achieved by setting all the positions within a region to zero. In this work, template$_{33\alpha\beta}$ and template$_{33\alpha\beta\gamma\delta}$ derived types have two and four regions, respectively: one for α and one for β substituents and one each for α, β, γ, and δ substituents. The variant sets differ *only* in the altered regions and, with the original set, are assessed using the same network configurations. The configurations are obtained by training a network with the unmodified training set of a partition. These variant experiments demonstrated that, in terms of their importance to the network's predictive method, the regions were ordered $\alpha > \beta > \gamma$. Experiments with different environment sizes and different network architectures demonstrated that the greater the environment size, the better the network's predictions, and that a large component of this predictive ability was linearly dependent.

Other earlier work investigated the effects of using SRF files with different "properties", to determine if performance was affected by the values in an SRF file. Three different "properties" were used: (a) scaled atomic numbers, each value uniquely identifying an element, (b) a single global value, making elements indistinguishable, and (c) scaled electronegativity values, using a property known to be partially correlated to the shift. The results showed types derived from SRF files containing some scaling of electronegativity values consistently out-performed any other "property". This was interpreted as evidence that one factor governing performance was the replacement values. If this postulate is accepted, then it infers that at least one set of optimum replacement values must exist.

The overall conclusions were that template based prediction is governed by factors with direct parallels in the ^{13}C additivity rules, where there is also (a) a correlation between predictive accuracy and environment size (b) a correlation between the proximity of a substituent to the focus and the magnitude of its contribution, (c) a high degree of linear dependence, and (d) set(s) of optimum replacement values/parameters for environmental features. Such a clear correspondence between a network-evolved and an existing solution is both unusual and fortuitous. It is unusual in that typically it is extremely difficult to relate network solutions to actual problem parameters.[31] It is fortuitous in that consideration of the existing solution(s) can suggest immediate refinements for the network method. This work focuses on some of these refinements.

Intermediary Unpublished Work. In previous works input vectors contained a five component ring vector which preceded the MAGS code. The purpose of a ring vector was to encode the size of the smallest ring system containing the focus, if any. They were necessary as analysis indicated that focal ring membership had an independent effect on the shift value. The effect is greatest for smaller rings, with five- and six-membered ring systems altering the shift by approximately ±9 ppm, respectively (compared to an environment(s) with identical substituents with no focal ring membership). Some 33% of 3c3v foci are ring members with the majority found in five- or six-membered rings. The effectiveness of the ring vectors was measured using variants where the ring vector data had been removed. As previously, networks were trained and monitored on unmodified sets and also monitored using the variant. Performance was assessed at epoch intervals, by measuring the percentage of monitoring or variant set vectors whose predicted shifts were within or without ±**X** ppm of the actual shift. For these ring variant experiments, within ±4 (<4) and ±8 (<8) and without ±16 (>16) and ±20 (>20), **tolerance bands** were used. The results showed virtually no difference between the performance of the variant and unmodified monitoring sets. It was concluded that this means of giving the network information on focal ring membership was ineffective. Hence for this work any environments where the foci are ring members have been excluded, and the input vectors have no ring vector.

The Purpose of the Experiments. As in previous work, the term "experiment" refers to *computer based* experiments, which, like their laboratory equivalents, are distinguished by experimental parameters. Here these are the following: (1) the 3c3v subclass used, (2) the different neural network

Table 2. The 19 Component Binary Vectors in the SRF_{19E_LRFU} File

element symbol	19E_LRFU file replacement
H	1000000000000000000
D	1000000000000000000
T	1000000000000000000
B	0100000000000000000
C	0010000000000000000
N	0001000000000000000
O	0000100000000000000
F	0000010000000000000
Si	0000001000000000000
P	0000000100000000000
S	0000000010000000000
Cl	0000000001000000000
X[a]	0000000001000000000
Ga	0000000000100000000
Ge	0000000000010000000
As	0000000000001000000
Se	0000000000000100000
Br	0000000000000010000
Sn	0000000000000001000
Sb	0000000000000000100
Te	0000000000000000010
I	0000000000000000001
–	0000000000000000000

[a] In some environments the symbol X is used to represent any Halogen. X is treated as Cl.

architectures used, (3) the different types of compound vector used, and (4) where networks are cross-evaluated, the subclass of the monitoring set used. To counteract the effects of data errors, experiments are repeated using different partitions of the unique set, which are then bundled together into experiment **categories** and average performance values calculated. In this work, categories belong to one of two groups. Group A repeated the work where symbols were replaced with different "properties", but this time on two separate 3c3v subclasses. One purpose here was to establish if the order of the "property" performances was (a) consistent with the order found previously and/or (b) consistent between the subclasses. A further purpose was to allow comparisons to the performances of ***element-value optimizing*** (EVO) architectures, which were designed to derive an internal set of element associated values (The scaling between, rather than the values of, the replacements is the important factor.[29]). These architectures should therefore theoretically give the best performances, assuming that the values of the other network-given "properties" are suboptimal. In group B, EVO networks were trained using vectors from one subclass and cross-evaluated with vectors from the other. The purpose here was to establish if the scope of the network derived values was universal or subclass specific.

Element Value Optimizing Network Architectures. Until now, input vectors have contained real numbers, and network architectures have been fully connected. To allow a network to derive an internal value for each element requires altering both these factors by (a) replacing each symbol with a binary vector and (b) using networks where the input layer is only partially connected. Table 2 shows the contents of the $SRF_{19E-LRFU}$ file which contains the 19 most common elements and their binary vector replacements. For this work environments were further restricted to those containing only these 19 elements. Each element is assigned to a unique position in the binary vector, with the associated binary vector having a value of 1 at, and only at, this position. Thus each vector has 19 components, 18 zeros and a one,

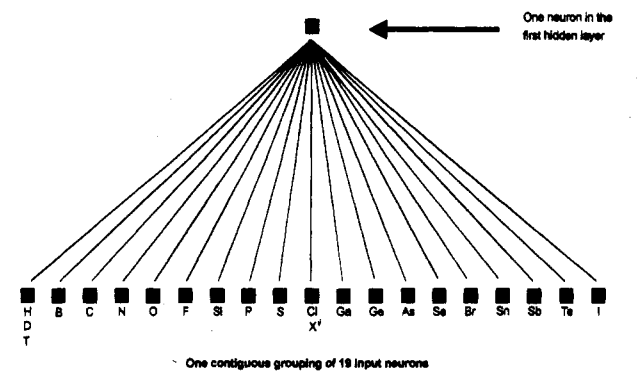


Figure 3. A 19 component limited receptive field unit [19C–LRFU].

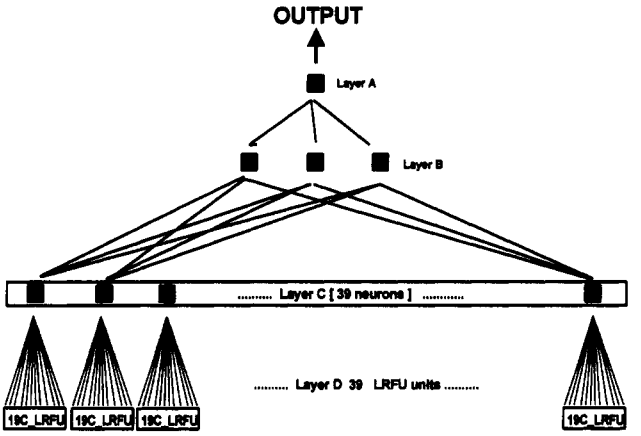


Figure 4. Network Architectures.

and $^{33}\alpha\beta\gamma linear_{19E-LRFU}$ types have an input vector with 741 (39 × 19) components.

The partial connectivity required in an EVO network is achieved by tiling the input layer with **limited receptive field units** (LRFU). An EVO network requires as many LRFU "tiles" as there are binary vectors in its input vectors. Hence networks used with the $^{33}\alpha\beta\gamma linear_{19E-LRFU}$ type have 39 LRFU tiles. Each of these LRFU consists of 19 input neurons and one hidden layer neuron. The structure of such a 19 component LRFU or **19C_LRFU** unit is shown in Figure 3. Figure 4 illustrates how EVO networks can be formed with reference to the fully connected architectures used up to now. A fully connected 39 3 1 architecture consists of layers A, B, and C, whilst a 39 1 Perceptron consists of only layers A and C. Tiling the input layer C of these progenitor networks with the 39 19C–LRFU of Layer D forms the ^{39}19C–LRFU 39 3 1 and ^{39}19C–LRFU 39 1 EVO architectures, respectively. The modus operandi of these networks can be explained with reference to Figure 4 and analogy to ^{13}C additivity rules. Until now, the combination of input vector and architecture used only allowed the network to derive an average weighting of the contribution from each input vector position. The combination of binary vector replacements *and* LRFU structure, however, also allows an optimum value to be derived for each of the 19 elements that may occupy a position. These element optima are derived using, and stored in, the weighted connections between a LRFU's input and hidden layer neurons. As back-propagation only adjusts the weights of connections with nonzero inputs, during training, each LRFU has only one weight adjusted per input vector.

Using the EVO networks described above is precluded by their large dimensionality, as a general tenet of network

Table 3. Conditions for Incuding Environments in the Two 3c3v Subclasses

subclass	Defining conditions after the restrictions (a) $^{31}P^{FO} \notin$ any ring system and (b) all $\alpha, \beta, \& \gamma \in$ (H,B,C,N,O,F,Si,P,S,Cl, Ga,Ge,As,Se,Br,Sn,Sb,Te,I)	total no. of unique vectors	no. of training set vectors	no. of monitoring set vectors	actual range spanned by the shifts of members of the subclass
$3c3v^{E}$	all α substituents $\in$ {H,C,Si,Ge,Sn}	672	500	172	[−450.0 to 87.0]
$3c3v^{NE}$	$3c3v^{E} \cap 3c3v^{NE} = \{\}$	683	500	183	[−211.8 to 293.7]

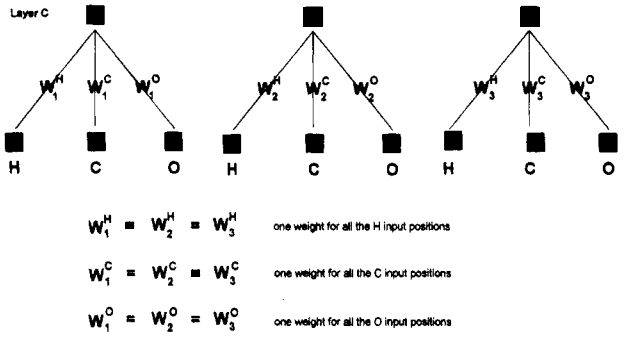


Figure 5. Reduciong network dimensionality using "shared weights".

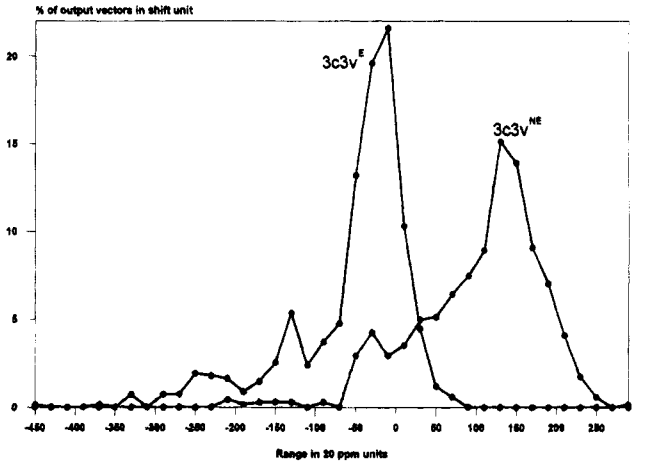


Figure 6. Distribution of $3c3v^{E}$ and $3c3v^{NE}$ subclass shifts.

training is to keep the training set: network dimensionality ratio high (at about 6:1). This problem was exacerbated by the diminished size of the 3c3v class, and the intended use of ^{31}P subclasses. Low training set: dimensionality ratios can have adverse effects on what is learned by the network. This problem was surmounted by using a shared weighting scheme to reduce the network dimensionality. These schemes can be used when the input layer contains highly correlated features and work by pooling the weights associated with these features across the input layer. Figure 5 illustrates how such a scheme works for an input layer containing a three component LRFU or 3C–LRFU unit. Pooling the weights in a ^{39}19C–LRFU 39 3 1 network forms a ^{39}19C–LRFU^{SW} 39 3 1 network and reduces the dimensionality by 722 (38 × 19). A ^{39}19C–LRFU^{SW} 39 3 1 network actually contains only 58 weights more than a 39 3 1 network, 19 from the pooled LRFUs and 39 arising from the internalization of the 39 layer C neurons, as internal neurons have a bias weight. In such a pooled scheme, instead of deriving a value for each of the 19 elements at each of the 39 positions, the ^{39}-19C_LRFU^{SW} 39 3 1 network derives a value for each of the 19 elements which is an *average* of those across the 39 positions.

It is also possible to construct architectures where the scope of a weighting scheme(s) only covers part(s) of the input layer. Designed for use only with linear format types derived from template$_{33\alpha\beta\gamma}$, the^{3,9,27}19C_LRFU^{SW} 39 3 1 architecture has three shared weighting schemes, one each for the α, β, and γ substituents. These extra schemes mean a ^{3,9,27}-19C_LRFU^{SW} 39 3 1 network has 38 more weights than a ^{39}19C_LRFU^{SW} 39 3 1 network.

EXPERIMENTAL SECTION

1. ^{31}P Subclasses. The 31-Phosphorus Handbook[32] contains many examples correlating its α (substituent) index codes with specific ranges of ^{31}P shifts. It seemed appropriate therefore to initially define a subclass as a group with identical α index codes. Unfortunately, the high environmental diversity of the 3c3v class[26] denied using this approach, as it would have created many small-sized subclasses. The adverse effects of small sizes of training set were mentioned previously. Less stringent criteria were therefore applied to divide the 3c3v class into two approximately equal size subclasses: (a) the $3c3v^{E}$ subclass where the α substituents are either periodic table group IV elements or hydrogen and (b) the $3c3v^{NE}$ subclass which contains the remaining environments. Table 3 gives summary information on each subclass, and Figure 6 shows the distribution of their output vectors.

2. Network Architectures and Training. All networks have a bias on each noninput neuron, and use the $(1/[1 + e^{-NET}])$ transfer function throughout. The learning and momentum rates are kept at values of 0.02 and 0.5 to avoid weight vector oscillations during training (See ref 26 for an explanation of the cause of such oscillations.). All network simulations were generated using the Neural Network Generator kindly made available by the MITRE Corporation.[33] All networks were trained for 40 000 epochs.

3. Evaluating Network Performances. Unless otherwise indicated, trained networks are monitored at ten Epoch intervals on a partition's single monitoring sub-set. The small subclass sizes make only one monitoring set feasible. Categories consist of either five or ten partitions. Performance is expressed as the percentage of monitoring set vectors where the difference between predicted and actual shifts is (a) within ±4 (<4) ppm, (b) within ±8 (<8) ppm, (c) without ±16 (>16) ppm, and (d) without ±20 (>20) ppm. The individual performance indicators are referred to as ***tolerance bands***. Four bands are used to check that any improved performance is general. In earlier work the averages for these bands were calculated using an ***average by epoch*** (ABE) method. Here, at every sampled epoch, an average was calculated for each tolerance band. The resulting file of averages is useful as it shows performance as a function of training time. Previous works obtained the performance maxima in tables of results from this file of averages at the epoch when the narrowest tolerance band's performance was at a maximum. This method of calculating category averages leads to some smoothing of the results as

the performance and peaks (and troughs) are normally not coincident between the individual experiments. For this work performance maxima were calculated using an ***average by value*** (ABV) method. Here, performance values are taken when the <4 ppm band is at a maximum in each *individual* experiment and the average(s) calculated from these. The formulae for both routines is given below.

A. The Average by Epoch Method

1. *make file of average performances in each band at each epoch*

$$CP^E = \frac{1}{k}\sum_{i=1}^{i=k} YP_i^E$$

where

CP^E is the category average performance at epoch E

YP_i^E is the band Y value at epoch E in the *i*th category partition

k is the number of partitions or individual expt per category

2. *get values in all bands at the epoch when band X is maximal*

$$av_{ABE} = XMAX(CP^E)$$

B. The Average by Value Method

$$av_{ABV} = \frac{1}{k}\sum_{i-1}^{i=k} XMAXP_i$$

where

$XMAXP_i$ is the maximum value in band X for the *i*th category partition

k is the number of partitions or individual expt per category

4. Meaningful Performance Differences. As in previous work a value is derived, below which any performance differences are deemed insignificant. To statistically derive this ***significance metric*** would have required a considerable increase in the number of experiments per category. As the time taken by some experiments can be quite lengthy (24 h on a 6 000 000 floating point operations per second silicon graphics IRIS D/35) the statistical approach could only be achieved by curtailing the investigation's breadth. The metric is heuristically derived, therefore, by using the difference(s) in the performance between members of ***category pairs.*** Within a pair each member has identical experimental parameters apart from using types with different structured formats, one using linear and the other branched. In this work four such pairs can be formed from the categories in group A. Theoretically, there should be no difference between the average performances of pair members providing there are sufficient experiments per category to negate the performance deviations caused by data errors.

5. Common Conditions. All types used have had their shifts scaled from [−455.5 to 293.7] to [0.1 to 0.9].

6. Conditions Unique to Each Group. In group A an experiment category consists of ten partitions, with and three factors uniquely identifying each category: (a) the subclass from which the data was obtained, (b) the type of the input vectors used, and (c) the network architecture. As identical experiments are carried out on each subclass, the 38 in this group can be treated as two sets of 19.

The nine group B categories are distinguished by (a) the subclass of the training set, (b) the subclass of the set used to evaluate the trained network, as the networks were cross-evaluated, and (c) the network architecture used. In group B due to the lengthy time required for the cross-evaluations, these categories consist of only five partitions. The original intention was to only use the ^{3,9,27}19C_LRFU^{SW} 39 3 1 architecture in group B. Valid cross evaluations cannot be obtained when such an architecture is trained with 3c3v^E types, however, as 14 weights in the α substituent weighting scheme are never adjusted. These weights correspond to elements whose presence as α substituents **excludes** the environment from being included in 3c3v^E subclass. Thus a further three categories were added. In these 3c3v^E types are trained on a ^{39}19C_LRFU^{SW} 39 3 1 architecture. Here all the weights in the weighting scheme are adjusted.

RESULTS

Group A, which compared the use of network-given and network-derived symbol replacements, has its results presented in Tables 4 and 5 for the 3c3v^E and 3c3v^{NE} subclasses, respectively. This group contains the four category pairs used to derive the significance metric of ±1 ppm, and these categories can be identified by their performances values given to one decimal place. Besides the performance maximum values in these tables, viewing the average performance as a function of training time (the ABE file) can also be revealing. Figures 7 (3c3v^E) and 8 (3c3v^{NE}) compare categories where 39 3 1, ^{39}19C_LRFU 39 3 1, and ^{3,9,27}19C_LRFU 39 3 1 architectures were trained with structured types. The legends of Figures 7 and 8 identify a category either by its SRF file "property", in which case a 39 3 1 architecture was used; or by its EVO architecture, in which case a $^{33\alpha\beta\gamma}$linear$_{19E_LRFU}$ type was used. In Figures 7 and 8 performance is shown using an *under* 20 (<20) ppm band, which was derived from the >20 ppm tolerance band.

Group B, where the scope of network derived values was investigated, has its results presented in Table 6. Part A contains the results from the six categories where a ^{3,9,27}-19C_LRFU 39 3 1 architecture was used. Part B contains the results from the categories where a ^{39}19C_LRFU 39 3 1 architecture was used.

DISCUSSION AND CONCLUSIONS

The groups are discussed individually first and then collectively.

Group A. With reference to the group A results in Tables 4 and 5 the following observations were made and conclusions drawn.

1. In each subclass, irrespective of whether the symbol replacements were given to, or derived by, a network, types with structured formats performed significantly better than the corresponding random format controls. This further supports the hypothesis that, for template based prediction, the positional equivalence property has a general scope in terms of its importance.

Table 4. Group A Results: $3c3v^{E}$ Subclass

		av_{ABV} performance values in each band when the <4 ppm band reaches a max[a]			
		% abs (target-actual) shifts *inside* the tolerance band		% ab (target-actual) shifts *outside* the tolerance band	
network architecture	type used	<4 ppm	<8 ppm	>16 ppm	>20 ppm
39 3 1	$^{33\alpha\beta\gamma}linear_{map\_eneg}$	25.5 (4)	40 (7)	33 (84)	26 (81)
39 3 1	$^{33\alpha\beta\gamma}branch_{map\_eneg}$	26.5 (4)	42 (8)	33 (83)	26 (80)
39 3 1	$^{33\alpha\beta\gamma}linear_{map\_atno}$	19 (4)	32 (7)	47 (85)	39 (82)
39 3 1	$^{33\alpha\beta\gamma}linear_{all\_1.0}$	16 (4)	26 (7)	55 (85)	48 (82)
^{39}19C_LRFU^{SW} 39 3 1	$^{33\alpha\beta\gamma}linear_{19E\_LRFU}$	30 (4)	47 (8)	30 (84)	22 (80)
^{3,9,27}19C_LRFU^{SW} 39 3 1	$^{33\alpha\beta\gamma}linear_{19E\_LRFU}$	32 (4)	50 (8)	25 (84)	18 (80)
39 1	$^{33\alpha\beta\gamma}linear_{map\_eneg}$	22.6 (11)	38 (21)	37 (59)	29 (51)
39 1	$^{33\alpha\beta\gamma}branch_{map\_eneg}$	23.6 (11)	37 (23)	38 (58)	29 (50)
39 1	$^{33\alpha\beta\gamma}linear_{map\_atno}$	12 (3)	19 (8)	63 (84)	56 (80)
39 1	$^{33\alpha\beta\gamma}linear_{all\_1.0}$	14 (10)	24 (21)	58 (62)	51 (54)
^{3,9,27}19C_LRFU^{SW} 39 1	$^{33\alpha\beta\gamma}linear_{19E\_LRFU}$	26 (5)	42 (9)	33 (83)	25 (79)
39 3 1	$^{33\alpha\beta\gamma}random_{map\_eneg}$	14 (4)	22 (8)	64 (84)	57 (81)
39 3 1	$^{33\alpha\beta\gamma}random_{map\text{-}atno}$	9 (4)	13 (7)	74 (84)	68 (80)
39 3 1	$^{33\alpha\beta\gamma}random_{all\_1.0}$	12 (4)	18 (6)	67 (85)	60 (83)
^{39}19C_LRFU^{SW} 39 3 1	$^{33\alpha\beta\gamma}random_{19E\_LRFU}$	13 (5)	19 (8)	66 (84)	59 (81)
^{3,9,27}19C_LRFU^{SW} 39 3 1	$^{33\alpha\beta\gamma}random_{19E\_LRFU}$	14 (5)	22 (8)	63 (84)	54 (81)
39 1	$^{33\alpha\beta\gamma}random_{map\_eneg}$	4 (2)	7 (4)	86 (89)	81 (85)
39 1	$^{33\alpha\beta\gamma}random_{map\_atno}$	6 (5)	10 (9)	82 (84)	77 (81)
39 1	$^{33\alpha\beta\gamma}random_{all\_1.0}$	6 (3)	9 (7)	81 (81)	76 (76)

[a] Parenthesized values are those at ten Epochs.

Table 5. Group A Results: $3c3v^{NE}$ Subclass

		av_{ABV} performance values in in each band when the <4 ppm band reaches a max[a]			
		% abs (target-actual) shifts *inside* the tolerance band		% abs (target-actual) shifts *outside* the tolerance band	
network architecture	type used	<4 ppm	<8 ppm	>16 ppm	>20 ppm
39 3 1	$^{33\alpha\beta\gamma}linear_{map\_eneg}$	15.8 (4)	25 (7)	55 (86)	47 (82)
39 3 1	$^{33\alpha\beta\gamma}branch_{map\_eneg}$	15.8 (4)	24 (8)	57 (86)	50 (82)
39 3 1	$^{33\alpha\beta\gamma}linear_{map\_atno}$	8 (3)	11 (7)	78 (86)	72 (82)
39 3 1	$^{33\alpha\beta\gamma}linear_{all\_1.0}$	11 (3)	17 (7)	69 (86)	62 (82)
^{39}19C_LRFU^{SW} 39 3 1	$^{33\alpha\beta\gamma}linear_{19E\_LRFU}$	20 (3)	32 (7)	47 (86)	38 (81)
^{3,9,27}19C_LRFU^{SW} 39 3 1	$^{33\alpha\beta\gamma}linear_{19E\_LRFU}$	19 (3)	31 (7)	43 (86)	34 (81)
39 1	$^{33\alpha\beta\gamma}linear_{map\_eneg}$	10.7 (8)	17 (16)	70 (70)	65 (63)
39 1	$^{33\alpha\beta\gamma}branch_{map\_eneg}$	10.4 (8)	17 (15)	70 (70)	63 (65)
39 1	$^{33\alpha\beta\gamma}linear_{map\_atno}$	7 (3)	10 (7)	81 (86)	75 (82)
39 1	$^{33\alpha\beta\gamma}linear_{all\_1.0}$	9 (6)	14 (13)	72 (73)	64 (66)
^{3,9,27}19C_LRFU^{SW} 39 1	$^{33\alpha\beta\gamma}linear_{19E\_LRFU}$	14 (4)	22 (7)	59 (85)	51 (80)
39 3 1	$^{33\alpha\beta\gamma}random_{map\_eneg}$	8 (4)	11 (8)	82 (85)	78 (81)
39 3 1	$^{33\alpha\beta\gamma}random_{map\text{-}atno}$	6 (4)	9 (7)	83 (85)	79 (82)
39 3 1	$^{33\alpha\beta\gamma}random_{all\_1.0}$	8 (3)	11 (7)	82 (86)	79 (82)
^{39}19C_LRFU^{SW} 39 3 1	$^{33\alpha\beta\gamma}random_{19E\_LRFU}$	10 (3)	14 (7)	75 (86)	69 (81)
^{3,9,27}19C_LRFU^{SW} 39 3 1	$^{33\alpha\beta\gamma}random_{19E\_LRFU}$	9 (3)	14 (7)	77 (86)	71 (81)
39 1	$^{33\alpha\beta\gamma}random_{map\_eneg}$	5 (3)	8 (6)	84 (89)	81 (87)
39 1	$^{33\alpha\beta\gamma}random_{map\_atno}$	6 (4)	9 (7)	85 (87)	81 (83)
39 1	$^{33\alpha\beta\gamma}random_{all\_1.0}$	5 (3)	8 (6)	85 (88)	81 (85)

[a] Parenthesized values are those at ten Epochs.

2. When replacement "properties" were given to the networks, the best performances were obtained from electronegativity, a "property" known to be correlated to the value of the NMR shifts. This order, consistent (a) with that found previously and (b) between the subclasses, gives further support to the hypothesis that one factor governing performance is the replacement values. The most compelling supporting evidence is obtained from the EVO architecture performances, however. These clearly outperform the best fully connected progenitor architectures, as is most convincingly shown in Figures 7 and 8.

3. In both subclasses significant differences occur between Perceptron (Figure 4, layers A and C) and multilayer (Figure 4, layers A, B, and C) architectures. One conclusion here is that, in either subclass, a significant percentage of a network derived mapping is nonlinear. Interestingly such differences are mirrored in their EVO network descendants, suggesting that these architectures may indeed be functioning as theorized above, i.e., that the LRFU units serve the *discrete* role of deriving and storing the element associated values.

4. Between the equivalent categories of each subclass, those in the more topologically specific $3c3v^{E}$ subclass always have a higher performance. Here equivalence is with respect to a category's type and architecture. This suggests that the topological diversity of the environments is a further factor governing template based prediction (in its current form).

5. The linearity of the network derived mapping in the more topologically specific $3c3v^{E}$ subclass is higher. Here the amount of linearity is calculated by taking the ratio of ^{3,9,27}19C_LRFU^{SW} 39 1 and ^{3,9,27}19C_LRFU^{SW} 39 3 1

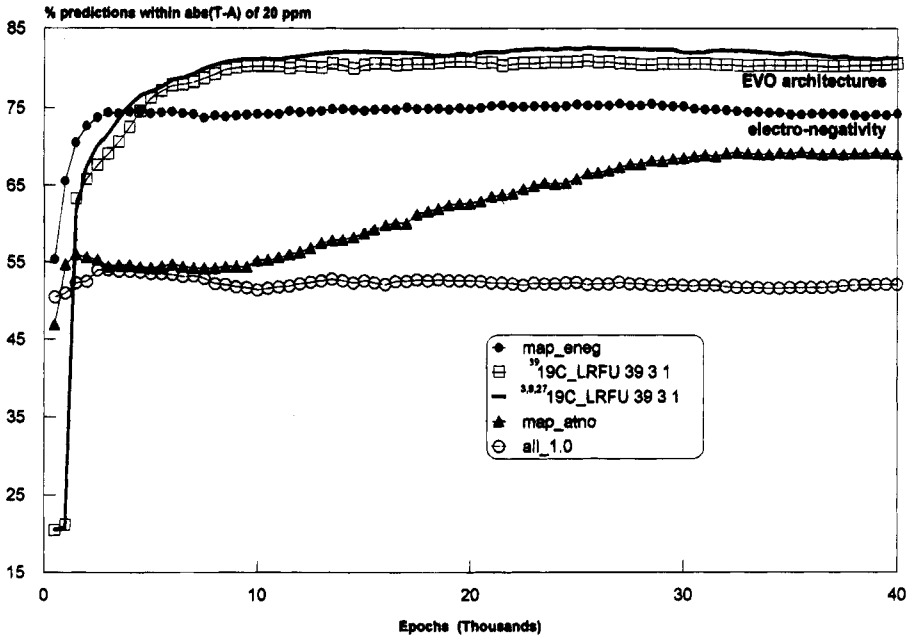


Figure 7. Types and architectures 3c3v^{E}.

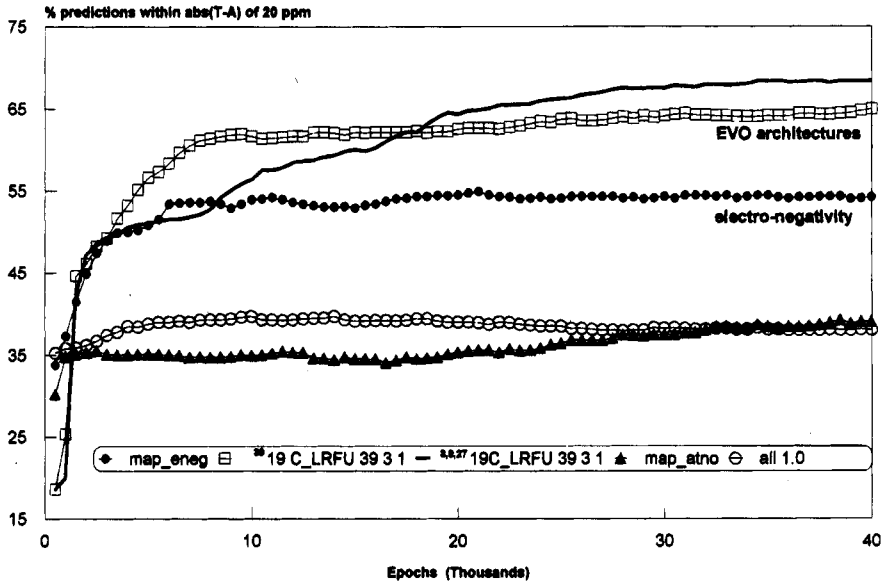


Figure 8. Types and Architectures 3c3v^{NE}.

Table 6. Group B Results: Cross-Evaluations on the 3c3v^{E} and 3c3v^{NE} Subclasses

training set subclass	subclass and partition subset used to evaluate the trained network[a]	av$_{ABV}$ performance values in each band when the <4 ppm band reaches a max[a]			
		% abs (target-actual) shifts *inside* the tolerance band		% abs (target-actual) shifts *outside* the tolerance band	
		<4 ppm	<8 ppm	>16 ppm	>20 ppm
	Part A: Network Architecture = ^{3,9,27}19C_LRFU^{SW} 39 3 1				
E	E training (500)	31 (4)	51 (9)	20 (85)	13 (80)
E	E monitoring (172)	32 (3)	51 (7)	25 (84)	19 (79)
E	NE all (683)	2 (1)	2 (2)	96 (97)	95 (96)
NE	NE training (500)	19 (4)	36 (7)	33 (86)	23 (81)
NE	NE monitoring (183)	18 (3)	31 (7)	44 (87)	36 (82)
NE	E all (672)	10 (0)	19 (0)	64 (100)	56 (100)
	Part B: Network Architecture = ^{39}19C_LRFU^{SW} 39 3 1				
E	E training (500)	27 (4)	47 (9)	25 (85)	17 (80)
E	E monitoring (172)	30 (3)	49 (7)	26 (85)	20 (79)
E	NE all (683)	2 (1)	3 (2)	94 (97)	93 (96)

[a] Parenthesized values are the number of unique vectors in the subset. Parenthesized values are those at ten Epochs.

architectures trained using $^{33\alpha\beta\gamma}$linear$_{19E_LRFU}$ types. The concurrence of higher performances and linearity in the more topologically specific subclass suggests that, as in the additivity rules, maximum performance will be obtained when the structure-shift relationship within a subclass can be linearly approximated.

6. No firm conclusions can be drawn on the relative merits of using multiple or single shared weighting schemes. With all other factors equal, the number of weight adjustments, and hence training time, using either is identical. The single scheme has the advantage of a slightly reduced network size. Comparing between the individual experiments using identical data sets, for the $3c3v^{E}$ subclass the multiple scheme consistently gives better performances in the $3c3v^{NE}$ subclass equivalent ones.

Group B. With reference to Table 6 the following observations were made and conclusions drawn.

1. In either subclass, while training and monitoring set performances are very similar, performance on vectors from the other subclass is greatly reduced. From the near equivalent performance of training and monitoring subsets within each subclass it was concluded that, in terms of characteristics relevant to the predictions, the subsets are very similar. From this, together with the attenuated performance on vectors from the other subclass, it was concluded that the network derived mappings were restricted in scope to subsets with similar characteristics to the training set.

2. Mappings derived from the more general $3c3v^{NE}$ subclass contain some predictive ability for the $3c3v^{E}$ subclass, whereas mappings derived from the more specific $3c3v^{E}$ subclass contain no predictive ability for the $3c3v^{NE}$ subclass. This suggests that the more topologically specific the training set, the more specific the mapping derived from it.

General Conclusions. The results indicate that template based prediction and additivity rules share further characteristics in that both have (a) optimum replacement values for topological features which are highly specific to the subset from which they were derived, (b) greater predictive accuracy when applied to subsets of the topological domain, (c) greater predictive accuracy when applied to more topologically specific subsets, (d) higher linearity when applied to more topologically specific subsets, and (e) mappings whose scope is correlated to the topological specificity of the subset used to derive them.

The increasing similarities between the two methods strongly suggest that, in its current form, template based prediction is a good candidate for a graphically based version of the additivity rules. If this is the case, then all the criteria required of a hybrid prediction method have a high probability of being fulfilled.

ACKNOWLEDGMENT

The author would like to thank Miller West Associates for supporting this research and Staffordshire University for providing the computer resources to run the network simulations.

REFERENCES AND NOTES

(1) Ranc, L. M.; Jurs, P. C. Simulation of carbon-13 nuclear magnetic resonance spectra of uinolines and isoquinolines. *Anal. Chim. Acta* **1991**, *248*, 183−193.

(2) Bernassau, J. M. ; Fetizon, M.; Maia, E. R. Prediction of Carbon-13 NMR Spectra. 1. Rigid Alkanes. *J. Phys. Chem.* **1986**, *96*, 6129−6134.

(3) Grant, D. M.; Paul, E. G. Carbon-13 Magnetic Resonance II. Chemical Shift Data for the Alkanes. *J. Am. Chem. Soc.* **1964**, *86*, 2984−2990.

(4) Pretsch, E.; Clerc, J. T.; Seibl, J.; Simon, W. *Tables of Spectral Data for Structural Elucidation of Organic Compounds*, 2nd ed.; Springer: Berlin, 1989.

(5) Brown, D. W. A short set of ^{13}C-NMR Correlation Tables. *J. Chem. Ed.* **1985**, *62*, 209−212.

(6) Munk, M. E.; Lind, R. J.; Clay, M. E. Computer mediated Reduction of Spectral Properties to Molecular Structures: General Design and Structural Building Blocks. *Anal. Chim. Acta* **1986**, *184*, 1−19.

(7) Bremser, W. HOSE −A Novel Substructure Code. *Anal. Chim. Acta* **1978**, *103*, 355−365.

(8) Gray, N. A. B.; Nourse, J. G.; Crandell, C. W.; Smith, D. H.; Djerassi, C.; Stereochemical Substructure Codes For ^{13}C Spectral Analysis. *Org. Magn. Reson.* **1981**, *15*, 375−389.

(9) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533−536.

(10) Bryson, A. E.; Ho. Y.-C. *Applied Optimal Control;* Hemisphere Publishing: New York, 1975 (revision of 1969 edition).

(11) Werbos, P. J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioural Sciences; Ph.D. Thesis, Harvard University, 1974.

(12) Parker, D. B. Learning Logic; Invention Report S81-64, File 1, Office of Technology Licensing: Stanford University, 1982.

(13) Parker, D. B. Optimal Algorithms for Adaptive Networks: Second Order Back Propagation, Second Order Direct Propagation, and Second Order Hebbian Learning. In *IEEE First International Conference on Neural Networks* (San Diego); Caudill, M., Butler, C. Eds.; Pubs IEEE: New York, 1987; Vol. II, pp 593−600.

(14) Fahlman, S. E. Faster-Learning Variations on Back-Propagation: An Empirical Study. *Proceedings of the 1988 Connectionist Models Summer School (Pittsburgh 1988)*; Touretzky, D., Hinton, G., Sejnowski, T. Eds.; Pubs Morgan Kaufmann: pp 38−51.

(15) Smieja, F. J.; Richards, G. D. Hard Learning the Easy Way: Backpropagation with Deformation. *Complex Systems* **1988**, *2*, 671−704.

(16) Gawthrop, P. J.; Sbarbaro, D. Stochastic Approximation and Multilayer Perceptrons: The Gain Backpropagation Algorithm. *Complex Systems* **1990**, *4*, 51−74.

(17) Davidge, R. Predicting Spectra Using Rule Induction and Neural Nets. AISBQ Postgraduate AI Workshop, 1990; pp 16−21.

(18) Kvasnicka, V. An Application of Neural Networks in Chemistry. Prediction of ^{13}C NMR Shifts. *J. Math. Chem.* **1991**, *6*, 63−76.

(19) Kvasnicka, V.; Skelenak, S.; Pospichal, J. Application of Recurrent Neural networks in Chemistry. Prediction and Classification of C13 NMR Shifts in a series of Monosubstituted Benzenes. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 742−747.

(20) Anker, L. S.; Jurs, P. C. Prediction of Carbon 13 Nuclear Magnetic Resonance Chemical Shifts by Artificial Neural Networks. *Anal. Chem.* **1992**, *64*, 10, 217−219.

(21) Doucet, J. P.; Panaye, A.; Feuilleaubois, E.; Ladd, P. Neural Networks and ^{13}C NMR Shift Prediction. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 320−324.

(22) Kolmogorov, A. N. On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition. *Dokl Akad Nauk USSR* **1957**, *114*, 953−956.

(23) Sprecher, D. A. On the Structure of continuous functions of several variables; *Trans. Am. Math. Soc.* **1965**, *115*, 340−355.

(24) Cybenko, G. Approximation by Superpositions of a Sigmoidal Function. *Math. Control, Signals Systems* **1989**, *2*, 337−341.

(25) Funahahi, K.-i. On the Approximate Realisation of Continuous Mappings by Neural Networks. *Neural Networks* **1989**, *2*, 183−192.

(26) West, G. M. J. Predicting Phosphorus NMR Shifts Using Neural Networks. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 577−589.

(27) West, G. M. J. Predicting ^{13}C NMR Shifts using Neural Networks and Graph Templates. submitted to *Anal. Chim. Acta* Jun 1995.

(28) Lindeman, L. P.; Adams, J. Q. Carbon-13 Nuclear Magnetic Resonance Spectroscopy: Chemical Shifts for the Paraffins through C_9. *Anal. Chem.* **1971**, *43*, 1245−1252.

(29) West, G. M. J. Predicting Phosphorus NMR Shifts Using Neural Networks II: Factors Influencing The Accuracy of Predictions. *J. Chem. Inf. Comput. Sci.* **1995**, *1*, 21−33.

(30) Morgan, H. L. Generation of Unique Machine Description for Chemical Structures, a Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107−113.

(31) Denker, J.; Schwartz, D.; Wittner, B.; Solla, S.; Howard, R.; Jackel, L.; Hopfield, J. Large Automatic Learning, Rule Extraction and Generalisation. *Complex Systems* **1987**, *1*, 877−922.

(32) Tebby, J. C. *Handbook of Phosphorus NMR Data*, CRC Press: U.S.A., 1991.

(33) Leighton, R.; Wieland, A. *The Aspirin/ MIGRANES Software Tools User's Manual Release V4.0*; MITRE Washington Neural Network Group: 1991.

CI940303A