

Information Processing in Negative Feedback Neural Networks

M D Plumbley

Department of Electronic and Electrical Engineering, King's College London, Strand,
London WC2R 2LS, United Kingdom

Abstract. Information theory suggests that extraction of the principal subspace from data is useful when the input to a neural network is corrupted with additive noise. A number of neural network algorithms exist which can find this principal subspace, many of which also extract the principal components of the input. However, when there is noise on both input and output of a network, simply extracting the principal subspace (or components) is not sufficient to optimize information capacity. An approximate solution to maximizing information capacity would be to extract the principal subspace of components with variances above a certain threshold, and then ensure that these are uncorrelated and that they have equal variance at the output. A neural network is described which uses negative feedback connections to achieve this uncorrelated, equal variance solution.

Short title: Information Processing in Negative Feedback Neural Networks

24 January 1996

1. Introduction

Information theory has been used as a tool towards the understanding of perception, almost as soon as the concept of *mutual information* was introduced by Shannon [1]. For example, Barlow [2] suggested that neural networks with *lateral inhibition* connections could be used to achieve an economical description of perceptual information. More recently, work by authors such as Linsker [3] and Atick and Redlich [4] have applied information theory more directly to neural network learning. In this article we shall use this approach to develop an algorithm for a two-stage neural network with negative feedback.

2. Principal subspace networks

Suppose we have an N -dimensional zero-mean input random vector \mathbf{x}_a and an M -dimensional output vector $\mathbf{y}_a = [y_1, \dots, y_M]$ where $\mathbf{y}_a = \mathbf{W}\mathbf{x}_a$ and \mathbf{W} is an $M \times N$ weight matrix with $M < N$ (figure 1(a)). If we set the M successive rows of \mathbf{W} to be the M largest eigenvectors of the input covariance matrix $\Sigma_{\mathbf{x}_a} = E(\mathbf{x}_a\mathbf{x}_a^T)$, then we have performed a *principal component analysis* (PCA) of the input \mathbf{x}_a .

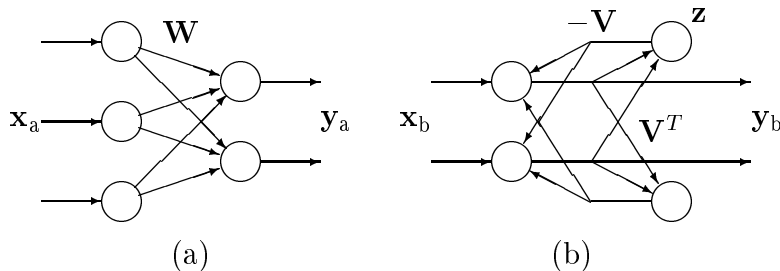


Figure 1. Linear networks for performing (a) principal subspace analysis, and (b) decorrelation.

If the input \mathbf{x}_a is corrupted by equal-variance additive gaussian noise, then PCA optimizes the information capacity from \mathbf{x}_a to \mathbf{y}_a . In fact, for optimum information capacity it is sufficient for the rows of \mathbf{W} to span the same subspace as the first M eigenvectors of $\Sigma_{\mathbf{x}_a}$: we call this principal *subspace* analysis. Any reversible rotation or scaling in the output \mathbf{y}_a -space does not affect the information capacity [3, 5].

A number of neural network algorithms have been developed to perform PCA or principal subspace analysis [6, 7, 5], many based on the Oja [8] principal component finding neuron. Often these use a learning algorithm of the form

$$\Delta \mathbf{W} = \eta_{\mathbf{W}} (\mathbf{y}_a \mathbf{x}_a^T - \mathbf{K} \mathbf{W}) \quad (1)$$

where $\eta_{\mathbf{W}}$ is a small update factor, and \mathbf{K} is a weight decay matrix which is typically a function of \mathbf{W} and \mathbf{x}_a . The form of \mathbf{K} is chosen to prevent the output variances increasing without bound, and to prevent all of the network output converging on the largest single principal component. For example, Williams' SEC algorithm [6] uses $\mathbf{K} \triangleq \mathbf{y}_a \mathbf{y}_a^T$, which results in an orthonormal weight set \mathbf{W} which finds the principal subspace.

3. Decorrelation networks

Suppose that instead of input noise, we were faced with noise on the output of a network, together with a maximum power limit (or cost) which prevents us from simply amplifying the network output to overcome the noise. In this case, information capacity will be optimized when the outputs are uncorrelated, and have equal variance (i.e. *orthonormalized*) [9].

For a neural network and algorithm to achieve this, suppose that we have a network with an input vector \mathbf{x}_b , output vector \mathbf{y}_b , and interneuron vector \mathbf{z} , where these are all M -dimensional (figure 1(b)). We further suppose that \mathbf{y}_b and \mathbf{z} adapt over a fast timescale so that

$$\mathbf{z} \leftarrow \eta_z (\mathbf{V}^T \mathbf{y}_b) + (1 - \eta_z) \mathbf{z} \quad \mathbf{y}_b \leftarrow \eta_y (\mathbf{x}_b - \mathbf{V} \mathbf{z}) + (1 - \eta_y) \mathbf{y}_b \quad (2)$$

where $-\mathbf{V}$ is a matrix of inhibitory connections from \mathbf{z} to \mathbf{y}_b , which are equal and opposite to the connections \mathbf{V}^T from \mathbf{y}_b to \mathbf{z} , and η_y and η_z are the update factors for \mathbf{y}_b and \mathbf{z} respectively. The equilibrium activity is then given by

$$\mathbf{z} = \mathbf{V}^T \mathbf{y}_b \quad \mathbf{y}_b = \mathbf{x}_b - \mathbf{V} \mathbf{z} \quad (3)$$

so that we have

$$\mathbf{y}_b = (\mathbf{I} + \mathbf{V} \mathbf{V}^T)^{-1} \mathbf{x}_b \quad (4)$$

provided the term $(\mathbf{I} + \mathbf{V} \mathbf{V}^T)$ is positive definite. This settling is assumed to operate at a timescale much faster than any change in the weights \mathbf{V} .

In [9], it was shown that the simple local algorithm

$$\begin{aligned} \Delta \mathbf{V} &= \eta_{\mathbf{V}} (\mathbf{y} \mathbf{z}^T - \beta \mathbf{V}) \\ &= \eta_{\mathbf{V}} (\mathbf{y} \mathbf{y}^T - \beta \mathbf{I}) \mathbf{V} \end{aligned} \quad (5)$$

where $\eta_{\mathbf{V}}$ is a small learning rate, produces the decorrelated equal variance outputs required, provided this can be achieved by *reducing*, and not increasing, the variance of all the input components.

We shall now see how we can combine these two networks into one which approximates an optimal solution where there is both input *and* output noise.

4. Orthonormalized principal subspace network

We have seen that if a network has noise on the input only, extracting the principal subspace optimizes information capacity. On the other hand, if a network has noise on the output only and a output power limit, then its outputs should be orthonormalized.

If we have noise on both the input and the output of a network with limited output power, the situation is rather more complicated, and depends on the relative levels of signal and noise variance of each principal component at the input to the network [10]. If the input signal is smaller than a certain threshold (determined by an operating point parameter), then that component should be suppressed completely (i.e. its output variance set to zero). If the input signal is large, it should be left uncorrelated from other components and set to the same variance as other non-suppressed components at the output. Between these two extremes, the output variance of the component should gradually increase approximately with the square root of the input variance. We will construct a network which approximates this solution, ignoring the mid-range graded behaviour requirement. This we should extract the principal subspace, containing input components with variance above the threshold, and orthonormalize these at the output.

While we could do this by extracting the principal components using e.g. a PCA network, and then scaling the outputs to give equal variance outputs (or suppress them if they are too small), here we present a neural network and algorithm with feedback connections which does not require the principal components to be calculated explicitly.

The network structure we use is a combination of the principal subspace network (figure 1(a)) with $\mathbf{x}_a = \mathbf{x}$ and $\mathbf{y}_a = \mathbf{y}$, followed by the orthonormalizing network (figure 1(b)), with $\mathbf{x}_b = \mathbf{W}\mathbf{x}$ and $\mathbf{y}_b = \mathbf{y}$ as shown in figure 2. When the activations have settled to their equilibrium values, we have $\mathbf{y} = \mathbf{W}\mathbf{x} - \mathbf{V}\mathbf{V}^T\mathbf{y}$ which gives us an output which satisfies

$$\mathbf{y} = (\mathbf{I} + \mathbf{V}\mathbf{V}^T)^{-1}\mathbf{W}\mathbf{x}. \quad (6)$$

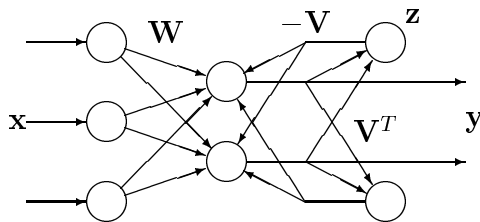


Figure 2. Network with combined principal subspace and decorrelating stages.

The algorithm for \mathbf{V} in the decorrelation stage is left unaltered, but the principal subspace stage needs a different learning algorithm for \mathbf{W} . Our new requirement for the

outputs y_i to be uncorrelated and have equal variance (if not suppressed) is incompatible with most principal subspace (and PCA) algorithms, which retain the relative scaling of input components at the output. These often produce an orthonormal set of *weights* \mathbf{W} rather than orthonormal outputs.

In fact, the algorithm for \mathbf{W} can be simplified, since the orthonormalizing output stage prevents the output variances increasing. We therefore propose the two-stage algorithm

$$\Delta\mathbf{W} = \eta_{\mathbf{W}}(\mathbf{y}\mathbf{x}^T - \alpha\mathbf{W}) \quad (7)$$

$$\Delta\mathbf{V} = \eta_{\mathbf{V}}(\mathbf{y}\mathbf{z}^T - \beta\mathbf{V}) \quad (8)$$

where α is a constant weight decay which fixes the input variance threshold, β is a constant weight decay which determines the output variance from the network, and we assume that that $\eta_{\mathbf{W}} \ll \eta_{\mathbf{V}}$. This is a generalization of a 1-input, 1-output algorithm by Barrow and Budd [11].

Analysis of the behaviour of the output stage proceeds as for the orthonormalizing stage alone [9], since \mathbf{V} is adapting with approximately fixed \mathbf{W} , so we simply have $\mathbf{W}\mathbf{x}$ in place of \mathbf{x}_a .

For \mathbf{W} we analyze the behaviour of algorithm (7) under the assumption that \mathbf{V} continually adapts to keep (8) stationary. When (7) is stationary, we find that $\Sigma_{\mathbf{x}}$ and $\mathbf{W}\mathbf{W}^T$ commute, and hence share eigenvectors [10]. Further, if only $L < M$ input components have variance greater than α , then \mathbf{W} can span at most these L components. If $L \geq M$, then some M of these L can be spanned. A rather involved perturbation analysis confirms that the matrix $\mathbf{W}^T\mathbf{W}$ is stable only when the rows of \mathbf{W} span the *principal* subspace of dimension $\min(L, M)$ of the input. (\mathbf{W} itself is never fully “stable”, since it is possible to make a rotation-type perturbation to the outputs of \mathbf{W} while \mathbf{W} continues to span the principal subspace. Analysis shows that the algorithm for \mathbf{W} will not resist this perturbation, and hence \mathbf{W} is theoretically “unstable”, in that it is confined to a sub-manifold rather than a single point. For a detailed analysis, see [10].)

Note that while this network effectively measures and equalizes (or suppresses) the variances of the input principal components as required, it does not perform PCA. There is no particular alignment of input principal components with output nodes. Figure 3 gives a simple example of below-threshold components being suppressed.

5. Discussion and conclusions

It is possible to construct other related feedback networks. For example, an additional interneuron stage can be used in the feedback loop [12], or a double feedback loop can be used to give a square-root variance characteristic which is closer to the theoretical optimum in the mid range of input variance [13]. The algorithms required for these

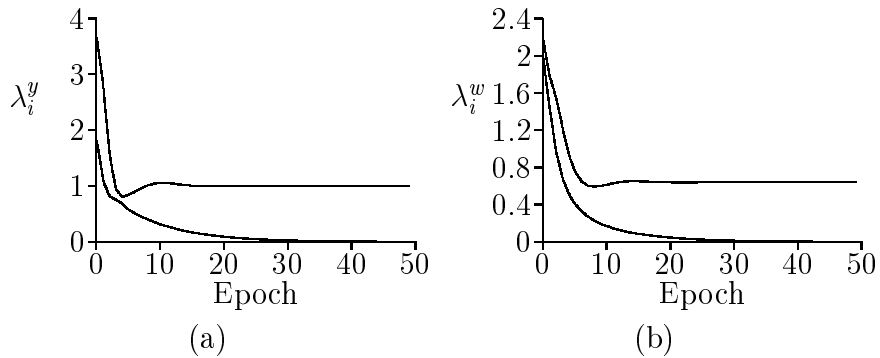


Figure 3.

Learning curves for algorithm (7,8) applied to a 4-input, 2-output network with input variances 4.0, 2.0, 1.0 and 0.0, and the parameters $\alpha = 2.5$ and $\beta = 1.0$ (chosen so that only one input component is above the threshold α). Plot (a) shows that the eigenvalues λ_1^y, λ_2^y of output covariance matrix converge to $\beta = 1.0$ and 0 respectively, confirming that only one component is above threshold and not suppressed. Plot (b) shows the eigenvalues λ_1^w, λ_2^w of $\mathbf{W}\mathbf{W}^T$, which analysis shows converge to the input component variance scaled by β/α^2 (0.64 for the first component), or 0 if suppressed.

feedback networks are particularly simple in form, typically consisting of a local Hebbian algorithm with weight decay.

Feedback inhibitory connections are thought to be important in many biological sensory and motor systems [14]. Investigating this type of artificial feedback network may help us to learn about biological information processing in these systems.

In addition, recent work by Harpur and Prager [15] indicates that simple rectification non-linearities in feedback networks of this type can perform useful unsupervised feature extraction. The extension of these linear networks, with their information-theoretic justification, into the non-linear domain, is a very promising area for future study.

Acknowledgments

Part of this work was supported by a grant (GR/J38987) from the UK Engineering and Physical Sciences Research Council. The author would also like to thank two anonymous referees, whose comments helped to improve this article.

References

- [1] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423,623–656, 1948.

- [2] H. B. Barlow. Three points about lateral inhibition. In W. Rosenblith, editor, *Sensory Communication*, pages 782–786. MIT Press, 1961.
- [3] R. Linsker. Self-organization in a perceptual network. *IEEE Computer*, 21(3):105–117, March 1988.
- [4] J. J. Atick and A. N. Redlich. Convergent algorithm for sensory receptive field development. *Neural Computation*, 5:45–60, 1993.
- [5] M. D. Plumbley. Lyapunov functions for convergence of principal component algorithms. *Neural Networks*, 8:11–23, 1995.
- [6] R. J. Williams. Feature discovery through error-correction learning. ICS Report 8501, University of California, San Diego, 1985.
- [7] E. Oja and J. Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications*, 106:69–84, 1985.
- [8] E. Oja. A simplified neuron model as a principal component analyser. *Journal of Mathematical Biology*, 15:267–273, 1982.
- [9] M. D. Plumbley. Efficient information transfer and anti-Hebbian neural networks. *Neural Networks*, 6:823–833, 1993.
- [10] M. D. Plumbley. A subspace network that determines its own output dimension. Technical Report 94/08, Department of Computer Science, King’s College London, UK, 1994.
- [11] H. G. Barrow and Julian M. L. Budd. Automatic gain control by a basic neural circuit. In I. Aleksander and J. Taylor, editors, *Artificial Neural Networks, 2*. Elsevier, 1992.
- [12] M. D. Plumbley. Analysis of an unsupervised indirect feedback network. In M. Marianaro and P. G. Morasso, editors, *Proceedings of the International Conference on Artificial Neural Networks, ICANN’94, Sorrento, Italy*, pages 397–400. Springer-Verlag, London, 1994.
- [13] M. D. Plumbley. Approximating optimal information transmission using local Hebbian algorithms in a double feedback loop. In S. Gielen and B. Kappen, editors, *Proceedings of the International Conference on Artificial Neural Networks, ICANN’93, Amsterdam, The Netherlands*, pages 435–440. Springer-Verlag, 1993.
- [14] S. W. Kuffler, J. G. Nicholls, and A. R. Martin. *From Neuron to Brain: A Cellular Approach to the Function of the Nervous System*. Sinauer Associates Inc., Sunderland, MA, second edition, 1984.
- [15] G. F. Harpur and R. W. Prager. A fast method for activating competitive self-organizing neural networks. In *Proceedings of the ISANN’94 Conference*, 1994.