ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 363-368

Canonical Correlation Analysis using Artificial Neural Networks

Pei Ling Lai and Colin Fyfe,

Artificial Neural Networks Research Group,
The University of Paisley,
Scotland.
email: lai–ci0@paisley.ac.uk, fyfe0ci@paisley.ac.uk
fax : +141 848 3542.

Abstract. We derive a new method of performing Canonical Correlation Analysis with Artificial Neural Networks. We demonstrate its capability on a simple artificial data set and then on a real data set where the results are compared with those achieved with standard statistical tools. We then extend the method to deal with a situation where there are two equal competing correlations within the datasets and show that this extension is effective on the previous data sets.

1. Introduction

Artificial Neural Networks (ANNs) are well known as being capable of performing powerful transformations. Some of the first demonstrations of this power came from the family of networks (e.g. [5, 6, 7, 2] 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 non-linear extensions of PCA networks have been shown to be capable of more sophisticated statistical techniques such as Exploratory Projection Pursuit [3] and Factor Analysis [1].

In this paper, we investigate a neural network implementation of Canonical Correlation Analysis (CCA). Canonical Correlation Analysis is used when we have two data sets which we believe have some underlying correlation. Consider two sets of input data; $\mathbf{x}_1$ and $\mathbf{x}_2$. Then in classical CCA, we attempt to find that linear combination of the variables which gives us maximum correlation between the combinations. Let

$$\begin{aligned} y_1 &= \mathbf{w}_1\mathbf{x}_1 = \sum_j w_{1j}x_{1j} \\ y_2 &= \mathbf{w}_2\mathbf{x}_2 = \sum_j w_{2j}x_{2j} \end{aligned}$$

Then we wish to find those values of $\mathbf{w}_1$ and $\mathbf{w}_2$ which maximise the correlation between y_1 and y_2 . Whereas Principal Components 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.

One way to view canonical correlation analysis is as an extension of multiple regression. 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 $\mathbf{x}_1$ which is most highly correlated with $\mathbf{x}_2$.

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 appropriate weights, $\mathbf{w}_1$ and $\mathbf{w}_2$.

$$\begin{aligned} y_1 &= \mathbf{w}_1\mathbf{x}_1 = \sum_j w_{1j}x_{1j} \\ y_2 &= \mathbf{w}_2\mathbf{x}_2 = \sum_j w_{2j}x_{2j} \end{aligned}$$

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 $g_1(\mathbf{w}_1|\mathbf{w}_2) = E(y_1y_2)$ as a function of the weights, $\mathbf{w}_1$. This is an unconstrained maximisation problem which clearly 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,

$$\begin{aligned} J_1 &= E(y_1y_2) + \frac{1}{2}\lambda_1(1 - y_1^2) \text{ and} \\ J_2 &= E(y_1y_2) + \frac{1}{2}\lambda_1(1 - y_2^2) \end{aligned}$$

These 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. Noting that

$$\frac{\partial g(\mathbf{w}_1|\mathbf{w}_2)}{\partial \mathbf{w}_1} = \frac{\partial(y_1y_2)}{\partial \mathbf{w}_1} = \frac{\partial(\mathbf{w}_1\mathbf{x}_1y_2)}{\partial \mathbf{w}_1} = \mathbf{x}_1y_2 \qquad (1)$$

these yield respectively

$$\begin{aligned}\frac{\partial J_1}{\partial \mathbf{w}_1} &= \mathbf{x}_1 y_2 - \lambda_1 y_1 \mathbf{x}_1 = \mathbf{x}_1(y_2 - \lambda_1 y_1) \\ \frac{\partial J_1}{\partial \lambda_1} &\propto (1 - y_1^2)\end{aligned}$$

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

$$\begin{aligned}\Delta w_{1j} &= \eta x_{1j}(y_2 - \lambda_1 y_1) \\ \Delta \lambda_1 &= \eta_0(1 - y_1^2) \\ \Delta w_{2j} &= \eta x_{2j}(y_1 - \lambda_2 y_2) \\ \Delta \lambda_2 &= \eta_0(1 - y_2^2)\end{aligned}$$

It has been found empirically that best results are achieved when $\eta_0 >> \eta$.

3. Experimental Results

We report simulations on both real and artificial data.

3.1. Artificial Data

Our first experiment comprises an artificial data set: $\mathbf{x}_1$ is a 4 dimensional vector, each of whose elements is drawn from the zero-mean Gaussian distribution, N(0,1); $\mathbf{x}_2$ is a 3 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 a learning rate of 0.0001 and 500000 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.

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).

3.2. Real data

Our second experiment uses a data set reported in [4]; it comprises 88 students' marks on 5 module exams. The exam results can be partitioned into two data sets: two exams were given as open book exams while the other three were closed book exams. The exams were on the subjects of Mechanics(C), Vectors(C), Algebra(C), 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).

Using a learning rate of 0.0001 and 500000 iterations, the maximal correlation our network finds is 0.696. The weights converge to the vectors (0.026, 0.052) and (0.082, 0.009, 0.004). These compare with reported results [4] of 0.663 and (0.026, 0.052) and (0.082, 0.008, 0.004) which were found by standard statistical batch methods.

3.3. Equal Correlations

We now create artificial data which contains two independent correlations of equal magnitude. We repeat experiment 1 with the same artificial data but this time create correlations between x_{11} and x_{21} and correlations of equal magnitude between x_{12} and x_{22} by drawing two independent samples from N(0,1) one to x_{11} and x_{21} and the other to x_{12} and x_{22}. The above network failed to converge to either of the correlations presumably because the correlations were of equal but independent magnitude. However by introducing an asymmetry to the network in our constraints - we allow the outputs to have different power originally -

$$\begin{aligned} J_1 &= E(y_1 y_2) + \frac{1}{2}\lambda_1(k_1 - y_1^2) \text{ and} \\ J_2 &= E(y_1 y_2) + \frac{1}{2}\lambda_1(k_2 - y_1^2) \end{aligned}$$

our weights converge to the CCA directions. We found that it is not necessary that $k_1 \neq k_2$ for all time, but merely ensure that during the first phase of convergence there is an inequality between these values (0.367666, 0.677243, 0.00431447, -0.0733304) and (0.313141, 0.582314, 0.000111589). We also now report that the most accurate results (such as those reported in the first section on artificial data) are achieved when there is some asymmetry between the parameters k_1 and k_2 even when there is only one correlation between the vectors.

4. Conclusion

We have investigated a neural network implementation of Canonical Correlation Analysis (CCA) and demonstrated its power on simple problems. We have reported simulations on both real and artificial data. We have shown that

1. Artificial Data - Starting with different initial distributions we can reliability find different canonical correlation.

2. Real Data - We have shown that the same network structure is capable of finding the same correlations in real data which were found by standard statistical batch methods.

3. Equal Correlation - We create artificial data containing two independent correlation of equal magnitude and extended our method to force the weights to converge to the CCA directions. We found that it is not necessary that $k_1 \neq k_2$ for all time. We also now report that most accurate results are achieved when there is some asymmetry between the parameters $\mathbf{k}_1$ and $\mathbf{k}_2$ even when there is only one correlation between the vectors.

Future work will concentrate on finding nonlinear correlations in data sets- a task for which standard statistical methods is not in general well-suited.

References

[1] D. Charles and C. Fyfe. Modelling multiple cause structure using rectification constraints. *(accepted for publication)*, 1997.

[2] C. Fyfe. Pca properties of interneurons. In *From Neurobiology to Real World Computing, ICANN 93*, pages 183–188, 1993.

[3] C. Fyfe and R. Baddeley. Non-linear data structure extraction using simple hebbian networks. *Biological Cybernetics*, 72(6):533–541, 1995.

[4] K. V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

[5] E. Oja. A simplified neuron model as a principal component analyser. *Journal of Mathematical Biology*, 16:267–273, 1982.

[6] E. Oja. Neural networks, principal components and subspaces. *International Journal of Neural Systems*, 1:61–68, 1989.

[7] T.D. Sanger. Analysis of the two-dimensional receptive fields learned by the generalized hebbian algorithm in response to random input. *Biological Cybernetics*, 1990.