LETTER Communicated by Robert Jacobs

Discriminant Component Pruning: Regularization and Interpretation of Multilayered Backpropagation Networks

Randal A. Koene
Yoshio Takane
Department of Psychology, McGill University, Montreal, PQ, Canada H3A 1B1

Neural networks are often employed as tools in classification tasks. The use of large networks increases the likelihood of the task's being learned, although it may also lead to increased complexity. Pruning is an effective way of reducing the complexity of large networks. We present discriminant components pruning (DCP), a method of pruning matrices of summed contributions between layers of a neural network. Attempting to interpret the underlying functions learned by the network can be aided by pruning the network. Generalization performance should be maintained at its optimal level following pruning. We demonstrate DCP's effectiveness at maintaining generalization performance, applicability to a wider range of problems, and the usefulness of such pruning for network interpretation. Possible enhancements are discussed for the identification of the optimal reduced rank and inclusion of nonlinear neural activation functions in the pruning algorithm.

1 Introduction

Feedforward neural networks have become commonplace tools for classification. A network containing sufficient neurons will learn a function distinguishing patterns from a well-separable data set. Because the nature of the function is not known a priori, the necessary size and complexity of the trained neural network are not known in advance. Consequently we tend to employ a neural network that can learn a greater variety of functions. We may then encounter the problem of overparameterization, which reduces reliability and generalization performance, as well as complicating interpretation of functions represented by the trained network. A plausible means of reducing the degree of overparameterization is to prune or regularize the complexity of the network.

A variety of approaches to pruning have been proposed: elimination of connections associated with small weights is one of the earliest and fastest methods; early stopping monitors performance on a test set during training; ridge regression penalizes large weights; skeletization (Mozer & Smolensky, 1989) removes neurons with the least effect on the output error; Optimal Brain Damage (Le Cun, Denker, & Solla, 1990) removes weights that least

affect the training error; Optimal Brain Surgeon (Hassibi, Stork, & Wolff, 1992) is an improvement of Optimal Brain Damage. Each method has advantages and disadvantages (Hanson & Pratt, 1989; Reed, 1993) in its approach to minimizing pruning errors, its applicability to different types of problems, or its computational efficiency. Principal Components Pruning (PCP) (Levin, Leen, & Moody, 1994) uses principal component analysis to determine which components to prune and will be used as a benchmark for comparison.

Discriminant components pruning (DCP), the pruning method we present, reduces the rank of matrices of summed contributions between the layers of a trained neural network. We describe DCP and demonstrate its effectiveness by comparing it with PCP in terms of their respective ability to reduce the ranks of weight matrices. Fisher's IRIS data are used as an initial benchmark for comparison, and two empirical data sets with specific complexities verify particular performance issues. The first of the latter two sets contains sparse data in which groups are not easily separable. The second demonstrates DCP's ability to cope with data of varying scales across individual inputs, and hence discriminant components that differ from the principal components of the data set. A brief demonstration of the usefulness of optimal DCP rank reduction to the interpretation of underlying functions represented in trained neural networks follows. The discussion summarizes our results and points out directions for future work.

2 Discriminant Components Pruning

We write the original trained function of a complete layer i of the network

$$\mathbf{Z}_{i+1} = \sigma(\mathbf{Z}_i\mathbf{W}_i) \equiv \sigma(\mathbf{X}_i), \tag{2.1}$$

where rows of the $N \times m_i$ matrix $\mathbf{Z}_i$ are the input vectors $\mathbf{z}_i(k)$ at layer i including a bias term, of N samples $k = 1, \ldots, N$. The $\mathbf{W}_i$ is the $m_{i-1} \times m_i$ matrix of weights that scale inputs to the m_i nodes in layer i, where $i = 1, \ldots, l$. Layer 1 is the first hidden layer, and layer l is the output layer of the network. The matrix $\mathbf{X}_i$ represents the input contributions, and $\sigma(\cdot)$ is the (often sigmoidal) activation function, also called the *squashing function*, that transforms elements of $\mathbf{X}_i = \mathbf{Z}_i\mathbf{W}_i$ into bounded output activations. Outputs at layer i, $\mathbf{Z}_{i+1}$, form the inputs to layer $i+1$, or network outputs when $i = l$. When $i = 1$, $\mathbf{Z}_i = \mathbf{Z}_1$ is the matrix of N input patterns. We can describe the pruned function as

$$\mathbf{Z}^{(r)}_{i+1} = \sigma(\mathbf{Z}_i\mathbf{W}^{(r)}_i), \tag{2.2}$$

where $\mathbf{W}^{(r)}_i$ is the weight matrix with reduced-rank r_i.

The parameter space is pruned by consecutive rank reduction of the layers, obtaining $\mathbf{W}^{(r)}_i$ at ranks $r_i = 1, \ldots, m_i$. To achieve good generalization

performance, we choose the optimal combination of reduced ranks at successive layers yielding the lowest sum of squared errors for the network output of the test set,

$$SS(\mathbf{Y} - \mathbf{Z}_{l+1}^{(r)}), \tag{2.3}$$

where $\mathbf{Y}$ is the matrix of test set target values, and $\mathbf{Z}_{l+1}^{(r)}$ is the matrix of predicted outputs for test samples from the pruned network.

The reduced rank approximation $\mathbf{W}_i^{(r)}$ of the weight matrix $\mathbf{W}_i$ is derived by minimizing the sum of squares,

$$SS(\mathbf{Z}_i\mathbf{W}_i - \mathbf{Z}_i^{(r)}\mathbf{W}_i^{(r)}), \tag{2.4}$$

subject to $\text{rank}(\mathbf{W}_i^{(r)}) = r_i$, where $\mathbf{Z}_i^{(r)}$ is the matrix of outputs from the previous pruned layer (see equation 2.2), with the special case of $\mathbf{Z}_1^{(r)} = \mathbf{Z}_1$ at $i = 1$.

Equation 2.4 can be minimized by standard reduced-rank regression analysis (Anderson, 1951). Let $\mathbf{P}_{Z_i^{(r)}}\mathbf{Z}_i\mathbf{W}_i = \mathbf{U}_i^*\mathbf{D}_i^*\mathbf{V}_i^{*\prime}$ be the singular value decomposition (SVD) of $\mathbf{P}_{Z_i^{(r)}}\mathbf{Z}_i\mathbf{W}_i$, where

$$\mathbf{P}_{Z_i^{(r)}} = \mathbf{Z}_i^{(r)}(\mathbf{Z}_i^{(r)\prime}\mathbf{Z}_i^{(r)})^{-1}\mathbf{Z}_i^{(r)\prime} \tag{2.5}$$

is an orthogonal projector onto the space spanned by column vectors of $\mathbf{Z}_i^{(r)}$. Then the best rank r_i approximation to $\mathbf{Z}_i\mathbf{W}_i$ is given by

$$\mathbf{Z}_i^{(r)}\mathbf{W}_i^{(r)} = \mathbf{U}_i^{*(r)}\mathbf{D}_i^{*(r)}\mathbf{V}_i^{*(r)\prime}. \tag{2.6}$$

If for some reason $\mathbf{W}_i^{(r)}$ is required, it can be obtained by

$$\mathbf{W}_i^{(r)} = (\mathbf{Z}_i^{(r)\prime}\mathbf{Z}_i^{(r)})^{-1}\mathbf{Z}_i^{(r)\prime}\mathbf{U}_i^{*(r)}\mathbf{D}_i^{*(r)}\mathbf{V}_i^{*(r)\prime}. \tag{2.7}$$

A more detailed derivation of equation 2.6 is in the appendix.

The diagonal elements in $\mathbf{D}_i^*$ reflect the importance of corresponding discriminant components (DCs). The best rank r_i approximation $\mathbf{W}_i^{(r)}$ of $\mathbf{W}_i$ is obtained by retaining the first r_i columns of $\mathbf{U}_i$ and $\mathbf{V}_i$, and the first r_i rows and columns of $\mathbf{D}_i^*$ corresponding to the r_i largest singular values. The new weights $\mathbf{W}_i^{(r)}$ serve to implement $\mathbf{X}_i^{(r)} = \mathbf{Z}_i^{(r)}\mathbf{W}_i^{(r)}$.

Due to the requirement that network topology be maintained, a reduced-rank approximation to each layer must be derived separately, which impedes optimal regularization to some degree. Another factor affecting the precision of the approximation lies in the exclusion, in the derivation of $\mathbf{W}_i^{(r)}$, of effects of the nonlinear transformation of propagated contributions. This

component is added to the approximation error. Where generalization performance of the pruned network is required to remain at least as good as that of the original network, the presence or absence of the additional error component could on occasion be significant to the minimum rank that can be achieved. (But see a further discussion in section 5.) Both of the above are factors that DCP shares with all similar methods, however. DCP's main advantage is efficiency in computation time and the number of components necessary to approximate the original network. The optimal fixed-rank approximation to $\mathbf{Z}_i\mathbf{W}_i$ on individual layers for the training samples is ensured through DCP's direct reduction of the matrix of summed contributions using SVD.

3 Effectiveness of Rank Reduction with DCP

DCP's ability to achieve low optimal ranks and its broad applicability is demonstrated by theoretical and empirical comparison with PCP, a comparable technique proposed by Levin, Leen, and Moody (1994).

3.1 Theoretical Advantages over Principal Components Pruning. PCP is a method of rank reduction based on principal component analysis (PCA). As such, it is similar to DCP, and it serves as a useful benchmark for comparison. PCP seeks a rank r_i approximation to the input matrix $\mathbf{Z}_i$ at each layer. This approximation can be found in a manner similar to that employed by DCP, with the SVD of $\mathbf{Z}_i$ denoted as

$$\mathbf{Z}_i = \mathbf{U}_i\mathbf{D}_i\mathbf{V}_i'. \tag{3.1}$$

The reduced-rank weight matrix is given by

$$\mathbf{W}_i^{(r)} = \mathbf{V}_i^{(r)}\mathbf{V}_i^{(r)'}\mathbf{W}_i, \tag{3.2}$$

where r_i principal components (PCs) to be retained in $\mathbf{V}_i^{(r)}$ do not necessarily correspond to the largest singular values. (The specific procedure is described below.) We can now write

$$\mathbf{Z}_i\mathbf{W}_i^{(r)} = \mathbf{U}_i\mathbf{D}_i\mathbf{V}_i'\mathbf{V}_i^{(r)}\mathbf{V}_i^{(r)'}\mathbf{W}_i = \mathbf{Z}_i^{(r)}\mathbf{W}_i, \tag{3.3}$$

for the new contributions at layer *i*, where

$$\mathbf{Z}_i^{(r)} = \mathbf{U}_i^{(r)}\mathbf{D}_i^{(r)}\mathbf{V}_i^{(r)'} \tag{3.4}$$

is a rank r_i approximation to $\mathbf{Z}_i$. The $\mathbf{U}_i^{(r)}$, $\mathbf{D}_i^{(r)}$, and $\mathbf{V}_i^{(r)}$ retain r_i columns of $\mathbf{U}_i$ and $\mathbf{V}_i$, and r_i rows and columns of $\mathbf{D}_i$. Composing a matrix of contributions with the reduced-rank weight matrix $\mathbf{W}_i^{(r)}$ in layer *i* in equation 3.3 is equal to

the matrix of contributions composed of pruned inputs $\mathbf{Z}_i^{(r)}$ and the original weight matrix.

Salient PCs in equation 3.2 may not be relevant DCs, since input parameters with relatively small variance may well be important factors for discrimination (Flury, 1995). PCP uses the following technique to rank-order principal components (PCs) according to their importance for discrimination. Since the total sum of squares in $\mathbf{Z}_i\mathbf{W}_i$ is

$$SS(\mathbf{Z}_i\mathbf{W}_i) = SS(\mathbf{U}_i\mathbf{D}_i\mathbf{V}_i'\mathbf{W}_i) = \sum_{j=1}^{m_i} d_{ij}^2 \tilde{\mathbf{w}}_{ij}'\tilde{\mathbf{w}}_{ij}, \tag{3.5}$$

where d_{ij} is the jth diagonal element of $\mathbf{D}_i$, and $\tilde{\mathbf{w}}_{ij}'$ is the jth row of $\tilde{\mathbf{W}}_i = \mathbf{V}_i'\mathbf{W}_i$, we may use each term in the summation of equation 3.5, namely,

$$d_{ij}^2 \tilde{\mathbf{w}}_{ij}'\tilde{\mathbf{w}}_{ij}, \tag{3.6}$$

to reflect the importance of the jth component. That is, r_i components are chosen according to the size of $d_{ij}^2\tilde{\mathbf{w}}_{ij}'\tilde{\mathbf{w}}_{ij}$.

DCP has advantages over PCP in that it is scale invariant. It also prunes more efficiently, which leads to a lower optimal reduced rank. The fewer number of effective parameters in the pruned network aid identification and interpretation efforts, while reducing instability of weight estimates. Scale invariance cannot be achieved as far as we deal with the input matrix alone, since SVD($\mathbf{Z}$) $\neq$ SVD($\mathbf{Z}\Delta$), where Δ is a diagonal scaling matrix. Scaled inputs are compensated in the neural net by inversely scaled connection weights, $\Delta^{-1}\mathbf{W}$. Thus, the matrix of summed contributions, $\mathbf{ZW}$, whose SVD we obtain in DCP, is invariant over the choice of Δ, as $\mathbf{ZW} = (\mathbf{Z}\Delta)(\Delta^{-1}\mathbf{W})$.

PCP deals with this problem by combining the salience in PCs (d_{ij}^2) with salience in discrimination ($\tilde{\mathbf{w}}_{ij}'\tilde{\mathbf{w}}_{ij}$), as in principal component discriminant analysis (Jolliffe, 1986). Scaling or additive offsets alter the very PCs extracted from $\mathbf{Z}$, however. Although such scaling may be quite common in natural data sets, the situation cannot be adequately dealt with by the individual salience measures. PCP's ability to prune a correspondingly trained network effectively is therefore impaired.

More efficient pruning can be expected as a direct consequence of rank reduction of $\mathbf{ZW}$, in comparison with rank reduction of $\mathbf{Z}$ only. In PCP, the effects of pruning in previous layers are not taken into account when pruning in following layers. Despite the linear simplification, DCP's $\mathbf{P}_{Z_i^{(r)}}\mathbf{Z}_i$ propagation maintains optimality at least relative to PCP.

3.2 Empirical Evaluation. PCP and DCP results were compared for pruning three-layer backpropagation networks (Rumelhart, Hinton, & Williams, 1986) on empirical data sets, the IRIS data set (Fisher, 1936), and two sets obtained or adapted from Toyoda (1996).

Pruning methods strive to reduce rank while approximating the original function as much as possible. A measure of the effect on the linear system at a single approximated layer can be obtained as the sum of squares of $\mathbf{Z}_i\mathbf{W}_i - \mathbf{Z}_i\mathbf{W}_i^{(r)}$. Performance of the combined layers in the neural network must be measured to determine the divergence of the approximated network function from the original, where generalization performance is indicated by performance on test set samples.

The sum of squared errors (SSE) for the pruned network output $SS(\mathbf{Y} - \mathbf{Z}_{l+1}^{(r)})$ does not show monotonic decrease for an increase in the rank of individual layers. The adjusted $\mathbf{W}_1^{(r)}$ affects inputs to the following layer, although differences are usually small. Even small differences can be significant at times, especially when they are mediated by a nonlinear transfer function. The sigmoid function leads to a situation in which relatively small differences on large positive or negative contributions are harmless, since they are bounded by the output-limiting asymptotes of the sigmoid function. Yet the same differences on contributions near zero, where the sigmoid function is steepest, can lead to significant changes in $\mathbf{Z}_2$. In such cases, pruning according to original inputs $\mathbf{Z}_2$ may not be optimal with PCP, demonstrating the importance of propagating these differences through $\mathbf{P}_{Z_i^{(r)}}\mathbf{Z}_i$.

R. A. Fisher's IRIS data set has been used widely as a benchmark for discriminant analysis methods. The data set consists of 150 samples reporting measurement of four characteristics—sepal width, sepal length, petal width, and petal length—of three species of iris flower: *Iris setosa*, *Iris versicolor*, and *Iris verginica*. The four characteristics are represented by inputs $z_{1,1}$ to $z_{1,4}$ of the neural network, where the first index indicates the layer and the second a node in that layer, with the additional bias term $z_{1,5} = 1$. Each iris species is given a corresponding output node $y_1 = z_{3,1}$ to $y_3 = z_{3,3}$. With the split-half method, separate training and test sets were created with 75 samples each and an equal number of samples (25) for each target class. A backpropagation neural network with 5 input units ($z_{1,1}$ to $z_{1,5}$), 5 hidden units (4+ bias), and 3 output units, one for each species, achieved an SSE of 0.34, correctly classifying 100% of the training set and 93.3% of the test set, with an SSE of 7.69.

In our second example, which we call academic aptitude requirement data, interviews were conducted with professors in six different faculties—Arts, Medicine, Engineering, Education, Agriculture, and Science—to determine academic aptitude requirements for students in their particular field or specialty. The frequencies with which particular qualifications were mentioned by professors comprise the data set: math-science ability, interest in people and/or children, interest in the field of study, interest in humanitarianism, interest in fieldwork, discussion ability, ability to work with computers, knowledge in foreign languages, reading ability, and logical thinking. Each is represented by an input node $z_{1,1}$ to $z_{1,10}$ of the neural network, with bias $z_{1,11} = 1$. The faculty to which each professor belonged was used

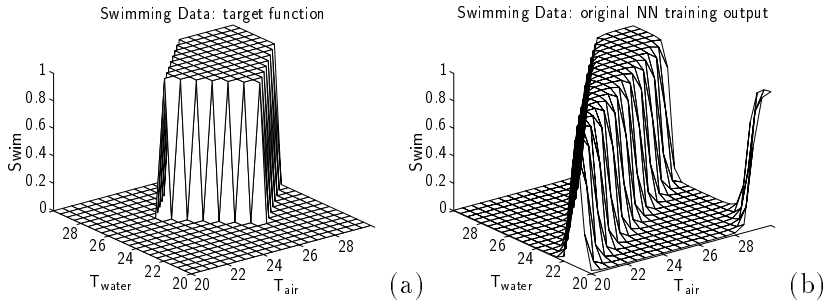

Figure 1: (a) The swimming decision target function, $z_{3,1} = 1$ when $z_{1,1} + z_{1,2} \geq 50$ and $|z_{1,1} - z_{1,2}| \leq 3$, $z_{3,1} = 0$ otherwise. (b) The corresponding trained output function, with training sample responses indicated.

as corresponding classification target, y_1 to y_6. Separate training (120 samples) and test (116 samples) sets were created with the split-half method. A backpropagation neural network with 11 input units ($z_{1,1}$ to $z_{1,11}$), 16 hidden units (15+ bias), and 6 output units (classes) achieved an SSE of 26.3, classifying 81.7% of the training samples correctly. Performance on the test set was 41.4% correct, with an SSE of 118.9.

In the third example, which we call school swimming decision data, there are 4 inputs with bias $z_{1,5}$, $z_{1,1}$ (air temperature) and $z_{1,2}$ (water temperature) from statistics on the decision to allow schoolchildren to swim, and a single target output y, classes: "no" $y = 0$, "yes" $y = 1$. Irrelevant inputs $z_{1,3}$ and $z_{1,4}$ are generated by normal random numbers with a relatively large variance and a mean offset of 50, imposing a clear distinction between PCs and DCs. The training and test sets each contain 24 samples, with 12 from each of the two target classes. A backpropagation neural network, with 5 units ($z_{1,1}$ to $z_{1,5}$) in the input layer, 5 units (4+ bias unit) in the hidden layer, and 1 unit in the output layer, achieved an SSE of nearly 0 on the training data, correctly classifying 100% of the training set and 79.1% of the test set (with an SSE of 4.99).

Figure 1a depicts the target function in terms of the relevant temperature inputs. Figure 1b depicts the function obtained from the trained network, where the surface mesh shows the response for temperature combinations when $z_{1,3} = z_{1,4} = 0$. Numbers indicate network outputs for training samples with target values 1 and 0.

3.2.1 Performance on Iris Data. The PCP rank-reduction procedure produced a combination of reduced ranks deemed optimal at 4×2, recognizable as the peak in Figure 2a. The corresponding ratio of correctly classified test samples was 96.0% with an SSE of 29.1.

Results of DCP rank reduction were restricted by the size of the contribu-

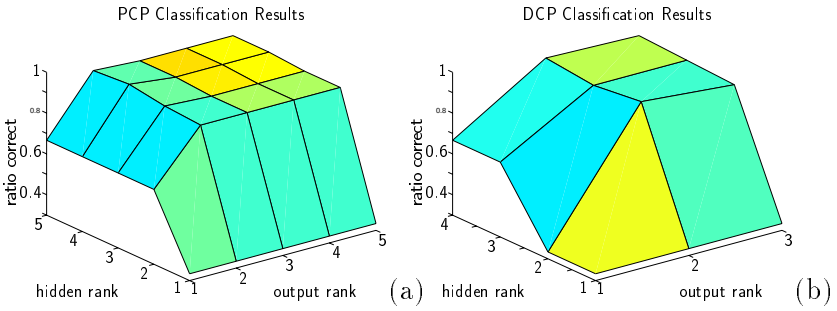

Figure 2: Test set classification ratios for Fisher's IRIS Data at (a) PCP and (b) DCP reduced ranks.

tion matrices in the two layers: 4 on the hidden layer (4 hidden units) and 3 on the output layer (3 output units). Optimal pruning was achieved at rank combination 2×2, with a test set classification ratio of 0.95% and an SSE of 23.8 (note these lowest ranks to which the plateau in Figure 2b extends).

The usefulness of Fisher's IRIS data as a benchmark for discriminant analysis was borne out in the clear distinction between optimal pruning ranks achieved by PCP and DCP, respectively. Although both methods managed to prune the parameter space considerably and a slight improvement of generalization performance in terms of the test set classification ratio was observed in both cases, PCP was unable to reduce the rank of the first layer as rigorously as DCP.

3.2.2 Performance on Academic Aptitude Data. Optimal performance for binary classification on the test set of the second example was determined at PCP reduced-rank combination 11×14 (see Figure 3a), with a ratio of 43.1% correctly classified samples and an SSE of 117.4.

In our second example, the DCP target rank is restricted by the rank of the matrix of summed contributions—hence, 11 (10 inputs + 1 bias unit) on the hidden layer and 6 (6 output classes) on the output layer. The optimal test set classification ratio was 44.0% at rank $r_1 = 5$ or $r_1 = 8$ with rank $r_2 = 4$ in the hidden and output layers, respectively, visible as the two peaks at output rank $r_2 = 4$ in the center of Figure 3b. Combination 8×4 was chosen over 5×4, because the test set SSE was better—110.2 instead of 117.4—as was performance on the training set.

PCP was able to maintain generalization performance of the neural network, but was unable to prune the hidden layer at that level of performance, so that rank $r_1 = 11$ remained unaltered. DCP managed to attain slightly better generalization performance at a much lower rank.

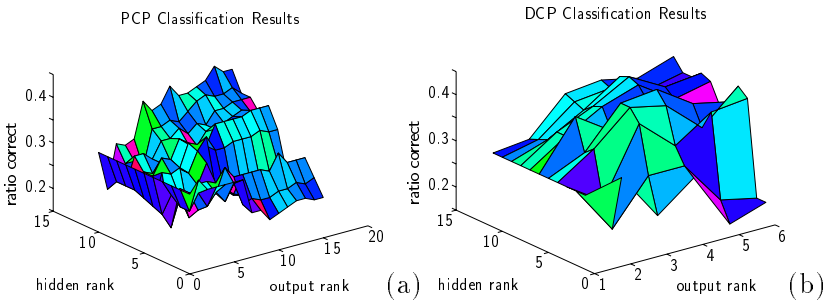

Figure 3: Test set classification ratios for the academic aptitude data at (a) PCP and (b) DCP reduced ranks.

3.2.3 Performance on Swimming Decision Data. The third example was chosen to demonstrate wider applicability of DCP. Not surprisingly, PCP's linear pruning error leaps from 0 to 13,633 even at rank $r_1 = 4$ and remains at approximately that level for all reduced ranks, expressing the detrimental effect of PCP's focus on the largest PCs of $\mathbf{Z}_1$. This translates into an output performance error for which only combinations with full rank on the hidden layer approximate original performance reasonably well. PCP's ability to prune the two meaningless input parameters was impaired. Optimal generalization pruning was determined to be 5×1 at 79.2% and an SEE of 4.82 (note that the training set ratio dropped to 87.5%). The output function becomes a near constant value with chance level (50%) performance below that rank, as shown in Figures 4 and 6a.

The largest DCP ranks for the swimming decision network are restricted by the number of hidden nodes ($m_1 = 4$) and the single output on the second layer, fixing rank $r_2 = 1$. The individual linear pruning error with a maximum SSE of 6702 at $r_1 = 1$ shows no PCP-like step function characteristics. Classification ratios and $SS(\mathbf{Y} - \mathbf{Z}_{l+1}^{(r)})$ errors show optimal performance up to reduced-rank combination 3×1, recognizable as the maximum plateau in Figure 6b, achieving 75.0% correct classifications, with an SSE of 6.14, and perfect training set performance. The resulting output function (see Figure 5) closely resembles the original output function of the trained neural network.

PCP failed to prune the hidden layer and identify the two salient parameters governing this classification task, where PCs and DCs are not the same. DCP correctly identifies the two and the necessary compensation bias for the mean offset on $z_{1,3}$ and $z_{1,4}$, retaining relevant DCs and successfully approximating the implicit function. In these and other empirical applications, DCP was consistently shown to prune to significantly lower ranks than the benchmark PCP method.

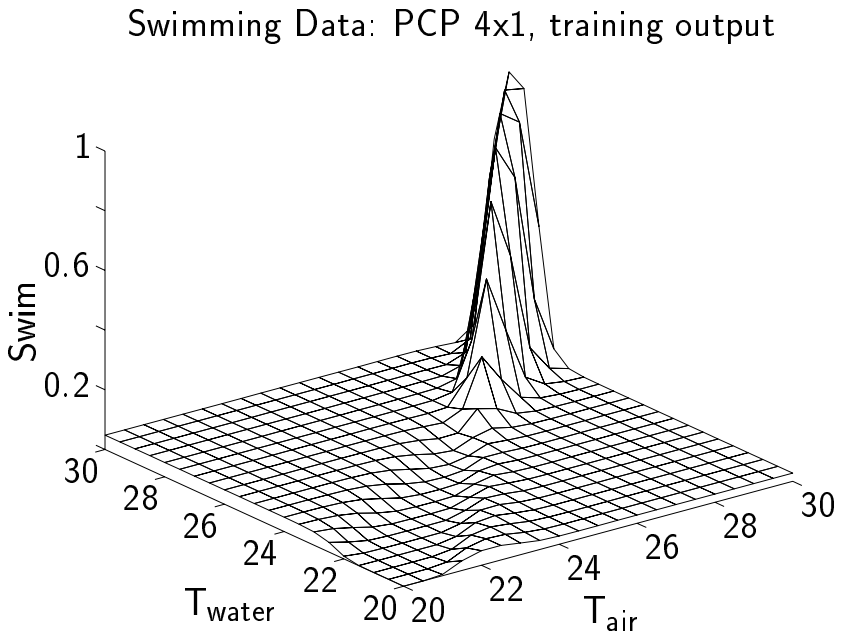

Figure 4: Output function of the swimming decision example after attempting to prune with PCP to ranks 4×1, in the absence of the two irrelevant inputs.

4 Neural Network Interpretation with DCP

We present interpretations of the classification functions of our two representative examples in which the dimensionality was reduced with DCP by pruning the number of parameters involved in the neural computation to optimal combined ranks.

4.1 Interpretation of Academic Aptitude Network. Our optimal DCP solution maintains generalization performance and retains a network of ranks 8×4 on the hidden and output layers, respectively. There is no known target function for this example.

The SVD of hidden-layer and output-layer matrices of contributions, $\mathbf{Z}_1\mathbf{W}_1 = \mathbf{U}_1^*\mathbf{D}_1^*\mathbf{V}_1^{*\prime}$ and $\mathbf{P}_{Z_1^{(r)}}\mathbf{Z}_2\mathbf{W}_2 = \mathbf{U}_2^*\mathbf{D}_2^*\mathbf{V}_2^{*\prime}$, are used to determine the relative importance of components and parameters. The first 11 and 6 diagonal elements of $\mathbf{D}_1^*$ and $\mathbf{D}_2^*$, respectively, are nonzero. Proportions of sums of squares explained by these are: 33.1%, 19.3%, 13.9%, 11.8%, 8.4%, 5.5%, 3.7%, 2.0%, 1.8%, 0.3%, and 0.2%, for the hidden layer; and 58.8%, 14.9%, 11.8%, 7. 6%, 4.5%, and 2.5% for the output layer. The 8×4 components re-

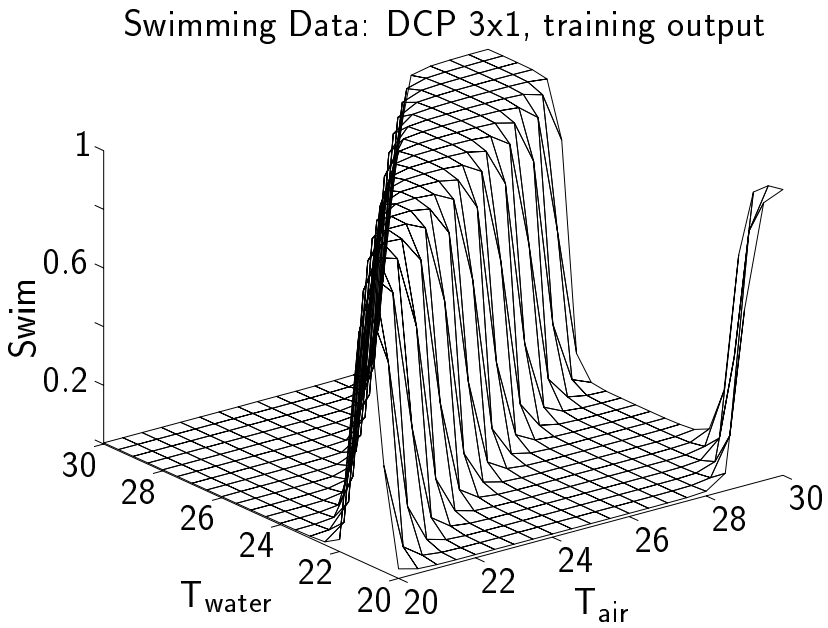

Figure 5: The output function of the swimming decision example after DCP pruning to ranks 3×1, in the absence of the two irrelevant inputs.

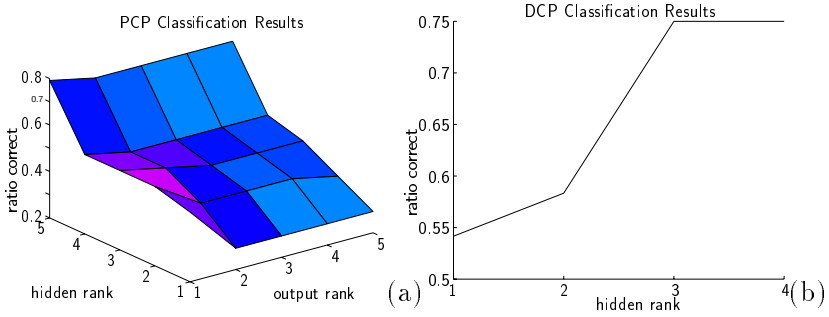

Figure 6: Test set correct classification ratios for the swimming decision data at combinations of (a) PCP and (b) DCP reduced ranks.

tained represent 97.7% and 93.0% of the original component contributions, respectively.

To understand the meaning of the retained components at the hidden and output layers, $\mathbf{U}_1$ and $\mathbf{U}_2$ are correlated with normalized input ($\mathbf{Z}_1$) and targets ($\mathbf{Y}$), respectively. The correlation matrices are subsequently rotated

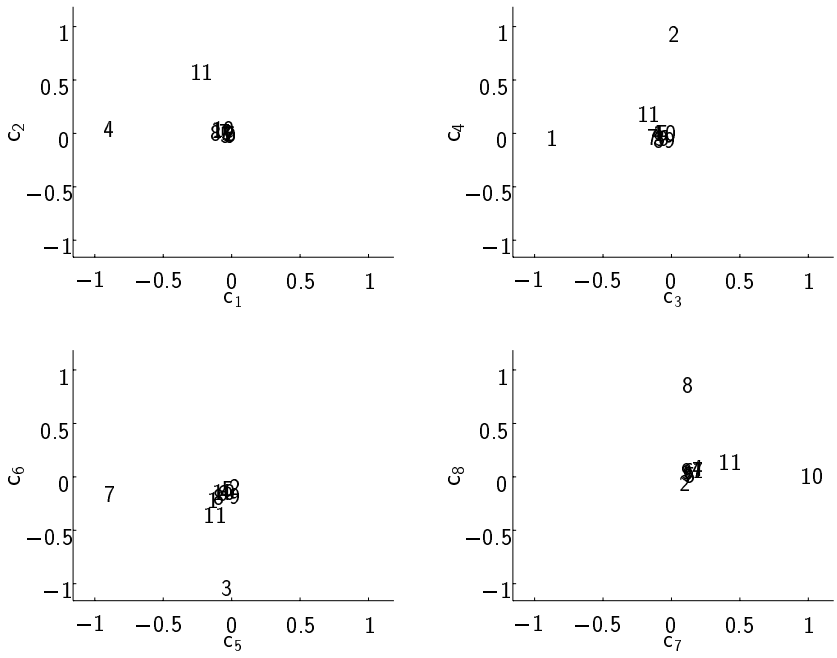

Figure 7: Correlations of aptitude requirements with rotated components in the hidden layer.

by a varimax (Mulaik, 1972) simple structure rotation (see Figure 7).

On the hidden layer, each of the eight rotated components is closely related to one input variable. We note significant correlations with these aptitude requirements used in the network for discrimination: math-science ability ($z_{1,1}$), interest in people and/or children ($z_{1,2}$), interest in field of study ($z_{1,3}$), interest in humanitarianism ($z_{1,4}$), ability to work with computers ($z_{1,7}$), knowledge in foreign languages ($z_{1,8}$), and logical thinking ($z_{1,10}$), as well as the bias input ($z_{1,11}$). Input variables not important to the discrimination task are interest in fieldwork ($z_{1,5}$), discussion ability ($z_{1,6}$), and reading ability ($z_{1,9}$). These abilities are all fairly basic to any fields of study and perhaps not recognized as particularly important in any specific fields.

Four target classes are highly correlated with the four remaining components of the rotated matrix in Figure 8. They are the faculties discriminated by the DCP-reduced, trained network: Arts (y_1), Medicine (y_2), Engineering (y_3), and Education (y_4). Aptitude requirements for Agriculture (y_5) and Science (y_6) are not discriminated by the network, which is consistent with the classification results.

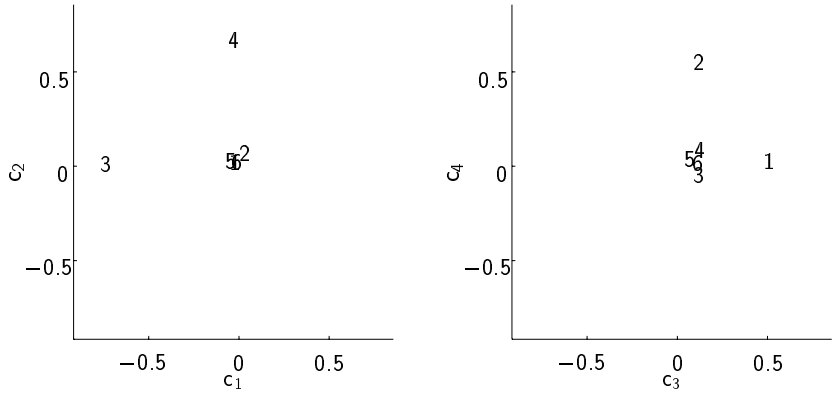

Figure 8: Correlations of target faculties with rotated components in the output layer.

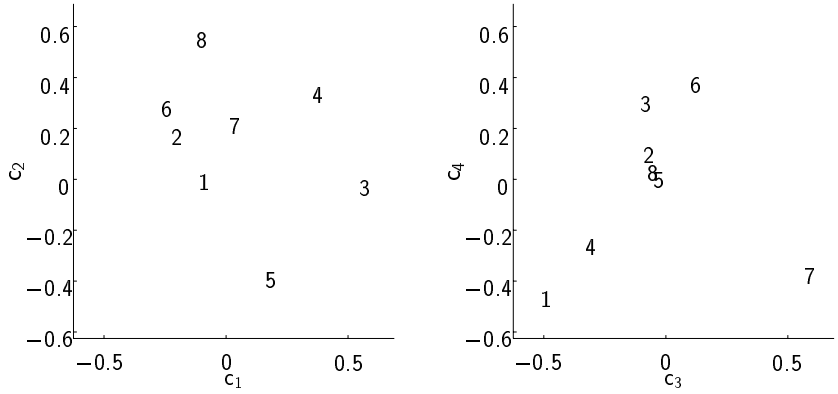

Figure 9: Correlations of output layer components with the eight components of the pruned hidden layer.

Correlations of components in the hidden layer with components in the output layer are shown in Figure 9. Only component number 2 in the hidden layer is not strongly correlated with any components in the output layer, which is explained by the fact that this component was correlated with bias input above.

Combining the correlations found for the inputs (see Figure 7) and outputs (see Figure 8) with those for the components in the two layers gives us the important qualifications for the four discriminated fields of study: Arts values logical thinking, interest in the field of study, and interest in human-

itarianism; Medicine values logical thinking, interest in people and/or children, and humanitarianism; Engineering values interest in people and/or children and math-science ability; Education values knowledge in foreign languages, ability to work with computers, and interest in people and/or children.

The distributed nature of the trained neural network complicates rule-based (formalist) interpretations of its inner workings. A number of hidden units contribute to each output unit to varying degrees, so that a distribution of (binary) component tasks cannot easily be obtained. DCP scales down the number of PCs requiring attention during interpretation. We were able to focus on significant discriminant components and the input and output variables they refer to.

4.2 Interpretation of Swimming Decision Network. The activation functions of the individual nodes in the hidden layer after applying DCP to the trained network are depicted in Figure 10. The surfaces depict the network hidden unit responses to the first two inputs, which are the only relevant variables. Target labels 0 and 1 in the contour plot indicate the locations of test set patterns, where the response does not match in a few cases due to the influence of $z_{1,3}$ and $z_{1,4}$. DCP retains only components of the two salient parameters and bias, lowering the dimensionality to allow interpretation of the hidden layer. We can compare the known target function for this example with the components of the interpreted function.

The output function does not show the abrupt cutoff at low temperatures seen in the target function. This is a result of not having our training samples in the region $z_{1,1} + z_{1,2} < 50$, but close to $z_{1,1} + z_{1,2} = 50$, a good example of approximations resulting from training under natural circumstances. We did not find evidence of the first rule "$z_{3,1} = 1$ when $z_{1,1} + z_{1,2} \geq 50$" among hidden unit functions. The output function appears to move from $z_{1,1} = z_{1,2}$ at low temperatures to $|z_{1,1} - z_{1,2}| \leq 3$ at high temperatures, corresponding to the second component of our target function. The adjusted weight matrix for connections to the output layer after optimal DCP is $\mathbf{W}_2^{(r)} = [9.78, 8.43, 2.39, -8.31, -6.95]$. A combination of the functions performed by the two hidden units with $w_{2,1} = 9.78$ and $w_{2,2} = 8.43$ suffices to generate the output function in Figure 5. The response of hidden unit 4 and the bias combine to form a constant offset of -15.26 on the output layer. The weight $w_{2,2}$ is too small to affect the output. The steepest gradient of the decision boundary of the first hidden unit (see the top of Figure 10) goes from air and water temperatures of 20 and 20.6 degrees (suggesting $z_{1,1} \leq z_{1,2}$ at low temperatures) to 30 and 27.3 degrees (suggesting $(z_{1,1} - z_{1,2}) \leq 3$ at high temperatures), respectively. Similarly, the second hidden unit (on the lower half of Figure 10) approximates $z_{1,1} \geq z_{1,2}$ at low and $(z_{1,2} - z_{1,1}) \leq 3$ at high temperatures.

A detailed expression of the binary equivalent of the output and the two

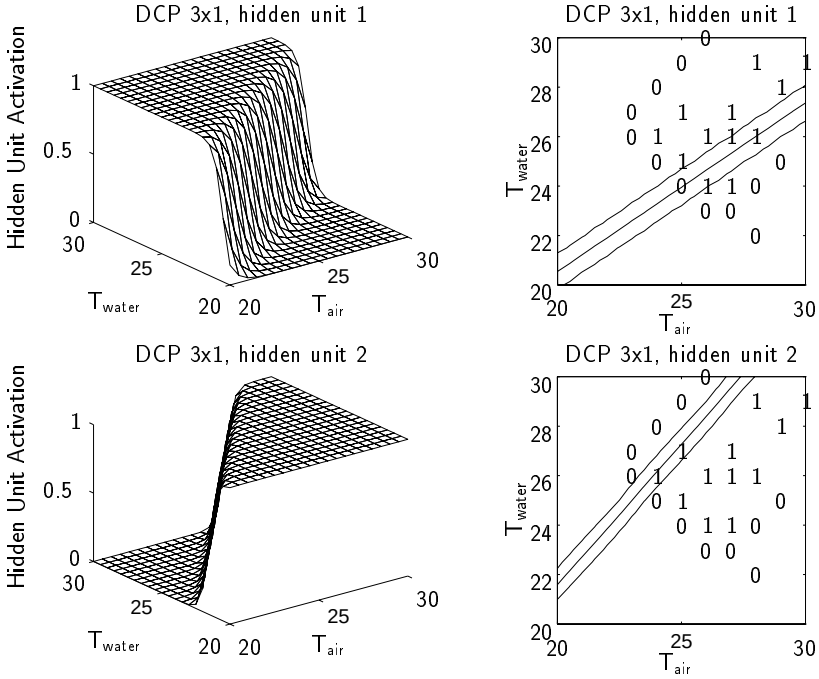

Figure 10: Activations and contour plots of the first two hidden units of the DCP pruned network as a function of the two relevant temperature parameters. Training samples are indicated in the contour plots.

decisive hidden unit responses can be derived from contour plots of hidden and output unit responses in Figures 11a and 11b, in terms of line equations in the parameter space of inputs $z_{1,1}$ and $z_{1,2}$. The lowest, middle, and upper diagonal lines in the contour plots indicate decision boundaries at 0.1, 0.5, and 0.9 response values, respectively.

The equations for the decision boundary at hidden unit responses of 0.5 are approximately,

$$z_{1,2} = 0.67 z_{1,1} + 7.2, \tag{4.1}$$

and,

$$z_{1,2} = 1.15 z_{1,1} - 1.4, \tag{4.2}$$

where $z_{1,1}$ and $z_{1,2}$ are the air and water temperature inputs, respectively. These correspond well with the equations for the decision boundaries of the network output in Figure 11. The two hidden units do indeed contribute the significant component functions of the network.

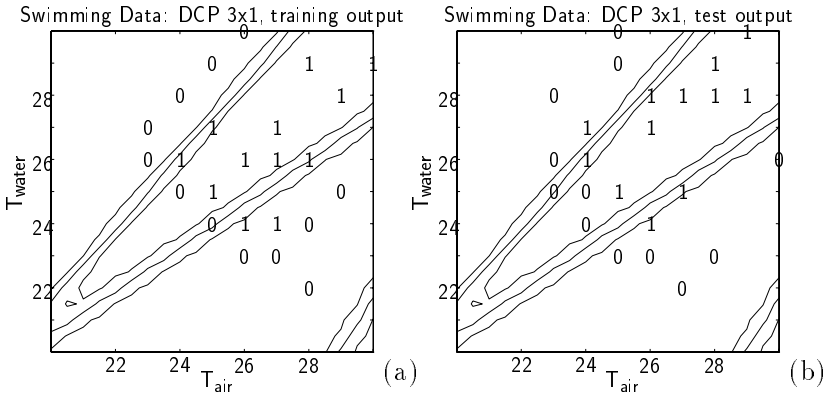

Figure 11: Contour plots of output unit responses to air and water temperature, including (a) training and (b) test patterns with their target values. Outer, central, and inner lines represent 0.1, 0.5, and 0.9 values of the decision boundary, respectively.

We can express the output function in the form of a rule,

$$\text{if } z^b_{2,1} \text{ and } z^b_{2,2} \text{ then } z^b_{3,1} = \textit{yes}(\textit{swim}),$$

where $z^b_{2,1}$ and $z^b_{2,2}$ are low (< 0.5) or high (> 0.5) binary hidden unit outputs, and $z^b_{3,1}$ is the binary network output. Similarly, equations 4.1 and 4.2 give us the rules,

$$\text{if } z_{1,2} \geq 0.67 z_{1,1} + 7.2 \text{ then } z^b_{2,1} = \textit{true}(\textit{high}),$$

and,

$$\text{if } z_{1,2} \leq 1.15 z_{1,1} - 1.4 \text{ then } z^b_{2,2} = \textit{true}(\textit{high}).$$

Consequently, the rule governing the complete function of the network is,

$$\begin{aligned} &\text{if } (z_{1,2} \geq 0.67 z_{1,1} + 7.2) \text{ and } (z_{1,2} \leq 1.15 z_{1,1} - 1.4) \\ &\qquad \text{then } z^b_{3,1} = \textit{yes}(\textit{swim}). \end{aligned} \tag{4.3}$$

The distributed and connectionist implementation of the function of this network is interpreted by equation 4.3 in terms of formalized functions of the significant components remaining after regularization with DCP.

Two conceptually distinct approaches dealing with the relationship between formalist or rule-based knowledge and distributed knowledge in neural networks are notable in this context. On the one hand, there are

those that begin with a formalist representation, attempting to define a sensible topology and initial weights for a neural network on the basis of prior rule-based knowledge. One such technique is KBANN (Towell, Shavlik, & Noordewier, 1990; Maclin & Shavlik, 1991). On the other hand, there are algorithms for the automatic extraction of rules from trained feedforward neural networks, such as KT (LiMin, 1994; Koene, 1995). KBANN has been shown to improve on regular artificial neural network (ANN) methods for complex tasks. KT, in turn, has been shown to generate rule-based representations that can outperform the original neural network for specific tasks. It may be useful to combine methods when seeking to preserve rule structure. In this way, KBANN can provide the initial rules, DCP prunes the trained neural network to fundamental components, and an extraction technique such as KT returns the resulting set of rules implicit in the function learned by the network. DCP simplifies the extraction of representative rules by identifying significant components and reducing the number of parameters involved in the learned function. DCP performance is greatest when there is complete freedom in the design of resulting $\mathbf{W}_i^{(r)}$. If a requirement is specified that rules initialized by KBANN must be preserved, the degree of pruning achievable by DCP may be affected, similar to the manner in which it is constrained by the requirement that a layered network topology be maintained.

5 Discussion

We have shown that the error resulting from the use of DCP for rank reduction is consistently lower than that of PCP at the same rank. This is helpful for interpretation efforts. DCP recognizes the components relevant for discrimination, achieving scale invariance and handling offsets in $\mathbf{Z}_i$. The propagation of changes in $\mathbf{Z}_i$ due to $\mathbf{W}_{i-1}^{(r)}$ through $\mathbf{P}_{Z_i^{(r)}}\mathbf{Z}_i$ allows for compensation of potentially cumulative individual divergences.

Generally, computational efficiency of DCP can be achieved compared to PCP, as a result of $\text{rank}(\mathbf{Z}_i\mathbf{W}_i) \leq \text{rank}(\mathbf{Z}_i)$, and because DCP finds discriminant components in a single phase, whereas PCP requires two phases (finding PCs and determining their order of significance) and time-consuming verifications of results when a particular PC is pruned.

Classification performance of networks regularized with DCP and PCP at their respective optimal reduced-rank combinations is maintained. At equal rank combinations, performance after DCP is significantly better than after PCP. Pruning by $\text{SVD}(\mathbf{Z}_i\mathbf{W}_i)$ clearly gives the best approximation of $\mathbf{Z}_i\mathbf{W}_i$. We place emphasis on the lower rank that can be achieved in view of its usefulness for the interpretation of distributed functions by minimizing neural network complexity.

Among the less satisfactory elements, the effect of nonlinear squashing functions can be dealt with by generalizing the criterion for the sum of

squared errors $SS(\mathbf{Z}_i\mathbf{W}_i - \mathbf{Z}_i^{(r)}\mathbf{W}_i^{(r)}) = tr(\mathbf{E}_i)$ for linear PCA to include a metric matrix $\mathbf{M}_{ij}$ (Jolliffe, 1986, p. 224),

$$\sum_{j=1}^{m_i}[\mathbf{Z}_i\mathbf{w}_{ij} - \mathbf{Z}_i^{(r)}\mathbf{w}_{ij}^{(r)}]'\mathbf{M}_{ij}[\mathbf{Z}_i\mathbf{w}_{ij} - \mathbf{Z}_i^{(r)}\mathbf{w}_{ij}^{(r)}]. \tag{5.1}$$

The sigmoid function $\sigma(\cdot)$ restricts its outputs to a given range. The particular metric matrix to be used is determined by the differential $\partial\sigma(\mathbf{x}_{ij})/\partial\mathbf{x}_{ij}$ of the sigmoidal activation function at neuron j in layer i,

$$\mathbf{M}_{ij} = \text{diag}(\sigma(\mathbf{Z}_i\mathbf{w}_{ij})(1 - \sigma(\mathbf{Z}_i\mathbf{w}_{ij}))). \tag{5.2}$$

The desire to account for nonlinear activation functions mainly addresses the possibility of even greater rank reduction. DCP does not obtain a linear approximation of the nonlinear function represented by the network, but rather of the summed contributions, which are subsequently nonlinearly transformed. Since this method accepts only solutions that lead to equal or better generalization performance, the nonlinear transformation of pruned summed contributions does not impede the performance of the network. Taking nonlinear propagation into account in future implementations may allow for even more rigorous pruning of combined layers.

An important implementational issue is the desire to determine the optimal rank of layers without having to compute all possible combinations of reduced ranks. The nonmonotonic nature of the combined error of concatenated layers makes it impossible to determine the optimal rank separately in each layer using the linear $\mathbf{Z}_i\mathbf{W}_i$ matrix. Future work is aimed at investigating the possibility of a maximum likelihood approach that enables the use of the Akaike information criterion (AIC) (Kurita, 1989), for efficiency at pruning individual layers. However, the difficulty of pruning on a layer-by-layer basis remains. An iterative technique for finding the optimal rank combination may prove to be the most rewarding, since an analytical solution examining all layers simultaneously remains an unlikely prospect due to the structure imposed on pruned network matrices by topological restrictions. In the absence of these restrictions, pruning of a neural network in a single DCP step is conceivable. An interesting development for problem domains where prior rule-based knowledge is available, or where a formalist representation of the function inherent in a trained neural network is desirable, might be the sequential application or the integration of KBANN, DCP, KT, and similar methods.

In summary, application of DCP decreases variance and subsequently maintains reliability and generalization performance at the smallest possible rank, while only the least significant components with regard to the discriminant behavior of the neural network are pruned. Propagating the effect of pruning at previous layers and adjusting the pruned matrix of contributions accordingly further improves the approximation. DCP achieves

greater pruning precision to a lower optimal reduced rank, resulting in a greater simplification of the network function in terms of the number of parameters to be identified during analysis and interpretation.

Appendix: Solution of the Reduced-Rank Regression Problem

Define $\mathbf{P}_{Z_i^{(r)}} = \mathbf{Z}_i^{(r)}(\mathbf{Z}_i^{(r)'}\mathbf{Z}_i^{(r)})^{-1}\mathbf{Z}_i^{(r)'}$. We then have the following identity (Takane & Shibayama, 1991):

$$SS(\mathbf{Z}_i\mathbf{W}_i - \mathbf{Z}_i^{(r)}\mathbf{W}_i^{(r)}) = SS(\mathbf{Z}_i\mathbf{W}_i - \mathbf{P}_{Z_i^{(r)}}\mathbf{Z}_i\mathbf{W}_i) + SS(\mathbf{P}_{Z_i^{(r)}}\mathbf{Z}_i\mathbf{W}_i - \mathbf{Z}_i^{(r)}\mathbf{W}_i^{(r)}). \quad \text{(A.1)}$$

The value of the first term on the right-hand side of equation A.1 is independent of $\mathbf{W}_i^{(r)}$. Therefore, the criterion in equation 2.4 can be minimized by minimizing the second term, which can be done by SVD of $\mathbf{P}_{Z_i^{(r)}}\mathbf{Z}_i\mathbf{W}_i$. This means that to obtain $\mathbf{Z}_i^{(r)}\mathbf{W}_i^{(r)}$ that minimizes equation 2.4, we first obtain the unconstrained least-squares estimate $\mathbf{P}_{Z_i^{(r)}}\mathbf{Z}_i\mathbf{W}_i$ (without the rank restriction) of $\mathbf{Z}_i^{(r)}\mathbf{W}_i^{(r)}$, and then obtain the reduced-rank approximation of this unconstrained estimate, given by equation 2.6. Note that $\mathbf{P}_{Z_i^{(r)}}\mathbf{Z}_i = \mathbf{Z}_i$ when $\mathbf{Z}_i^{(r)} = \mathbf{Z}_i$, as in the first hidden layer.

References

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

Fisher, R. (1936). The use of multiple measurements in axonomic problems. *Annals of Eugenics, 7*, 179–188.

Flury, B. (1995). Developments in principal component analysis. In W. J. Krzanowski (Ed.), *Recent advances in descriptive multivariate analysis*, pp. 14–33. Oxford: Oxford Science Publications.

Hanson, S., & Pratt, L. (1989). Comparing biases for minimal network construction with back-propagation. In D. Touretzky (Ed.), *Advances in neural information processing, 1* (pp. 177–185). San Mateo, CA: Morgan Kauffman.

Hassibi, B., Stork, D., & Wolff, G. (1992) *Optimal brain surgeon and general network pruning* (Tech. Rep. No. 9235). Menlo Park, CA: RICOH California Research Center.

Jolliffe, I. (1986). *Principal component analysis*. New York: Springer-Verlag.

Koene, R. *Extracting knowledge in terms of rules from trained neural networks*. Unpublished master's thesis, Department of Electrical Engineering, Delft University of Technology, Delft, Netherlands.

Kurita, T. (1989). A method to determine the number of hidden units of three layered neural networks by information criteria. *Transactions of the Insti-*

tute of Electronics, Information and Communication Engineers, J73-D-II, No. 11 (pp. 1872–1878). (in Japanese)

Le Cun, Y., Denker, J., & Solla, S. (1990). Optimal Brain Damage. In D. Touretzky (Ed.), *Advances in neural information processing systems, 2* (pp. 598–605). San Mateo, CA: Morgan Kauffman.

Levin, A., Leen, T., & Moody, J. (1994). Fast pruning using principal components. In J. D. Cowan, G. Tessauro, & J. Alspector (Eds.), *Advances in neural information processing systems, 6* (pp. 35–42). San Mateo, CA: Morgan Kauffman.

LiMin, F. (1994). Rule generation from neural networks. *IEEE Transactions on Systems, Man and Cybernetics, 24*, 1114–1124.

Maclin, R., & Shavlik, J. (1991). Refining domain theories expressed as finite-state automata. *Machine Learning: Proceedings of the Eighth International Workshop* (pp. 524–528).

Mozer, M., & Smolensky, P. (1989). Skeletonization: A technique for trimming the fat from a network via relevance assessment. In D. Touretzky (Ed.), *Advances in neural information processing systems, 1* (pp. 107–115). San Mateo, CA: Morgan Kauffman.

Mulaik, S. A. (1972). *The foundations of factor analysis*. New York: McGraw-Hill.

Reed, R. (1993). Pruning algorithms—a survey. *IEEE Transactions on Neural Networks, 4–5*, 740–747.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition, 1* (pp. 318–362). Cambridge, MA: MIT Press.

Takane, Y., & Shibayama, T. (1991). Principal component analysis with external information on both subjects and variables. *Psychometrika, 56*, 97–120.

Towell, G., Shavlik, J., & Noordewier, M. (1990). Refinement of approximate domain theories by knowledge-based neural networks. *AAAI90*, 861–866.

Toyoda, H. (1996). *Nonlinear multivariate analysis by neural network models*. Tokyo: Asokura Shoten. (in Japanese)

Received November 5, 1997; accepted July 20, 1998.