

Approximating Optimal Information Transmission using Local Hebbian Algorithms in a Double Feedback Loop

Mark D. Plumbley

Centre for Neural Networks, King's College London
Department of Mathematics, Strand, London WC2R 2LS, UK
Tel: +44 (0)71-873 2214; Email: M.Plumbley@oak.cc.kcl.ac.uk

Abstract

Maximising mutual information (MI) under various constraints has been suggested as a goal for neural networks in a perceptual system. Networks using Hebbian algorithms have been found to be suitable for optimising MI with either input or output noise. In this paper we show that a double feedback loop network, using local Hebbian algorithms, can approximate the characteristics required for optimizing MI with *both* input and output noise. This represents a better approximation than simply orthonormalising the principal subspace.

1 Introduction

Recently, interest has been expressed in using information theory to investigate unsupervised learning in perceptual systems. Linsker [3] suggested that a perceptual system should adapt itself to maximize the mutual information (MI) between its input and its output.

The type of network which optimizes information transmission depends considerably on the noise sources assumed to be present in the system. If the only noise source is one of independent equal-variance (*spherical*) Gaussian noise on the input components, then a linear N -input M -output network maximizes MI when the M outputs span the same subspace as the M principal eigenvectors of the input [5]. On the other hand, if the only noise source is one of spherical Gaussian noise on the *output* components, with a power cost preventing the output from being amplified without bound, MI is optimized when the output components are decorrelated and of equal variance [6].

The case of noise on both input and output has been analyzed in the frequency domain [1, 5]. This predicts a *whitening filter* which equalizes output power spectral density (equivalent to equalizing output variance) when the input signal-to-noise ratio (SNR) is large, but which cuts off completely below a certain critical input SNR, with intermediate behaviour between these two extremes. The close match that Atick and Redlich [1] found between this theoretical approach and the measured sensitivity curves for the human visual system is very suggestive that optimization of information transmission, with built-in power-like costs, is important for the efficient operation of real perceptual systems.

Linear unsupervised neural networks using local Hebbian algorithms have already been discovered which find optimum MI condition for single noise sources, i.e. with noise on *either* the input or the output. In this paper we attempt to construct a network with a local Hebbian algorithm which approximates an optimization of MI with both input *and* output noise.

2 Previous Network Algorithms

Consider a linear network with N -dimensional input vector \mathbf{x} , weight matrix W , and N -dimensional output vector \mathbf{y} such that $\mathbf{y} = W\mathbf{x}$ (Fig. 1(a)). Several neural network algorithms have been proposed over the last decade or so to allow such a network to extract one or more principal components of the input, or at least the subspace spanned by those principal components. Most of these are based on a principal component analysis (PCA) neuron due to Oja (see e.g. [4]). Any

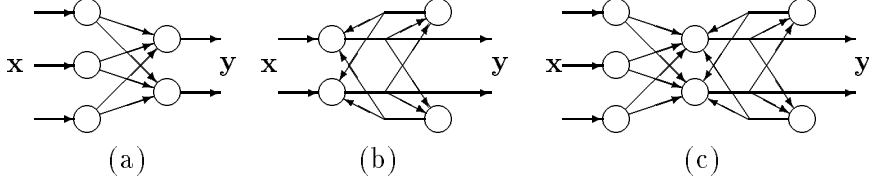


Figure 1: Previous information optimising networks.

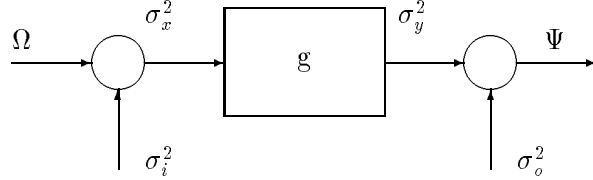


Figure 2: Block diagram showing information source Ω , destination Ψ , and noise sources.

one of these algorithms is sufficient to maximize MI in a network subjected to (spherical Gaussian) input noise only.

With spherical Gaussian noise on the *output*, and an output power cost proportional to $\sum_i \sigma_{y_i}^2 = \text{Tr}(\Sigma_y)$, where $\Sigma_y = E(\mathbf{y}\mathbf{y}^T)$ is the output covariance matrix, MI is optimized if the output signal is also spherical Gaussian [6], i.e. $\Sigma_y = \beta I$. This can be achieved using e.g. a network with inhibitory interneurons (Fig. 1(b)), where we have $\mathbf{z} = V^T \mathbf{y}$ and $\mathbf{y} = \mathbf{x} - V \mathbf{z}$ i.e. $\mathbf{y} = (I + VV^T)^{-1} \mathbf{x}$ after an initial transient. In this case, the algorithm

$$\Delta V = \eta_V (\mathbf{y}\mathbf{z}^T - \beta V) = \eta_V (\mathbf{y}\mathbf{y}^T - \beta I)V \quad (1)$$

is suitable, since it converges when $\Sigma_y = \beta I$ as required, provided the variances of all the input components (the eigenvalues of Σ_x) are greater than β . The variance of any input component which is less than β will be left unchanged at the output: in effect the network ‘decouples’ itself from these components [6].

In an earlier paper [7], the author approached the problem of combining these solutions for single noise sources for the case of both input and output noise. A combination of a modified PCA stage and the inhibitory interneuron stage generalized from the Barrow and Budd [2] Automatic Gain Control (AGC) network was suggested (Fig. 1(c)) with the local Hebbian algorithms

$$\Delta W = \eta_W (\mathbf{y}\mathbf{x}^T - \alpha W) \quad \Delta V = \eta_V (\mathbf{y}\mathbf{z}^T - \beta V). \quad (2)$$

This network does cut off small variance components, and fixes large variance input components to $\sigma_y^2 = \beta$, as required for an optimal filter, but the mid-range characteristics need improving.

3 Mid-range Optimal Filtering

Fig. 2 shows a 1-input 1-output linear system with input and output noise and gain g . We use the Lagrange multiplier technique to maximize the mutual information (MI) $I(\Psi, \Omega)$ across this system with power cost σ_y^2 . Thus we should maximize the function

$$J = I(\Psi, \Omega) - (\lambda/2)\sigma_y^2 = (1/2) \left(\log(\sigma_y^2 + \sigma_o^2) - \log(g^2\sigma_i^2 + \sigma_o^2) - \lambda\sigma_y^2 \right) \quad (3)$$

where λ is a Lagrange multiplier. Differentiating and equating to zero, and scaling so that $\sigma_i^2 = \sigma_o^2 = 1$, the optimal value for σ_y^2 satisfies

$$\sigma_y^2 \left[(\sigma_x^2)^2 - (\sigma_x^2 + \sigma_y^2)(1 + \lambda(\sigma_y^2 + 1)\sigma_x^2) \right] = 0 \quad (4)$$

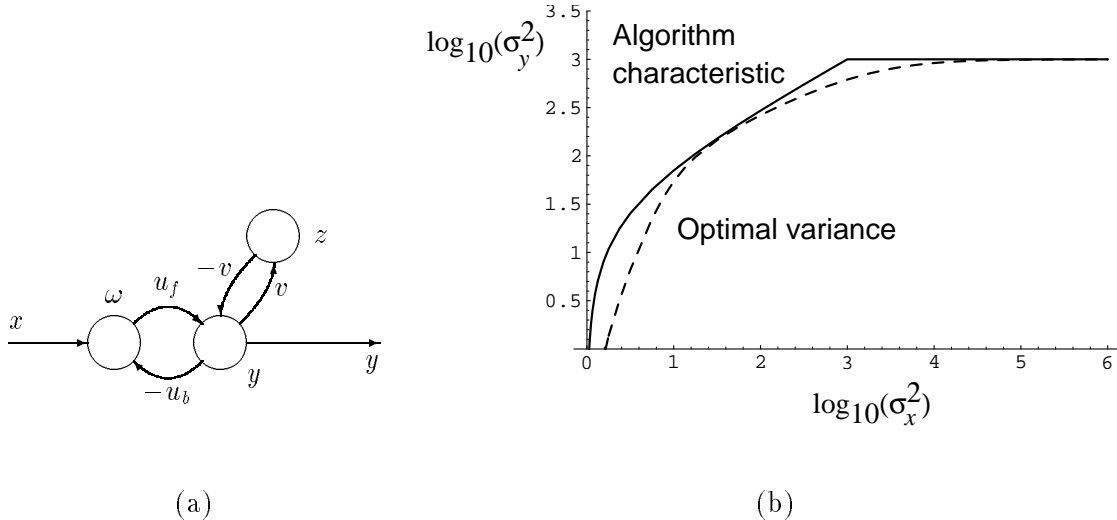


Figure 3: Double feedback loop network: (a) layout; (b) characteristic

If $\sigma_x^2 \gg 1/\lambda$ and $\sigma_x^2 \gg 1$ then the optimal value for the output noise is approximately constant for a given value of λ , i.e.

$$\sigma_y^2 \approx 1/\lambda - 1 \approx 1/\lambda \quad \text{if } \lambda \ll 1 \quad (5)$$

Alternatively, if $\sigma_x^2 \leq 1 + \lambda$, the output variance σ_y^2 should be zero, so the system should shut off completely. Both of these ranges are already handled by the PCA/inhibitory interneuron combination (Fig. 1(c)) considered in the previous section.

For $\lambda \ll 1$ a mid range appears between these two extremes. If we consider the case $1/\lambda \gg \sigma_x^2 \gg 1$, we find that the optimal value for output variance approximates

$$\sigma_y^2 \approx (\sigma_x^2/\lambda)^{1/2}. \quad (6)$$

In a previous paper [6] the author observed that the variance of the inhibitory interneuron in a decorrelating network approximately varies with the square root of the input variance, which is precisely the behaviour we require for this mid range. This leads us to suggest a *double* feedback loop (Fig. 3(a)): the first to generate the low input variance cutoff and square-root-variance behaviour in the mid range, with the second to limit the upper bound to the output variance in the high input variance regime.

4 Double Feedback Loop Operation

When the activations in the network of Fig. 3(a) have settled, we have that

$$z = vy \quad (7a)$$

$$y = u_f \omega - vz \quad (7b)$$

$$\omega = x - u_b y \quad (7c)$$

from which we can deduce, for example, that

$$(1 + v^2)y = u_f \omega. \quad (8)$$

We propose to use the following local Hebbian algorithms with weight decay

$$\Delta u_f = \eta_u (\omega y - \alpha_f u_f) \quad (9a)$$

$$\Delta u_b = \eta_u (\omega y - \alpha_b u_b) \quad (9b)$$

$$\Delta v = \eta_v (yz - \beta v) \quad (9c)$$

where η_u and η_v are update rates for the weights in the first and second loops respectively. To simplify the analysis for the purposes of this paper, we shall assume that $\eta_v \gg \eta_u$, so v adapts much faster than u_f and u_b (this makes stability conditions simpler, but does not affect stationarity conditions).

Consider now the conditions for (9a)–(9c) to be stationary on the average: we approximate this by setting their expectations to zero. (See e.g. [4] for an example of a more rigorous approach to this type of problem using stochastic approximation.) Firstly, equating the expected value of (9c) to zero and substituting in (7a) gives us

$$E(y^2)v - \beta v = 0$$

i.e.

$$\sigma_y^2 = \beta \quad \text{or} \quad v = 0. \quad (10)$$

Now, equating the expected value of (9a) to zero we get

$$E(\omega y) = \alpha_f u_f$$

which, multiplying both sides by $(1 + v^2)$ and substituting in (8) gives us

$$E(\omega^2)u_f = \alpha_f u_f (1 + v^2)$$

so

$$\sigma_\omega^2 = \alpha_f (1 + v^2) \quad \text{or} \quad u_f = 0. \quad (11)$$

Finally, equating the expected value of both (9a) and (9b) to zero, we get $\alpha_f u_f = \alpha_b u_b$, i.e.

$$u_b = (\alpha_f / \alpha_b) u_f. \quad (12)$$

Let us now consider the implications of these convergence conditions (10)–(12). One thing to notice immediately is that there is a *choice* of conditions, due to the ‘or’s in (10) and (11). We shall see in a moment that these lead to three (rather than four) convergence regimes, depending on the input variance σ_x^2 .

Firstly, choose the condition $u_f = 0$ from (11). We can immediately see from (7c) that $\sigma_\omega^2 = \sigma_x^2$ in this case. In addition, we also have $u_b = 0$ from (12), and also that

$$\sigma_y^2 = 0 \quad (13)$$

from the stationarity of (9a) (or (9b)). Thus, provided $\beta \neq 0$, in (10) we must have $v = 0$ since $\sigma_y^2 = 0 \neq \beta$: this has removed one of our choices. Perturbation analysis reveals that this regime is only stable for $\sigma_x^2 \leq \alpha_f$.

This time, choose $\sigma_\omega^2 = \alpha_f (1 + v^2)$ from (11) but $v = 0$ from (10): consequently $\sigma_\omega^2 = \alpha_f$. Substituting (7b) into (7c) and rearranging, we get $\omega(1 + u_b u_f) = x$ so

$$\sigma_x^2 = \sigma_\omega^2 (1 + u_b u_f)^2 = \alpha_f (1 + u_b u_f)^2. \quad (14)$$

Substituting in (12) gives us

$$(\sigma_x^2 / \alpha_f)^2 = 1 + (\alpha_f / \alpha_b) u_f^2 \approx (\alpha_f / \alpha_b) u_f^2 \quad (15)$$

for $\sigma_x \gg \alpha_f$, which with (7b) gives us

$$\sigma_y^2 \approx \alpha_b (\sigma_x^2 / \alpha_f)^{1/2}. \quad (16)$$

Our final choice gives us

$$\sigma_y^2 = \beta \quad (17)$$

which implies (after a little manipulation) that the input variance must satisfy

$$\sigma_x^2 = \alpha_f (1 + \beta / \alpha_b^2) (1 + v^2) \quad (18)$$

i.e. (17) is valid for $\sigma_x^2 \geq \alpha_f (1 + \beta / \alpha_b^2)$.

To summarize, this algorithm (9a)–(9c) produces three regimes of behaviour for the output variance σ_y^2 , depending on the range of the input variance σ_x^2 . These are:

1. Small input variance $\sigma_x \leq \alpha_f$, for which $u_f = u_b = 0$ so $\sigma_y^2 = 0$;
2. Intermediate input variance $\alpha_f \leq \sigma_x^2 \leq \alpha_f(1 + \beta/\alpha_b^2)$, for which $v = 0$ and $\sigma_y^2 \approx \alpha_b(\sigma_x^2/\alpha_f)^{1/2}$; and
3. Large input variance $\sigma_x^2 \geq \alpha_f(1 + \beta/\alpha_b^2)$, for which $v \neq 0$, leading to $\sigma_y^2 = \beta$.

These are similar to the three regimes of behaviour required for the optimal filter described in section 3, especially for small λ where the approximations are valid.

For a particular value of λ in (4), a reasonable approximation can be obtained using $\alpha_f = 1 + \lambda$ to fix the same cutoff condition, $\beta = 1/\lambda$ to fix the same asymptote, and $\alpha_b = (\alpha_f/\lambda)^{1/2} = (1 + 1/\lambda)^{1/2}$ to get similar mid-range behaviour. Fig. 3(b) compares the output variance produced by this algorithm with the optimal value of σ_y^2 for an example case of $\lambda = 10^3$.

5 Discussion

In sections 3 and 4 we have concentrated on the single-input, single-output case to investigate in detail. However, the same approach can be extended in a relatively straightforward manner to a network with N inputs and M outputs (with $M \leq N$).

With N inputs, we need to consider an orthogonal basis for the input components, perform a principal component analysis on the inputs, and optimize the orthogonal components independently according to (4), using the same value of λ for each component. If the number of outputs M is less than the number of inputs N , we simply ignore the $N - M$ input components with smallest input variance (i.e. the *minor* components). This will give us the optimal transform for the N input to M output transform.

The network of Fig. 3(a) can also be modified into an N -input M -output network in a similar manner. The two convergence regimes where the output loop of weights v units z is decoupled (the equivalent of $v = 0$) reduces to the single feedback loop that has already been considered by the author in a previous paper [6]. It was observed that such a network will reject input components below the cut-off limit, and produce an approximately square root variance output for others, but without explicitly performing a principal component analysis at any stage. It is possible to show that the second feedback loop does indeed limit larger components of the input to have variance no larger than β , but again without necessarily performing a principal component analysis to do so. We shall consider this in more detail in a later paper.

6 Conclusions

We have seen that a double feedback loop network, using only local Hebbian algorithms with weight decay, can learn to reasonably approximate the characteristics required for optimal transmission of information with both input and output noise, and an output power cost. This represents an improvement for mid-range input variance values over simply orthonormalising the principal subspace of the input.

Acknowledgements

I would like to thank John Taylor and Guido Bugmann for many useful comments and discussions about this paper and related work. The author is supported by a Temporary Lectureship from the Academic Initiative of the University of London.

References

- [1] J. J. Atick and A. N. Redlich. What does the retina know about natural scenes? *Neural Computation*, 4:196–210, 1992.

- [2] H. G. Barrow and J. M. L. Budd. Automatic gain control by a basic neural circuit. In I. Alexander and J. Taylor, editors, *Artificial Neural Networks, 2*. Elsevier, 1992.
- [3] R. Linsker. Self-organization in a perceptual network. *IEEE Computer*, 21(3):105–117, Mar. 1988.
- [4] E. Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5:927–935, 1992.
- [5] M. D. Plumbley. On information theory and unsupervised neural networks. Technical Report CUED/F-INFENG/TR.78, Cambridge University Engineering Department, UK, 1991.
- [6] M. D. Plumbley. Efficient information transfer and anti-Hebbian neural networks. *Neural Networks*, 1993. (in press).
- [7] M. D. Plumbley. A Hebbian/anti-Hebbian network which optimizes information capacity by orthonormalizing the principal subspace. In *Proceedings of the IEE Conference on Artificial Neural Networks, ANN'93*, Brighton, UK, May 1993. (To appear).