Neural Network Simulations by Cascade Correlation and Knowledge-Based Cascade Correlation Networks

Yoshio Takane, Yuriko Oshima-Takane, and Thomas R. Shultz

Department of Psychology, McGill University, 1205 Dr. Penfield Avenue, Montreal, QC H3A 1B1 Canada. *takane@takane2.psych.mcgill.ca, yuriko@hebb.psych.mcgill.ca, thomas.shultz@mcgill.ca*

Summary. Cascade correlation (CC) has proven to be an effective tool for simulating human learning. One important class of problem solving tasks can be thought of as establishing appropriate connections between inputs and outputs. A CC network initially attempts to solve the task with a minimal network configuration, but when the task cannot be solved, it is powered up by recruiting a hidden unit to capture the uncaptured aspects of the input-output relationship until a satisfactory degree of performance is reached. Knowledge-based CC (KBCC) has a similar mechanism, but instead of recruiting hidden units, it can recruit other networks previously trained with similar tasks. In this paper we demonstrate the usefulness of these network tools for simulating learning behavior by human subjects.

Key words: Pronoun learning, Nonlinear function learning, Constructive algorithm, Knowledge transfer, Environmental bias.

1. Introduction

Many of the learning tasks we face day to day can be thought of as establishing appropriate connections between input and output variables. As an example, consider the learning of first- and second-person pronouns. When the mother talks to her child, *me* refers to herself, while *you* refers to the child. When the child talks to the mother, however, *me* refers to the child, while *you* refers to the mother. How can the child learn the semantic rule of these pronouns? There are three important input variables, indicating who the speaker is, who the addressee is, and who the referent is. When the speaker and the referent represent the same person, *me* should be used, and when the addressee and the referent represent the same person, *you* should be used (Oshima-Takane, 1988, 1992). The output variable should elicit *me* when the values on the speaker and the referent variables agree, and it should elicit *you* when the values on the addressee and the referent variables agree. The learning of pronouns can thus be considered a mapping problem from inputs to outputs.

This mapping is nonlinear and involves certain kinds of interaction effects among the input variables. In particular, the mapping should be able to identify on which two of the three input variables values agree. Multi-layered feed-forward neural networks (NN) are particularly good at capturing nonlinear and interaction effects that govern the input-output relationship without being told which nonlinear and interaction effects are important. They automatically create the necessary components by observing examples of the input-output

relationship.

It is interesting to see how the networks create the necessary components. Cascade correlation (CC) networks (Fahlman & Lebiere, 1990) are particularly attractive in simulating the process of creating the necessary components. A CC network starts with the simplest network topology, in which there are only input and output units. However, when no further improvement is possible within the current network topology, it changes its configuration by recruiting a hidden unit which captures the unpredicted part of the input-output relationship. This process is repeated until a satisfactory degree of performance is reached. The dynamic growth in problem solving capability in CC networks looks similar to that of human subjects gradually accumulating relevant knowledge to become able to solve more complicated tasks (Shultz, 2003).

CC networks, however, usually start from scratch. CC accumulates knowledge within a particular task, but that knowledge does not carry over to other situations in which related tasks are to be solved. Human subjects, on the other hand, acquire knowledge from their previous experiences and actively apply the knowledge to solve subsequent problems. Knowledge-based cascade correlation (KBCC) (Shultz & Rivest, 2001) has been developed to bridge this gap. It can incorporate prior knowledge (acquired elsewhere) to solve a current task by recruiting other networks previously trained on similar tasks. This feature of KBCC may allow more realistic simulations of human learning.

2. Cascade Correlation Learning Algorithm

Cascade correlation (CC) network (Fahlman & Lebiere, 1990) is a constructive algorithm, which allows the network to grow dynamically, starting with only the input and output units. Each input unit is directly connected to the output units by adjustable weights. Initial weights are selected randomly, and are adjusted based on target activations given in the training patterns (Initial Output Training). When performance cannot be improved any further by weight adjustments, a hidden unit with a sigmoid activation function is recruited, producing nonlinear and interaction effects in the mapping of inputs to outputs. The new hidden unit is trained in such a way that it has an activation pattern that maximally "correlates" with the current network error (Input Training). After the hidden unit has been trained, output weights are readjusted to optimize performance (Output Training). This cycle of error reduction is repeated until an acceptable performance is reached. No network topology has to be prescribed except input and output. An "optimal" network configuration is automatically determined, tailored to the level of difficulty of the task.

To illustrate, let us look at part of Figure 1 labelled "Source Net" (enclosed in a circle), which depicts a simple CC network. There are four input units labelled b', sp', ad', and rf', and one output unit labelled o'. CC starts from this

minimal configuration (without a hidden unit). It initially estimates connection weights from the input units to the output unit (Initial Output Training). This part is equivalent to the linear logistic discrimination method. It turns out that the linear effects of the input variables are not sufficient for predicting the output variable, and so a hidden unit (labelled h') is recruited. Incoming weights (input weights) to this unit are determined by maximizing the "correlation" between the unit's activation and network's current error (Input Training). Once the hidden unit is recruited and trained, it is immediately used to predict the output variable in the next phase. The weights for connections leading to o' (output weights) are re-estimated (Output Training) so that the network predictions best agree with the target outputs. In CC networks, the input weights, once estimated, are fixed throughout the remainder of the training period. Thus error is not propagated back across different levels of the network, resulting in quicker, more stable convergence. Units are connected in a cascaded manner; the input units and all previously recruited hidden units are connected to more recently recruited hidden units as well as to the output units. This helps to capture higher order nonlinearities and interaction effects among the input variables most efficiently. The input units are directly connected to the output units (cross connections). The cross connections capture linear effects of the input variables, which often account for major portions of the variability in the target function.

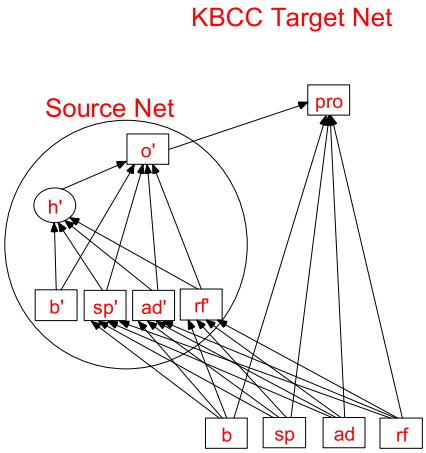

Figure 1.

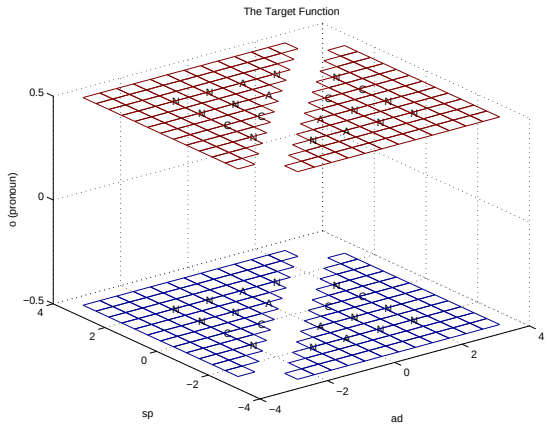

Figure 2.

The CC algorithm thus proceeds in two alternate phases, Input and Output Training phases. More specifically:

Input Training. For generality, we assume that there are K output units in the network. At stage g, the vector of input weights $\boldsymbol{v}^{(g-1)}$ to a new hidden unit is determined by maximizing

$$\phi^{(g)}(\boldsymbol{v}^{(g-1)}) = \sum_{k=1}^{K} |\boldsymbol{e}_k^{(g-1)\prime} \boldsymbol{J} \boldsymbol{z}^{(g-1)}|,$$

where $\boldsymbol{e}_k^{(g-1)}$ is the vector of current errors in output unit k, $\boldsymbol{J}$ is the matrix of centering operator, and $\boldsymbol{z}^{(g-1)}$ is a vector of activations at the new hidden unit, defined by

$$\boldsymbol{z}^{(g-1)} = \sigma(\boldsymbol{Z}^{(g-1)} \boldsymbol{v}^{(g-1)}),$$

where σ is a sigmoid transformation.

Output Training. The matrix of activations at stage g is obtained by appending the matrix of activations in the previous stage, $\boldsymbol{Z}^{(g-1)}$, by $\boldsymbol{z}^{(g-1)}$. That is, $\boldsymbol{Z}^{(g)} = [\boldsymbol{Z}^{(g-1)}, \boldsymbol{z}^{(g-1)}]$. The matrix of output weights, $\boldsymbol{W}^{(g)}$, is determined by minimizing

$$\psi^{(g)}(\boldsymbol{W}^{(g)}) = \mathrm{SS}(\boldsymbol{O} - \hat{\boldsymbol{O}}^{(g)}).$$

where for any matrix $\boldsymbol{X}$, $\mathrm{SS}(\boldsymbol{X}) = \mathrm{tr}(\boldsymbol{X}'\boldsymbol{X})$, $\boldsymbol{O}$ is the matrix of target outputs, and $\hat{\boldsymbol{O}}^{(g)}$ is the matrix of output predictions defined by

$$\hat{\boldsymbol{O}}^{(g)} = \sigma(\boldsymbol{Z}^{(g)} \boldsymbol{W}^{(g)}).$$

The algorithm starts with the second phase with $\boldsymbol{Z}^{(g)}$ equated to the matrix of inputs. See Shultz (2003), and Takane, Oshima-Takane, and Shultz (1999) for more detail.

3. Pronoun Learning Problem

The learning of first and second pronouns presents an interesting problem to psychologists because of the shifting reference of these pronouns. Oshima-Takane and her collaborators (Oshima-Takane, 1988, 1992; Oshima-Takane, Goodz, & Derevensky, 1996) have conducted a series of extensive studies on this topic with human subjects. As noted earlier, the problem can be regarded as a special type of rule (nonlinear function) learning, where the rule to be learned is: Use *me* when the speaker and the referent agree, and use *you* when the addressee and the referent agree.

To analyze the task more closely, let us look at Table 1. There are three input variables: Speaker (sp), Addressee (ad) and Referent (rf), and there is one output variable (o) indicating the pronoun to be used (*me* or *you*). For the moment, we assume that there are only three persons involved: Child, Mother and Father. (This number will be increased to five in the simulations to be reported later.) The three input variables can take either one of these three values. There are two constraints in forming input patterns: (1) Speaker and Addressee can never agree, and (2) Either Speaker and Referent agree, or Addressee and Referent agree. These constraints limit the number of possible patterns to 12. It can be verified that the rules mentioned in the previous paragraph indeed hold for all the 12 patterns. For example, when Father talks

to Child referring to himself, he uses *me*, while when Mother talks to Child referring to Child, she uses *you*, etc. The child has to learn the three relevant input variables and be able to identify which two of the three variables take identical values.

Table 1. Training Patterns in the Three-Person Situation

Condition		Input Variables			Output Variable
		Speaker	Addressee	Referent	Pronoun
Addressee patterns	1)	Father	Child	Father	me
	2)	Father	Child	Child	you
	3)	Mother	Child	Mother	me
	4)	Mother	Child	Child	you
Nonaddressee patterns	5)	Father	Mother	Father	me
	6)	Father	Mother	Mother	you
	7)	Mother	Father	Mother	me
	8)	Mother	Father	Father	you
Child-speaking patterns	9)	Child	Father	Child	me
	10)	Child	Father	Father	you
	11)	Child	Mother	Child	me
	12)	Child	Mother	Mother	you

Notice that there are three distinct groups of input patterns in the table. The first group, called addressee patterns, consists of patterns in which the addressee is always Child. The second group, called nonaddressee patterns, consists of patterns in which Child is neither the speaker nor the addressee. The third group, called child-speaking patterns, consists of patterns in which the speaker is always Child. Oshima-Takane (1988) hypothesized that relevant information necessary for learning the correct use of the pronouns is not provided in the speech addressed to the child, and that the child has to be exposed to the nonaddressee patterns (i.e., to pay attention to overheard speech) to learn their correct use. Her hypotheses have been empirically verified in both experimental and observational studies (Oshima-Takane, 1988, 1992; Oshima-Takane, et al., 1996). However, for obvious ethical reasons children cannot be tested under the pure addressee or nonaddressee condition. This is where simulation studies will be important because neural nets can be trained under these pure conditions. According to Oshima-Takane's hypotheses, networks will learn an incorrect function (or rule) when trained with only addressee patterns, but arrive at a correct function when trained with nonaddressee patterns. The child-speaking patterns provide a test of whether the correct function is learned or not. Changes in learning as a result of changes in inputs are generally known as the problem of environmental bias.

4. Simulations of Pronoun Learning by CC

We investigate the effects of environmental bias in pronoun learning by CC network simulations. Since the pronoun learning problem is equivalent to finding a nonlinear function that connects inputs to outputs, we are in effect investigating changes in function approximations due to environmental bias. For the purpose of simulations we arbitrarily assigned the values of 0, 2, and -2 to Child, Mother, and Father on the three input variables, and .5 to *me* and $-.5$ to *you* on the output variable. However, a previous study (Oshima-Takane, Takane, and Shultz, 1999) indicated that three persons were not sufficient for CC networks to learn a correct function and generalize properly. To provide a richer learning environment (more examples), we added two other persons, who were coded 1 and -1 in the simulation studies. With five persons in total, we obtain 8 addressee patterns, 24 nonaddressee patterns, and 8 child-speaking patterns. The target function can be formally stated as

$$o = (ad - rf)/(ad - sp) - 0.5,$$

where sp, ad and rf are the values of the speaker, the addressee, and the referent variables, respectively. It can easily be verified that when $sp = rf$, $o = .5$, and when $ad = rf$, $o = -.5$.

Figure 2 presents a graphical display of the target function. The *me* surface is presented at the top ($o = .5$), and that of *you* is presented at the bottom ($o = -.5$). Note that for the *me* surface, the axis labelled sp represents both the speaker and the referent variables which should agree, while the axis labelled ad represents the addressee variable. For the *you* surface, on the other hand, the axis labelled ad represents both the addressee and the referent variables which should agree, while the axis labelled sp represents the speaker variable. These surfaces were drawn for the values between -3.5 and 3.5 for all the input variables. The letter A on each surface indicates addressee patterns, the letter N nonaddressee patterns, and the letter C child-speaking patterns, used in the training. No other points on the grids were used in the training.

Two simulation studies were conducted to test Oshima-Takane's hypotheses using CC networks. In the first study, nets were trained under the pure addressee condition, while in the second study under the pure nonaddressee condition. We expect that the nets will learn an incorrect function under the former condition, and a correct function under the latter condition. We can graphically display the function learned in each simulation study. In both conditions one hidden unit was recruited. The two top figures in Figure 3 (3a and 3b) show the *me* and the *you* surfaces, respectively, constructed under the pure addressee condition. Both surfaces correctly discriminate the four addressee patterns for *me* from the four addressee patterns for *you*. (Read off the function values at ad = 0 in both figures.) However, they do not correctly discriminate the eight child-speaking patterns. (Read off the function values at sp = 0 in

the two figures.) Nonaddressee patterns are also not properly discriminated. As expected, further training was necessary to deal with the child-speaking and nonaddressee patterns. The bottom portions of Figure 3 (3c and 3d) show the *me* and *you* surfaces obtained from the pure nonaddressee condition. They correctly discriminate not only the nonaddressee patterns used in the training, but also the child-speaking and the addressee patterns. Generalizations (function values at untrained points) also seem quite good, although the *me* surface shows a sign of problems in generalization in the lower right corner. The surfaces look quite similar to the corresponding target functions depicted in Figure 2. The nonaddressee patterns are indeed crucial for pronoun learning, as hypothesized by Oshima-Takane (1988, 1992), indicating the importance of overheard speech. These results conform to the findings by Oshima-Takane, Takane, and Shultz (1999).

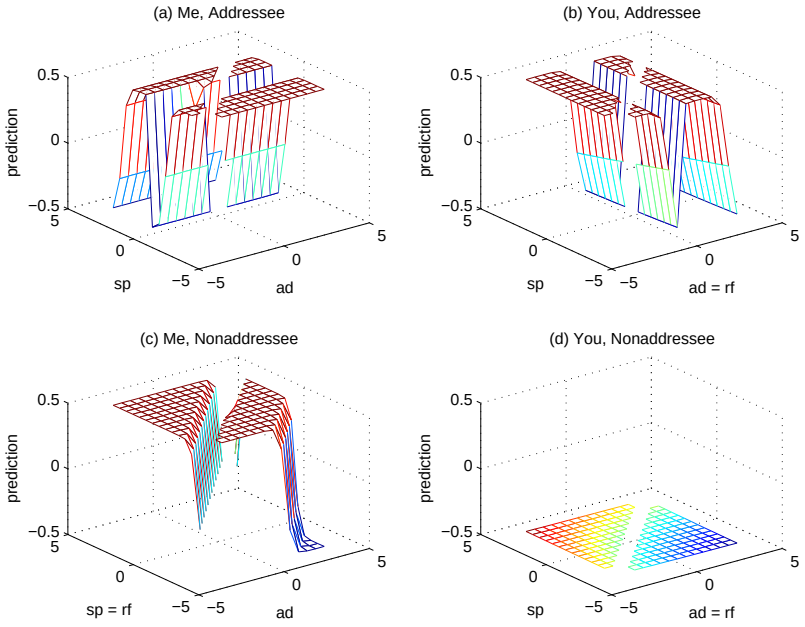

Figure 3. Approximated Functions: Addressee (a, b) and Nonaddressee Condition (c, d).

5. KBCC Network Simulations

Knowledge-based cascade correlation (KBCC) networks can incorporate prior knowledge in learning a new task by recruiting other networks previously trained with related tasks. Figure 1 depicts a KBCC target network, where a source net pre-trained by CC has been recruited instead of a single hidden unit. When the source net is being recruited, its inputs (except the bias unit) are connected to all existing units (except the output units) in the target network. The source inputs are trained (the weights leading to the source inputs are adjusted) in such a way that outputs from the source net are maximally cor-

related with current target network error. The input connections are assumed to have linear transfer functions. Once the source input training is done, the output training in the target network proceeds just as in ordinary CC. Shultz and Rivest (2001) describe the algorithm for KBCC in detail along with a report of numerical experiments assessing the capability of KBCC networks under a variety of conditions.

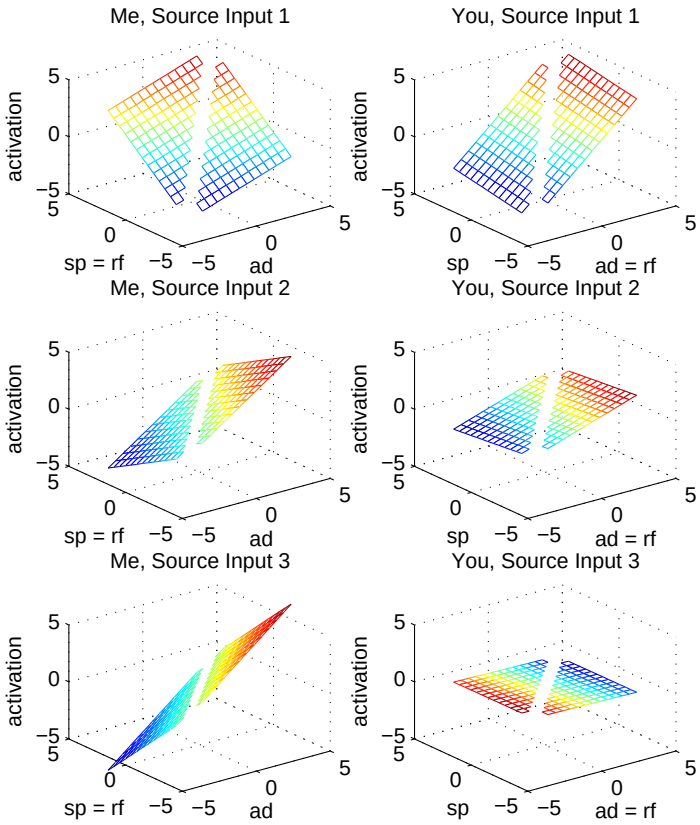

Figure 4. Input Recoding: Activations at Source Inputs

We examined effectiveness of KBCC networks in the pronoun learning task. Addressee nets learned an incorrect function in CC network simulations. They could capture only addressee patterns with which they were trained, but could not generalize correctly to untrained patterns. We may ask, however, whether they learned anything useful for eventual learning of the untrained patterns. What happens if an addressee net is used as a possible source net in a KBCC network? Would this source net facilitate the learning of child-speaking and nonaddressee patterns? We simulated the learning of addressee and child-speaking patterns by KBCC with the addressee nets as possible source nets. We were interested to find whether the addressee source net has any effects (either facilitating or interfering) on learning, and if the learned function gen-

eralizes to nonaddressee patterns.

We first examined how the input training of the addressee source net could change the output activations in the source net. Figure 4 shows activations at source net inputs (labelled Inputs 1, 2, and 3 which correspond with sp', ad', and rf' in Figure 1, but they no longer represent sp, ad, and rf as such) in a KBCC network, which are obtained by linear transformations of the input units (labelled b, sp, ad and rf) in the target network. Of the three source input units, the only crucial unit turned out to be Input 3 (rf') because in the original addressee net the only significantly nonzero weight was for the connection from rf' to h'. Other input units (sp' and ad') did not play any significant roles. Figure 5 indicates output predictions (activations) from the source net. As indicated, these predictions are totally different from the original outputs from the addressee source net which were displayed in Figure 3a and 3b. This shows how drastically the input recoding (linear transformations of the original inputs) can change the output predictions from a source net. The source output turned out to be nearly identical to those from the final target network (so much so that no separate figures are given for the target outputs). Somewhat surprisingly, the final output functions resemble the target function for the learning of first and second person pronouns depicted in Figure 2. The *you* surface is perfectly recovered including the untrained nonaddresse patterns, while the *me* surface has some observable departure from the target function. These surfaces look similar to those obtained by the nonaddressee CC nets (Figure 3c and 3d). (However, the prediction error for the nonaddressee patterns was somewhat larger for the KBCC nets than that of the nonaddressee CC nets.) The activation pattern at h' was also very similar to the hidden unit activations in the nonaddressee CC net. This means that source input training can make the source net work like a nonaddresse CC net by linear transformations of the original inputs. A closer inspection of the activation pattern at Input 3 (rf') in the source net has revealed that it represents a linear combination of the original input units similar to the sum of contributions of these units that goes into the hidden unit in the nonaddressee CC net.

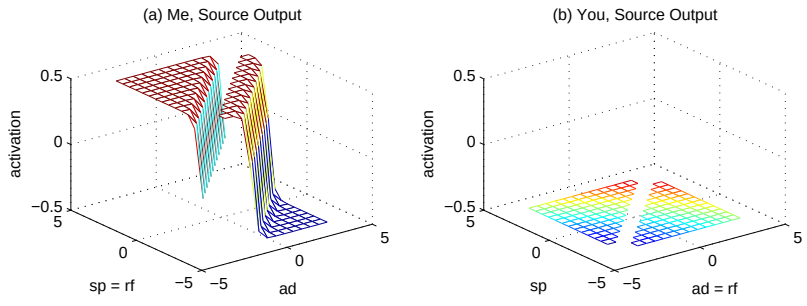

Figure 5. Output Predictions at Source (and Target) Outputs

6. Conclusion and Future Direction

KBCC presents an interesting paradigm for how prior knowledge may be used in human learning, although obviously more systematic investigations of knowledge representation in KBCC are in order.

When only the first and second person pronouns are used in a simulation, sp = rf implies ad ≠ rf, and ad = rf implies sp ≠ rf because no patterns in which sp ≠ rf ≠ ad are included in the training. This means that a network does not have to learn the true rule. Apparently correct behavior follows if it uses a degenerate rule: *me* if sp = rf and *you* otherwise, or *you* if ad = rf and *me* otherwise. Indeed, this was the case (Takane, 1998), and for the network to learn the true rule, pronouns other than *me* and *you*, e.g., *he* and *she*, have to be included (Oshima-Takane, Takane, & Takane, 1999). Similar simulation studies by KBCC involving pronouns other than *me* and *you* would undoubtedly be interesting.

References

Fahlman, S. E., and Lebiere, C. (1990). The cascade-correlation learning architecture. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems 2* (pp. 524-632). Los Altos, CA: Morgan Kaufmann.

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 pronominal errors: A case study. *Journal of Child Language*, **19**, 111-131.

Oshima-Takane, Y., Goodz, N., and Derevensky, J. L. (1996). Birth order effects on early language development: do secondborn children learn from overheard speech? *Child development*, **67**, 621-634.

Oshima-Takane, Y., Takane, Y., and Shultz, T. R. (1999). The learning of first and second person pronouns in English: network models and analysis. *Journal of Child Language*, **26**, 545-575.

Oshima-Takane, Y., Takane, M., and Takane, Y. (1999). Learning of first, second, and third person pronouns. *Proceedings of the Eighteenth Annual Meeting of the Cognitive Science Society* (pp. 800-805). Hillsdale, NJ: Erlbaum Associates.

Shultz, T. R. (2003). *Computational Developmental Psychology.* Cambridge, MA: MIT Press.

Shultz, T. R., and Rivest, F. (2001). Knowledge-based cascade-correlation: using knowledge to speed learning. *Connection Science*, **13**, 43-72.

Takane, Y. (1998). Nonlinear multivariate analysis by neural network mod-

els. In C. Hayashi, N. Ohsumi, K. Yajima, Y. Tanaka, H. H. Bock and Y. Baba (Eds.), *Data Science, Classification, and Related Methods* (pp. 527-538). Tokyo: Springer Verlag.

Takane, Y., Oshima-Takane, Y., and Shultz, T. R. (1999). Analysis of knowledge representations in cascade correlation networks. *Behaviormetrika*, **26**, 5-28.

In Higuchi, T., Iba, Y., and Ishiguro, M. (Eds.) *Proceedings of Science of Modeling: The 30^{th} Anniversary Meeting of the Information Criterion (AIC)*, (pp. 245-254). Tokyo: The Institute of Statistical Mathematics, 2003. Revised slightly on December 31, 2003.