Nonlinear Multivariate Analysis by Neural Network Models

Yoshio Takane

Department of Psychology, McGill University
1205 Dr. Penfield Ave., Quebec, H3A 1B1 CANADA

Summary: Feedforward neural network (NN) models approximate nonlinear functions that connect inputs to outputs by repeated applications of simple nonlinear transformations. By combining this feature of NN models with traditional multivariate analysis (MVA) techniques, nonlinear versions of the latter can readily be constructed. In this paper, we examine various properties of nonlinear MVA by NN models in two specific contexts: Cascade Correlation (CC) networks for nonlinear discriminant analysis simulating the learning of personal pronouns, and a five-layer auto-associative network for nonlinear principal component analysis (PCA) finding two defining features of cylinders. We analyze the mechanism of function approximations, focussing, in particular, on how interaction effects among input variables are captured by superpositions of sigmoidal transformations.

1. Introduction

Feed-forward neural network (NN) models and statistical models have much in common. The former can be viewed as approximating nonlinear functions that connect inputs to outputs. Many statistical techniques can be viewed as approximating functions (often linear) that connect predictor variables to criterion variables. It is thus beneficial to exploit various developments in NN models in nonlinear extensions of linear statistical techniques. There is one aspect of nonlinear transformations by NN models that is particularly attractive in developing nonlinear multivariate analysis (MVA). It allows joint multivariate transformations of input variables, so that interactions among them can be captured automatically in as much as they are needed for prediction. In this paper we examine various properties of nonlinear MVA by NN models in two specific contexts: Cascade Correlation (CC) networks for nonlinear discriminant analysis simulating the learning of personal pronouns, and a five-layer auto-associative network for nonlinear principal component analysis (PCA) recovering two defining attributes of cylinders. In particular, we analyze the mechanism of function approximations in these networks.

2. Cascade Correlation (CC) Network

NN models consists of a set of units, each performing a simple operation. Units receive contributions from other units, computes activations by summing the incoming contributions and applying prescribed (nonlinear) transformations (called transfer functions) to the summed contributions, and send out their contributions according to the activations and strengths of connections to other units. A network of such units can produce rather complicated and interesting effects. It can produce almost any kind of nonlinear effects and interactions among input variables by looking at examples that show specific input–output relationships.

The CC learning network is capable of dynamically growing nets (Fahlman & Lebiere, 1990). It starts as a net without hidden units, and it adds hidden units to improve its performance until a satisfactory degree of performance is reached. Thus, no *a priori* net topology has to be specified. Hidden units are recruited one at a time in such a way that all pre-existing units are connectd to the new one. Input units are directly connected to output units (cross conneions) as well as to all hidden units. The cross connections capture linear effects of input variables. Hidden units, on the other hand, produce nonlinear and interaction effects among the input variables that are necessary to connect inputs to outputs in some tasks. When a new hidden unit is recruited, the connection weights associated with input connections are determined so as to maximize the correlation between residuals from network predictions at the particular stage and projected outputs from the recruited hidden unit, and are fixed throughout the rest of the learning process. This avoids the necessity of back-propagating error across different levels of the network, and leads to faster and more stable convergence. The weights associated with output connections are, however, re-estimated after each new hidden unit is recruited.

The CC algorithm constructs a net and estimates connection weights based on a sample of training patterns. For each input pattern, a unit in a trained net sends contributions to units it is connected to. A contribution is defined as the product of the activation for the pattern at the sending unit and the weight associated with the connection between the sending unit and the receiving unit. The receiving unit forms its activation by summing up the contributions from other units and applying the sigmoid transformation to the summed contribution. An activation is computed at each unit and for each input pattern in the training sample. Let $\mathbf{a}_1$ denote an input pattern (a vector of activations at input units and bias, which acts like a constant term in regression analysis), and let $\mathbf{w}_1$ represent the vector of weights associated with the connections from the input and bias units to hidden unit 1 (h_1). Then, the activation for the input pattern at h_1 is obtained by $b_1 = f(\mathbf{a}_1'\mathbf{w}_1) - .5$, where f is a sigmoid function, i.e., $f(t) = 1/\{1 + \exp(-t)\}$. Now h_1 as well as the input and bias units send contributions to h_2. The activation at h_2 is then obtained by $b_2 = f(\mathbf{a}_2'\mathbf{w}_2) - .5$. A similar process is repeated until an activation at the output unit is obtained, which is the network prediction for the output. In the training phase, connection weights are determined so that the network prediction closely approximates the output corresponding to the input pattern.

3. Two-Number Identification

The CC network algorithm was first applied to the two-number identification problem, in which there are two input variables, x_1 and x_2 (excluding the bias). Pairs of x_1 and x_2 are classified into group 1 (indicated by output variable y equal to .5) when the two numbers are identical, and are otherwise classified into group 2 (indicated by y = -.5). This is a simple two-group discrimination problem, but the function to be approximated is highly nonlinear, as can be seen in Figure 1(a). The problem is interesting because of its implication to real psychological problems; identifying two objects underlies many psychological phenomena, as exemplified by an example given in the next section.

One hundred training patterns, generated by facorially combining x_1 and x_2 varied systematically from 1 to 10 in the step of 1, were used in the training. The CC network algorithm constructed a network depicted in Figure 1(b). This net has three input units (including the bias), one output unit, and two recruited hidden units. Network predictions are computed in a manner described above. Figure 1(c) displays the function approximated by the CC net (the set of network predictions as a function of x_1 and x_2). The approximation looks quite good, although the ridge at $x_1 = x_2$ in the approximated function is not as "sharp" as in the original target function. This is due to the "crudeness" of the training sample. The minimum difference between two distinct numbers in the training sample is 1, so that the net was not required

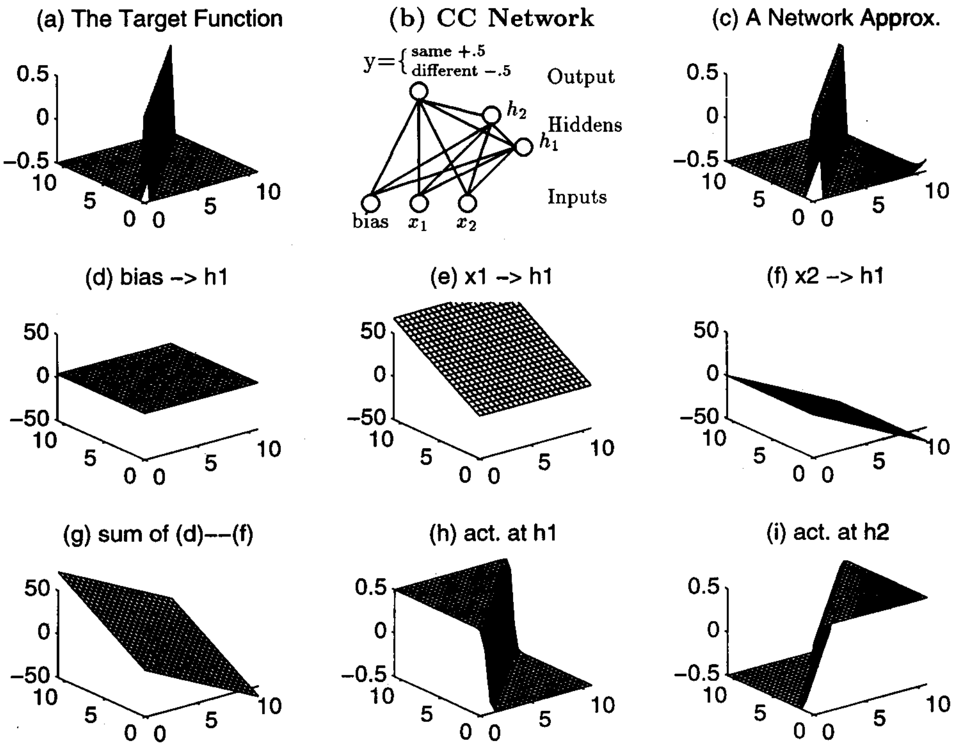


Figure 1: The mechanism of a function approximation for the two-number identification problem by CC network. (a) depicts the target function approximated by the CC network, (b), with the approximated function displayed in (c). (d) through (f) are contributions from three input units to h_1, which are summed to obtain (g), which is sigmoid-transformed to obtain the activation function (h) at h_1. The activation function at h_2 (i) is similarly derived. These activation functions are used to define contributions of the units to other units.

to discriminate beween two numbers whose differences are less than 1. The ridge in the approximated function can be made sharper if pairs of numbers with smaller differences are included in the training sample. Note that interpolations are done quite nicely. That is, although numbers like 5.5 were not included in the training sample, the identification involving such numbers are handled as expected. Extrapolation, on the other hand, seems a bit difficult, as indicated by a slight increase in function values toward the righthand side corner. Note that the target function involves a form of interaction between x_1 and x_2, where the word " interaction" is construed broadly; the meaning of a specific value on one variable, say x_1, depends on the value on the other variable, x_2.

It is interesting to see how the approximated function is bulit up and what roles the two hidden units play. Figure 1(d) through (f) present contributions of the three input units to h_1. As described above, contributions are defined as products of activations at the input units and the weights associated with the connections leading to h_1. The contributions are summed up (Figure 1(g)), and further sigmoid-transformed to obtain the activation function at h_1 (Figure 1(h)). It seems that h_1 is identifying if $x_1 \geq x_2$. The activation function at h_2 (Figure 1(i)) is similarly derived. Contributions now come from four units (three input units plus h_1). h_2 seems to be identifying if $x_2 \geq x_1$. The output unit (y) receives contributions from all other units. However, h_1 and h_2 seem to play particularly important roles. y stands out to take .5, when and only when input patterns satisfy both $x_1 \geq x_2$ and $x_2 \geq x_1$, but otherwise -.5. Interestingly, this is essentially how we prove $x_1 = x_2$ in mathematics.

4. Pronoun Learning

We were interested in the two-number identification problem because of its implication to a real psychological problem, that is, the learning of first and second person pronouns. When the mother talks to her child, *me* refers to the mother and *you* to the child. However, when the child talks to the mother, *me* refers to the child, and *you* to the mother. The child has to learn the shifting reference of these pronouns. There are three relevant input variables in this problem (excluding bias) and one output variable indicating *me* ($y = .5$) or *you* ($y = -.5$). The three input variables are speaker (sp), addressee (ad), and referent (rf). The rule (or the function) to be learned is: "Use *me* when the speaker and the referent agree (i.e., $y = .5$, when sp = rf)", and "use *you* when the addressee and the referent agree (i.e., $y = -.5$, when ad = rf)." The network should be able to judge which two of the three input variables agree in their values. The two-number identification problem is thus a prerequisite to the pronoun learning problem.

How children learn the correct use of these pronouns has been studied by Oshima-Takane (1988, 1992) and her collaborators (Oshima-Takane, et al., 1996). Simulation studies by CC networks have also been reported in Oshima-Takane, et al. (1995), and in Takane, et al. (1995). All previous simulation studies, however, presupposed the existence of only two pronouns, *me* and *you*. This severely limits the scope of these studies. In particular, the operating rule may not coincide with the one assumed above. That is, seemingly correct behavior can follow from rules other than the one described above. For example, a rule such as *me* if sp = rf and *you* otherwise, or *you* if ad = rf and *me* otherwise, works equally well so far as only *me* and *you* are considered. That is, ad = rf is equivalent to sp $\neq$ rf, and sp = rf is equivalent to ad $\neq$ rf when only *me* and *you* are to be distinguished.

We, therefore, first investigate what rule is in fact learned under the me–you–only condition. Forty training patterns were created by systematically varying the three input variables from -2 to 2 in the step of 1, and by discarding all but *me* and *you* patterns. Forty patterns were retained. (Remember that sp and ad cannot agree, and such patterns were also discarded.) The CC network algorithm recruited two hidden units to perform the task. The approximated function is depicted in Figure 2 in terms of ad on the y-axis and rf on the x-axis for nine different values of sp (-2, -1.5, -1. -.5, 0. .5, 1, 1.5, and 2). It looks like the output variable, y, takes the value of -.5, as it is supposed to (see the diagonal "ditch" observed in each graph), but it also takes the value of .5 in all other cases, including sp = rf and sp $\neq$ rf $\neq$ ad. This is correct for sp = rf, but not for sp $\neq$ rf $\neq$ ad. Remember that no training patterns were given for the latter and so it is quite natural that the net responded rather arbitrarily to the latter patterns. This implies, however, that pronouns other than *me* and *you* are necessary to learn the correct use of these two pronouns. That is, to learn to discriminate between sp = rf and sp $\neq$ rf $\neq$ ad, patterns involving other pronouns such as *he* and *she* have to be included in the training sample.

To verify the above assertion, another simulation study was conducted, this time, with pronouns other than *me* and *you* also included in the training sample. This condition, called the me-you-others condition, had 100 training patterns with 40 *me-you* patterns plus 60 *others* patterns. The net was trained to take the value of 0 ($y = 0$) when sp $\neq$ rf $\neq$ ad in addition to $y = .5$ when sp = rf and $y = -.5$ when ad = rf. Figure 3 shows the approximated function under this condition, which looks as it is supposed to. The task is appreciably more complicated than before, and the CC network algorithm recruited five hidden units to perform the task.

5. Five-Layer Auto-Associative Network

The next example pertains to a five-layer auto-associative network. A simplified version of this network is depicted in Figure 4(a). There are five layers of units including the input layer at the bottom and the output layer at the top. Units are interconnected between adjacent layers, but not within same layers or between nonadjacent layers. It is well known (e.g., Baldi & Hornik, 1989) that a three-layer neural network with linear transfer functions at both middle (hidden) and output layers has a rank reducing capability when the number of units in the hidden layer is smaller than both the number of input units and that of output units. This is a network version of (linear) reduced-rank regression (Anderson, 1951), also known as PCA with instrumental variables (Rao, 1964) and redundancy analysis (Van den Wollenberg, 1977). The usual (linear) PCA results when inputs and outputs coincide, as in Figure 4(a). The name "auto-associative" derives from the fact that this net attempts to reproduce X from input X with a reduced number of components (the number of units in the middle layer). The network version of PCA is not interesting in itself since there are more efficient and accurate algorithms to do linear PCA. It becomes interesting when the model is extended to nonlinear PCA by including two additional hidden layers with nonlinear transfer functions (most often, with sigmoidal transformations), one between the input layer and the middle layer, and the other between the middle layer and the output layer, resulting in a five-layer network. Layer 2 (hidden layer 1) and layer 4 (hidden layer 3) perform nonlinear input encoding and nonlinear output decoding, respectively. Unlike the CC network, the network topology (the number of layers, the number of units in each layer and how the units in different layers are connected) is *a priori* specified and fixed throughout the learnig process, in which only connection weights are adjusted using the backpropagation (BP) algorithm.

The five-layer auto-associative network was proposed (apparently independently) by several authors at about the same time (e.g., Irie & Kawato, 1989; Oja, 1991), and

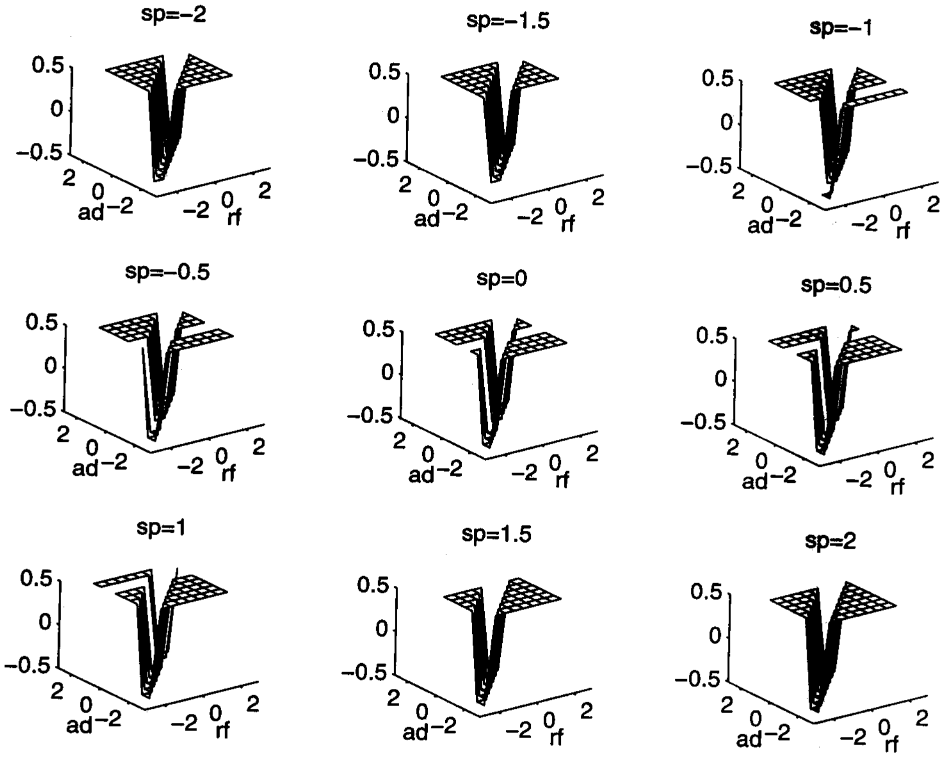


Figure 2: The approximation function for the pronoun learning problem obtained under the me-you-only condition. The function is depicted as functions of 'addressee' (y-axis) and 'referent' (x-axis) at several values of 'speaker'. Function values (z-axis) at ad=rf should indicate *you* (y = +.5), and those at sp=rf *me* (y = −.5), if the pronouns are correctly learned. The problem is that the function takes the assumed value for *me* even if sp≠rf≠ad for which no examples were given in the training. Discontinuities in the function correspond to points where sp=ad which never occurs.

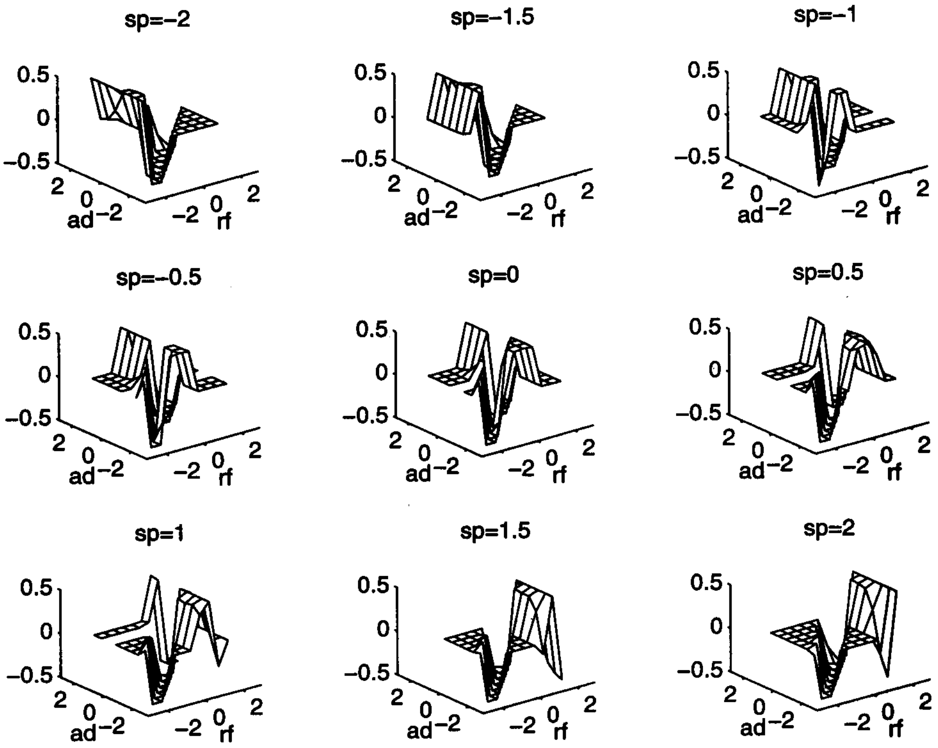


Figure 3: The same as Figure 2, but under the me-you-others condition. When the correct learning occurs, the function takes the value of *you* ($y = -.5$) if and only if ad=rf, the value of *me* ($y = +.5$) if and only if sp=rf, and $y = 0$ if and only if sp≠rf≠ad.

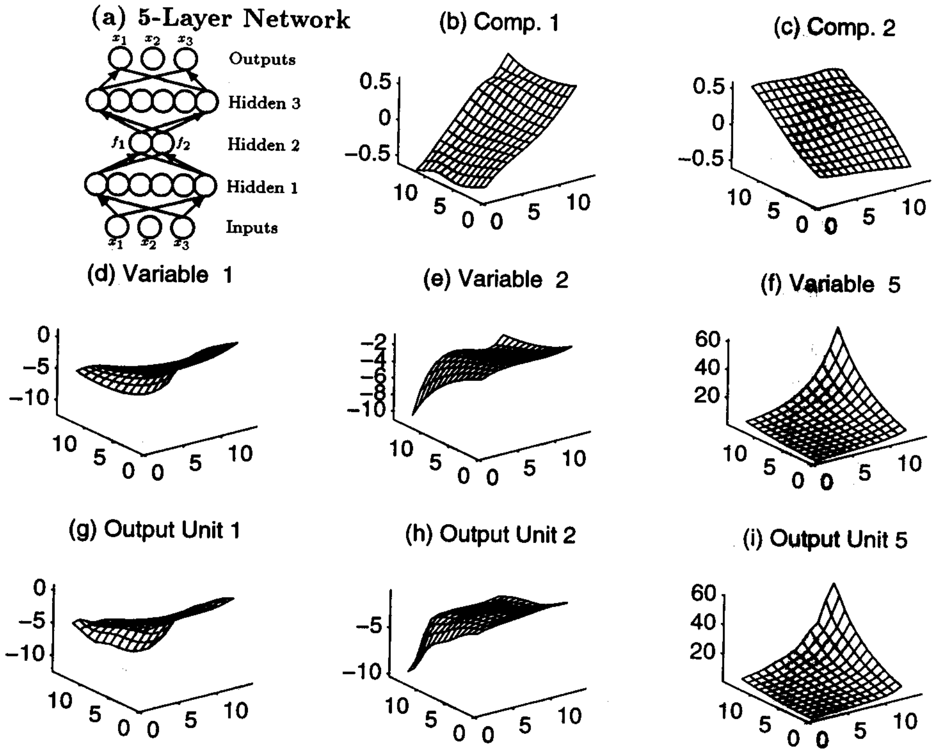


Figure 4: The mechanism of a function approximation in the five-layer auto-associative network. (a) depicts the basic construction of the network. (b) and (c) represent recovered components as functions of the original components (*ln a* on the y-axis, *ln b* on the x-axis). (d) through (f) display a sample of input functions (out of 12 altogether), and (g) through (i) recovered functions at the output units corresponding to (d)—(f).

has been applied to extracting components that determine facial expressions of emotion (DeMers & Cottrell, 1993) and internal color representation (Usui, et al., 1991). Takane (1995) examined recovery properties of nonlinear PCA by the NN models using several artificial data sets.

6. The Cylinder Problem

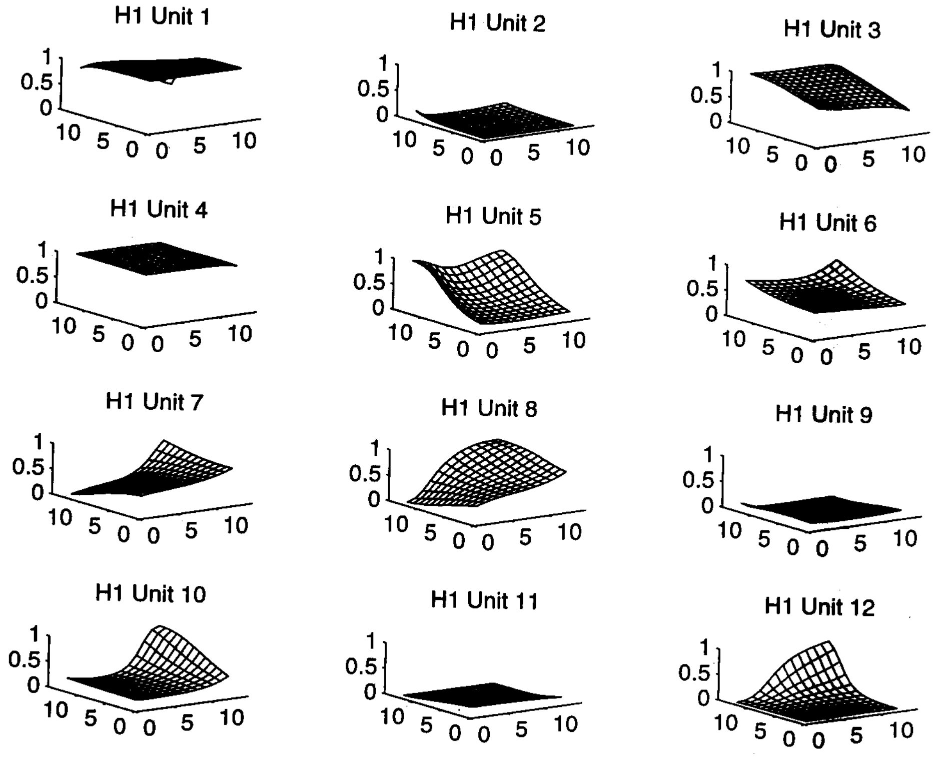
The example used to demonstrate nonlinear PCA by NN models was adapted from Kruskal & Shepard (1974), who generated a set of cylinders by systematically varying log altitude ($\ln a$) and log base area ($\ln b$) of the cylinders. These two variables serve as two assumed components to be recovered by nonlinear PCA. Kruskal & Shepard then measured the cylinders with respect to twelve variables which are all monotonic functions of $\ln a$ and $\ln b$: 1. altitude, 2. base area, 3. circumference, 4. side area, 5. volume, 6. moment of inertia, 7. slenderness ratio, 8. diagonal–base angle, 9. diagonal–side angle, 10. electrical resistance, 11. conductance, and 12. torsional deformability.

The training patterns used in the present study were generated in a similar way, except that 1) $\ln a$ and $\ln b$ were systematically varied from -.6 to .6 in the step of .1 to obtain 13 equally spaced levels, which were factorially combined to obtain 169 cylinders (as opposed to 30 prescribed cylinders in Kruskal & Shepard), and 2) after the same twelve variables were used to measure the cylinders, they were further transformed by an arbitrary linear transformation to define a completely new set of twelve variables, which may no longer be monotonic with either $\ln a$ or $\ln b$. Three examples of these variables are shown in Figure 4(d)—(f), as functions of $\ln a$ and $\ln b$. These variables are joint multivariate nonlinear transfomations of $\ln a$ and $\ln b$. Nonmetric PCA allowing only variablewise monotonic transformations is expected to have great difficuties in recovering the original components from such data. However, nonlinear PCA by means of a five-layer auto-associative network with 12 units in each of the the first and third hidden layers (this number is the same as the number of input units and that of output units) could almost perfectly recover the input data. The recovered data are shown in Figure 4(g)—(i) for the variables corresponding to those in Figure 4(d)—(f). The recovered variables at the output units look remarkably similar to the corresponding input variables, except that small wiggles are observed in the former. Figure 4-(b) and (c) give two recovered components plotted against $\ln a$ and $\ln b$. In both cases, recovered components are fairly linear with the original components.

It is interesting to see how input variables are approximated (recovered) at the ouput units with a reduced number of components in the middle layer (Hidden layer 2). Figures 5 and 6 display the activation functions created at hidden layers 1 and 3 (H_1 and H_3), respectively. The activation functions at H_1 were obtained by sigmoid transformations of linear combinations of input unit activations. They are in turn linearly combined to obtain the two recovered components at H_2. The recovered components are then linearly combined and sigmoid-transformed to obtain the activation functions at H_3. They were then linearly combined to obtain the approximated input functions at the output units.

7. Discussion

NN models present interesting perspectives to nonlinear multivariate analysis by allowing joint multivariate nonlinear transformations of input variables. In this paper, we highlighted the mechanisms of these transformations in two specific context: CC



H1 Unit 1

H1 Unit 2

H1 Unit 3

H1 Unit 4

H1 Unit 5

H1 Unit 6

H1 Unit 7

H1 Unit 8

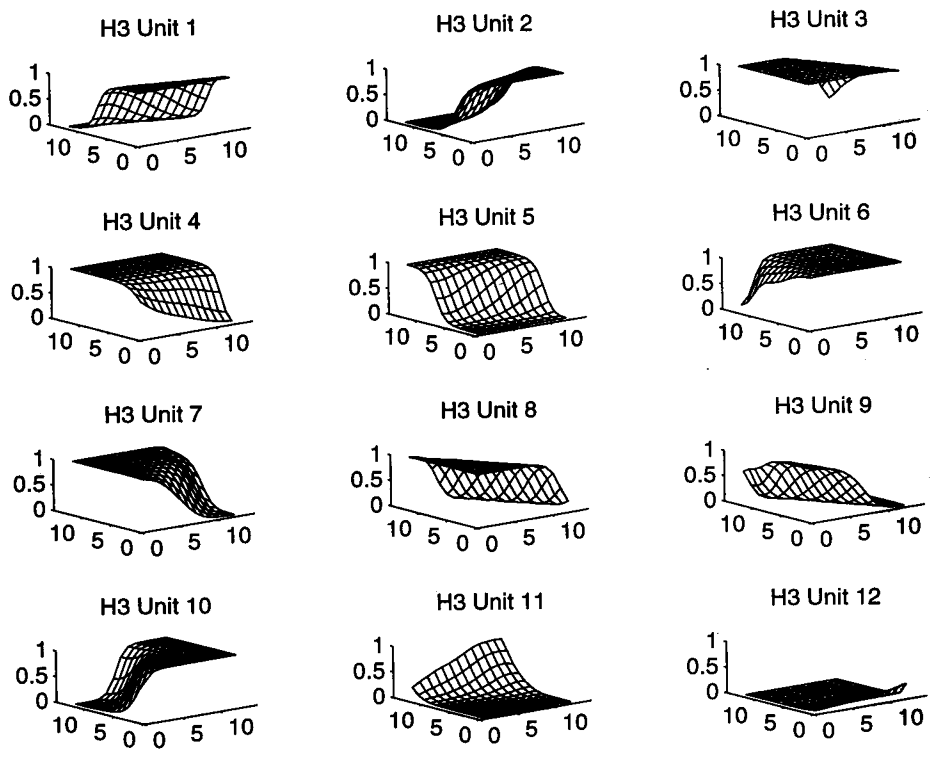
H1 Unit 9

H1 Unit 10

H1 Unit 11

H1 Unit 12

Figure 5: The activation functions created at units in hidden layer 1. These activation functions are linearly combined to obtain the activation functions ((b) and (c) of Figure 4), which are recovered component scores.



H3 Unit 1 H3 Unit 2 H3 Unit 3 H3 Unit 4 H3 Unit 5 H3 Unit 6 H3 Unit 7 H3 Unit 8 H3 Unit 9 H3 Unit 10 H3 Unit 11 H3 Unit 12

Figure 6: The same as Figure 5, but for hidden layer 3. These activation functions are linearly combined to obtain the output functions (activation functions at output units), some of which are given in (g)—(i) of Figure 4.

networks for nonlinear discriminant analysis and a five-layer auto-associative network for nonlinear PCA. In the present studies, no random errors were added in the data generation process. Investigating how the networks cope with random errors in the data is an important next step to evaluate the viability of the approach as a general method for developing nonlinear multivariate analysis techniques.

References

Anderson, T. W. (1951). Estimating linear restrictions on regression coefficients for multivariate normal distributions. *Annals of Mathematical Statistics*, **22**, 327–351.

Baldi, P. & Hornik, K. (1989). Neural network and principal component analysis: Learning from examples without local minima. *Neural Network*, **2**, 53–58.

DeMers, D., & Cottrell, G. (1993). Non-linear dimension reduction. In: *Neural Information Processing Systems 5*, Hanson, S. J. et al. (eds.), 580–587, Morgan Kaufmann, San Mateo, CA.

Fahlman, S. E., & Lebiere, C. (1990). The cascade correlation learning architecture. In: *Neural Information Processing Systems 2*, Touretzky, D. S. (eds.), 524–532, Morgan Kaufmann, San Mateo, CA.

Irie, B. & Kawato, M. (1989). Tasō pāseputoron ni yoru naibu hyōgen no kakutoku. [Acquisition of internal representation by multi-layered perceptron.] *Shingakugihō*, **NC89-15**.

Kruskal, J. B., & Shepard, R. N. (1974). A nonmetric variety of liner factor analysis. *Psychometrika*, **39**, 123–157.

Oja, E. (1991). Data compression, feature extraction, and autoassociation in feedforward neural networks. In: *Artificial Neural Networks*, Kohonen, T. et al. (eds.), 737–745.

Oshima-Takane, Y. (1988). Children learn from speech not addressed to them: The case of personal pronouns. *Journal of Child Language*, **15**, 94–108.

Oshima-Takane, Y. (1992). Analysis of pronomial errors: A case study. *Journal of Child Language*, **19**, 111-131.

Oshima-Takane, Y. et al, (1995). The learning of personal pronouns: Network models and analysis. *McGill University Cognitive Science Center Technical Report*, **2095**, McGill University.

Oshima-Takane, Y. et al. (1996). Birth order effects on early language development: Do second born children learn from overheard speech? *Child Development*, **67**, 621–634.

Rao, C. R. (1964). The use and interpretation of principal component analysis in applied research. *Sankhyā A*, **26**, 329–358.

Takane, Y. (1995). Nonlinear multivariate analysis by neural network models. *Proceedings of the 63rd Annual Meeting of the Japan Statistical Society*, 258–260.

Takane, Y. et al. (1995). Network analyses: The case of first and second person pronouns. *Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics*, 3594–3599.

Usui, S. et al. (1991). Internal color representation acquired by a five-layer neural network. In: *Artificial Neural Network*, Kohonen, T. et al. (eds.), 867–872.

Van den Wollenberg, A. L. (1977). Redundancy analysis: An alternative for canonical analysis. *Psychometrika*, **42**, 207–219.