# Analysis of Unscaled Contributions in Cross Connected Networks

**Thomas R. Shultz** and **Yuriko Oshima-Takane**
shultz@psych.mcgill.ca yuriko@hebb.psych.mcgill.ca
Department of Psychology
McGill University
Montréal, Québec, Canada H3A 1B1

**ABSTRACT**

Contribution analysis is a useful tool for the analysis of cross-connected networks such as those generated by the cascade-correlation learning algorithm. Networks with cross connections that supersede hidden layers pose particular difficulties for standard analyses of hidden unit activation patterns. A contribution is defined as the product of an output weight and the associated activation on the sending unit. Previously such contributions have been multiplied by the sign of the output target for a particular input pattern. The present work shows that a principal components analysis (PCA) of unscaled contributions yields more interesting insights than comparable analyses of contributions scaled by the sign of output targets.

## 1 INTRODUCTION

Solutions learned by neural networks are often quite difficult to understand because of the complex non-linear properties of neural nets and the common use of distributed representations. Standard techniques of network analysis, based either on a network's weights or its hidden unit activations have been somewhat limited. The most notable features of weight diagrams are often the complexity of the pattern of weights and its variability across multiple networks learning the same problem. Statistical analysis of activation patterns on hidden units is limited to nets with a single hidden layer without cross-connections.

Cross connections are direct connections that bypass intervening hidden layers. They are known to increase learning speed in back-propagation networks (Lang & Witbrock, 1988) and are a standard feature of some generative learning algorithms, such as cascade-correlation (Fahlman & Lebiere, 1990). Because such cross connections carry so much of the work load, any analysis restricted to hidden unit activations provides at best a partial picture of the network solution.

Contribution analysis appears to be a useful technique for multi-layer, cross connected nets. Sanger (1989) defined a contribution as the triple product of an output weight, the activation of a sending unit, and the sign of the output target for that input. Contributions are potentially more informative than either weights alone or hidden unit activations alone since they take account of both weight and sending activation. Shultz and Elman (1994) used principal components analysis to reduce the dimensionality of such contributions in several different types of cascade-correlation nets.

The present work explores whether it is preferable to employ contributions that are scaled by the sign of their output targets or to use unscaled contributions in network analysis. Sanger (1989) recommended scaling contributions by the signs of output targets in order to determine whether the contributions helped or hindered the network's solution. However, since target signs are not available to networks except as error correction signals, it could be argued that it is more natural to use unscaled contributions in analyzing knowledge representations.

Understanding the knowledge representations in network solutions may be useful in a variety of contexts. It is surely useful in the area of cognitive modeling, where the mere ability of nets to simulate psychological phenomena does not suffice. It is also critically important to determine whether the representations developed by networks bear any systematic relation to the representations employed by human subjects (McCloskey, 1991).

## 2 PRINCIPAL COMPONENTS ANALYSIS OF CONTRIBUTIONS

In contrast to Sanger's (1989) three-dimensional array of contributions (output unit x hidden unit x input pattern), we begin with a two-dimensional output weight x input pattern array of contributions. This is more efficient than the slicing technique used by Sanger to focus on particular output or hidden units and yet allows identification of the roles of specific contributions (Shultz & Elman, 1994).

We subject the correlations among contributions across input patterns to PCA, a statistical technique that identifies dimensions of variation (Flury, 1988). A component is a line of closest fit to a set of points in multi-dimensional space. PCA summarizes a multivariate data set in a few components by capitalizing on correlations among the variables.

Here we apply PCA to contributions taken from networks learning either continuous XOR or arithmetic comparisons. The contribution matrix for each net is subjected to PCA with 1.0 as the minimum eigenvalue for retention. Varimax rotation is applied to improve the interpretability of the solution. Component scores are plotted to indicate the function of the components and component loadings are examined to determine the roles of particular contributions.

## 3 APPLICATION TO THE CONTINUOUS XOR PROBLEM

The classical binary XOR problem has too few training patterns (four) to require contribution analysis. We construct a continuous version of the XOR problem by dividing the input space into four quadrants. Input values are incremented in steps of 0.1 starting from 0.1, yielding 100 *x*, *y* input pairs. Quadrant *a* has values of *x* less than 0.55 combined with values of *y* above 0.55. Quadrant *b* has values of *x* and *y* greater than 0.55. Quadrant *c* has values of *x* and *y* less than 0.55. Quadrant *d* has values of *x* greater than 0.55 combined with values of *y* below 0.55. Problems from quadrants *a* and *d* produce a positive output target, whereas problems from quadrants *b* and *c* yield a negative output target.

Three cascade-correlation nets are trained on continuous XOR. Each net generates a unique solution, recruiting either five or six hidden units and taking from 541 to 765 epochs. PCA of unscaled contributions yields three components rather than the two yielded by PCA of scaled contributions (Shultz & Elman, 1994). Plots of rotated component scores for the 100 training patterns are less dense but more interesting for unscaled than for scaled contributions.

Two-dimensional plots of component scores for net 1 are shown in Figure 1 and labeled according to their respective quadrant. Figure 1a, plotting scores on components 1 and 3, shows that component 1 reflects the second input dimension (quadrants *a* and *b* vs. quadrants *c* and *d*). Figure 1b, plotting scores on components 2 and 3, shows that component 2 reflects the first input dimension (quadrants *b* and *d* vs. quadrants *a* and *c*). Both Figures 1a and 1b reveal that component 3 separates the quadrants with a positive output target (*a* and *d*) from those with a negative output target (*b* and *c*). Similar results were obtained for the two other nets. In contrast, plots of component scores for scaled contributions indicated interactive separation of the four quadrants, but with no clear individual roles for the two components (Shultz & Elman, 1994).

Figure 2 plots the rotated component scores for this net. Such plots can be examined to determine the role of each contribution in the network. For example, input 2 and hidden units 1, 5, and 6 all participate in the job done by component 1, namely the representation of the second input dimension.

## 4 APPLICATION TO COMPARATIVE ARITHMETIC

Arithmetic comparison tasks require nets to compare sums or products to some value and then output whether the sum or product is greater than, less than, or equal to that comparative value. The fact that several psychological simulations using neural nets involve problems of linear and non-linear arithmetic operations enhances interest in this sort of problem (McClelland, 1989; Shultz, Schmidt, Buckingham, & Mareschal, in press).

Addition and multiplication tasks each involve three linear input units. The first two input units each code a randomly selected integer in the range from 0 to 9, inclusive, and the third input unit codes a randomly selected comparison integer. For addition problems, comparison values range from 0 to 19, inclusive; for multiplication, comparison values range from 0 to 82, inclusive. Two output units code the results of the comparison. Target outputs of +- represent that the results of the arithmetic operation are *greater than* the comparison value, targets of -+ represent *less than*, and targets of ++ represent *equal to*. For problems involving both addition and multiplication, a fourth input unit codes the type of arithmetic operation to be performed: 0 for addition, 1 for multiplication.

Nets trained on either addition or multiplication have 100 randomly selected training patterns, with the restriction that 45 of them have correct answers of *greater than*, 45 have correct answers of *less than*, and 10 have correct answers of *equal to*. These constraints reduce the skew of comparative values in the high direction on multiplication problems. Nets trained on both addition and multiplication receive 100 randomly selected addition problems and 100 randomly selected multiplication problems. There are three addition nets, three multiplication nets, and three nets trained on both addition and multiplication.

### 4.1 ADDITION RESULTS

Each of the three nets learning addition problems recruited a single hidden unit. They took between 155 and 169 epochs to learn. PCA of unscaled contributions in each net yields three significant components, unlike the two components obtained with scaled contributions.

Component score plots, such as that for net 1 in Figure 3, indicate that component 1 distinguishes *less than* from *greater than* answers. Problems with *equal to* answers were not isolated by the three components. Components 2 and 3 are

particularly sensitive to variation in the size of the first and second integers to be added, respectively. This was revealed by examining extreme component scores on these components, either greater than 1.0 or less than -1.0. Problems with extremely negative component 2 scores had a mean of 8.41 for the first integer and 5.36 for the second integer. Problems with extremely positive component 2 scores had a mean of 1.00 for the first integer and 5.52 for the second integer. This indicates that component 2 is primarily sensitive to the size of the first integer input. In contrast, component 3 was sensitive to the size of the second integer input with means of 1.48 for extremely negative component scores and 8.36 for extremely positive component scores. The means on the first integer input did not vary much with extremity of component 3 score: 4.70 vs. 4.05. Similar results obtained for the other two nets.

PCA of scaled contributions had produced two components that were sensitive only to answer type and not to variation in integer input. As with the continuous XOR problem, the plots of component scores were denser for scaled contributions, but not as revealing (Shultz & Elman, 1994).

### 4.2 MULTIPLICATION RESULTS

Multiplication is a much more difficult problem for nets with additive activation functions, as revealed by the fact that the nets learning multiplication comparisons required from 832 to over 1000 epochs and recruited between six and eight hidden units. Runs were terminated when they reached 1000 epochs. PCA applied to the contributions in these nets yields from 4 to 6 significant components. Plots of rotated component scores for two of the four components from net 3 are presented in Figure 4. This plot shows that most of the separation of *greater than* from *less than* outputs was accomplished by component 2. Component 1 served to make this distinction for the remaining problems. Problems with *equal to* answers were not isolated by any of the four components.

Component 1 also served to represent variation in the second input. Problems with extremely high scores on component 1 have a mean second input of 8.57; those with extremely low scores on component 1 have a mean second input of 0.56. Component 3 serves a similar role for the first input. Problems with extremely high scores on component 3 have a mean first input of 8.11; those with extremely low scores on component 3 have a mean first input of 1.10. The role of component 4 is opaque. Basically similar results were obtained for the other two multiplication nets. In contrast, PCA of scaled components were less revealing, except for offering a clear separation of answer types (Shultz & Elman, 1994).

### 4.3 RESULTS FOR NETS DOING BOTH ADDITION AND MULTIPLICATION

Learning to do both addition and multiplication is even more difficult than multiplication alone. None of the three nets quite reached victory by 1000 epochs, but each did come close. Either seven or eight hiddens units were recruited. PCA of contributions yields five components in each of the three nets. Besides the familiar distinctions between problem types and variation in integer inputs found in nets doing either addition or multiplication, it is of interest to determine whether nets doing both operations distinguish between adding and multiplying.

Figure 5 shows rotated component scores for three components from net 3. Component 1 separates *greater than* from *less than* answers. Component 5 and, to a lesser extent, component 4 separate adding from multiplying. The role of component 4 is not very clear from Figure 5, but various two-dimensional plots of component 4 reveal that it separates adding vs. multiplying for problems with *less than* answers.

Components 2 and 3 handle variation in the first and second input integers, respectively. Problems with extremely positive component 2 scores have a mean first input integer of 8.53; problems with extremely negative component 2 scores have a mean first input integer of 0.84. Problems with extremely positive component 3 scores have a mean second input integer of 8.55; problems with extremely negative component 3 scores have a mean second input integer of 1.05. Problems with *equal to* answers are not isolated by any of the components. Results for the other two nets learning both multiplication and addition comparisons are essentially similar to these. In contrast, PCA of scaled contributions had produced three components that interactively separated the three answer types and operations, but did not represent variation in input integers (Shultz & Elman, 1994).

## 5 DISCUSSION

As with continuous XOR, there is considerable variation among networks learning comparative arithmetic problems. Yet with all of this variation, it is apparent that the nets learn to separate arithmetic problems according to features afforded by the training set. Nets learning either addition or multiplication differentiate the problems according to answer types and nets learning both arithmetic operations supplement these answer distinctions with the operational distinction between adding and multiplying. Variation along the integer input dimensions is also well represented.

This research confirms earlier conclusions that PCA of network contributions is a useful technique for understanding the performance of networks constructed by the cascade-correlation learning algorithm (Shultz & Elman, 1994). Because cascade-correlation nets typically possess multiple hidden layers and are fully cross connected, they are difficult to analyze with more standard methods emphasizing activation patterns on the hidden units alone. Examination of their weight patterns is also problematic, particularly in larger networks, because of the highly distributed nature of the net's representations.

Analyzing contributions, in contrast to either hidden unit activations or weights, is a naturally appealing solution. Contributions capture the influence coming into output units both from adjacent hidden units and from distant, cross connected hidden and input units.

The present work also suggests that analyzing unscaled contributions yields more useful results than does the analysis of contributions that are scaled by the output targets. This is particularly true in terms of sensitivity to various input dimensions and to operational distinctions between adding and multiplying. Plots of component scores based on unscaled contributions are typically not as dense as those based on scaled contributions but seem to be more revealing of what information the network is representing. Including target outputs in these analyses is not only unrealistic, but also obscures at least part of what networks represent, such as variation along important input dimensions. A drawback of using unscaled contributions is that contributions from the bias unit are ignored for lack of variation. This may explain why the present analyses fail to isolate arithmetic problems with *equal to* outcomes.

Because PCA of contributions can identify the role of contributions from particular hidden units, it should be useful in predicting the results of lesioning experiments with neural nets. Once the role of a hidden unit has been identified by its association with a particular principal component, then it could be predicted that lesioning this unit would impair whatever function is served by the component. PCA of network contributions obtained from cognitive modeling could also be a useful source of psychological hypotheses.

**Acknowledgments**

This research was supported by the Natural Sciences and Engineering Research Council of Canada.

**References**

Fahlman, S. E., & Lebiere, C. (1990.) The Cascade-Correlation learning architecture. In D. Touretzky (Ed.), *Advances in neural information processing systems 2*, (pp. 524-532). Mountain View, CA: Morgan Kaufmann.

Flury, B. (1988). *Common principal components and related multivariate models*. New York: Wesley.

Lang, K. J., & Witbrock, M. J. (1988). Learning to tell two spirals apart. In D. Touretzky, G. Hinton, & T. Sejnowski (Eds)., *Proceedings of the Connectionist Models Summer School*, (pp. 52-59). Mountain View, CA: Morgan Kaufmann.

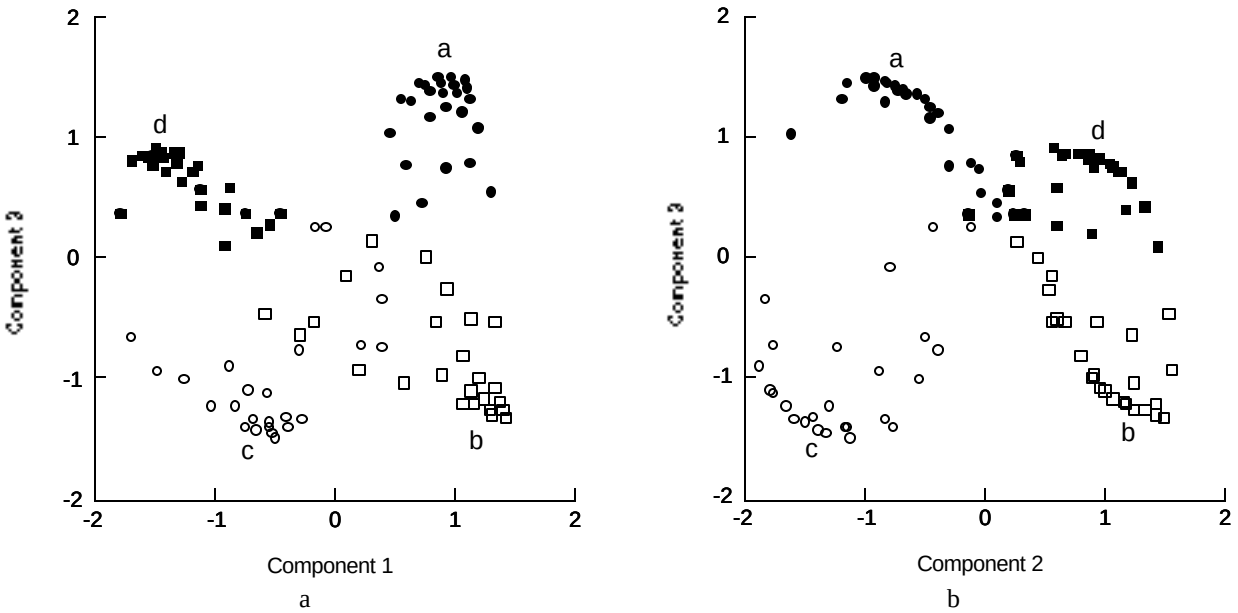
McClelland, J. L. (1989). Parallel distributed processing: Implications for cognition and development. In Morris, R. G. M. (Ed.), *Parallel distributed processing: Implications for psychology and neurobiology*, pp. 8-45. Oxford University Press.

McCloskey, M. (1991). Networks and theories: The place of connectionism in cognitive science. *Psychological Science*, **2**, 387-395.

Sanger, D. (1989). Contribution analysis: A technique for assigning responsibilities to hidden units in connectionist networks. *Connection Science*, **1**, 115-138.

Shultz, T. R., & Elman, J. L. (1994). Analyzing cross connected networks. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in Neural Information Processing Systems 6*. San Francisco, CA: Morgan Kaufmann.

Shultz, T. R., Schmidt, W. C., Buckingham, D., & Mareschal, D. (In press). Modeling cognitive development with a generative connectionist algorithm. In G. Halford & T. Simon (Eds.), *Developing cognitive competence: New approaches to process modeling*. Hillsdale, NJ: Erlbaum.




a b

Figure1. Rotated component scores for a continuous XOR net. 1a. Components 1 and 3. 1b. Components 2 and 3.

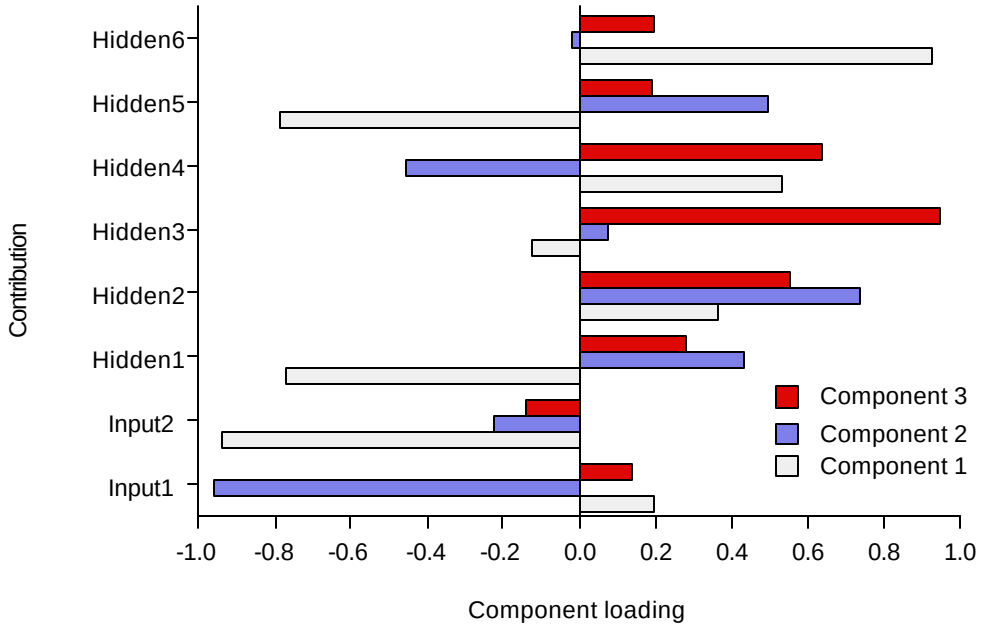



Figure 2. Component loadings for a continuous XOR net.

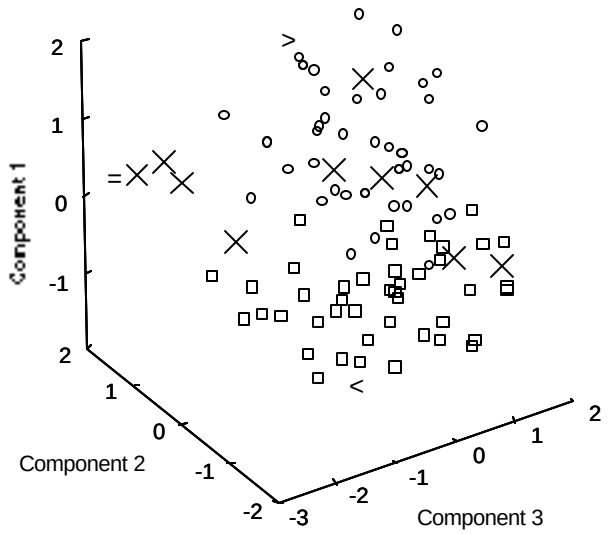



Figure 3. Rotated component scores for an addition net.

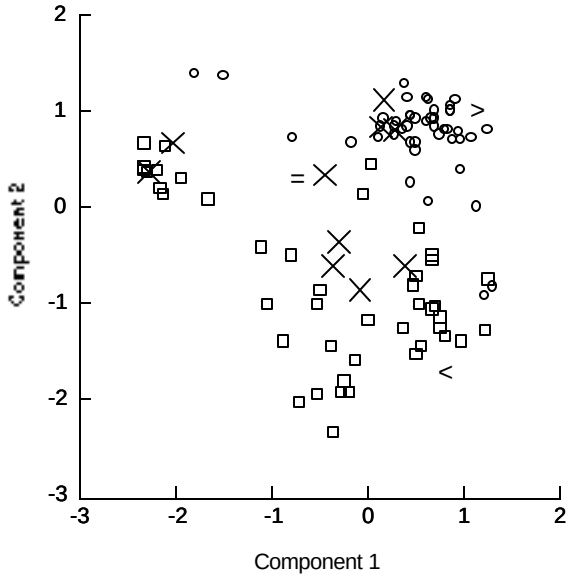



Figure 4. Rotated component scores for a multiplication net.

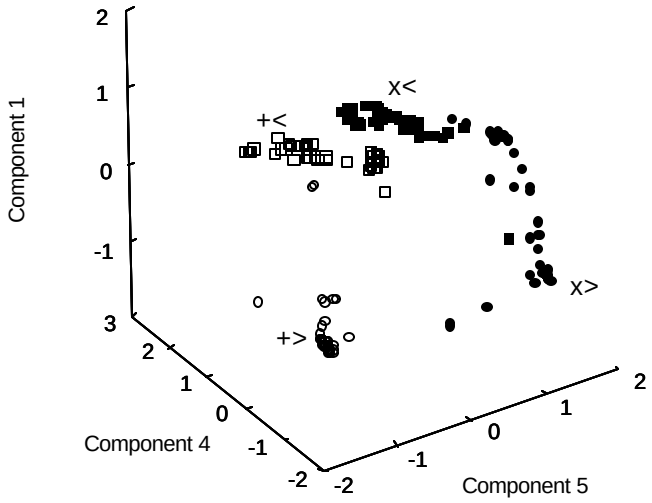



Figure 5. Rotated component scores for a net doing both addition and multiplication.