PERGAMON

Neural Networks 12 (1999) 1391–1397

Neural Networks

www.elsevier.com/locate/neunet

Contributed article

A neural implementation of canonical correlation analysis

P.L. Lai*, C. Fyfe

Department of Computing and Information Systems, Applied Computational Intelligence Research Unit, The University of Paisley, Paisley, Scotland, UK

Received 18 May 1998; accepted 6 April 1999

Abstract

We derive a new method of performing Canonical Correlation Analysis with Artificial Neural Networks. We demonstrate the network's capabilities on artificial data and then compare its effectiveness with that of a standard statistical method on real data. We demonstrate the capabilities of the network in two situations where standard statistical techniques are not effective: where we have correlations stretching over three data sets and where the maximum nonlinear correlation is greater than any linear correlation. The network is also applied to Becker's (*Network: Computation in Neural Systems*, 1996, 7:7–31) random dot stereogram data and shown to be extremely effective at detecting shift information. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Canonical correlation analysis; Nonlinear correlations; Random dot stereograms

1. Introduction

Artificial Neural Networks (ANNs) are well known for their capacity as implementations of powerful statistical transformations. Some of the first demonstrations of this power came from the family of networks (e.g. Oja, 1982, 1990; Sanger, 1990) which extract the Principal Components of the input data. These give the best (in the sense of least mean square error) linear compression of a data set. Recently nonlinear extensions of PCA networks have been shown to be capable of more sophisticated statistical techniques such as Exploratory Projection Pursuit (Fyfe & Baddeley, 1995) and Factor Analysis (Charles & Fyfe, 1998).

In this paper, we review and extend a recently proposed neural network (Lai & Fyfe, 1998) implementation of Canonical Correlation Analysis (CCA). Canonical Correlation Analysis (Mardia, Kent & Bibby, 1979) is used when we have two data sets which we believe have some underlying correlation. Consider two sets of input data, from which we draw iid samples to form a pair of input vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$. Then in classical CCA, we attempt to find the linear combination of the variables that gives us maximum correlation between the combinations. Let

$$y_1 = \mathbf{w}_1\mathbf{x}_1 = \sum_j w_{1j}x_{1j}, \tag{1}$$

$$y_2 = \mathbf{w}_2\mathbf{x}_2 = \sum_j w_{2j}x_{2j}. \tag{2}$$

Then we wish to find those values of $\mathbf{w}_1$ and $\mathbf{w}_2$ that maximise the correlation between y_1 and y_2. Whereas Principal Components Analysis and Factor Analysis deals with the interrelationships within a set of variables, CCA deals with the relationships between two sets of variables. If the relation between y_1 and y_2 is believed to be causal, we may view the process as one of finding the best predictor of the set $\mathbf{x}_2$ by the set $\mathbf{x}_1$, and similarly of finding the most predictable criterion in the set $\mathbf{x}_2$ from the $\mathbf{x}_1$ data set. Thus, we later review a data set in which a set of exam results are split into those achieved by students when they had access to their books and those marks obtained when the students were denied their books during the exam. We might wish to use a student's open book exams to predict how well he/she might do in the closed book exams.

One way to view canonical correlation analysis is as an extension of multiple regression (see Mardia et al., 1979, p. 281). Recall that in multiple regression analysis the variables are partitioned into an $\mathbf{x}_1$-set containing q variables and a $\mathbf{x}_2$-set containing $p = 1$ variable. The regression solution involves finding the linear combination of $\mathbf{x}_1$ that is most highly correlated with $\mathbf{x}_2$.

Let $\mathbf{x}_1$ have mean μ_1 and $\mathbf{x}_2$ have mean μ_2. Then the

* Corresponding author. Tel.: + 44-141-848-3328; fax: + 44-141-848-3542.

E-mail addresses: lai--ci0@paisley.ac.uk, Peiling.lai@paisley.ac.uk (P.L. Lai), fyfe0ci@paisley.ac.uk (C. Fyfe)

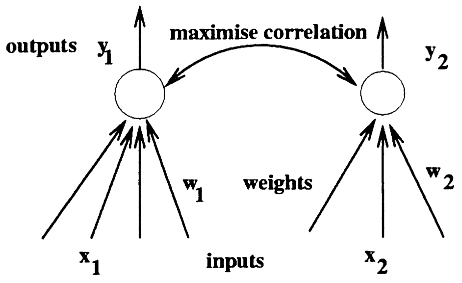


Fig. 1. The CCA network. By adjusting weights, w_1 and w_2, we maximise correlation between y_1 and y_2.

standard statistical method (see Mardia et al., 1979) lies in defining

$$\Sigma_{11} = E\{(\mathbf{x}_1 - \mu_1)(\mathbf{x}_1 - \mu_1)^{\mathrm{T}}\}, \tag{3}$$

$$\Sigma_{22} = E\{(\mathbf{x}_2 - \mu_2)(\mathbf{x}_2 - \mu_2)^{\mathrm{T}}\}, \tag{4}$$

$$\Sigma_{12} = E\{(\mathbf{x}_1 - \mu_1)(\mathbf{x}_2 - \mu_2)^{\mathrm{T}}\} \tag{5}$$

and

$$K = \Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1/2} \tag{6}$$

where T denotes the transpose of a vector. We then perform a Singular Value Decomposition of K to get

$$K = (\alpha_1, \alpha_2, \ldots, \alpha_k)D(\beta_1, \beta_2, \ldots, \beta_k)^{\mathrm{T}} \tag{7}$$

where α_i and β_i are the standardised eigenvectors of KK^{T} and $K^{\mathrm{T}}K$, respectively, and D is the diagonal matrix of eigenvalues.

Then the first canonical correlation vectors (those which give greatest correlation) are given by

$$\mathbf{w}_1 = \Sigma_{11}^{-1/2}\alpha_1, \tag{8}$$

$$\mathbf{w}_2 = \Sigma_{22}^{-1/2}\beta_1 \tag{9}$$

with subsequent canonical correlation vectors defined in terms of the subsequent eigenvectors, α_i and β_i.

In this paper, we present a neural implementation of CCA that adaptively learns the optimal weights to maximise correlations between the data sets.

2. The canonical correlation network

The input data comprises two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. Activation is fed forward from each input to the corresponding output through the respective weights, $\mathbf{w}_1$ and $\mathbf{w}_2$ (see Fig. 1 and Eqs. (1) and (2) to give outputs y_1 and y_2).

We wish to maximise the correlation $E(y_1y_2)$ where $E(\,)$ denotes the expectation which will be taken over the joint distribution of $\mathbf{x}_1$ and $\mathbf{x}_2$. We may regard this problem as that of maximising the function $g_1(\mathbf{w}_1 \mid \mathbf{w}_2) = E(y_1y_2)$ which is defined to be a function of the weights, $\mathbf{w}_1$ given the other set of parameters, $\mathbf{w}_2$. This is an unconstrained maximisation problem that has no finite solution and so we must constrain the maximisation. Typically in CCA, we add the constraint that $E(y_1^2 = 1)$ and similarly with y_2 when we maximise $g_2(\mathbf{w}_2|\mathbf{w}_1)$. Using the method of Lagrange multipliers, this yields the constrained optimisation functions,

$$J_1 = E(y_1y_2) + \tfrac{1}{2}\lambda_1(1 - y_1^2)$$

and

$$J_2 = E(y_1y_2) + \tfrac{1}{2}\lambda_2(1 - y_2^2).$$

We may equivalently use

$$J = E(y_1y_2) + \tfrac{1}{2}\lambda_1(1 - y_1^2) + \tfrac{1}{2}\lambda_2(1 - y_2^2)$$

but it will be more convenient in the following sections to regard these as separate criteria which can be optimised independently by implicitly assuming that $\mathbf{w}_1$ is constant when we are changing $\mathbf{w}_2$ and vice versa. We wish to find the optimal solution using gradient ascent and so we find the derivative of the instantaneous version of each of these functions with respect to both the weights, $\mathbf{w}_1$ and $\mathbf{w}_2$, and the Lagrange multipliers, λ_1 and λ_2. By changing the Lagrange multipliers in proportion to the derivates of J we are changing the relative strength of the constraint compared to the function we are optimising; this allows us to smoothly maximise that function in the region in which we are satisfying the constraint.

Noting that

$$\frac{\partial g_1(\mathbf{w}_1 \mid \mathbf{w}_2)}{\partial \mathbf{w}_1} = \frac{\partial(y_1y_2)}{\partial \mathbf{w}_1} = \frac{\partial(\mathbf{w}_1\mathbf{x}_1y_2)}{\partial \mathbf{w}_1} = \mathbf{x}_1y_2 \tag{10}$$

these yield, respectively,

$$\frac{\partial J_1}{\partial \mathbf{w}_1} = \mathbf{x}_1y_2 - \lambda_1y_1\mathbf{x}_1 = \mathbf{x}_1(y_2 - \lambda_1y_1),$$

$$\frac{\partial J_1}{\partial \lambda_1} \propto (1 - y_1^2).$$

Similarly with the J_2 function, $\mathbf{w}_2$ and λ_2. This gives us a method of changing the weights and the Lagrange multipliers on an online basis. We use the joint learning rules

$$\Delta w_{1j} = \eta x_{1j}(y_2 - \lambda_1y_1), \tag{11}$$

$$\Delta\lambda_1 = \eta_0(1 - y_1^2),$$

$$\Delta w_{2j} = \eta x_{2j}(y_1 - \lambda_2y_2),$$

$$\Delta\lambda_2 = \eta_0(1 - y_2^2)$$

where w_{1j} is the jth element of weight vector, $\mathbf{w}_1$, etc. It has been found empirically that best results are achieved when $\eta_0 \gg \eta$.

Table 1
The converged weights from the neural network are compared with the values reported from a standard statistical technique (Mardia et al., 1979)

Standard statistics maximum correlation	0.6630		
$\mathbf{w}_1$	0.0260	0.0518	
$\mathbf{w}_2$	0.0824	0.0081	0.0035
Neural network maximum correlation	0.6962		
$\mathbf{w}_1$	0.0264	0.0526	
$\mathbf{w}_2$	0.0829	0.0098	0.0041

However, just as a neural implementation of PCA (e.g. Oja, 1982) may be very interesting but not a generally useful method of finding Principal Components, so a neural implementation of CCA may be only a curiosity. However, it has been shown that nonlinear extensions of PCA networks are able to search for the independent components of a data set (ICA, Oja, 1997) and that such extensions are therefore justifiable as engineering tools for investigating data sets. We therefore extend our neural implementation of CCA by maximising the correlation between outputs when such outputs are a nonlinear function of the inputs. We investigate a particular case of maximisation of $E(y_1 y_2)$ when the values y_i are a nonlinear function of the inputs, $\mathbf{x}_i$.

For the nonlinear optimisation, we use, e.g. $y_3 = \sum_j w_{3j} f_3(v_{3j} x_{1j}) = \mathbf{w}_3 \mathbf{f}_3$ where the function $f_3(\,)$ is applied on an element-wise basis to give the vector $\mathbf{f}_3$.

The equivalent optimisation function for the nonlinear problem is then

$$J_3(\mathbf{w}_3 \mid \mathbf{w}_4) = E(y_3 y_4) + \tfrac{1}{2}\lambda_3(1 - y_3^2),$$

$$J_4(\mathbf{w}_4 \mid \mathbf{w}_3) = E(y_4 y_3) + \tfrac{1}{2}\lambda_4(1 - y_4^2).$$

One solution is discussed in Section 3.5.

3. Experimental results

We report simulations on both real and artificial data sets of increasing complexity. We begin with data sets in which there is a linear correlation and we demonstrate the effectiveness of the network on random dot stereogram data of Becker (1996). We then extend the method in two ways not possible with standard statistical techniques:

1. We maximise correlations between more than two input data sets.
2. We consider maximising correlations where such correlations may be on nonlinear projections of the data.

3.1. Artificial data

Our first experiment comprises an artificial data set: $\mathbf{x}_1$ is a four-dimensional vector, each of whose elements is drawn from the zero-mean Gaussian distribution, $N(0,1)$; $\mathbf{x}_2$ is a three-dimensional vector, each of whose elements is also drawn from $N(0,1)$. In order to introduce correlations between the two vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$, we generate an additional sample from $N(0,1)$ and add it to the first elements of each vector. Thus there is no correlation between the two vectors other than that existing between the first element of each.

Using an initial learning rate of 0.0001 that is decreased linearly to 0 over 50 000 iterations, the weights converge to the vectors (**0.679**, 0.023, −0.051, −0.006) and (**0.681**, 0.004, 0.005). This clearly illustrates the high correlation between the first elements of each of the vectors and also the fact that this is the only correlation between the vectors. We may compare the reported results with the optimal values of $(\sqrt{0.5}, 0, 0, 0)$ and $(\sqrt{0.5}, 0, 0)$.

The effect of the constraint on the variance of the outputs is clearly seen when we change the distribution from which all samples are drawn to $N(0, 5)$. The weight vectors converge to (**0.141**, 0.002, 0.003, 0.002) and (**0.141**, 0.002, −0.001)—the optimal results are $(\sqrt{0.02}, 0, 0, 0)$ and $(\sqrt{0.02}, 0, 0)$. The differences in magnitude are due to the constraint, $E(y_i^2) = 1$ since

$$E(y_i^2) = 1 \Leftrightarrow E(\mathbf{w}_i \mathbf{x}\mathbf{x}^{\mathrm{T}} \mathbf{w}_i^{\mathrm{T}}) = \mathbf{w}_i R_{xx} \mathbf{w}_i^{\mathrm{T}} = 1 \qquad (12)$$

where R_{xx} is the covariance matrix of the input data.

3.2. Real data

Our second experiment uses a data set reported in Mardia et al. (1979, p. 290); it comprises 88 students' marks on five module exams. The exam results can be partitioned into two data sets: two exams were given as closed-book exams (C), while the other three were opened-book exams (O). The exams were on the subjects of Mechanics(C), Vectors(C), Algebra(O), Analysis(O), and Statistics(O). We thus split the five variables (exam marks) into two sets—the closed-book exams (x_{11}, x_{12}) and the open-book exams (x_{21}, x_{22}, x_{23}). One possible quantity of interest here is how highly a student's ability on closed-book exams is correlated with his ability on open-book exams. Alternatively, one might try to use the open-book exam results to predict the closed-book results (or vice versa).

The results shown in Table 1 were found using a learning rate of 0.0001 and 50 000 iterations. We have reported our results to four decimal places in this section to facilitate comparison with those reported in Mardia et al. (1979), which were found by standard statistical batch methods (Mardia et al., 1979). The $\mathbf{w}_1$ vector consists of the weights from the closed-book exam data to y_1 while the $\mathbf{w}_2$ vector consists of the weights from the open-book exam data to y_2. We note the excellent agreement between the methods. The

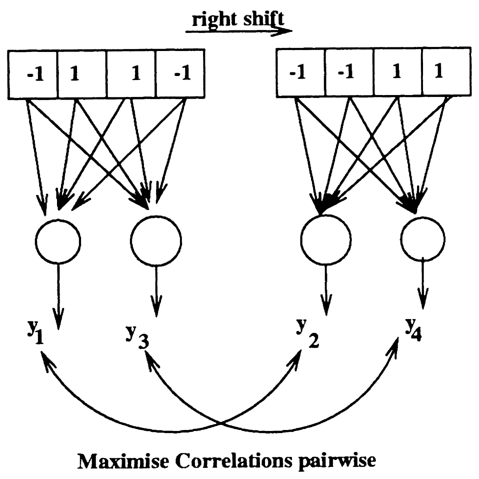


Fig. 2. The random dot stereogram data and network. The $\mathbf{x}_2$ set is either a left-shifted or a right-shifted (as shown) version of the $\mathbf{x}_1$ set. We find that $\mathbf{w}_1$ and $\mathbf{w}_2$ reliably find the left shift and $\mathbf{w}_3$ and $\mathbf{w}_4$ the right shift or vice versa.

highest correlations are given by a weighted average of x_{11} and x_{12} with the former receiving half the weight of the latter (since $\mathbf{w}_{11} \approx \mathbf{w}_{12}$) and the average of x_{21}, x_{22} and x_{23} heavily weighted on x_{21} (since $\mathbf{w}_{21} \gg \mathbf{w}_{22}$, $\mathbf{w}_{23}$).

3.3. Random dot stereograms

It has been suggested (Becker, 1996) that one of the goals of sensory information processing may be the extraction of common information between different sensors or across sensory modalities. One reason that this is possible is because of the coherence that exists in time and place in sensory input data. We may view the above network as a means of merging two different data streams—$\mathbf{x}_1$ and $\mathbf{x}_2$—which may be either representatives of two different modalities or as different representatives of the same modality where such representatives may be different in either time or place. Becker (1996) has developed this idea and experimented on a data set which is an abstraction of random dot stereograms: an example is shown graphically in Fig. 2. The central idea behind this is that two different neural units or neural network modules should learn to extract features that are coherent across their inputs. If there is any feature in common across the two inputs, it should be discovered, while features that are independent across the two inputs will be ignored.

Each input vector consists of a one-dimensional random strip which corresponds to the left image and a shifted version of this which corresponds to the right image. The left image has components drawn with equal probability from the set $\{-1,1\}$ and the right image is generated by choosing a randomly chosen global shift—either one pixel left or one pixel right—and applying it to the left image. We wish to find the maximum linear correlation between y_1 and y_2 which are themselves linear combinations of $\mathbf{x}_1$ and $\mathbf{x}_2$. Because the shifts are chosen with equal probability, there are two equal sets of correlations corresponding to left-shift and right-shift. In order to find these, we require two pairs of outputs and the corresponding pairs of weights ($\mathbf{w}_1$,$\mathbf{w}_2$) and ($\mathbf{w}_3$,$\mathbf{w}_4$). The learning rules for $\mathbf{w}_3$ and $\mathbf{w}_4$ in this experiment are analagous to those for $\mathbf{w}_1$ and $\mathbf{w}_2$; at each presentation of a sample of input data, a simple competition between the products y_1y_2 and y_3y_4 determine which weights will learn on the current input samples: if $y_1y_2 > y_3y_4$, $\mathbf{w}_1$, $\mathbf{w}_2$ are updated, else $\mathbf{w}_3$, $\mathbf{w}_4$ are updated.

Using a learning rate of 0.001 and 100 000 iterations, the weights converge to the vectors shown in Table 2.

The first pair of weights $\mathbf{w}_1$ and $\mathbf{w}_2$ have identified the second element of $\mathbf{x}_1$ and the third element of $\mathbf{x}_2$ as having maximum correlation while other inputs are ignored (the weights from these are approximately 0). This corresponds to a right shift. This first pair of outputs has a (sample) correlation of 0.512. Similarly the second pair of weights has identified the third element of $\mathbf{x}_1$ and the second element of $\mathbf{x}_2$ as having maximum correlation while other inputs are ignored. The second pair has a (sample) correlation of 0.530 and corresponds to an identification of left shift.

Now for some patterns there will be ambiguity since it is possible that by chance a right-shifted pattern will happen to match the weights $\mathbf{w}_3$, $\mathbf{w}_4$, and therefore a bank of pairs of pairs of neurons is required to perform as well as Becker's IMAX network (Becker, 1996). For best results, each set of four neurons as above should see as input a slightly different part of the input data (though with a global left or right shift). We have shown elsewhere (Lai & Fyfe, 1999) that a Factor Analysis network which takes as input the output of such a network can identify left and right shifts from this data set.

The table thus demonstrates that these high correlations come from one pair learning the shift-left transformation while the other learns the shift-right.

It should be noted at this point that Becker (1996) only uses a subset of 12 of the 16 possible patterns. Those which are ambiguous (such as $(-1,1,-1,1)$) are removed whereas we are drawing our data from all 16 possible patterns. In addition, the method in Becker (1996) uses computationally expensive backpropagation of the derivatives of mutual information. We are able to find both correlations with a very simple network.

Kay and Phillips (1997) have developed a neural network method for integrating contextual and input data. We might consider, e.g. from the first data set, $\mathbf{x}_1$ as the input data to the network and $\mathbf{x}_2$ as the contextual input. The results from

Table 2
The converged weights clearly show that the first pair of neurons has learned a right shift while the second pair has learned a left shift

$\mathbf{w}_1$	−0.002	**1.110**	0.007	−0.009
$\mathbf{w}_2$	0.002	0.025	**0.973**	0.020
$\mathbf{w}_3$	−0.014	0.026	**1.111**	−0.002
$\mathbf{w}_4$	0.013	**0.984**	0.003	−0.007

Table 3
The weights of the converged three input vectors network. The network has clearly identified the correlation between the first element in each vector

$\mathbf{w}_1$	**0.812**	0.013	0.027
$\mathbf{w}_2$	**0.777**	−0.014	0.030
$\mathbf{w}_3$	**0.637**	0.007	0.012

the CCA network and the Kay and Phillips network are similar too, though Kay and Phillips use a probabilistic network with a nonlinear activation function designed to manage the effect of contextual information on the response of the network to input data.

Finally, the CCA network presented here may be criticised as a model of biological information processing in that it appears as a nonlocal implementation of Hebbian learning, i.e. the $\mathbf{w}_1$ weights use the magnitude of y_2 as well as y_1 to self-organise. One possibility is to postulate nonlearning connections which join y_2 to y_1 thus providing the information that $\mathbf{w}_1$ requires for learning. Alternatively we may describe the λ_1 parameter as a lateral weight from y_2 to y_1 and so the learning rules become

$$\Delta w_{1j} = (\eta\lambda_1)x_{1j}\left(\frac{y_2}{\lambda_1} - y_1\right),$$

$$\Delta\lambda_1 = \eta_0(1-y_1^2)$$

where we have had to incorporate a λ_1 term into the learning rate. Perhaps the second of these suggestions is more plausible than the first as a solution to the nonlocal feature of the previous learning rules, since nonlearning connections hardwires some activation passing into the cortex. This is an area of further investigation.

3.4. More than two data sets

In integrating information from different sensory modalities, the cortex may be presented with the problem of integrating that from more than two data sets simultaneously. Therefore we extend the algorithm without introducing unnecessarily complex activation passing which would become biologically implausible.

We create an artificial data set which comprises three vectors each of whose first elements have correlations of equal magnitude: $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ are each three-dimensional vectors, each of whose elements is initially independently drawn from $N(0,1)$. We now draw a sample from $N(0,1)$ and add it to the first element of each of $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ and attempt to maximise the correlation between y_1, y_2 and y_3. We opt to maximise three separate constrained objective functions:

$$J_1 = E(y_1y_2) + \tfrac{1}{2}\lambda_1(1-y_1^2),$$

$$J_2 = E(y_2y_3) + \tfrac{1}{2}\lambda_2(1-y_2^2)$$

and

$$J_3 = E(y_3y_1) + \tfrac{1}{2}\lambda_3(1-y_3^2).$$

We use gradient ascent on the instantaneous version of each of these functions with respect to both the weights, $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$, and the Lagrange multipliers, λ_1, λ_2 and λ_3. This gives us the learning rules

$$\frac{\partial J_1}{\partial \mathbf{w}_1} = \mathbf{x}_1y_2 - \lambda_1y_1\mathbf{x}_1 = \mathbf{x}_1(y_2-\lambda_1y_1),$$

$$\frac{\partial J_2}{\partial \mathbf{w}_2} = \mathbf{x}_2y_3 - \lambda_2y_2\mathbf{x}_2 = \mathbf{x}_2(y_3-\lambda_2y_2),$$

$$\frac{\partial J_3}{\partial \mathbf{w}_3} = \mathbf{x}_3y_1 - \lambda_3y_3\mathbf{x}_3 = \mathbf{x}_3(y_1-\lambda_3y_3).$$

The derivates with respect to the Lagrange multipliers are similar to the previous rules (11) though we have found empirically that the best result are achieved when the λ's learning rate is again very greatly increased (now $\eta_0 \approx 200$–1000η).

Using $\eta = 0.0001$, $\eta_0 = 0.05$ and 100 000 iterations, the weights converge to the values shown in Table 3. The

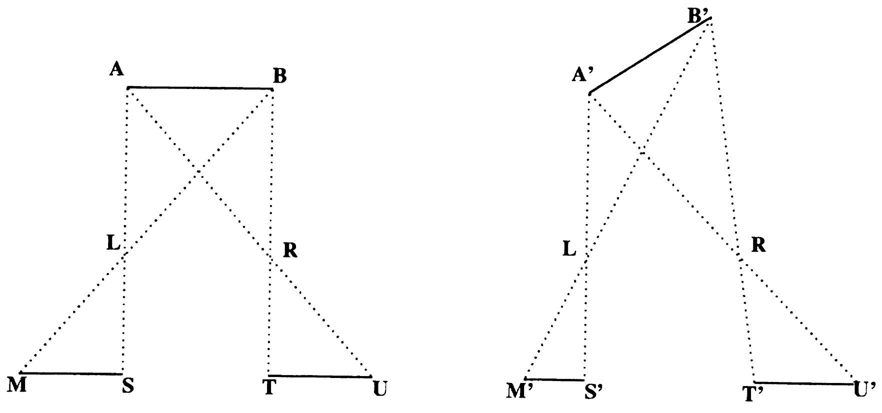
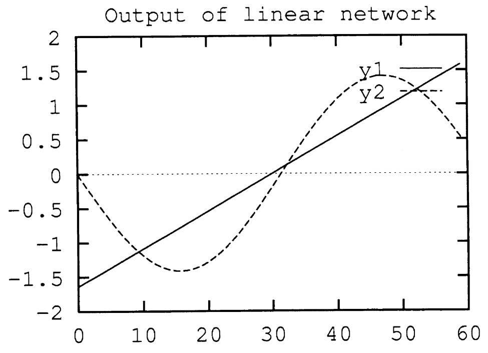


Fig. 3. The left figure represents visual information from a surface AB which is passed through pupils R and L to flat "retinas" MS and TU. The right figure represents the same scene when the external surface A′B′ is not parallel to the plane of the retinas.



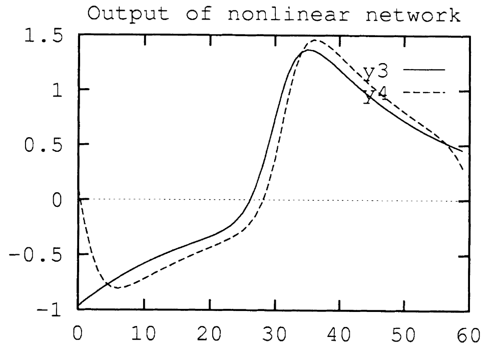


Fig. 4. The top diagram shows the outputs, y_1 and y_2, of the linear network, while the lower diagram shows the outputs, y_3 and y_4, of the nonlinear network. Visual inspection would suggest that the outputs from the nonlinear network are more correlated. The actual sample corrrelation values achieved were 0.623 (linear) and 0.865 (nonlinear).

three-way correlation derived by this method is equal to three pairwise correlations.

3.5. Nonlinear correlations

As the data set in Section 3.3 provides us with an abstraction of random dot stereograms, it is not a complete and accurate abstraction of how the cortex extracts depth information from surfaces: at any one time, we view not just single points but segments of (at least) a surface. Consider Fig. 3.

We see that the relationship between the projection of the surface on the retinas is a function of the angle between the plane of the surface and the plane of the retinas. We do not wish to determine the precise relationship in any specific case since we wish to create a general purpose depth analyser which does not depend on, e.g. both the retinas and the surface being flat, the pupils having a precise relationship with the limits of the viewed surfaces.

Therefore in our final experiment, we investigate the general problem of maximising correlations between two data sets when there may be an underlying nonlinear relationship between the data sets: we generate artificial data according to the prescription:

$$x_{11} = 1 - \sin\theta + \mu_1, \tag{13}$$

$$x_{12} = \cos\theta + \mu_2, \tag{14}$$

$$x_{21} = \theta - \pi + \mu_3, \tag{15}$$

$$x_{22} = \theta - \pi + \mu_4 \tag{16}$$

where θ is drawn from a uniform distribution in $[0, 2\pi]$ and μ_i, $i = 1,\dots,4$ are drawn from the zero mean Gaussian distribution $N(0,0.1)$. Eqs. (13) and (14) define a circular manifold in the two-dimensional input space while Eqs. (15) and (16) define a linear manifold within the input space where each manifold is only approximate due to the presence of noise (μ_i, $i = 1,\dots,4$). The subtraction of π in the linear manifold equations is merely to centre the data.

Thus $\mathbf{x}_1 = \{x_{11}, x_{12}\}$ lies on or near the circular manifold $x_{11}^2 + x_{12}^2 = 1$ while $\mathbf{x}_2 = \{x_{21}, x_{22}\}$ lies on or near the line $x_{21} = x_{22}$.

We wish to test whether the network can find nonlinear correlations between the two data sets, $\mathbf{x}_1$ and $\mathbf{x}_2$, and test whether such correlations are greater than the maximum linear correlations. To do this we train two pairs of output neurons:

- we train one pair of weights $\mathbf{w}_1$ and $\mathbf{w}_2$ using rules (11);
- we train a second pair of outputs, y_3 and y_4 which are calculated from

$$y_3 = \sum_j w_{3j} \tanh(v_{3j} x_{1j}) = \mathbf{w}_3 \mathbf{f}_3, \tag{17}$$

$$y_4 = \sum_j w_{4j} \tanh(v_{4j} x_{2j}) = \mathbf{w}_4 \mathbf{f}_4. \tag{18}$$

This gives us another pair of adaptable parameters that may be changed to maximise the correlation. In particular, since the weights, $\mathbf{v}_3$ and $\mathbf{v}_4$ are used prior to the use of the nonlinearity, this gives us extra flexibility in maximising correlations.

The correlation between y_1 and y_2 neurons was maximised using the previous linear operation (11) while that between y_3 and y_4 used the functions

$$J_3 = E(y_3 y_4) + \tfrac{1}{2}\lambda_3(1 - y_3^2)$$

and

$$J_4 = E(y_3 y_4) + \tfrac{1}{2}\lambda_4(1 - y_4^2)$$

whose derivatives give us (taking into account the nonlinearities (17) and (18))

$$\frac{\partial J_3}{\partial \mathbf{w}_3} = \mathbf{f}_3 y_4 - \lambda_3 y_3 \mathbf{f}_3 = \mathbf{f}_3(y_4 - \lambda_3 y_3), \tag{19}$$

$$\frac{\partial J_3}{\partial \mathbf{v}_3} = \mathbf{w}_3 y_4 (1 - \mathbf{f}_3^2)\mathbf{x}_1 - \lambda_3 \mathbf{w}_3 (1 - \mathbf{f}_3^2)\mathbf{x}_1 y_3$$

$$= \mathbf{w}_3 (1 - \mathbf{f}_3^2)\mathbf{x}_1 (y_4 - \lambda_3 y_3).$$

Similarly with the J_4 function, $\mathbf{w}_4$, $\mathbf{v}_4$, and λ_4. This gives us a method of changing the weights and the Lagrange multipliers on an online basis. We use the joint learning rules

$$\Delta \mathbf{w}_3 = \eta \mathbf{f}_3(y_4 - \lambda_3 y_3),$$

$$\Delta \mathbf{w}_4 = \eta \mathbf{f}_4(y_3 - \lambda_4 y_4),$$

$$\Delta \mathbf{v}_{3i} = \eta \mathbf{x}_{1i} \mathbf{w}_{3i}(y_4 - \lambda_3 y_3)(1 - \mathbf{f}_3^2),$$

$$\Delta \mathbf{v}_{4i} = \eta \mathbf{x}_{2i} \mathbf{w}_{4i}(y_3 - \lambda_4 y_4)(1 - \mathbf{f}_4^2).$$

We use a learning rate of 0.001 for all weights and learn over 100 000 iterations. The network finds a sample linear correlation between the data sets of 0.623 (equal to the correlation between y_1 and y_2) while the nonlinear neurons, y_3 and y_4, have a sample correlation of 0.865. In putting the data sets through the nonlinear tanh( ) function we have created a relationship whose correlations are greater than the linear correlations of the original data set. We show a test of the outputs from both the linear and nonlinear networks in Fig. 4 in which we graph the output values of the trained network from each pair of neurons against inputs where the values in Eqs. (13)–(16) range from -3 to 3. We see that the linear network is aligning the outputs as well as may be expected but the nonlinear network's outputs are very closely aligned with each other over much more of the data set.

4. Conclusion

We have illustrated the performance of a linear canonical correlation neural network on a variety of data sets, both real and artificial. We have been able to extend the CCA method

- to deal with the case where the correlation extends to more than two input data sets;
- to deal with the case where the greatest correlation between data sets is nonlinear.

Neither of these extensions are possible using the standard statistical methods of Canonical Correlation Analysis.

The network was also successfully operated on a data set designed to abstract the essential concepts of random dot stereograms and was shown to be capable of finding whether a particular pair of inputs exhibited a left or a right shift. The actual data set used contained all possible pairs of inputs whether ambiguous or not and identified the appropriate shift extremely reliably.

Future work on this method will concentrate on nonlinear correlations since there is no simple closed form solution to the maximisation of such correlations: an iterative method such as a neural network weight learning method is necessary for such problems. Of most interest, is the situation where such correlations are to be found in real (visual) data.

Acknowledgements

The authors would like to gratefully acknowledge the many helpful discussions with Darryl Charles and the valuable comments of the reviewers.

References

Becker, S. (1996). Mutual information maximization: models of cortical self-organization. *Network: Computation in Neural Systems*, *7*, 7–31.

Charles, D., & Fyfe, C. (1998). Modelling multiple cause structure using rectification constraints. *Network: Computation in Neural Systems*, *9*, 167–182.

Fyfe, C., & Baddeley, R. (1995). Non-linear data structure extraction using simple hebbin networks. *Bilogical Cybernetics*, *72* (6), 533–541.

Kay, J., & Phillips, W. A. (1997). Activation functions, computational goals and learning rules for local processors with contextual guiding. *Neural Computation*, *9* (4), 895–910.

Lai, P.L., & Fyfe, C. (1998). Canonical correlation analysis using artificial neural networks. In: *European Symposium on Artificial Neural Networks, ESANN98*, pp. 363–367.

Lai, P.L., & Fyfe, C. (1999). A probalistic derivation of canonical correlation analysis. Submitted for publication.

Mardia, K. V., Kent, J., & Bibby, J. (1979). *Multivariate analysis*, New York: Academic Press.

Oja, E. (1982). A simplifed neuron model as a principal component analyser. *Journal of Mathematical Biology*, *16*, 267–273.

Oja, E. (1989). Neural networks, principal components and subspaces. *International Journal of Neural Systems*, *1*, 61–68.

Oja, E. (1997). The nonlinear pca learning rule in independent component analysis. *Neurocomputing*, *17*, 25–45.

Sanger, T. (1990). Analysis of the two-dimensional receptive fields learned by the generalized hebbian algorithm in response to random input. *Biological Cybernetics*, *63*, 221–228.